

FACULDADE DE TECNOLOGIA DE SÃO PAULO

**DANIEL SILVA FERREIRA BRUNO**

ÁRVORES DE CONHECIMENTO: UM ESTUDO DE CASO COM ANÁLISE

SÃO PAULO

2019

FACULDADE DE TECNOLOGIA DE SÃO PAULO

**DANIEL SILVA FERREIRA BRUNO**

ÁRVORES DE CONHECIMENTO: UM ESTUDO DE CASO COM ANÁLISE

Trabalho submetido como exigência parcial

para a obtenção do Grau de Tecnólogo em

Análise e Desenvolvimento de Sistemas

Orientador: Professora Mestre Sandra Harumi Tanaka

SÃO PAULO

2019

FACULDADE DE TECNOLOGIA DE SÃO PAULO

**DANIEL SILVA FERREIRA BRUNO**

ÁRVORES DE CONHECIMENTO: UM ESTUDO DE CASO COM ANÁLISE

Trabalho submetido como exigência parcial para a obtenção do Grau de  
Tecnólogo em Análise e Desenvolvimento de Sistemas.

Parecer do Professor Orientador

---

---

---

Conceito/Nota Final: \_\_\_\_\_

**Atesto o conteúdo contido na mídia entregue pelo aluno e assinada por mim  
para avaliação do TCC.**

**Estou ciente de que se o aluno não tiver entregado a mídia conforme regras do  
Roteiro ele estará reprovado na disciplina mesmo que esteja aprovado por mim.**

Orientador: Professora Mestre Sandra Harumi Tanaka

SÃO PAULO, \_\_\_\_ de \_\_\_\_\_ de 2019.

Assinatura do Orientador

## RESUMO

Devido à escassez de softwares móveis livres de código-fonte aberto, disponíveis sobre o tema de Árvores de conhecimento, foi feita nesta monografia, a análise de uma nova versão de um software típico deste tema. Este software foi inicialmente feito com um Banco local de dados, mas a versão futura dele correspondente a esta análise deve funcionar como um sistema distribuído via API REST, sendo acessível simultaneamente em vários smartphones distintos com o Sistema Operacional Android, ou iOS, por exemplo.

**Palavras-chave:** Android, API, Aplicativos, Árvores de conhecimento, Mobilidade, REST.

## **ABSTRACT**

Due to the scarcity of open source free softwares, available on the topic Knowledge Trees, this monograph was the analysis of a new version of a typical software for this theme. This software was initially made with a local database, but the future version of it corresponding to this analysis should work as a distributed system via REST API, being simultaneously accessible in several different smartphones with the Android Operating System, or iOS, for example.

**Keywords:** Android, API, Applications, Knowledge Trees, Mobility, REST.

# SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>7</b>
1.1 FIREBASE E O FIREBASE REALTIME DATABASE .....	8
1.2 BACK4APP .....	8
1.3 APIS REST.....	9
1.4 ANÁLISE DE SISTEMAS.....	9
<b>2. DESENVOLVIMENTO.....</b>	<b>11</b>
2.1 REQUISITOS FUNCIONAIS DA VERSÃO ANTERIOR DO SOFTWARE.....	11
2.2 REQUISITOS NÃO FUNCIONAIS DA VERSÃO ANTERIOR DO SOFTWARE .....	13
2.3 MODELO ENTIDADE RELACIONAMENTO (MER) DA VERSÃO ANTERIOR DO SOFTWARE .....	14
2.4 REQUISITOS FUNCIONAIS PARA A VERSÃO FUTURA DO SOFTWARE .....	14
2.5 REQUISITOS NÃO FUNCIONAIS PARA A VERSÃO FUTURA DO SOFTWARE.....	15
2.6 PROPOSTA DE MODELO ENTIDADE RELACIONAMENTO (MER) PARA A VERSÃO FUTURA DO SOFTWARE .....	16
2.7 PROPOSTA DE CASOS DE USO PARA A VERSÃO FUTURA DO SOFTWARE.....	19
2.8 DESCRIÇÃO DOS CASOS DE USO PARA A VERSÃO FUTURA DO SOFTWARE.....	21
2.9 SOBRE A VERSÃO IMPLEMENTADA DO SOFTWARE PARA ESTA MONOGRAFIA.....	26
2.10 ESPECIFICAÇÃO DA API REST PARA A VERSÃO FUTURA DO SOFTWARE.....	26
2.11 ESPECIFICAÇÃO DA API REST: GERÊNCIA DE USUÁRIOS E DE EMPRESAS .....	27
2.12 ESPECIFICAÇÃO DA API REST: GERÊNCIA DE CERTIFICAÇÕES E DE REQUISIÇÕES.....	27
2.13 ESPECIFICAÇÃO DA API REST: GERÊNCIA DE COMPETÊNCIAS.....	27
<b>3. CONCLUSÃO .....</b>	<b>28</b>

# 1. INTRODUÇÃO

Atualmente, não existem muitos apps livres e de código-fonte aberto, disponíveis sobre o tema de Árvores de Conhecimentos. Assim, o autor desta monografia fez uma parte do desenvolvimento inicial de um app para Android, em Java, que, usando um Banco de Dados gerido em memória RAM, servia para gerenciar uma Árvore de Conhecimentos, para somente uma Empresa (na disciplina de Interação Humano Computador, IHC, no segundo semestre de 2017). E, ele também fez um novo desenvolvimento completo deste app para Android, em Java, que, com o uso da biblioteca de Persistência, chamada Room, da Google, versão 1.1.1, gerenciava uma Árvore de Conhecimentos para uma Empresa que estivesse localizada em somente um único endereço físico (na disciplina de Laboratório de Engenharia de Software, LESW, no segundo semestre de 2018). Reforça-se aqui que este software refeito tem o inconveniente de servir somente para Empresas de localização física única (que seria o caso de micro e pequenas empresas, em geral), por ter sido feito para usar somente o banco de dados local de um único dispositivo móvel. Realmente, a versão posterior deste software (de LESW) é melhor do que a inicial (de IHC), por permitir o armazenamento permanente das mudanças feitas durante qualquer execução deste app, pois as mudanças feitas durante alguma execução deste app na versão inicial não eram gravadas, pois eram perdidas do Banco de Dados após esta execução.

Baseado nas características fornecidas do contexto atual, o objetivo deste projeto de formatura será o de analisar um app móvel, assim como, a parte de Backend deste Sistema de Informação como um todo, para gerência ubíqua de Árvores de Conhecimentos, dado que este app será uma nova versão do software feito em LESW. Assim, uma arquitetura distribuída para este novo software será completamente analisada.

Realmente, a não existência de muitos apps livres para gerência de Árvores de Conhecimentos justifica a criação de uma arquitetura distribuída na nova versão deste app.

A base das hipóteses desta monografia é descrita pelas seguintes ideias:

- Como uma versão futura do app feito em LESW poderia ser analisada.

Nesta monografia, a Revisão bibliográfica e o Estudo de Caso serão usados como Metodologias de Pesquisa.

A versão futura deste app tem como objetivo facilitar a visualização das competências dos funcionários de uma empresa, de modo que seja feita em diversos smartphones ou tablets, para uma mesma empresa, tanto localmente como remotamente.

Esta monografia sugere dois recursos de TI para construir um software distribuído escalável: Firebase e Back4App. Ressalta-se que a construção desta nova versão não é o foco desta monografia, pois ela está restrita somente à análise deste app.

## 1.1 Firebase e o Firebase Realtime Database

Segundo Damasceno (2019), o recurso Firebase Realtime Database é um Banco de Dados NoSQL, na forma de uma árvore de dados JSON, pertencente à plataforma Firebase. E segundo Duarte (2018), a Firebase é uma plataforma do tipo Backend-as-a-service da Google, gerida na nuvem, e sincronizada em tempo real. Duarte (2018) também sugere que hajam poucos níveis aninhados nos objetos JSON do Banco de Dados da Firebase, assim como a inexistência de uma normalização excessiva, de modo que espalhe demasiadamente os dados geridos.

Nota-se, segundo Jamin (2016), que a Firebase não é uma boa opção para casos específicos, como: o do app referente a esta monografia, por exemplo. As causas desta conclusão são:

- Código do tipo macarrão ou espaguete (Spaghetti code, em inglês), implicando em um desenvolvimento adicional da nova gerência dos dados neste app, e sabe-se que este app exigiu um razoável esforço no desenvolvimento da versão feita em LEWS, e assim, uma nova demanda considerável de tempo é esperada para desenvolver esta gerência via plataforma Firebase;
- A Firebase baixa todas as sub-árvores de dados JSON sempre que um app é iniciado num smartphone, implicando que este aparelho baixaria os dados de todas as Árvores de Conhecimentos armazenadas na nuvem, embora se saiba que ele servirá para gerir somente uma Árvore de Conhecimentos, que seria de somente uma única empresa, quando ele for executado;
- A Firebase pode apresentar inconsistências em casos de operações sem internet, implicando em possíveis inconsistências nas Árvores de Conhecimentos Empresariais armazenadas na nuvem para este app; As situações de uso sem internet deste app foram supostas como certas, ainda que, eventualmente;
- A Firebase é ruim para lidar com relações de chaves primárias no Banco de dados, e o app desta monografia usa este tipo de relação, implicando em eventual código adicional do tipo “Spaghetti code”;
- A Firebase é ruim para lidar com queries complexas no Banco de dados, implicando em mais “Spaghetti code” para simular este comportamento; O aplicativo desta monografia apresenta algumas queries relativamente complexas.

## 1.2 Back4App

Segundo Ribeiro (2017), a Back4App é uma alternativa viável à plataforma Firebase, pois ela oferece Segurança e Privacidade, assim como, Desempenho e Produtividade. Assim, um usuário não acessa os dados de outro usuário, e os dados coletados são previamente validados. A Back4App permite o uso de queries complexas e eficientes, e usa um recurso de Cache no lado do cliente.

Baseando-se nestas informações, nota-se que a Back4App pode ser melhor do que a Firebase, o que pode ser verificado em um Trabalho de Conclusão de Curso futuro envolvendo o app desta monografia. Segundo Klimkin (2017), a Back4App usa o Banco de dados MongoDB internamente, e o MongoDB foi sugerido como alternativa à Firebase pela orientadora desta monografia.



Mojica (2016) cita outros serviços de Back-end que podem usar o banco de dados MongoDB, como, por exemplo, o AWS, o Google App Engine, e o Heroku, para o servidor. E para o banco de dados, ele cita o MongoLab e o ObjectRocket.

### **1.3 APIs REST**

Segundo Redação (2019), uma API (Application Programming Interface, em inglês) é um conjunto de rotinas e padrões de programação para acesso a um aplicativo de software ou plataforma baseada na Web. Uma API pode ser criada por uma empresa de Software, com a intenção de que outros desenvolvedores de Software usem este serviço. Com as API's, não há necessidade de intervenção direta dos usuários.

Segundo Gomes (2008), REST é uma abreviação de REpresentational State Transfer, sendo um conjunto de princípios que definem como Web Standards do tipo HTTP (HyperText Transfer Protocol) e URI's (Uniform Resource Identifier) devam ser usados. Os cinco princípios da REST são:

- Todas as coisas devem ter um identificador;
- Todas as coisas devem ser vinculadas;
- Todos os métodos devem ser padronizados;
- Todos os recursos devem ter múltiplas representações;
- Todas as comunicações não devem ter estado.

Nota-se assim que uma API REST é uma API que segue as normas REST.

Ferreira (2017) explica sobre boas práticas de desenvolvimento de API's REST, como por exemplo, o uso de URI's legíveis e do mesmo padrão de URI na identificação dos recursos, a não alteração de URI's pré-existentis, o uso dos métodos HTTP para manipular, o suporte às diversas representações dados (HTML, XML ou JSON), o suporte à Comunicação sem Estado, e a segurança dos dados de autenticação ou autorização, além da correta utilização dos códigos HTTP.

### **1.4 Análise de Sistemas**

Segundo Batista (2012), um sistema é o conjunto estruturado ou ordenado de partes ou elementos que se mantém em interação, ou seja, em ação recíproca, na busca da consecução de um ou de vários objetivos. Assim, segundo Batista (2012), um sistema se caracteriza, sobretudo, pela influência que cada componente exerce nos demais e pela união de todos (globalismo ou totalidade), para gerar resultados que levam ao objetivo esperado.

Segundo Batista (2012), uma Empresa é um sistema aberto, pois ela sofre interação dos seus subsistemas (departamentos) e do ambiente externo (mercado em que atua).

Segundo Batista (2012), o Sistema de Informação, sendo um subsistema da empresa, pode ser definido como todo e qualquer sistema que apresente dados ou informações de entrada que tenham por fim gerar informações de saída para suprir determinadas necessidades empresariais.

Segundo Bezerra (2016), os Sistemas de Informações servem para gerenciar informações de uma forma adequada e eficiente, e para dar suporte, assim como

melhorar o processo de negócio de uma organização com relação às informações que nela fluem.

Segundo Batista (2012), a informação é o resultado do tratamento dos dados existentes acerca de alguém ou de alguma coisa, e ela aumenta a consistência e o conteúdo dos dados relacionados.

Nota-se que esta monografia trata de um Sistema de Informações, que será usado por Empresas, geralmente no Departamento de Recursos Humanos delas.

Segundo Júnior (2001), nota-se que a Análise de Sistemas é composta por várias atividades ou fluxos de trabalho de desenvolvimento de Software, citados, por exemplo, no Processo Unificado. Estas atividades são: a Análise de Requisitos, a Análise do Software, e o Projeto do Software.

Segundo Júnior (2001), a Análise de Requisitos visa especificar os requisitos de Software, via identificação das necessidades de usuários e clientes. Ainda segundo Júnior (2001), os requisitos funcionais são expressos via casos de uso, através do modelo de casos de uso, que é composto por todos os atores e casos de uso de um sistema, e estes requisitos especificam como este Software será usado sob a perspectiva de clientes, usuários e desenvolvedores. Júnior (2001) explica que a identificação de casos de uso é feita pelo estudo do uso deste Software pelos usuários para a realização de trabalhos ou tarefas.

Segundo Júnior (2001), a Análise do Software visa gerar o modelo de Análise, que tem a função de refinar os requisitos especificados na atividade de Análise de Requisitos, através da construção de diagramas de classes conceituais, e fornece mais poder expressivo e formalismo, como diagramas de interações e diagramas de gráficos de estados que representam a dinâmica do Sistema.

Segundo Júnior (2001), o Projeto do Software visa moldar o sistema de informação, de modo que sua forma seja definida para suprir as necessidades especificadas pelos requisitos funcionais e não funcionais, produzindo um modelo de projeto baseado no modelo de análise definido pela atividade de Análise do Software. Ainda segundo Júnior (2001), o modelo de projeto descreve este sistema no nível físico, com uma compreensão detalhada dos requisitos deste sistema, e ele leva em conta fatores como: Linguagens de Programação, Sistemas Operacionais, Tecnologias de Bancos de Dados, Interface com o Usuário etc. Júnior (2001) mostra também que o modelo de projeto descreve as realizações físicas de casos de uso, e outros detalhes relacionados aos ambientes de implantação deste sistema.

A Análise de Sistemas é geralmente feita para Sistemas de Informações, pois eles são uma combinação de pessoas, dados, processos, interfaces, redes de comunicação e tecnologia, segundo Bezerra (2016). Nota-se que esta atividade é geralmente restrita à área de Tecnologia da Informação, ainda que seja feita em uma organização que não tenha esta área como propósito final.

## 2. DESENVOLVIMENTO

Os requisitos funcionais e não funcionais da nova versão deste app serão elicitados mentalmente pelo Analista de Sistemas, que é o autor desta monografia.

A documentação detalhada do trabalho feito em LESW pode ser consultada no respectivo repositório do GitHub, citado nas referências bibliográficas desta monografia.

As seguintes questões de Análise de Sistema serão tratadas nesta seção:

- Requisitos Funcionais da versão anterior do Software;
- Requisitos Não Funcionais da versão anterior do Software;
- Modelo Entidade Relacionamento (MER) da versão anterior do Software;
- Requisitos Funcionais para a versão futura do Software;
- Requisitos Não Funcionais para a versão futura do Software;
- Proposta de Modelo Entidade Relacionamento (MER) para a versão futura do Software;
- Proposta de Casos de Uso para a versão futura do Software;
- Descrição dos Casos de Uso para a versão futura do Software;
- Especificação da API REST para a versão futura do Software.

### 2.1 Requisitos Funcionais da versão anterior do Software

Os requisitos funcionais (Casos de Uso) para a versão anterior deste Software são:

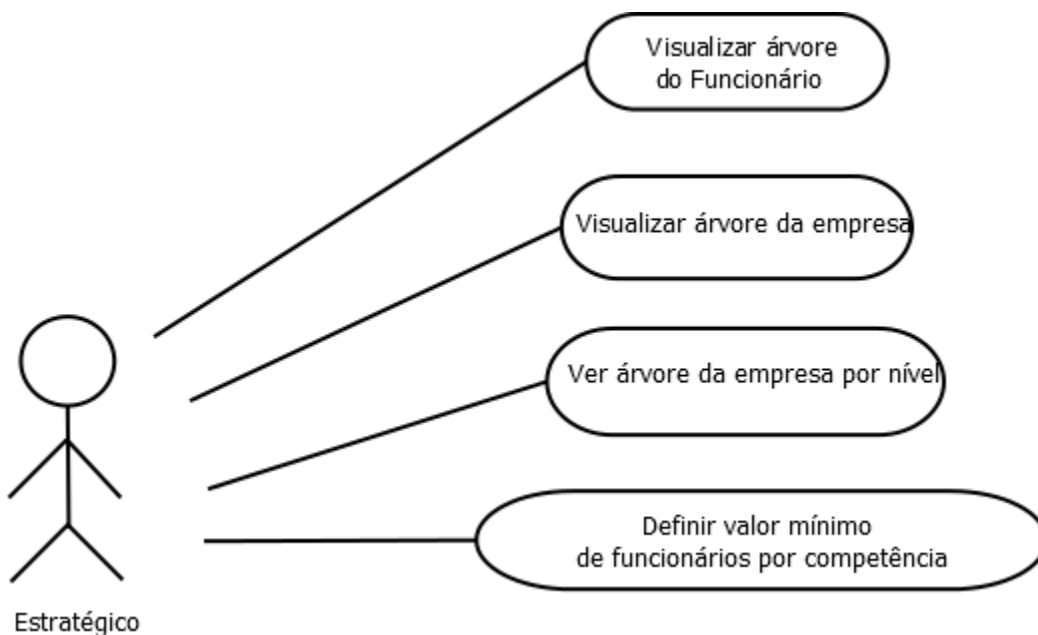


Figura 1 Casos de uso para o perfil Estratégico

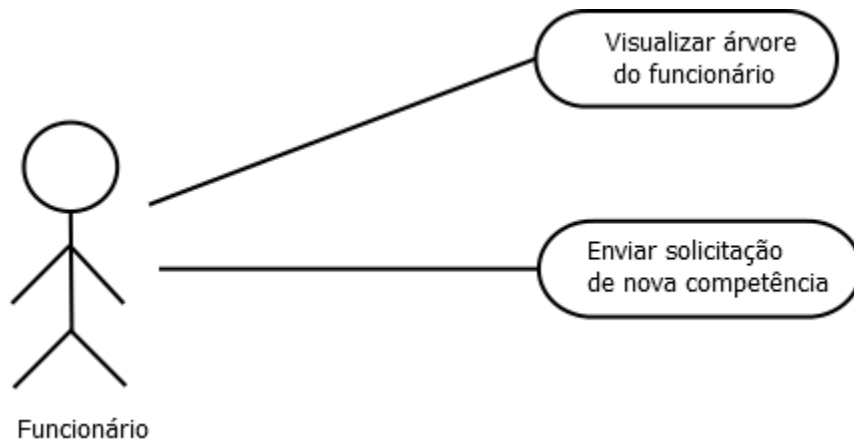


Figura 2 Casos de uso para o perfil Funcionário

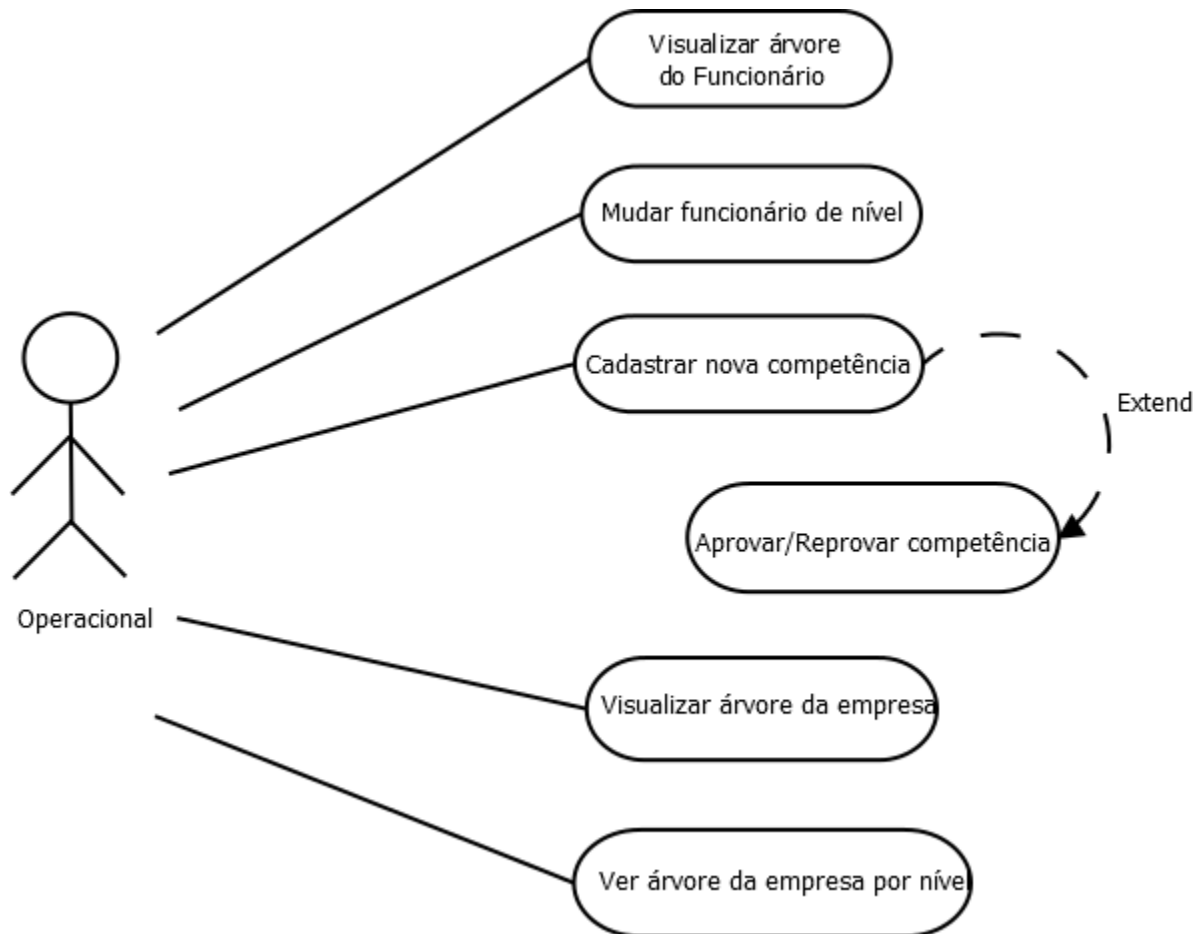
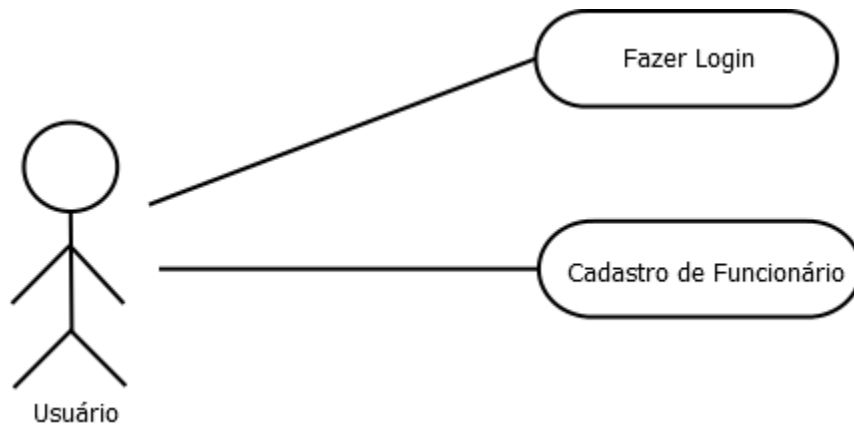


Figura 3 Casos de uso para o perfil Operacional



**Figura 4** Casos de uso para o perfil Usuário

Visava-se somente a custódia de competências formais, isto é, aquelas que fossem obtidas por meio de um certificado.

## **2.2 Requisitos Não Funcionais da versão anterior do Software**

Os Requisitos não funcionais desta versão são:

- Usar a linguagem Java para desenvolvimento;
- Usar o Android Studio versão 3.2.x;
- Usar smartphones ou tablets, com pelo menos a versão 4.2.x, JellyBean, do Sistema Operacional Android;
- Usar a biblioteca de Persistência Room, versão 1.1.1, da Google.

## 2.3 Modelo Entidade Relacionamento (MER) da versão anterior do Software

O MER desta versão antiga era:

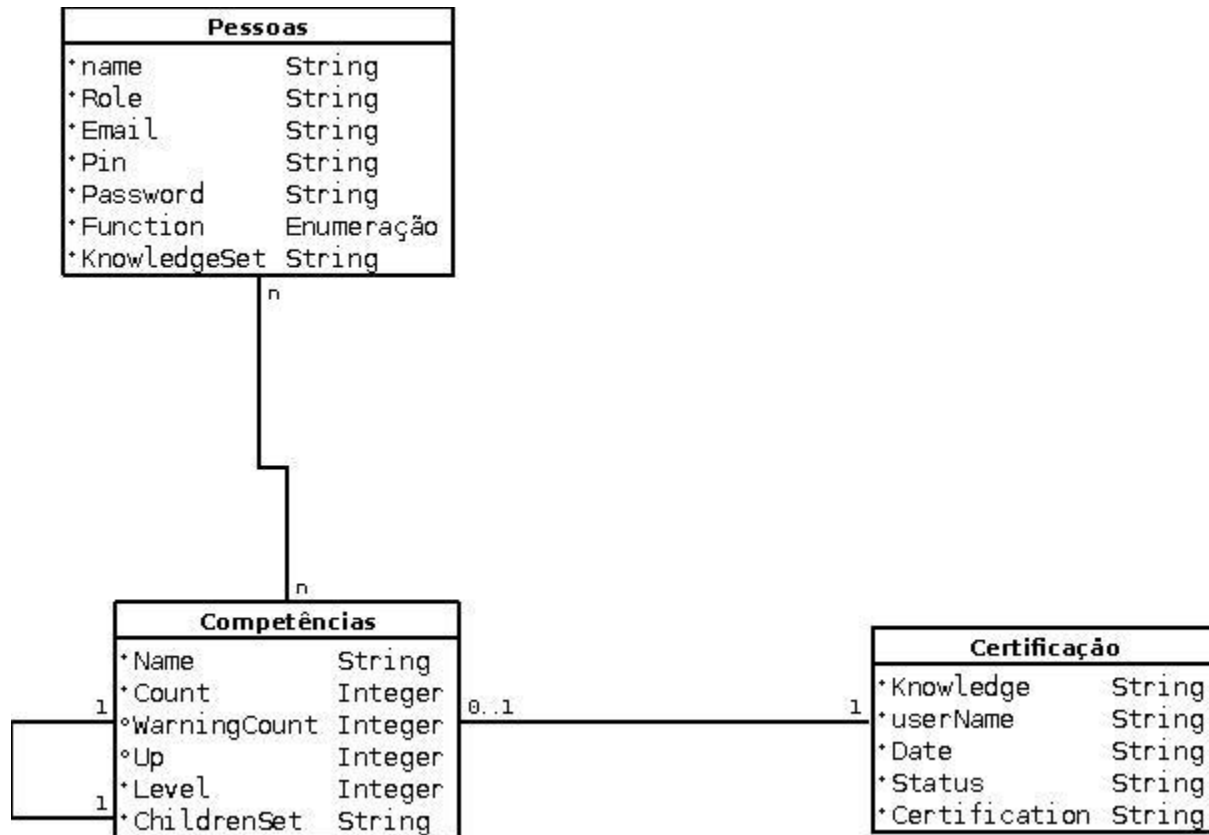


Figura 5 Modelo Entidade-Relacionamento do software antigo.

## 2.4 Requisitos Funcionais para a versão futura do Software

Os requisitos funcionais (Casos de Uso e regras de negócio) para a nova versão deste Software são:

1. O Software deve ser capaz de cadastrar todos os empregados de qualquer empresa, nos níveis: Funcionário, Operacional e Estratégico.
2. O Software deve ser capaz de cadastrar uma nova empresa, registrando somente o seu nome (que pode ser a Razão Social, o Nome Fantasia, ou mesmo o nome da Marca desta empresa), somente via usuário com perfil Estratégico, durante o cadastro deste usuário.
3. O Software deve ser capaz de associar o dado de identificação de uma empresa pré-cadastrada com um usuário de todos os níveis disponíveis, durante o cadastro deste usuário.
4. O Software deve ser capaz de cadastrar uma Competência, somente via um usuário com perfil Operacional (RH).
5. O Software deve ser capaz de cadastrar uma Certificação para um usuário com qualquer perfil possível.
6. O Software deve ser capaz de cadastrar uma Requisição para um usuário, independentemente do seu perfil.

7. O Software deve permitir o login de usuários pré-cadastrados, independentemente do seu perfil.
8. O Software deve permitir a visualização da árvore do funcionário para todos os perfis disponíveis.
9. O Software deve permitir a mudança de nível de funcionários pré-cadastrados, desde que via usuário com perfil Operacional (RH).
10. O Software deve permitir a aprovação ou a reprovação de competências para um usuário com qualquer perfil disponível, desde que via usuário com perfil Operacional (RH).
11. O Software deve permitir a visualização completa da árvore da empresa para usuários com perfil Operacional (RH) ou perfil Estratégico.
12. O Software deve permitir a visualização completa de um nível da árvore da empresa para usuários com perfil Operacional (RH) ou perfil Estratégico.
13. O Software deve permitir, a qualquer usuário com perfil Estratégico, a definição do valor mínimo de funcionários que possuam determinada competência na Árvore de Conhecimentos da Empresa.
14. O Software não deve guardar competências e requisições de um funcionário demitido, devendo ser as certificações deste funcionário guardadas com ele para uso na sua Árvore de conhecimentos em uma posterior empresa.

## **2.5 Requisitos Não Funcionais para a versão futura do Software**

Os requisitos não funcionais para este Software são:

1. O Software deve ser programado em Linguagem Java.
2. O Software deve ser feito no Android Studio 3.4.x (onde x representa o dígito da versão mais recente da família de versões 3.4 do Android Studio).
3. O Software rodará em dispositivos móveis com o Sistema Operacional Android da Google, com pelo menos a versão 4.2 JellyBean.

## 2.6 Proposta de Modelo Entidade Relacionamento (MER) para a versão futura do Software

Nota-se que a entidade Empresas deve ter somente um nome, e uma enumeração que represente o tipo desta empresa, como: Sociedade Limitada (M.E.), ou Sociedade Anônima (isto é, por Ações), etc. Assim, observa-se que, no LinkedIn (uma rede social trabalhista), as empresas são encontradas pelos nomes delas, e não por CNPJ ou Razão Social, por exemplo, pois, as empresas multinacionais não tem dado de CNPJ nos Estados Unidos da América.

Uma proposta de MER da nova versão do software poderia ser:

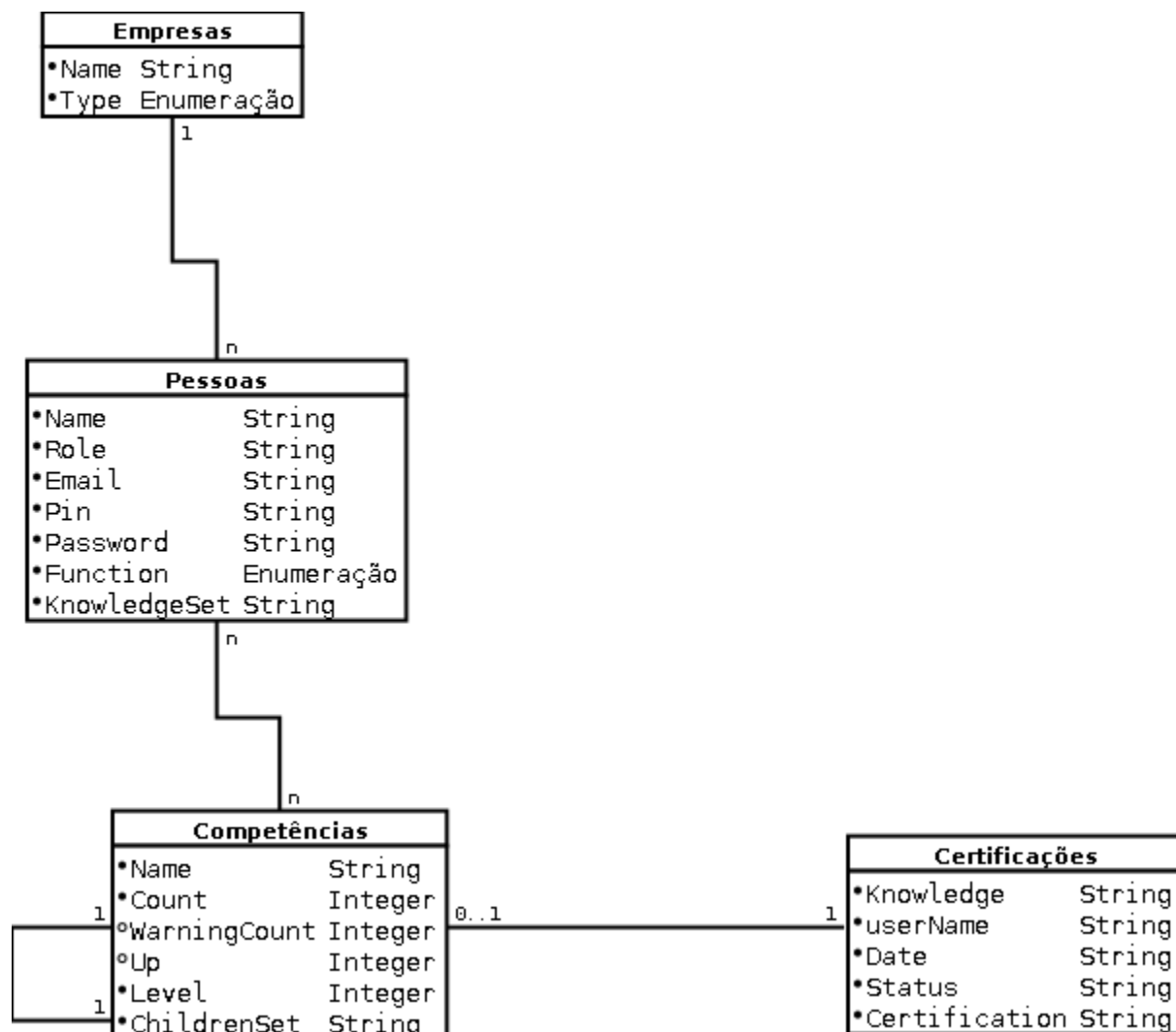


Figura 6 Primeira proposta do Modelo Entidade-Relacionamento do novo software



Realmente, esta proposta de MER não aborda o caso de um funcionário que adquiriu uma competência sem certificações, por exemplo, um programador Java que sabe programar em Java por ter aprendido em projetos Open Source ou em trabalhos Freelancer, mas que nunca obteve uma certificação em Java da Oracle (ou da SUN). Trata-se de um bug do Software feito em Laboratório de Engenharia de Software.

Esta proposta também não aborda o senso que o status de aprovação de uma Competência não deveria ser o status da Certificação associada a esta Competência. Assim, este status deveria ser o da própria Competência. Pois, a empresa não é apta para validar uma Certificação que já foi validada e entregue para o Funcionário por outra Organização competente. Trata-se de um erro na concepção do MER feita nos softwares anteriores.

Para resolver esta questão de um funcionário possuir uma Competência sem Certificações e a questão da validação da Certificação, propõe-se um novo MER:

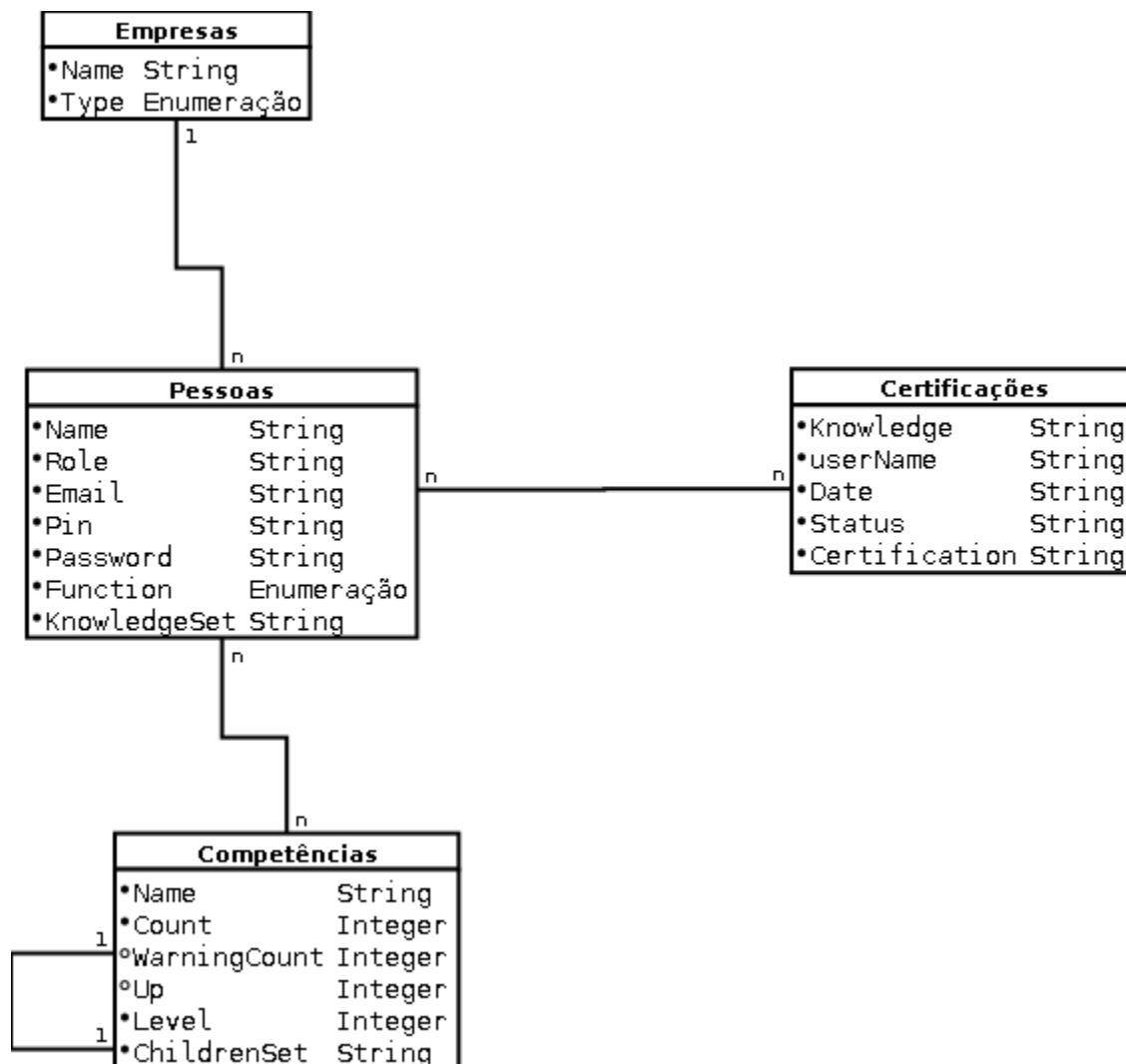


Figura 7 Nova proposta de MER para o novo Software

Pois nesta nova proposta de MER, os funcionários cadastram e usam suas Certificações para quaisquer empresas que eles quiserem trabalhar, e pedem alguma competência para o departamento de RH, que pode analisar o currículo e a Árvore de Conhecimentos deste funcionário, assim como, as Certificações dele, para decidir se eles

possuem ou não as competências pedidas por eles. Assim, esta nova proposta elimina a obrigatoriedade de apresentar-se uma certificação para pedir-se uma competência, por parte do funcionário. E a competência de um funcionário tem um atributo de Status para conferir se este funcionário possui esta competência de fato.

Mas, realmente, uma pessoa não pode ir obtendo a competência que quiser, devendo requisitá-la assim como faria se fosse mudar de cargo para uma posição de maior responsabilidade, como por exemplo, de Desenvolvedor Python Júnior para Desenvolvedor Python Pleno. Neste caso, a última proposta feita de MER deveria ser alterada para incorporar a Entidade de Requisição, que corresponde à formalização de um pedido de aquisição de competência por um empregado da organização representada pela Árvore completa de Conhecimentos dela. E esta nova proposta seria representada na figura 8, exibida a seguir.

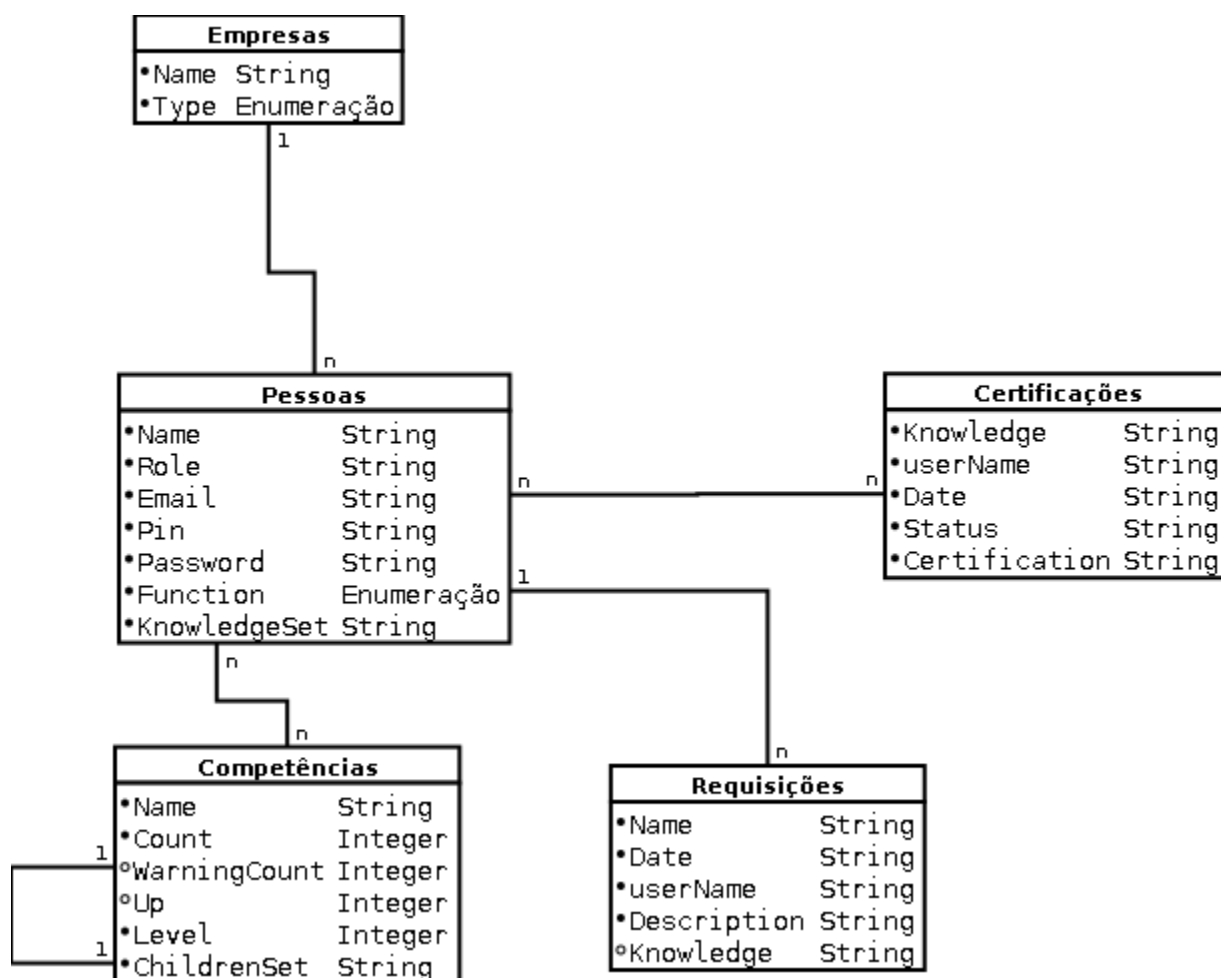


Figura 8 Proposta de MER com as Requisições, para o novo Software

Neste caso, o empregado faria uma requisição para o setor de Recursos Humanos, representado por usuários do SW com perfil operacional, e este setor analisaria esta requisição, eventualmente com os superiores hierárquicos deste empregado, e decidiria se este empregado deveria ou não adquirir a competência demandada. Na requisição, o nome da Competência não precisaria ser obrigatoriamente fornecido, embora fosse obrigatória, a apresentação de uma descrição da Competência exigida.

Supõe-se que tanto o serviço de Back-end como o aplicativo móvel irão seguir esta proposta de MER em seus respectivos Bancos de Dados.

## 2.7 Proposta de Casos de Uso para a versão futura do Software

E os Casos de Uso da nova versão do software são:

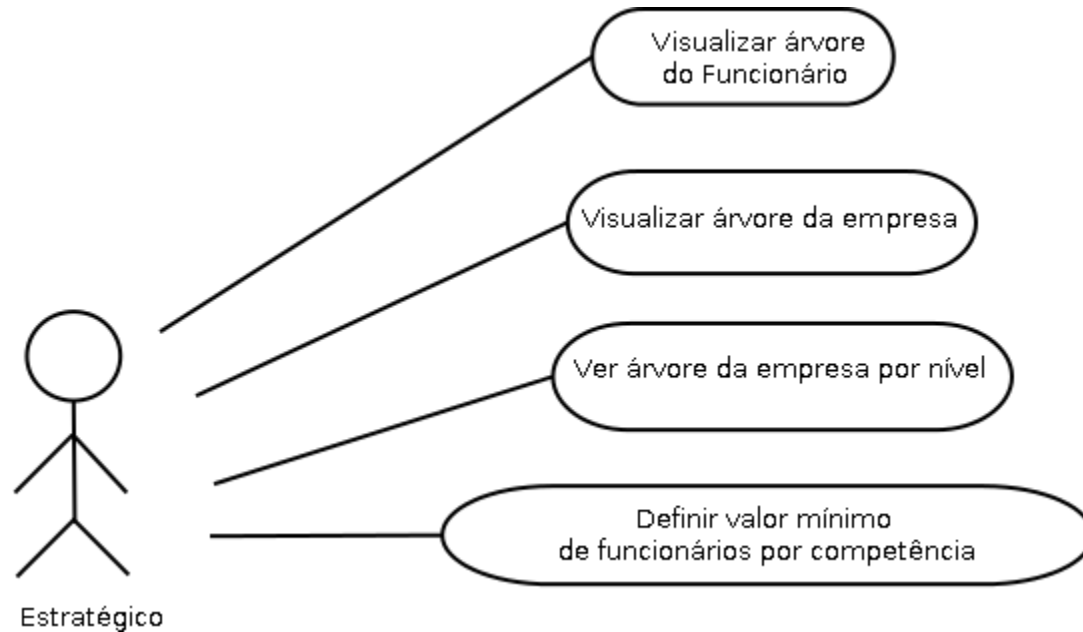


Figura 9 Caso de Uso para o perfil Estratégico

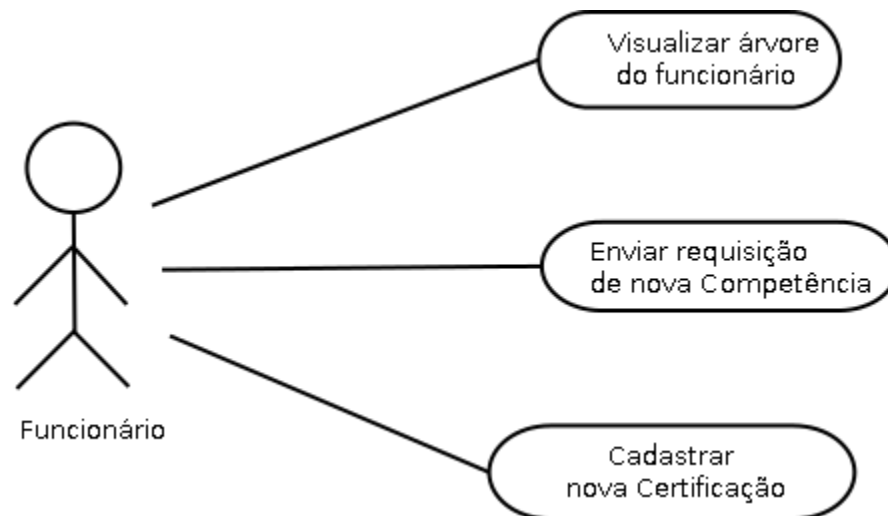


Figura 10 Caso de Uso para o perfil Funcionário

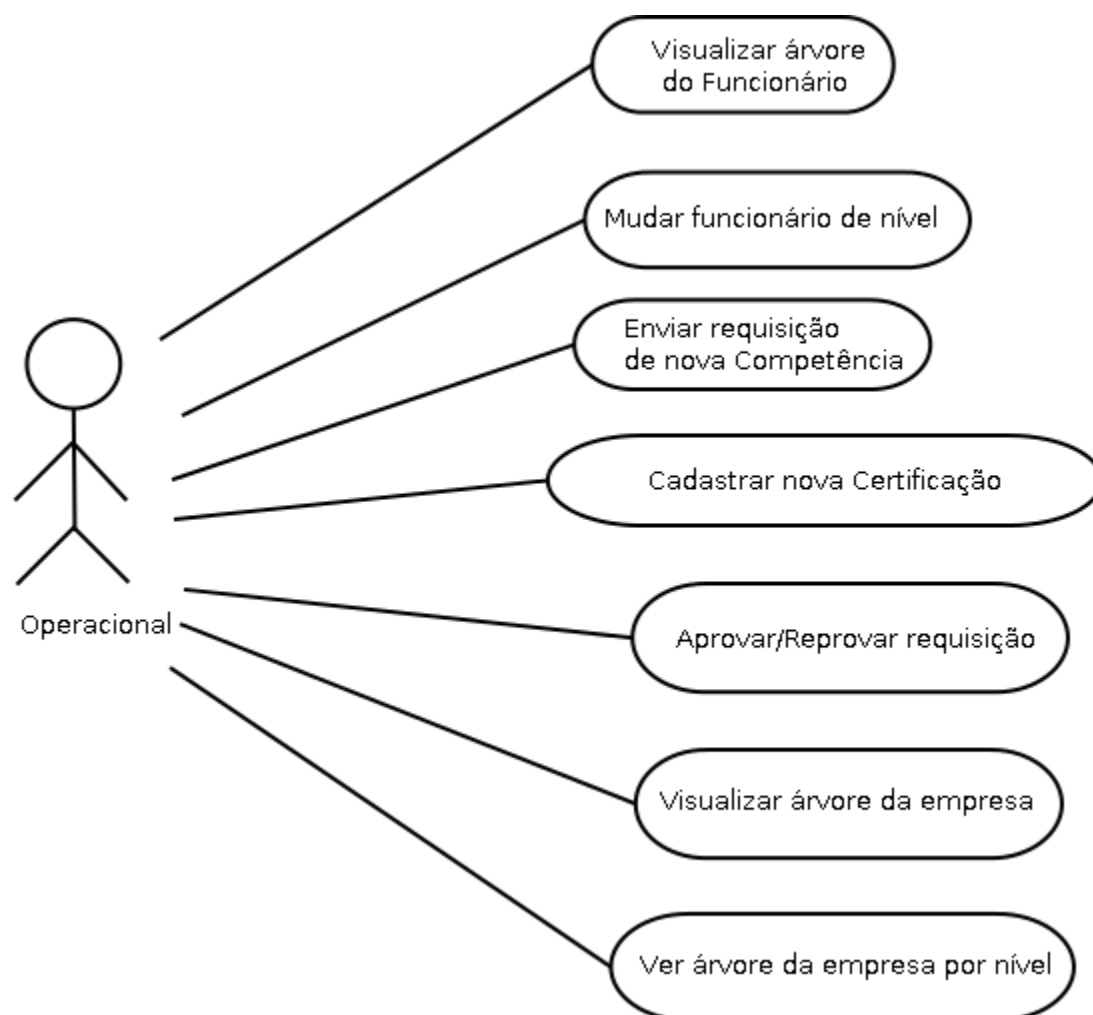


Figura 11 Caso de Uso para o perfil Operacional

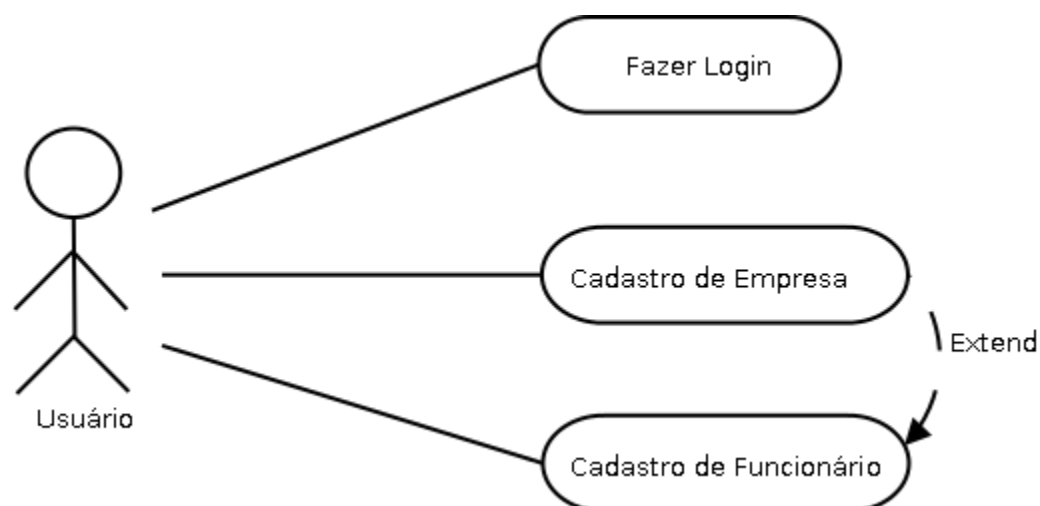


Figura 12 Caso de Uso para o perfil Usuário

## 2.8 Descrição dos Casos de Uso para a versão futura do Software

As descrições destes Casos de Uso seguem adiante:

### Casos de Uso do Perfil usuário:

#### **ID: CSU-FAZER-LOGIN**

NOME: Fazer Login

OBJETIVO: Acessar o sistema para uso

ATOR: Usuário do aplicativo deste software

PRÉ-CONDIÇÕES: Software iniciado e usuário deslogado neste software

PÓS-CONDIÇÕES: Usuário logado no software, com sistema liberado para uso

FLUXO PRINCIPAL:

- 1 - Ator informa o dado de username.
  - a) O dado de username aparece na tela do app.
- 2 - Ator informa o dado de senha.
  - a) O dado de senha aparece criptografado na tela do app.
- 3 - Ator aciona o botão de acesso ao sistema.
  - a) O software valida os dados inseridos.
  - b) O software exibe que os dados foram validados com sucesso.
  - c) O acesso ao sistema é liberado.

FLUXO ALTERNATIVO:

FA1 – 2a e 1a podem ser trocados de ordem.

FLUXO DE EXCEÇÃO:

FE1 – O ator informou credenciais inválidas. Neste caso o sistema solicita novamente os dados de login e senha do usuário, de modo a voltar para o início dos passos 1 ou 2 do Fluxo Principal, conforme for o caso.

#### **ID: CSU-CADASTRO-FUNCIONÁRIO**

NOME: Cadastro de Funcionário

OBJETIVO: Criação de perfil para um funcionário, dos três tipos possíveis

ATOR: Usuário do aplicativo

PRÉ-CONDIÇÕES: Software iniciado, e perfil não criado ainda

PÓS-CONDIÇÕES: Perfil criado de funcionário

FLUXO PRINCIPAL:

- 1 - O ator seleciona a opção para registrar
  - a) O software apresenta a tela de registro de novo perfil
- 2 - O ator preenche o formulário de cadastro de perfil, após escolher o tipo de perfil apropriado
- 3 - O ator aciona o botão de registro de novo perfil
  - a) O software salva o novo perfil
  - b) O software apresenta a mensagem de confirmação
  - c) O acesso ao sistema é liberado

FLUXO ALTERNATIVO:

FA1 – Se o perfil for do tipo Estratégico, então serão apresentadas as entradas pré-cadastradas de nome da empresa, e tipo da empresa, para que o usuário escolha alguma se quiser, mais um botão para realizar o cadastro de uma nova empresa (Outro caso de uso). Senão, estas entradas serão escondidas.

#### FLUXO DE EXCEÇÃO:

FE1 – No passo 2 do Fluxo Principal, o ator deixa algum dos campos em branco, e aciona o botão de efetuação do registro de novo perfil. Neste caso, o software apresenta a mensagem “Todos os campos devem ser preenchidos”, e não realiza o salvamento deste novo perfil. Obs.: no software feito em Laboratório de Engenharia de Software, aparece outra mensagem, o que representa um bug daquele software.

### **ID: CSU-CADASTRO-EMPRESA**

NOME: Cadastro de Empresa

OBJETIVO: Criação de empresa para perfil estratégico de um funcionário

ATOR: Usuário do aplicativo que pretenda ter perfil estratégico

PRÉ-CONDIÇÕES: Software foi iniciado, e o perfil estratégico está em processo de criação

PÓS-CONDIÇÕES: Perfil Estratégico criado de funcionário, e a empresa deste perfil foi criada.

#### FLUXO PRINCIPAL:

- 1 - O ator está na tela de criação de perfil, e seleciona a opção de perfil Estratégico
  - a) O software apresenta a opção de criação de empresa
- 2 - O ator seleciona a opção de criação de empresa
  - a) O software redireciona para a tela de criação de empresa
- 3 - O ator preenche o dado novo de nome da empresa
- 4 - O ator aciona o botão de registro de nova empresa
  - a) O software salva o dado novo da empresa
  - b) O software apresenta a mensagem de confirmação
  - c) O software redireciona para a tela anterior de criação de perfil

#### FLUXO ALTERNATIVO:

Não há fluxo alternativo.

#### FLUXO DE EXCEÇÃO:

FE1 – No passo 3 do Fluxo Principal, o ator deixa algum dos campos em branco, e aciona o botão de efetuação do registro de nova empresa. Neste caso, o software apresenta a mensagem “Todos os campos devem ser preenchidos”, e não realiza o salvamento desta nova empresa.

### **Casos de Uso do Perfil Funcionário:**

### **ID: CSU-VISUALIZAR-ARVORE-FUNCIONÁRIO**

NOME: Visualizar árvore do funcionário

OBJETIVO: Visualizar a árvore formada do funcionário que se logou no sistema com o perfil Funcionário

ATOR: Usuário que se logará no app com o Perfil Funcionário

PRÉ-CONDIÇÕES: Usuário não está logado no sistema, ainda. O software foi iniciado.

PÓS-CONDIÇÕES: A tela da árvore de perfil de funcionário que já está logado, aparece.

#### FLUXO PRINCIPAL:

- 1 - O usuário loga-se como Perfil Funcionário no sistema, com sucesso.

- a) O software realiza a operação e mostra a tela com a árvore deste funcionário.

**FLUXO ALTERNATIVO:**

FA1 – O ator já está logado no sistema, mas está em outra aba. Neste caso, ele muda de aba, e seleciona a aba “COMPETÊNCIAS” para visualizar a sua árvore de conhecimento.

**FLUXO DE EXCEÇÃO:**

FE1 – Se o usuário não fornecer as suas credenciais corretamente, ele não irá para a tela que exibe a sua árvore de conhecimento.

**ID: CSU-REQUISITAR-COMPETENCIA**

**NOME:** Enviar requisição de nova Competência

**OBJETIVO:** Requisitar uma Competência, para o perfil Funcionário

**ATOR:** Usuário que se logará no app com o Perfil Funcionário

**PRÉ-CONDIÇÕES:** O software foi iniciado e o usuário com perfil Funcionário já está logado no sistema. A Requisição não foi feita, ainda.

**PÓS-CONDIÇÕES:** A Requisição de nova Competência já foi feita.

**FLUXO PRINCIPAL:**

- 1 - O ator aciona o botão de requisição de nova competência na Árvore da Empresa.
  - a) O Software redireciona para uma nova tela com os dados a serem preenchidos.
- 2 - O ator preenche os dados solicitados.
- 3 - O ator clica no botão “REQUISITAR COMPETÊNCIA NA ÁRVORE”.
  - a) O software realiza a adição da Requisição.
  - b) O software mostra uma mensagem de confirmação.
  - c) O software reinicia a tela para permitir a realização de novas adições de Requisições.

**FLUXO ALTERNATIVO:**

FA1 – O ator pode realizar uma quantidade arbitrária de vezes os passos 2 e 3, nesta ordem, desde que realize todos eles.

**FLUXO DE EXCEÇÃO:**

FE1 – Se o ator não realizar o passo 2, e for do passo 1 direto para o passo 3, o software reclamará por causa de dados não preenchidos.

**ID: CSU- CADASTRAR-NOVA-CERTIFICAÇÃO**

**NOME:** Cadastrar nova Certificação

**OBJETIVO:** Cadastrar uma nova Certificação, para o perfil Funcionário

**ATOR:** Usuário que se logará no app com o Perfil Funcionário

**PRÉ-CONDIÇÕES:** O software foi iniciado e o usuário com perfil Funcionário já está logado no sistema. A Certificação não foi cadastrada, ainda.

**PÓS-CONDIÇÕES:** A Certificação já foi cadastrada.

**FLUXO PRINCIPAL:**

- 1 - O ator aciona a opção de cadastro de nova Certificação.
  - a) O software mostra a tela de cadastro de uma nova Certificação.
- 2 - O ator preenche os dados da nova Certificação.
  - a) O software mostra os dados preenchidos da nova Certificação.
  - b)
- 3 - O ator clica no botão “CADASTRAR CERTIFICAÇÃO”.

- a) O software realiza o cadastro da nova Certificação e reinicia a tela para permitir a realização de novas adições de Certificações.

**FLUXO ALTERNATIVO:**

FA1 – O ator pode realizar quantas vezes quiser os passos 2 e 3, desde que realize todos eles.

**FLUXO DE EXCEÇÃO:**

FE1 – Se o ator não realizar o passo 2, e pular do passo 1 para o passo 3, então o software reclamará por causa de dados não preenchidos.

**Casos de Uso do Perfil Operacional:**

**ID: CSU-VISUALIZAR-ARVORE-FUNCIONÁRIO-OPER**

**NOME:** Visualizar árvore do funcionário

**OBJETIVO:** Visualizar a árvore formada do funcionário que se logou no sistema com perfil Operacional

**ATOR:** Usuário que se logará no app com o Perfil Operacional

**PRÉ-CONDIÇÕES:** Usuário não está logado no sistema, ainda. O software foi iniciado.

**PÓS-CONDIÇÕES:** A tela da árvore de funcionário com Perfil Operacional que já está logado, aparece

**FLUXO PRINCIPAL:**

1 - O usuário loga-se como Perfil Operacional no sistema, com sucesso.

- a) O software realiza a operação e mostra a tela com a árvore deste funcionário.

**FLUXO ALTERNATIVO:**

FA1 – O ator já está logado no sistema, mas está em outra aba. Neste caso, ele muda de aba, e seleciona a aba “COMPETÊNCIAS” para visualizar a sua árvore de conhecimento.

**FLUXO DE EXCEÇÃO:**

FE1 – Se o usuário não fornecer as suas credenciais corretamente, ele não irá para a tela que exibe a sua árvore de conhecimento.

**ID: CSU-MUDAR-FUNCIONÁRIO-NIVEL-OPER**

**NOME:** Mudar funcionário de nível

**OBJETIVO:** Mudar um funcionário de nível, trocando o tipo de perfil dele

**ATOR:** Usuário que logou-se no sistema com Perfil Operacional

**PRÉ-CONDIÇÕES:** O ator não está logado no sistema, e o funcionário não teve o seu perfil alterado, ainda

**PÓS-CONDIÇÕES:** O ator já está logado no sistema, e o funcionário já teve o seu perfil alterado para o tipo desejado

**FLUXO PRINCIPAL:**

1 - O ator acessa o sistema, e loga-se

2 - O ator acessa a opção “MUDANÇA DE NÍVEL”

3 - O ator seleciona o nível de que deseja visualizar

4 - O ator seleciona o funcionário

5 - O ator seleciona o novo nível

6 - O ator clica no botão “MUDAR NÍVEL DE FUNCIONÁRIO”



- a) O sistema mostra a mensagem “Funcionário teve o nível de perfil alterado com sucesso!”

**FLUXO ALTERNATIVO:**

FA1 – O ator pode pular o passo 3, se ele já vir o nível desejado por ele.

FA2 – O ator pode pular o passo 4, se ele já vir o funcionário desejado por ele.

FA3 – O ator pode pular o passo 5, se ele já vir o novo nível desejado por ele.

FA4 – O ator pode fazer os passos 3, 4 e 5, em qualquer ordem, e não necessariamente na ordem sequencial pré-estabelecida no fluxo principal, desde que o passo 3 ocorra antes do passo 4.

**FLUXO DE EXCEÇÃO:**

Não há fluxo de exceção.

**ID: CSU-SOLICITAR-COMPETENCIA-SEM-CERTIFICADO-OPER**

**NOME:** Enviar solicitação de nova Competência sem Certificação

**OBJETIVO:** Solicitar uma Competência sem a necessidade de uma Certificação, para o perfil Operacional

**ATOR:** Usuário que se logará no app com o Perfil Operacional

**PRÉ-CONDIÇÕES:** O software foi iniciado e o usuário com perfil Operacional já está logado no sistema. A Competência sem Certificado não foi solicitada, ainda.

**PÓS-CONDIÇÕES:** A Competência sem Certificado já foi solicitada.

**FLUXO PRINCIPAL:**

4 - O ator seleciona o nível da Competência mãe na Árvore da Empresa.

5 - O ator seleciona a Competência mãe dentre as disponíveis para aquele nível selecionado no passo anterior.

6 - O ator preenche o nome da nova Competência filha.

7 - O ator clica no botão “ADICIONAR COMPETÊNCIA NA ÁRVORE”.

- a) O software realiza a adição da Competência e reinicia a tela para permitir a realização de novas adições de Competências.

**FLUXO ALTERNATIVO:**

FA1 – O ator pode realizar o passo 3 antes dos passos 1 e 2. E ainda assim, ao realizar o passo 4, o software responderá sempre do mesmo jeito.

**FLUXO DE EXCEÇÃO:**

FE1 – Se o ator não realizar o passo 3, o software reclamará por causa de dados não preenchidos.

**ID: CSU- SOLICITAR-COMPETENCIA-COM-CERTIFICADO-OPER**

**NOME:** Enviar solicitação de nova Competência com Certificação

**OBJETIVO:** Solicitar uma Competência que exija uma Certificação, para o perfil Operacional

**ATOR:** Usuário que se logará no app com o Perfil Funcionário

**PRÉ-CONDIÇÕES:** O software foi iniciado e o usuário com perfil Operacional já está logado no sistema. A Competência com Certificado não foi solicitada, ainda.

**PÓS-CONDIÇÕES:** A Competência com Certificado já foi solicitada.

**FLUXO PRINCIPAL:**

4 - O ator preenche o nome da nova Competência.

- a) O software mostra o nome preenchido da nova Competência.

5 - O ator preenche o nome da Certificação usada.

- a) O software mostra o nome preenchido da Certificação usada.

6 - O ator preenche a data da prova.

- a) O software mostra a data preenchida da prova.

7 - O ator clica no botão “ADICIONAR”.

- a) O software realiza a adição da Competência e reinicia a tela para permitir a realização de novas adições de Competências.

**FLUXO ALTERNATIVO:**

FA1 – O ator pode realizar os passos 1, 2 e 3 em qualquer ordem, desde que realize todos eles, e o software responderá sempre do mesmo jeito.

**FLUXO DE EXCEÇÃO:**

FE1 – Se o ator não realizar um dos passos 1, 2 e 3, e realizar o passo 4, então o software reclamará por causa de dados não preenchidos.

**ID: CSU- APROVAR-REPROVAR-NOVA-COMPETENCIA**

**NOME:** Enviar solicitação de nova Competência com Certificação

**OBJETIVO:** Solicitar uma Competência que exija uma Certificação, para o perfil Operacional

**ATOR:** Usuário que se logará no app com o Perfil Funcionário

**PRÉ-CONDIÇÕES:** O software foi iniciado e o usuário com perfil Operacional já está logado no sistema. A Competência com Certificado não foi solicitada, ainda.

**PÓS-CONDIÇÕES:** A Competência com Certificado já foi solicitada.

**FLUXO PRINCIPAL:**

1 - O ator preenche o nome da nova Competência.

- a) O software mostra o nome preenchido da nova Competência.

2 - O ator preenche o nome da Certificação usada.

- a) O software mostra o nome preenchido da Certificação usada.

3 - O ator preenche a data da prova.

- a) O software mostra a data preenchida da prova.

4 - O ator clica no botão “ADICIONAR”.

- a) O software realiza a adição da Competência e reinicia a tela para permitir a realização de novas adições de Competências.

**FLUXO ALTERNATIVO:**

FA1 – O ator pode realizar os passos 1, 2 e 3 em qualquer ordem, desde que realize todos eles, e o software responderá sempre do mesmo jeito.

**FLUXO DE EXCEÇÃO:**

FE1 – Se o ator não realizar um dos passos 1, 2 e 3, e realizar o passo 4, então o software reclamará por causa de dados não preenchidos.

## **2.9 Sobre a versão implementada do Software para esta Monografia**

A versão implementada do Software para esta monografia encontra-se no endereço do repositório indicado nas referências bibliográficas desta monografia. Esta versão começa como uma versão melhorada da versão deste software entregue em LESW, e eventualmente, pode ser reimplementada como uma versão de Software baseada na Análise feita nesta monografia.

## **2.10 Especificação da API REST para a versão futura do Software**

A API REST proposta para a versão do Software para esta monografia é definida pelos seguintes recursos:

- Gerência de Usuários e de Empresas;
- Gerência de Certificações e de Requisições;
- Gerência de Competências.

Esta API serve para sincronização automática e backup de dados locais geridos em cada dispositivo móvel relacionado com uma determinada empresa que use este software.

### **2.11 Especificação da API REST: Gerência de Usuários e de Empresas**

A Gerência de Usuários e de Empresas nesta API é feita com:

### **2.12 Especificação da API REST: Gerência de Certificações e de Requisições**

A Gerência de Certificações e de Requisições nesta API é feita com:

### **2.13 Especificação da API REST: Gerência de Competências**

A Gerência de Competências nesta API é feita com:

### 3. CONCLUSÃO

Concluiu-se que a versão futura deste Software foi inicialmente analisada, e esta análise está pronta para ser desenvolvida com suporte à gerência distribuída de dados via API REST. Provavelmente, a análise desta versão pode ser melhorada, por via de mudança de escopo de Software, ou por adição de Requisitos Funcionais, por exemplo.

Nota-se que esta análise não trata dos casos de uso envolvendo edição ou exclusão de dados e de Entidades, mas isto pode ser feito em uma monografia futura sobre este trabalho. Esta análise trata somente dos casos de cadastro de dados. Também, esta análise não trata de diplomas como substitutos para certificações, o que seria válido para uma Árvore de Conhecimentos com um nó pai de Engenharia, e nós filhos como: Engenharia Elétrica, Engenharia Mecânica, etc. Esta análise trata somente de funcionários com regime CLT..., pois os consultores PJ não fazem parte da Árvore de Conhecimentos de uma empresa, assim como, todos os outros prestadores de serviço PJ desta empresa.

Segundo Novicki (2018), surgirão apps que rodem em smartphones como em computadores Desktop, graças à tecnologia de PWAs, que permite que estes apps rodem nos browsers. E neste caso, supondo que o app desta monografia fosse feito como um PWA, então os requisitos não funcionais poderiam mudar para acomodar toda esta gama de dispositivos de Tecnologia da Informação, e certamente este app poderia ser feito em JavaScript ou JQuery...

## REFERÊNCIAS

BATISTA, E. DE O. **Sistemas de informação: o uso consciente da tecnologia para o gerenciamento.**, 2. Edição. São Paulo: Saraiva, 2012. 282 p.

BEZERRA, E. **Princípios de Análise e Projeto de Sistema com UML.**, 3. Edição. São Paulo: Elsevier Brasil. 2016. 416 p.

DAMASCENO, J. **Desenvolvimento Android – Aprenda a criar 15 apps.** Disponível em: <<https://www.udemy.com/curso-completo-do-desenvolvedor-android/>>. Acesso em: 04 mar. 2019.

DOCUMENTAÇÃO DA ROOM. **Documentação Online da Room.** Disponível em: <<https://developer.android.com/topic/libraries/architecture/room>>. Acesso em: 22 abr. 2019

DOCUMENTAÇÃO EXTRA DESTA MONOGRAFIA NO GITHUB. **Documentação extra desta monografia no GitHub.** Disponível em: <[https://github.com/dsfb/TCC\\_Docs\\_FatecSP\\_1sem\\_2019](https://github.com/dsfb/TCC_Docs_FatecSP_1sem_2019)>. Acesso em: 28 abr. 2019

DOCUMENTAÇÃO DO ANDROID. **Documentação Online do Android.** Disponível em: <<https://developer.android.com/docs/>>. Acesso em: 11 nov. 2018

DUARTE, L. **Tutorial app Android com Firebase – Realtime Database.**, 2018. Disponível em: <<https://www.luiztools.com.br/post/tutorial-app-android-com-firebase-realtime-database/>>. Acesso em: 04 mar. 2019.

FERREIRA, R. **REST: Princípios e boas práticas.**, 2017. Disponível em: <<https://blog.caelum.com.br/rest-principios-e-boas-praticas/>>. Acesso em: 22 abr. 2019

GOMES, A. F. **Uma rápida Introdução ao REST (Tradução).**, 2008. Disponível em: <<https://www.infoq.com/br/articles/rest-introduction>>. Acesso em: 22 abr. 2019

JAMIN, B. **Reasons Not To Use Firebase.**, 2016. Disponível em: <<https://crisp.chat/blog/why-you-should-never-use-firebase-realtime-database/>>. Acesso em: 22 abr. 2019

JÚNIOR, R. M. S. **Análise e Projeto orientado a objetos usando UML e o Processo Unificado.**, 2001. 113 p. Trabalho de Conclusão de Curso (Ciência de Computação) – Centro de Ciências Exatas e Naturais, Universidade Federal do Pará, Belém, 2001.

KLIMKIN, D. **Access Back4App mongodb.**, 2017. Disponível em: <<https://stackoverflow.com/questions/43024483/access-back4app-mongodb>>. Acesso em: 28 abr. 2019

LEVY, P. **As árvores de conhecimentos.**, 3. reimpressão. São Paulo: Editora Escuta. 2008. 188 p.

MOJICA, G. **How to Migrate from Parse to Self-hosted MongoDB**, 2016. Disponível em: <<https://www.appcoda.com/parse-server-migration/>>. Acesso em: 28 abr. 2019

NOVICKI, D. **Goodbye Electron, Hello Desktop PWAs**, 2018. Disponível em: <<https://medium.com/dailyjs/goodbye-electron-hello-desktop-pwas-f316b8f39882>>. Acesso em: 29 mai. 2019

REDAÇÃO. **O que é API?** Disponível em: <<https://canaltech.com.br/software/o-que-e-api/>>. Acesso em: 22 abr. 2019

REPOSITÓRIO DO SOFTWARE DESTA MONOGRAFIA NO GITHUB. **Repositório do Software desta monografia no GitHub**. Disponível em: <[https://github.com/dsfb/TCC\\_App\\_FatecSP\\_1sem\\_2019](https://github.com/dsfb/TCC_App_FatecSP_1sem_2019)>. Acesso em: 30 abr. 2019

RIBEIRO, I. **Parse Server: The Best Practices Guide.**, 2017. Disponível em: <<https://blog.back4app.com/2017/11/09/parse-server-best-practices/>>. Acesso em: 22 abr. 2019