

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, foram feitas, nesta monografia, a análise e o eventual desenvolvimento parcial de uma nova versão de um software feito inicialmente com banco de dados local sobre este tema (isto é, sem o suporte à API REST), de modo que o software completo futuro desta nova versão funcione como um sistema distribuído via API REST, sendo acessível em vários smartphones distintos com o sistema operacional Android (da Google).

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

ABSTRACT

Due to the scarcity of open source free mobile software, available on the subject of Knowledge Trees, this monograph was the analysis and possible partial development of a new version of software made initially with a local database on this theme (ie, without REST API support), so that the future full software of this new version works as a distributed system via the REST API, being accessible in several different smartphones with the Android (Google) operating system.

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

SUMÁRIO

1. INTRODUÇÃO	7
1.1 FIREBASE E O FIREBASE REALTIME DATABASE	8
1.2 BACK4APP	8
1.3 APIS REST.....	9
1.4 ANÁLISE DE SISTEMAS.....	9
2. DESENVOLVIMENTO.....	10
2.1 REQUISITOS FUNCIONAIS DA VERSÃO ANTERIOR DO SOFTWARE.....	10
2.2 REQUISITOS NÃO FUNCIONAIS DA VERSÃO ANTERIOR DO SOFTWARE	12
2.3 MODELO ENTIDADE RELACIONAMENTO (MER) DA VERSÃO ANTERIOR DO SOFTWARE	12
2.4 REQUISITOS FUNCIONAIS PARA A VERSÃO FUTURA DO SOFTWARE	13
2.5 REQUISITOS NÃO FUNCIONAIS PARA A VERSÃO FUTURA DO SOFTWARE.....	14
2.6 PROPOSTA DE MODELO ENTIDADE RELACIONAMENTO (MER) PARA A VERSÃO FUTURA DO SOFTWARE	15
2.7 PROPOSTA DE CASOS DE USO PARA A VERSÃO FUTURA DO SOFTWARE.....	17
2.8 DESCRIÇÃO DOS CASOS DE USO PARA A VERSÃO FUTURA DO SOFTWARE.....	20
3. CONCLUSÃO	26

1. INTRODUÇÃO

Atualmente, não existem muitos softwares móveis livres e de código-fonte aberto, disponíveis sobre o tema de Árvores de Conhecimento. Assim, o Daniel S. F. Bruno (autor desta monografia) fez uma parte do desenvolvimento de um software móvel para o sistema operacional Android (feito em linguagem Java) que, usando um Banco de Dados gerido em memória RAM, servia para gerenciar uma árvore de conhecimento para somente uma empresa (na disciplina de Interação Humano Computador, IHC, com o professor Shigueo, no segundo semestre de 2017), assim como, também ele fez um novo desenvolvimento completo deste software móvel para o sistema operacional Android (feito também em linguagem Java) que, com o uso da biblioteca de Persistência Room da Google (versão 1.1.1), gerencia uma árvore de conhecimento para uma empresa que esteja localizada somente em um único endereço físico (na disciplina de Laboratório de Engenharia de Software, LESW, com a professora Sandra Harumi Tanaka, professora orientadora desta monografia, 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 (LESW) é melhor do que a inicial (IHC), por permitir o armazenamento permanente das mudanças feitas durante qualquer execução deste aplicativo, pois as mudanças feitas durante uma execução deste aplicativo na versão inicial não eram gravadas após esta execução, e eram perdidas do Banco de Dados deste aplicativo.

Baseado nas características fornecidas do contexto atual, o objetivo deste projeto de formatura será o de analisar e desenvolver parcialmente um software distribuído móvel livre e de código-fonte aberto, para gerência ubíqua de árvores de conhecimento, feito também em linguagem Java, que seja uma nova versão do software feito na disciplina de Laboratório de Engenharia de Software. Assim, uma arquitetura distribuída para este novo software será completamente analisada e parcialmente desenvolvida.

Realmente, a não existência de muitos softwares livres móveis para gerência de árvores de conhecimentos justifica a criação de uma arquitetura distribuída na nova versão deste software.

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

- Qual recurso de TI serve melhor para construir um software distribuído escalável: Firebase ou Back4App;
- Como uma versão futura do software feito na disciplina Laboratório de Engenharia de Software poderia ser analisada;
- Como a versão do software entregue na disciplina Laboratório de Engenharia de Software poderia ser melhorada, eventualmente.

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

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

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 haja poucos níveis aninhados nos objetos JSON do Banco de Dados da Firebase, assim como, que não haja uma normalização excessiva, de modo a espalhar demasiadamente os dados geridos.

Nota-se, segundo Jamin (2016), que a Firebase não é uma boa opção para casos específicos, como o do software 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 software, e sabe-se que este software exigiu uma razoável tarefa de desenvolvimento do Banco local de Dados com a biblioteca Room, e então, uma nova demanda de tempo é esperada para desenvolver esta nova gerência; Nota-se que haveria uma troca da biblioteca Room pela plataforma Firebase, neste caso;
- A Firebase baixa todas as sub-árvores de dados JSON sempre que um aplicativo móvel é iniciado, implicando que um dispositivo móvel baixaria os dados de todas as árvores de conhecimento armazenadas na nuvem, mesmo que este dispositivo fosse destinado a gerir somente uma árvore de conhecimento, que seria de somente uma única empresa; Supõe-se que cada empresa teria somente uma árvore de conhecimento neste aplicativo;
- A Firebase pode apresentar inconsistências em casos de operações sem internet, implicando em possíveis inconsistências nas árvores de conhecimento empresariais armazenadas na nuvem para este aplicativo;
- A Firebase é ruim para lidar com relações de chaves primárias, e o aplicativo desta monografia usa este tipo de relação, implicando em eventual código adicional do tipo macarrão;
- A Firebase é ruim para lidar com queries complexas no Banco de dados, implicando em código adicional 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, 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 futuro envolvendo o aplicativo desta monografia. Segundo Klimkin (2017), a Back4App usa o banco de dados MongoDB internamente, e este banco de dados 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 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) devem 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, e a correta utilização dos códigos HTTP.

1.4 Análise de Sistemas

Segundo a Wikipédia (2019), a Análise de Sistemas é uma atividade que estuda processos com o objetivo de escolher o melhor caminho racional para realizar-se o Processamento de Dados (e de Informações). Assim, a partir desta informação, nota-se que o profissional de TI que exerce esta atividade define o melhor algoritmo que um Software deveria ter, para resolver um problema de TI, que envolva a transformação ou o processamento dos dados e das Informações, assim como, ele também define a melhor arquitetura de dados e de projeto de Software para este problema.

Segundo Bezerra (2016), 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.

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

2. DESENVOLVIMENTO

Os requisitos da nova versão deste software serão elicitados mentalmente pelo analista de sistemas Daniel S. F. Bruno, que é o autor desta monografia.

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

#TODO: descrever brevemente e levemente sobre a especificação do software feito em LESW.

#TODO: reorganizar as subseções (2.1, 2.2, 2.3, etc.) de acordo com a lista criada a seguir!

#TODO: Corrigir os diagramas de casos de uso da versão futura do software.

As seguintes questões de Análise 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.

2.1 Requisitos Funcionais da versão anterior do Software

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

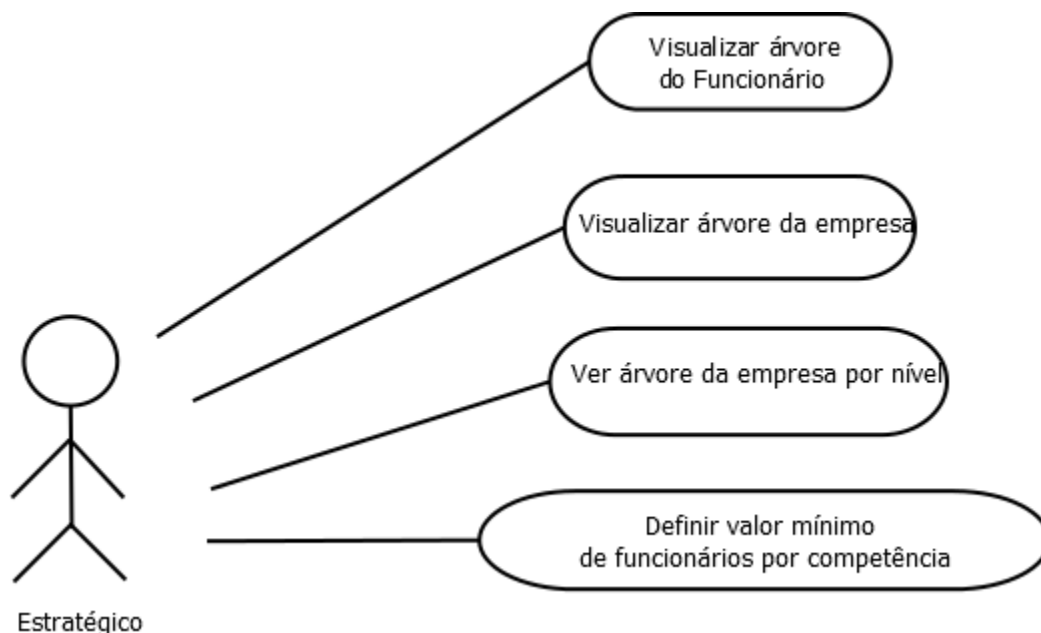


Figura 1 Casos de uso para o perfil Estratégico

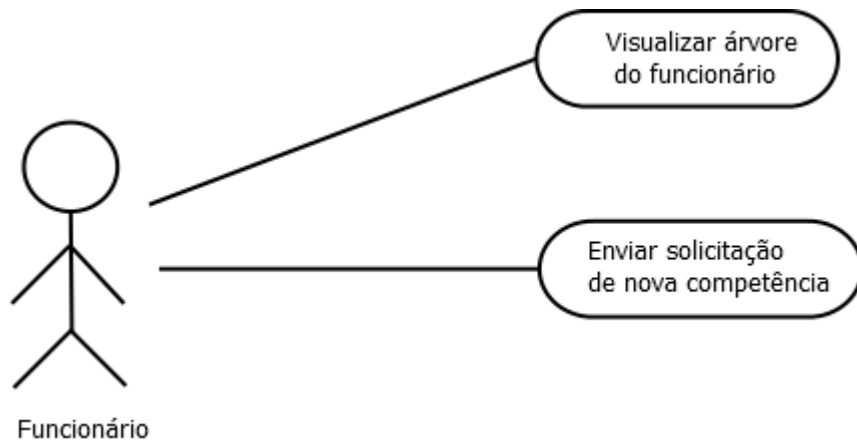


Figura 2 Casos de uso para o perfil Funcionário

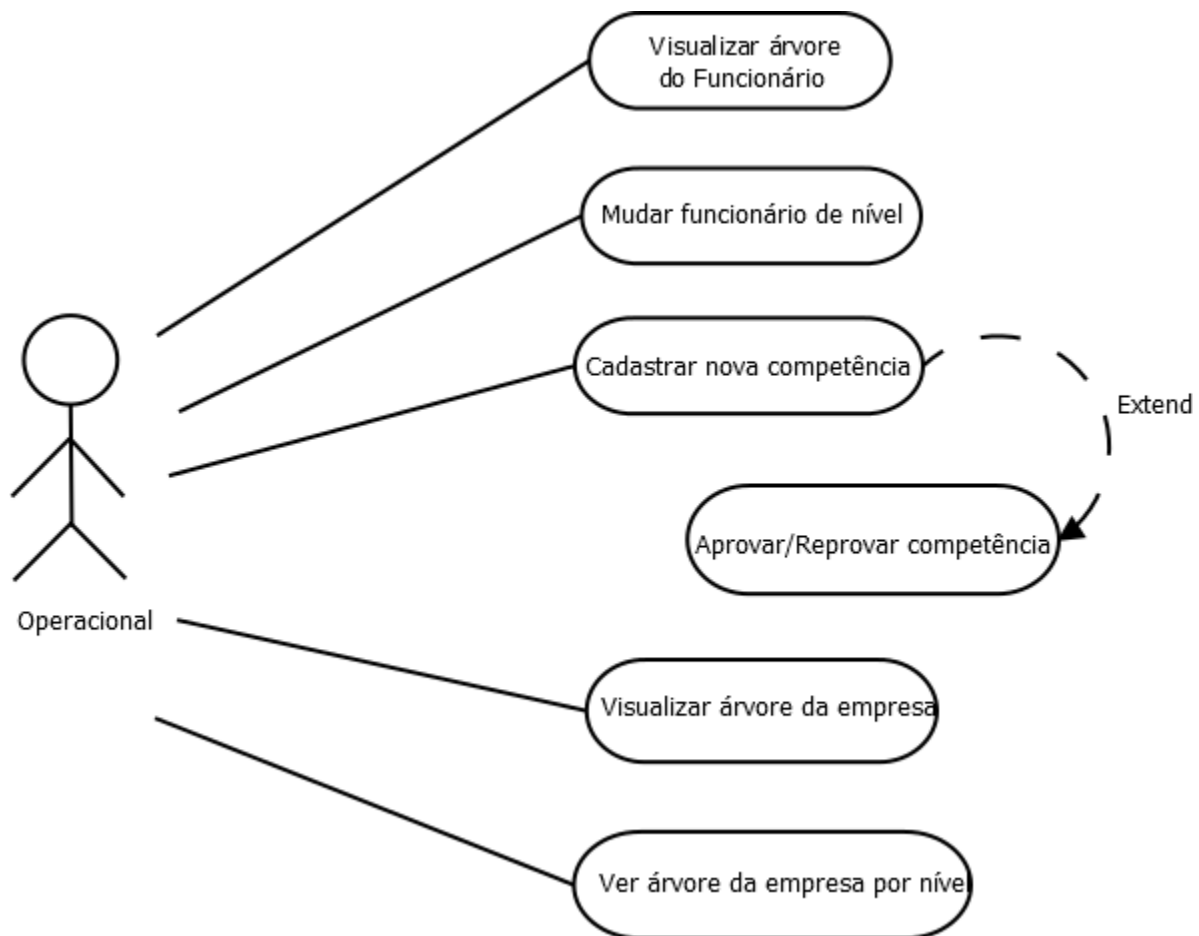


Figura 3 Casos de uso para o perfil Operacional

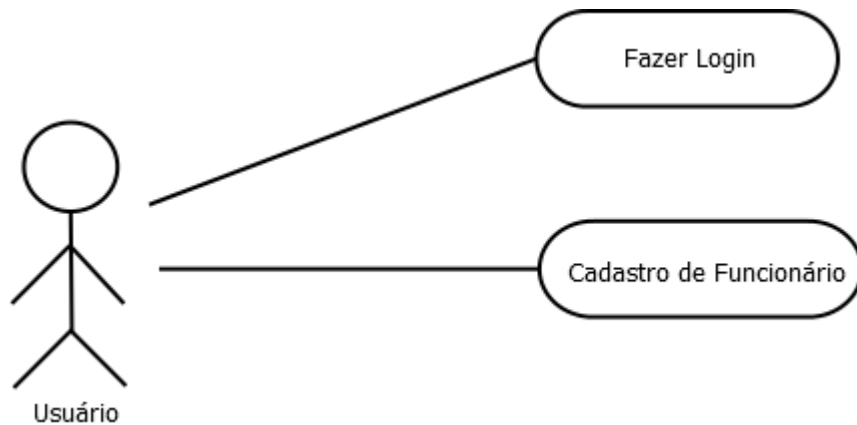


Figura 4 Casos de uso para o perfil Usuário

2.2 Requisitos Não Funcionais da versão anterior do Software

Este software antigo foi feito em Java, usando-se o Android Studio versão 3.2.1, e para smartphones com pelo menos a versão 4.2 JellyBean do Android. Qualquer informação era salva num arquivo compatível com o Banco de Dados SQLite, e era gerida pela biblioteca de Persistência Room, versão 1.1.1 da Google. Visava-se somente a custódia de competências formais, isto é, aquelas que fossem obtidas por meio de um certificado.

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

O MER deste software móvel antigo era:

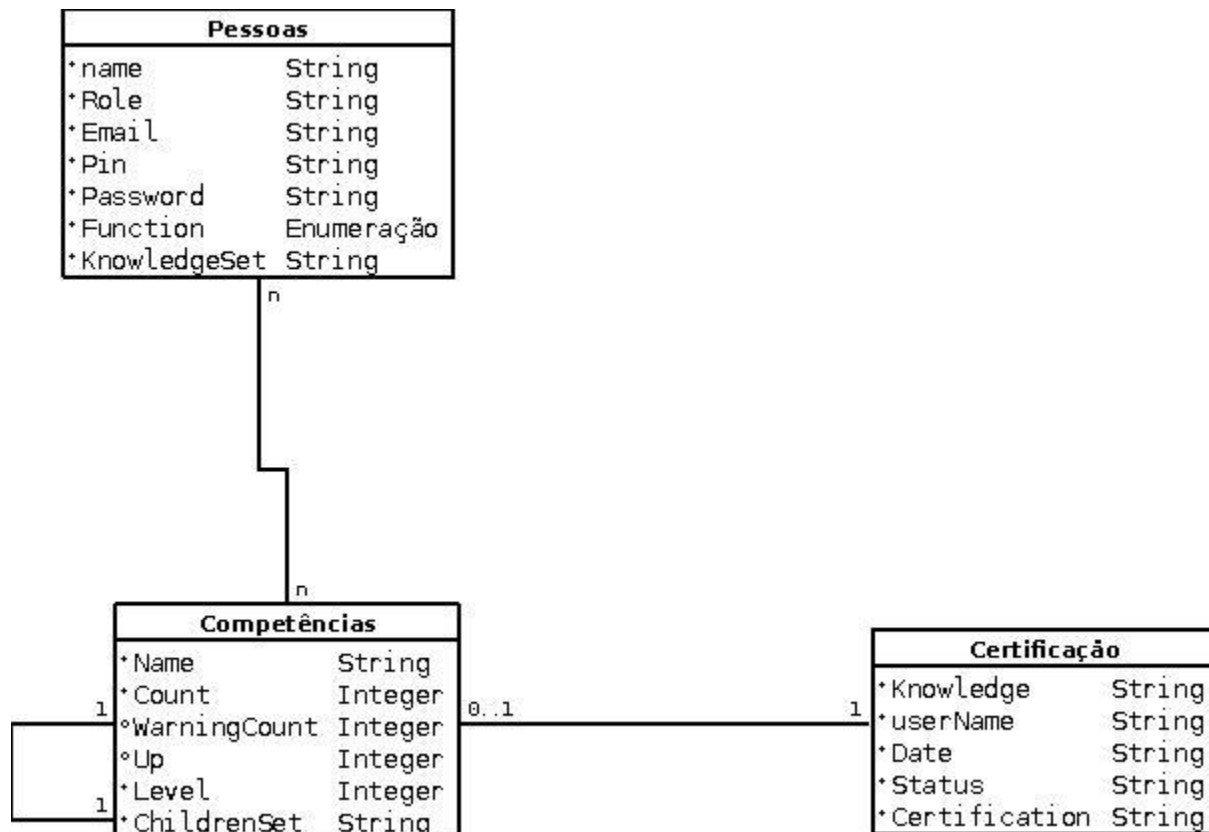


Figura 5 Modelo Entidade-Relacionamento do software antigo.

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 qualquer 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 com perfil Estratégico, durante o cadastro deste usuário.
4. O Software deve ser capaz de cadastrar uma Competência, 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 ser 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 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 que qualquer usuário com perfil Estratégico defina o valor mínimo de funcionários que possuam determinada competência na Árvore da 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

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

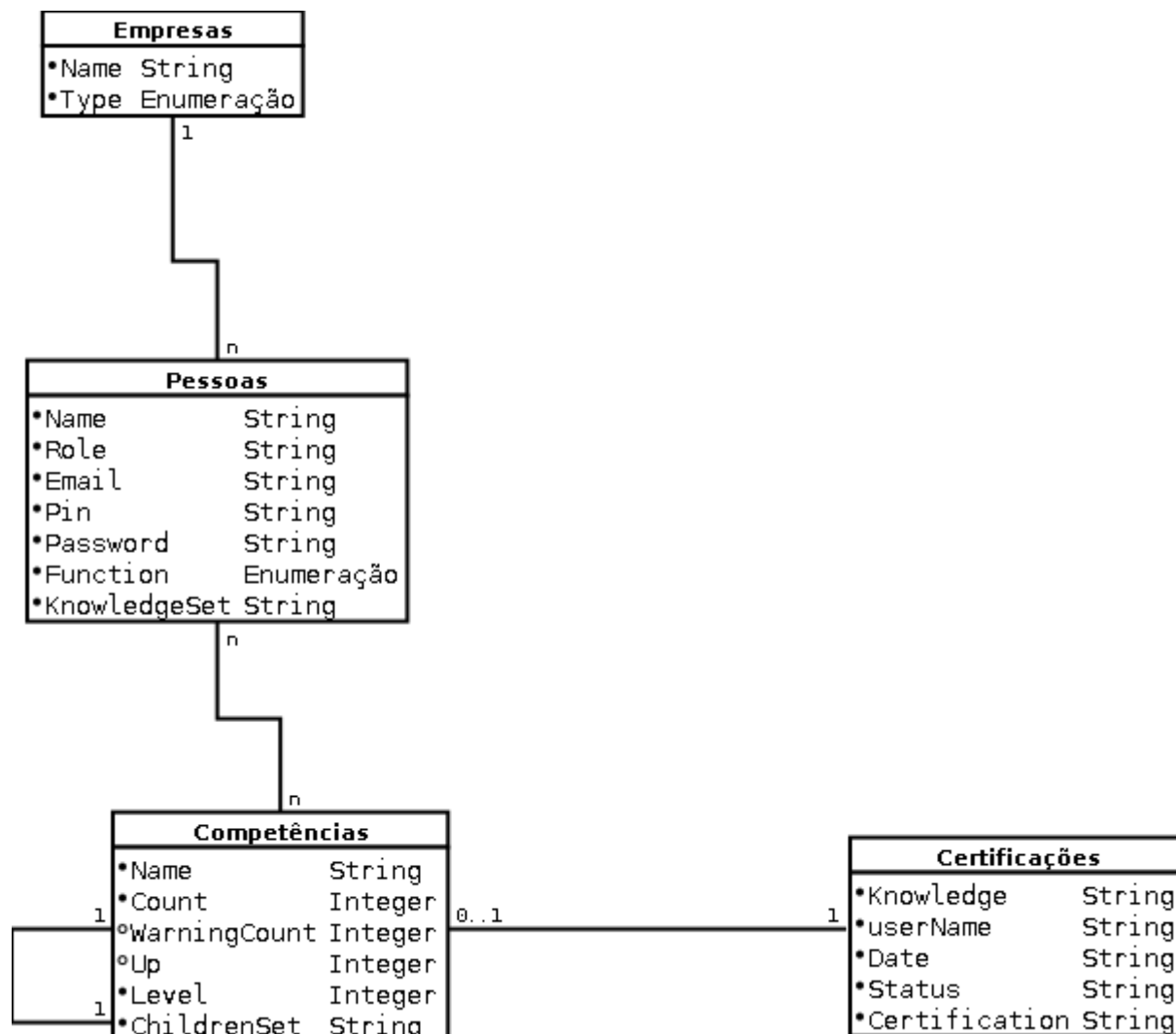


Figura 6 Primeira proposta do Modelo Entidade-Relacionamento do novo software

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 de 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:

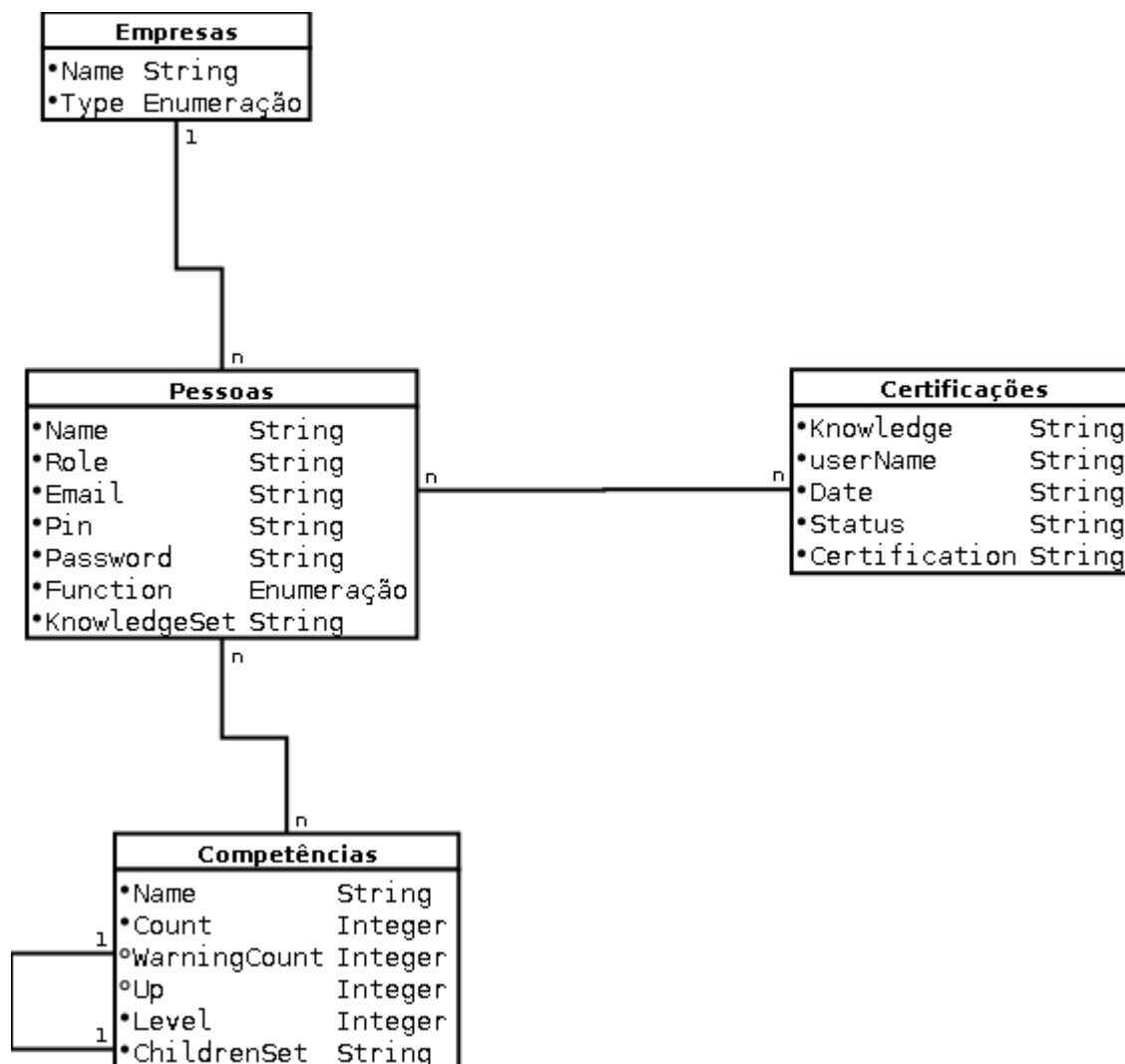


Figura 7 Nova proposta de MER para o novo Software

Pois nesta nova proposta de MER, os funcionários cadastram suas Certificações que os acompanham para quaisquer empresas que eles quiserem trabalhar, e pedem alguma competência para o departamento de RH, que pode analisar o currículo 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.

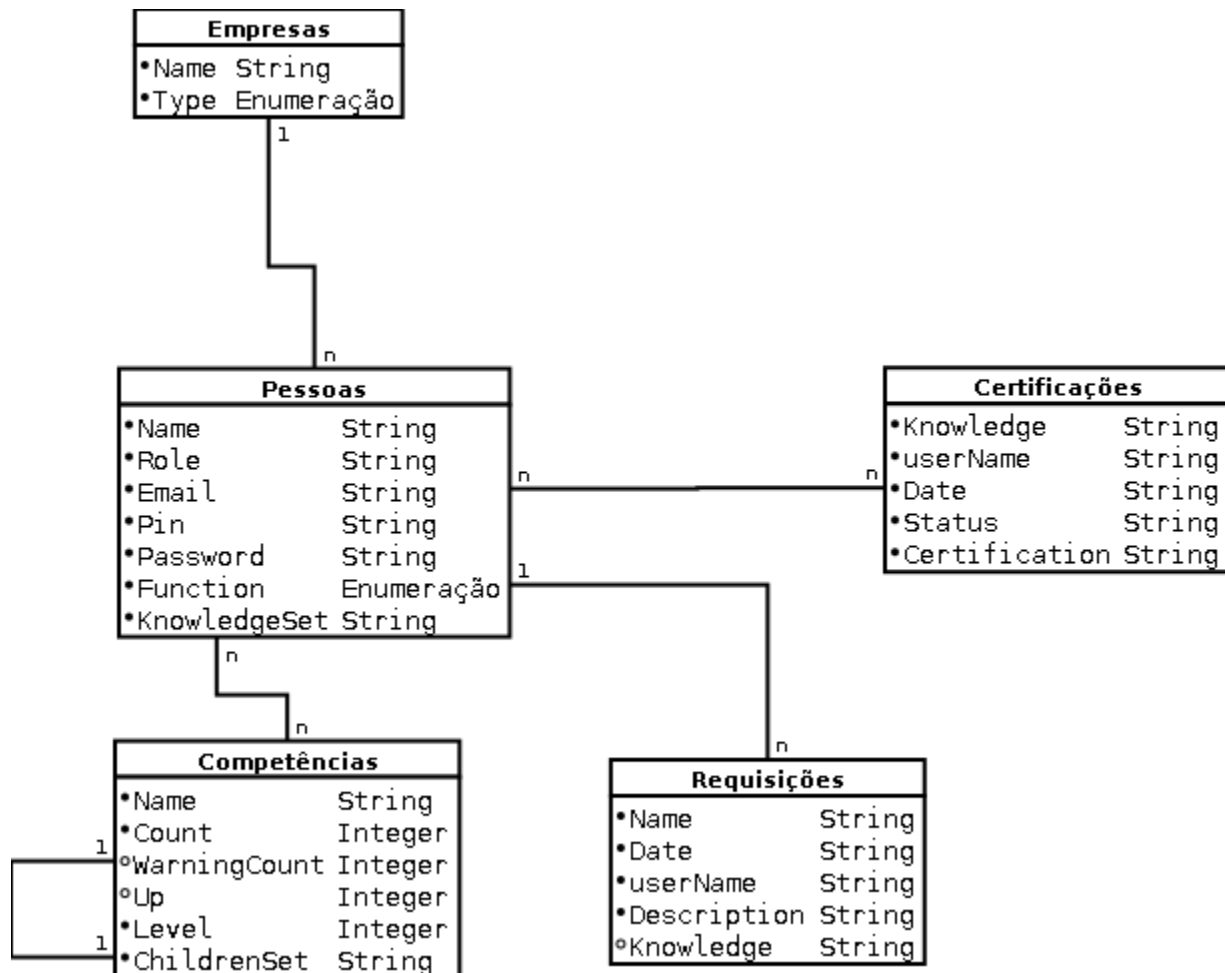


Figura 8 Proposta de MER com as Requisições, para o novo Software

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 BackEnd 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:

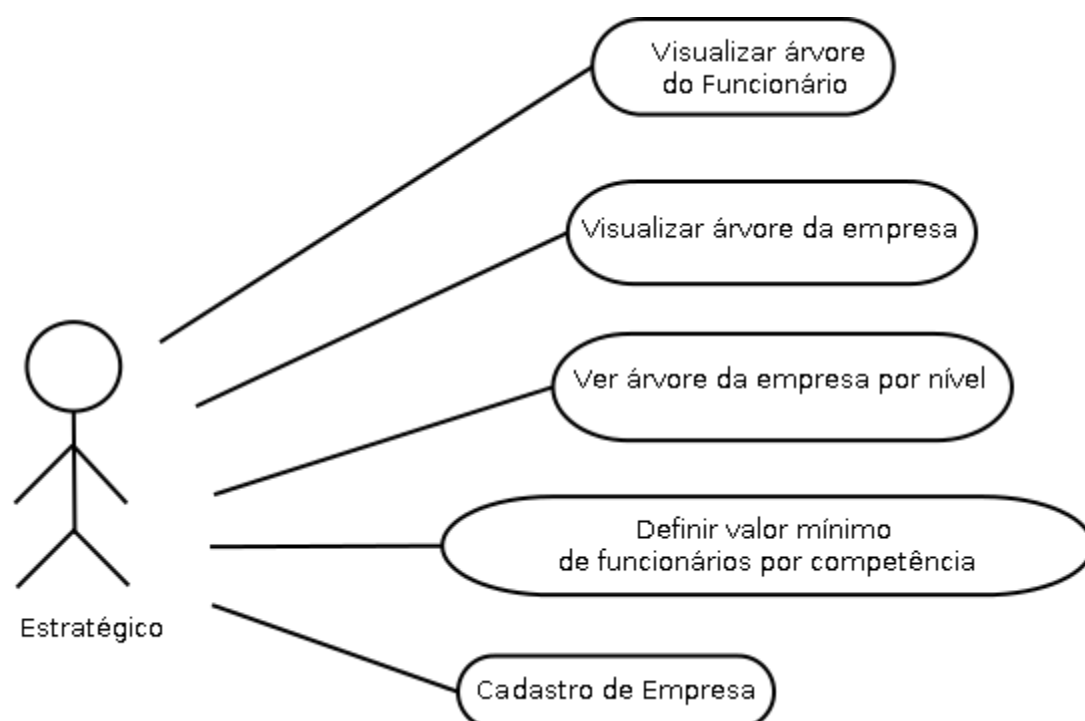


Figura 9 Caso de Uso para o perfil Estratégico

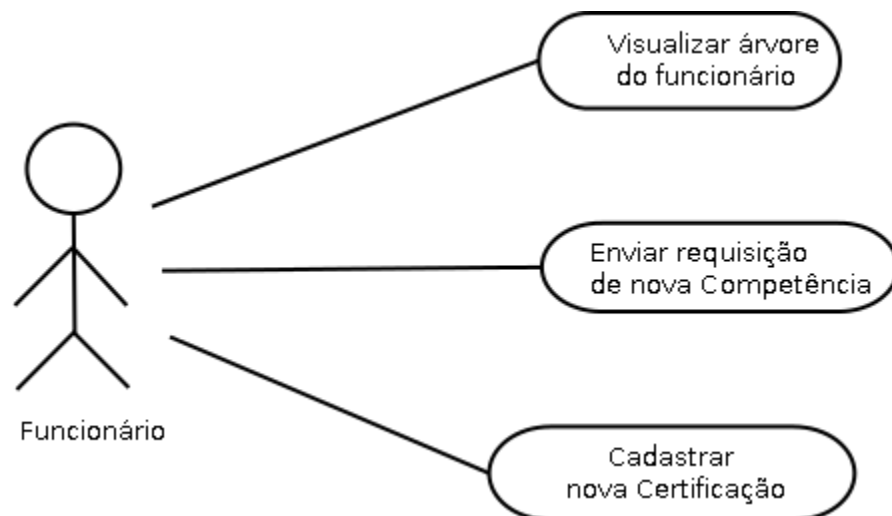


Figura 10 Caso de Uso para o perfil Funcionário

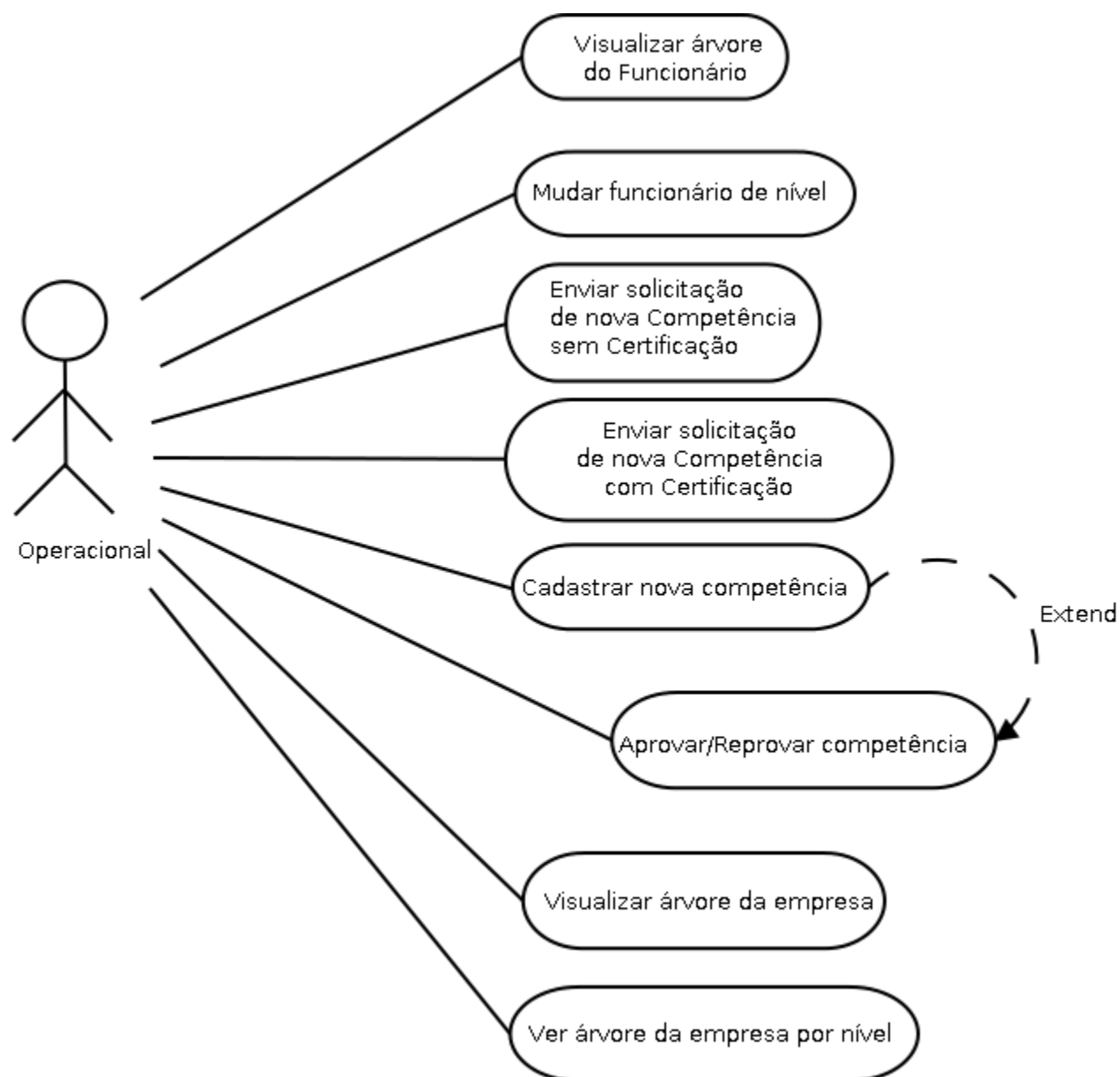


Figura 11 Caso de Uso para o perfil Operacional

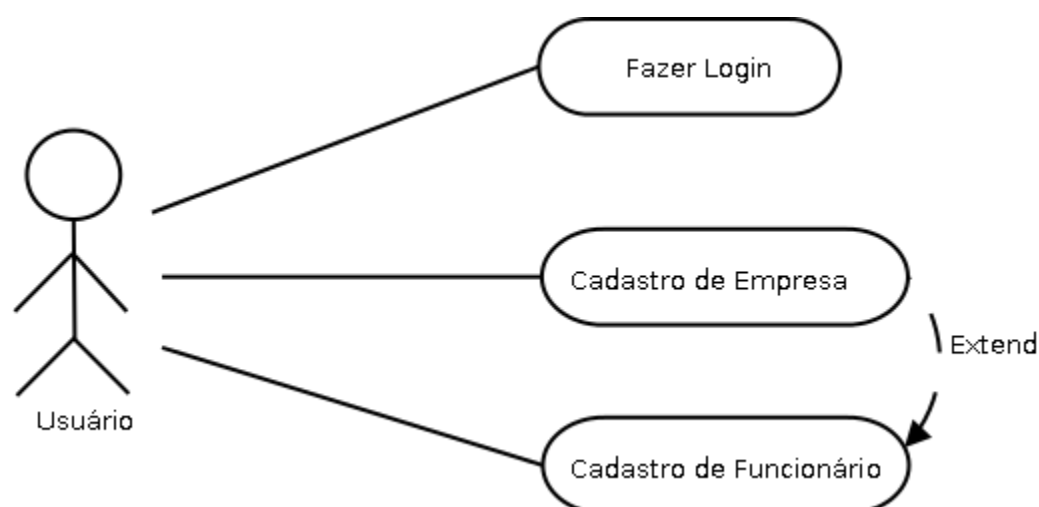


Figura 12 Caso de Uso para o perfil Usuário

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 Estratégico, então serão apresentadas as entradas de nome da empresa, e tipo da empresa, 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.

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-SOLICITAR-COMPETENCIA-SEM-CERTIFICADO

NOME: Enviar solicitação de nova Competência sem Certificação

OBJETIVO: Solicitar uma Competência sem a necessidade de uma 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 Competência sem Certificado não foi solicitada, ainda.

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

FLUXO PRINCIPAL:

- 1 - O ator seleciona o nível da Competência mãe na Árvore da Empresa.
- 2 - O ator seleciona a Competência mãe dentre as disponíveis para aquele nível selecionado no passo anterior.

- 3 - O ator preenche o nome da nova Competência filha.
- 4 - 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

NOME: Enviar solicitação de nova Competência com Certificação

OBJETIVO: Solicitar uma Competência que exija uma 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 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.

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 ver o nível desejado por ele.
FA2 – O ator pode pular o passo 4, se ele já ver o funcionário desejado por ele.
FA3 – O ator pode pular o passo 5, se ele já ver 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:

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

- 6 - O ator seleciona a Competência mãe dentre as disponíveis para aquele nível selecionado no passo anterior.
- 7 - O ator preenche o nome da nova Competência filha.
- 8 - 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:

- 5 - O ator preenche o nome da nova Competência.
 - a) O software mostra o nome preenchido da nova Competência.
- 6 - O ator preenche o nome da Certificação usada.
 - a) O software mostra o nome preenchido da Certificação usada.
- 7 - O ator preenche a data da prova.
 - a) O software mostra a data preenchida da prova.
- 8 - 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.

.

3. CONCLUSÃO

Concluiu-se que a versão futura deste Software foi inicialmente analisada e elaborada por completo, e está pronta para ser desenvolvida com suporte à gerência distribuída de dados. Embora, esta versão tenha sido implementada sem este suporte, eventualmente, nota-se que o desenvolvimento deste suporte pode ser feito em um Trabalho Futuro relacionado com esta Monografia. Provavelmente, a análise e a elaboração desta versão podem ser melhoradas, por via de mudança de escopo, ou por adição de requisitos funcionais, por exemplo.

REFERÊNCIAS

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>. 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

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

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

WIKIPEDIA. **Análise de Sistemas.**, 2019. Disponível em:
<https://pt.wikipedia.org/wiki/An%C3%A1lise_de_sistemas>. Acesso em: 22 abr. 2019