****

**CURSO SUPERIOR DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**PATRIK ZANOTTO GUERRA**

**MINHA EQUIPE**

**Caxias do Sul**

**2017**

**PATRIK ZANOTTO GUERRA**

**MINHA EQUPE**

**Trabalho apresentado para o Curso de Análise e Desenvolvimento de Sistemas, do Centro Universitário Uniftec como parte dos requisitos para avaliação da unidade curricular de TCC.**

**Orientador: Prof. Thiarlei Macedo**

**Caxias do Sul**

**2017**

RESUMO

O presente trabalho tem o objetivo de apresentar as especificações técnicas para o desenvolvimento de um aplicativo móvel denominado Minha Equipe, cujo propósito é ser uma ferramenta de apoio no gerenciamento de equipes afim de centralizar todas as informações relevantes em um único contexto. A proposta é possibilitar a criação de equipes na tarefas podem ser atribuídas e gerenciadas pelos membros, ainda, locais relevantes qual podem ser apontados em um mapa e estes serem vinculados a tarefas, servindo como um indicador de localização. Um chat entre a equipe será disponibilizado permitindo a troca de mensagens entre os membros. Por meio do aplicativo móvel, será coletada a localização de cada usuário e então disponibilizada para o restante dos membros da equipe de modo a possibilitar a rápida localização do membro. Nesse documento são apresentados os artefatos de análise do sistema, incluindo especificações de requisitos, protótipos de interface, diagramas de casos de uso, diagramas de classe, diagramas de sequência e diagramas de entidade-relacionamento. As tecnologias utilizadas para desenvolvimento foram Ionic Framework e Google Firebase.

**Palavras-chave:** gerenciamento de equipe. *Software*. Localização. Tarefas.

Lista de Tabelas

[Tabela 1 - Exemplo de Requisito Funcional 4](#_Toc497773071)

[Tabela 2 - Exemplo de documentação de Caso de Uso 7](#_Toc497773072)

[Tabela 3 - Modificadores de Acesso do Diagrama de Classes 8](#_Toc497773073)

[Tabela 4 - Multiplicidade do Diagrama de Classes 9](#_Toc497773074)

[Tabela 5 - Cardinalidades do Diagrama ER 12](#_Toc497773075)

[Tabela 6 - Exemplo de Tabela do Dicionário de Dados 13](#_Toc497773076)

[Tabela 7 - Requisito Funcional ‘Cadastro Inicial’ 16](#_Toc497773077)

[Tabela 8 - Requisito Funcional ‘Login’ 17](#_Toc497773078)

[Tabela 9 - Requisito Funcional ‘Recuperar Senha’ 17](#_Toc497773079)

[Tabela 10 - Requisito Funcional ‘Perfil’ 18](#_Toc497773080)

[Tabela 11 - Requisito Funcional ‘Alterar E-mail’ 19](#_Toc497773081)

[Tabela 12 - Requisito Funcional ‘Alterar Senha’ 20](#_Toc497773082)

[Tabela 13 - Requisito Funcional ‘Manter Equipe’ 20](#_Toc497773083)

[Tabela 14 - Requisito Funcional ‘Convidar’ 21](#_Toc497773084)

[Tabela 15 - Requisito Funcional ‘Gerenciar Membros’ 22](#_Toc497773085)

[Tabela 16 - Requisito Funcional ‘Mapa da Equipe’ 22](#_Toc497773086)

[Tabela 17 - Requisito Funcional ‘Localização do Membro’ 22](#_Toc497773087)

[Tabela 18 - Requisito Funcional ‘Chat’ 23](#_Toc497773088)

[Tabela 19 - Requisito Funcional ‘Manter Local’ 23](#_Toc497773089)

[Tabela 20 - Requisito Funcional ‘Manter Tarefa’ 24](#_Toc497773090)

[Tabela 21 - Mensagens da Aplicação 25](#_Toc497773091)

[Tabela 22 - Caso de uso ‘Login’ 34](#_Toc497773092)

[Tabela 23 - Caso de uso ‘Recuperar Senha’ 35](#_Toc497773093)

[Tabela 24 - Caso de uso ‘Cadastrar Usuário’ 36](#_Toc497773094)

[Tabela 25 - Caso de uso ‘Alterar Usuário’ 37](#_Toc497773095)

[Tabela 26 - Caso de uso ‘Alterar E-mail’ 37](#_Toc497773096)

[Tabela 27 - Caso de uso ‘Alterar Senha’ 38](#_Toc497773097)

[Tabela 28 - Caso de uso ‘Manter Equipe (Criar) ’ 39](#_Toc497773098)

[Tabela 29 - Caso de uso ‘Manter Equipe (Alterar) ’ **Erro! Indicador não definido.**](#_Toc497773099)

[Tabela 30 - Caso de Uso ‘Manter Equipe (Remover)’ 41](#_Toc497773100)

[Tabela 31 - Caso de uso ‘Remover Usuário’ 41](#_Toc497773101)

[Tabela 32 - Caso de uso ‘Sair da Equipe’ 42](#_Toc497773102)

[Tabela 33 - Caso de uso ‘Convidar Usuário’ 43](#_Toc497773103)

[Tabela 34 - Caso de uso ‘Aceitar Convite’ 43](#_Toc497773104)

[Tabela 35 - Caso de uso ‘Local do Membro’ 44](#_Toc497773105)

[Tabela 36 - Caso de uso ‘Chat’ 45](#_Toc497773106)

[Tabela 37 - Caso de uso ‘Manter Tarefa’ 46](#_Toc497773107)

[Tabela 38 - Caso de uso ‘Concluir Tarefa’ 47](#_Toc497773108)

[Tabela 39 - Caso de uso ‘Cancelar Tarefa’ 47](#_Toc497773109)

[Tabela 40 - Caso de uso ‘Localização do Local’ 48](#_Toc497773110)

[Tabela 41 - Caso de uso ‘Manter Local’ 49](#_Toc497773111)

[Tabela 42 - Dicionário de dados da tabela ‘Equipe’ 51](#_Toc497773112)

[Tabela 43 - Dicionário de dados da tabela ‘EquipeConvite 51](#_Toc497773113)

[Tabela 44 - Dicionário de dados da tabela ‘EquipePessoa’ 51](#_Toc497773114)

[Tabela 45 - Dicionário de dados da tabela ‘Local’ 51](#_Toc497773115)

[Tabela 46 - Dicionário de dados da tabela ‘LocalCordenada’ **Erro! Indicador não definido.**](#_Toc497773116)

[Tabela 47 - Dicionário de dados da tabela ‘Mensagem’ **Erro! Indicador não definido.**](#_Toc497773117)

[Tabela 48 - Dicionário de dados da tabela ‘Pessoa’ **Erro! Indicador não definido.**](#_Toc497773118)

[Tabela 49 - Dicionário de dados da tabela ‘PessoaLocalizacao’ **Erro! Indicador não definido.**](#_Toc497773119)

[Tabela 50 - Dicionário de dados da tabela ‘PessoaTag’ **Erro! Indicador não definido.**](#_Toc497773120)

[Tabela 51 - Dicionário de dados da tabela ‘Tag’ **Erro! Indicador não definido.**](#_Toc497773121)

[Tabela 52 - Dicionário de dados da tabela ‘Tarefa’ 51](#_Toc497773122)

[Tabela 53 - Dicionário de dados da tabela ‘TarefaUsuarioResponsavel’ **Erro! Indicador não definido.**](#_Toc497773123)

[Tabela 54 - Dicionário de dados da tabela ‘UsuarioAcesso’ 51](#_Toc497773124)

[Tabela 55 - Dicionário de dados da tabela ‘UsuarioAcessoRecuperarSenha’ **Erro! Indicador não definido.**](#_Toc497773125)

[Tabela 56 - Itens de validação 51](#_Toc497773126)

LISTA DE FIGURAS

[Figura 1 - Exemplo de protótipo 4](#_Toc497773127)

[Figura 2 - Exemplo de associação entre Ator e Caso de Uso 5](#_Toc497773128)

[Figura 3 - Exemplo de Generalização entre Casos de Uso 6](#_Toc497773129)

[Figura 4 - Exemplo de Generalização entre Atores 6](#_Toc497773130)

[Figura 5 - Exemplo de Include e Extend 6](#_Toc497773131)

[Figura 6 - Exemplo de Diagrama de Classes 9](#_Toc497773132)

[Figura 7 - Exemplo de Multiplicidade no Diagrama de Classes 10](#_Toc497773133)

[Figura 8 - Exemplo de implementação de Interface 10](#_Toc497773134)

[Figura 9 - Exemplo de Diagrama de Sequência de Logout 11](#_Toc497773135)

[Figura 10 - Exemplo de Diagrama ER 13](#_Toc497773136)

[Figura 11 - Protótipo de interface 14](#_Toc497773137)

[Figura 12 - Protótipo de 'Cadastro Inicial' 27](#_Toc497773138)

[Figura 13 - Protótipo de 'Login' 28](#_Toc497773139)

[Figura 14 - Protótipo de 'Menu' 28](#_Toc497773140)

[Figura 15 - Protótipo de 'Membros da Equipe' 29](#_Toc497773141)

[Figura 16 - Protótipo de 'Chat' 29](#_Toc497773142)

[Figura 17 - Protótipo de 'Contexto da Equipe' 30](#_Toc497773143)

[Figura 18 - Protótipo de 'Manter Tarefa' 31](#_Toc497773144)

[Figura 19 - Protótipo de 'Tarefas' 31](#_Toc497773145)

[Figura 20 - Protótipo de 'Manter Local' 32](#_Toc497773146)

[Figura 21 - Protótipo de 'Locais' 32](#_Toc497773147)

[Figura 22 - Diagrama de casos de uso do usuário Membro da Equipe 33](#_Toc497773148)

[Figura 23 - Diagrama de casos de uso do usuário Administrador da Equipe 33](#_Toc497773149)

[Figura 24 - Diagrama de casos de uso de ‘Usuário’ 34](#_Toc497773150)

[Figura 25 - Diagrama de casos de uso de ‘Gerenciamento de Equipe’ 39](#_Toc497773151)

[Figura 26 - Diagrama de casos de uso de ‘Membro’ 44](#_Toc497773152)

[Figura 27 - Diagrama de casos de uso de ‘Chat’ 45](#_Toc497773153)

[Figura 28 - Diagrama de casos de uso de ‘Tarefa’ 46](#_Toc497773154)

[Figura 29 - Diagrama de casos de uso de ‘Local’ 48](#_Toc497773155)

[Figura 30 - Classe Base utilizada no Diagrama de Classes **Erro! Indicador não definido.**](#_Toc497773156)

[Figura 31 - Diagrama de Classe da camada de Domínio 51](#_Toc497773157)

[Figura 32 - Diagrama de Classe da camada de Repositório 51](#_Toc497773158)

[Figura 33 - Diagrama de Classe da camada de Aplicação (Parte 1) 51](#_Toc497773159)

[Figura 34 - Diagrama de Classe da camada de Aplicação (Parte 2) 51](#_Toc497773160)

[Figura 35 - Diagrama de Classe da camada de Aplicação – Adapters 51](#_Toc497773161)

[Figura 36 - Diagrama de Classe da camada de Aplicação – DTO 51](#_Toc497773162)

[Figura 37 - Diagrama de Sequência ‘Autenticar’ 51](#_Toc497773163)

[Figura 38 - Diagrama de Sequência ‘Listar Equipes’ 51](#_Toc497773164)

[Figura 39 - Diagrama de Sequência ‘Sair da Equipe’ 51](#_Toc497773165)

[Figura 40 - Diagrama de Sequência ‘Criar Equipe’ 51](#_Toc497773166)

[Figura 41 - Diagrama de Sequência ‘Alterar Equipe’ 51](#_Toc497773167)

[Figura 42 - Diagrama de Sequência ‘Adicionar Membro na Equipe’ 51](#_Toc497773168)

[Figura 43 - Diagrama de Sequência ‘Remover Membro de Equipe’ 51](#_Toc497773169)

[Figura 44 - Diagrama de Sequência ‘Enviar Mensagem no Chat’ 51](#_Toc497773170)

[Figura 45 - Diagrama de Sequência ‘Locais’ 51](#_Toc497773171)

[Figura 46 - Diagrama de Sequência ‘Alterar Local’ 51](#_Toc497773172)

[Figura 47 - Diagrama de Sequência ‘Inserir Local’ 51](#_Toc497773173)

[Figura 48 - Diagrama de Sequência ‘Cadastre-se’ 51](#_Toc497773174)

[Figura 49 - Diagrama de Sequência ‘Tarefas’ 51](#_Toc497773175)

[Figura 50 - Diagrama de Sequência ‘Alterar Usuário’ 51](#_Toc497773176)

[Figura 51 - Diagrama de Sequência ‘Concluir Tarefa’ 51](#_Toc497773177)

[Figura 52 - Diagrama de Sequência ‘Desativar Tarefa’ 51](#_Toc497773178)

[Figura 53 - Diagrama de Sequência ‘Inserir Tarefa’ 51](#_Toc497773179)

[Figura 54 - Diagrama de Sequência ‘Alterar Tarefa’ 51](#_Toc497773180)

[Figura 55 - Diagrama ER **Erro! Indicador não definido.**](#_Toc497773181)

LISTA DE ABREVIATURAS

|  |  |
| --- | --- |
| UML | *Unified Modeling Language* (Linguagem de Modelagem Unificada) |
| RF | Requisito Funcional |
| PI | Protótipo de Interface |
| Obr. | Obrigatório |
| NF | Não Funcional |
| U | Usuário |
| S | Sistema |
| UC | *Use Case* (Caso de Uso) |
| PK | *Primary Key* (Chave Primaria) |
| FK | *Foreign Key* (Chave Estrangeira) |
| UK | *Unique Key* (Chave Unica) |
| ER | Entidade-relacionamento |

SUMÁRIO

[1 INTRODUÇÃO 1](#_Toc497773210)

[1.1 JUSTIFICATIVA 1](#_Toc497773211)

[2 METODOLOGIA 3](#_Toc497773212)

[2.1 REQUISITOS DE SOFTWARE 3](#_Toc497773213)

[2.2 DIAGRAMA DE CASOS DE USO 5](#_Toc497773214)

[2.2.1 Detalhamento dos Casos de Uso 7](#_Toc497773215)

[2.3 DIAGRAMA DE CLASSES 8](#_Toc497773216)

[2.4 DIAGRAMA DE SEQUÊNCIA 11](#_Toc497773217)

[2.5 DIAGRAMA DE ENTIDADE-RELACIONAMENTO 12](#_Toc497773218)

[2.6 PROTOTIPAÇÃO DE INTERFACES 14](#_Toc497773219)

[3 TECNOLOGIAS 15](#_Toc497773220)

[4 ESPECIFICAÇÃO DOS REQUISITOS 15](#_Toc497773221)

[4.1 ESQUEMA DE REQUISITOS FUNCIONAIS 16](#_Toc497773222)

[4.2 MENSAGENS DA APLICAÇÃO 25](#_Toc497773223)

[5 PROTÓTIPOS DE INTERFACE 27](#_Toc497773224)

[5.1 PROTÓTIPOS 27](#_Toc497773225)

[6 DIAGRAMAS DE CASOS DE USO 33](#_Toc497773226)

[6.1 DIAGRAMAS DE CASO DE USO DE USUÁRIO 33](#_Toc497773227)

[6.2 DIAGRAMAS DE CASOS DE USO DE GERENCIAMENTO DE EQUIPE 39](#_Toc497773228)

[6.3 DIAGRAMAS DE CASOS DE USO DE EQUIPE 44](#_Toc497773229)

[7 DIAGRAMAS DE CLASSES 50](#_Toc497773230)

[8 DIAGRAMAS DE SEQUÊNCIA 51](#_Toc497773231)

[9 DIAGRAMAS DE ENTIDADE-RELACIONAMENTO 51](#_Toc497773232)

[10 ESTRATÉGIA DE TESTES 51](#_Toc497773233)

[11 CONSIDERAÇÕES SOBRE A IMPLEMENTAÇÃO (REMOVER) 51](#_Toc497773234)

[12 CONSIDERAÇÕES PARA TRABALHOS FUTUROS 51](#_Toc497773235)

[13 CONCLUSÃO 51](#_Toc497773236)

[14 BIBLIOGRAFIA 51](#_Toc497773237)

# INTRODUÇÃO

O projeto Minha Equipe surgiu da observação e participação junto a uma equipe de comunicação ao realizar a cobertura de evento escoteiro. Durante a realização do evento foram identificadas dificuldades que por si só não causaram nenhum grande impacto negativo, porém, ao serem analisadas em um contexto macro, fica evidenciado que sendo melhor realizada uma gestão de pessoas, tarefas e recursos, o resultado final seria exponencial. Dos problemas identificados durante a realização do evento, destaca-se: a dificuldade de localizar os membros da equipe, mesmo que utilizando ferramentas de terceiros; as tarefas designadas aos membros acabaram por vezes não sendo cumpridas visto que não havia uma gestão correta disso; a falta de informação quanto aos locais; e de maior destaque, a dificuldade de comunicação com outros membros da equipe. Sendo assim, nota-se uma necessidade de centralização das informações relativas ao contexto em que a equipe está inserida. Diante disso, foi notada a oportunidade de desenvolver um sistema para gerenciamento de equipes apoiado por um aplicativo para dispositivos móveis.

## JUSTIFICATIVA

Mutirões Regionais Pioneiro são mutirões com fins sociais voltados para jovens do movimento Escoteiro entre as idades de 18 e 21 anos incompletos. No Rio Grande do Sul, estes eventos ocorrem duas vezes ao ano e em cidades distintas. Estes eventos não possuem qualquer objetivo financeiro sendo a sua organização feita por voluntários, ainda, o valor investido pelos participantes é somente para custeio de itens necessários tais como alimentação, estrutura do evento, transporte e materiais. Diante ao fato de que os eventos escoteiros estão sendo executados com maior frequência e o número de participantes está aumentando, fica evidente a necessidade de que a equipe de comunicação responsável pela cobertura do evento precisa crescer proporcionalmente.

Durante a realização do evento, ao mesmo tempo em que há um local central para realizar a gestão da equipe há também outros locais denominados ‘bases’ para onde os participantes são deslocados afim de realizar ações tais como reforma de praças, plantio de árvores, pintura de escolas dentre outros. Diante disso, e da grandiosidade do evento foi criada a necessidade de deslocamento dos membros da equipe entre as bases. Como os participantes da atividade desconheciam a localização dos colegas, determinados locais ficavam sem qualquer comunicador, ou ocorria o oposto o excesso de comunicadores em determinadas bases.

É notável assim, que há uma necessidade de melhorar a gestão do recurso humano e designação de tarefas para então estar efetivamente realizando uma melhor cobertura de eventos.

# METODOLOGIA

O desenvolvimento do presente projeto foi baseado em conceitos provenientes da orientação a objetos. Sommerville (2003) explica que o projeto orientado a objetos é uma estratégia em que os projetistas de sistema pensam em termos de ‘coisas’, em vez de operações ou funções, ainda o projeto orientado a objetos se dedica a desenvolver um modelo orientado a objetos do domínio de aplicação.

Para descrever a estrutura e funcionamento do sistema foi utilizada a UML (*Unified Modeling Language*). Esta é uma linguagem visual utilizada para modelar *softwares* baseados no paradigma de orientação a objetos e é aceita internacionalmente, facilitando a descrição do *software* projetado (GUEDES, 2009).

Neste projeto foram empregados os seguintes diagramas da UML: de classes, de casos de uso, de sequência e de entidade-relacionamento. Junto ao diagrama de casos de uso está o detalhamento de cada caso de uso, assim como o diagrama entidade-relacionamento está acompanhado do dicionário de dados.

## REQUISITOS DE SOFTWARE

Requisitos de *software* podem ser definidos como objetivos e restrições definidas pelos *stakeholders* (clientes, usuários, gerentes) do projeto. Assim, um requisito está relacionado intrinsicamente as propriedades de *software*.

Os requisitos são tradicionalmente classificados entre funcionais e não funcionais. Leite (2007) define os requisitos funcionais como

“[...] a descrição das diversas funções que clientes e usuários querem ou precisam que o *software* forneça. Eles definem a funcionalidade desejada do *software*. O termo função é usado no sentido genérico de operação que pode ser realizado pelo sistema, seja através de comandos dos usuários, seja pela ocorrência de eventos internos ou externos ao sistema”.

Para Leite (2007) requisitos não funcionais referem-se a qualidades globais do *software*, sendo que estes normalmente tendem a ser menos tangíveis tal como manutenibilidade, usabilidade, facilidade de uso dentre outros. Em geral, requisitos não funcionais são descritos de maneira informal e dependendo do contexto, de difícil validação.

O modelo utilizado representará um requisito funcional, contendo seu título e a descrição de sua funcionalidade seguido dos campos e interface relacionados. Abaixo do requisito funcional, serão detalhadas todas as restrições referentes à sua funcionalidade (requisitos não-funcionais). Conforme demonstrado na Tabela 1.

Tabela 1 - Exemplo de Requisito Funcional

|  |  |  |  |
| --- | --- | --- | --- |
| **RF1 - <Nome do Requisito Funcional>** | | | |
| **Descrição:**  <Descrição do requisito onde são explicados todos os detalhes importantes sobre a funcionalidade> | | | |
| **PI 1.1 - <Nome do Protótipo de Interface>** | | | |
| Uma imagem contendo captura de tela  Descrição gerada com muito alta confiança  Figura 1 - Exemplo de protótipo  Fonte: O autor (2017) | | | |
| **Campo** | | **Obr.** | **Descrição** |
| <Nome do Campo> | | <Sim> | <Descrição do campo> |
| **Requisitos não funcionais** | | | |
| **NF 1.1** | <Descrição e detalhamento dos requisitos não funcionais> | | |

Fonte: O autor (2017)

Escrever os requisitos de forma clara e de fácil entendimento é de suma importância para o projeto pois estabelece confiança e clareza no contato com o contratante e as demais partes envolvidas no projeto (desenvolvedores, analistas, usuário entre outros).

## DIAGRAMA DE CASOS DE USO

De acordo com Guedes (2011) o diagrama de casos de uso procura, por meio de uma linguagem simples, possibilitar a compreensão do comportamento externo do sistema por qualquer pessoa, tentando apresentar o sistema por intermédio de uma perspectiva do usuário.

Para Medeiros (2004), caso de uso é a parte mais importante da construção de *software* orientado a objetos utilizando a UML, ainda talvez seja o único instrumento que acompanha um *software* do seu início até a conclusão.

Os atores apresentados nos casos de uso podem ser pessoas, sistemas ou hardware e são representados como ‘bonecos magros’, já cada caso de uso é representado por uma elipse contendo dentro de si um texto a que funcionalidade o caso de uso se refere.

A Figura 2 apresenta uma associação entre um ator e um caso de uso.

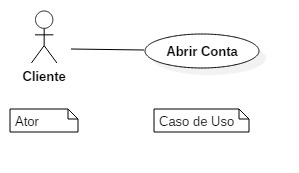


Figura 2 - Exemplo de associação entre Ator e Caso de Uso

Fonte: O autor (2017)

Entre um ator e um caso de uso pode existir apenas a associação, que é ilustrada com uma linha ligando ambos os objetos representando uma funcionalidade do sistema do ponto de vista do ator.

A generalização (ou especialização), é quando ocorre a herança de funcionalidades entre atores ou entre casos de uso, ou seja, com esta herança o elemento herdeiro possui todas as funcionalidades do elemento herdado mais as suas funcionalidades específicas. A Figura 3 apresenta um exemplo de generalização entre casos de uso, já a Figura 4 apresenta a generalização entre atores.

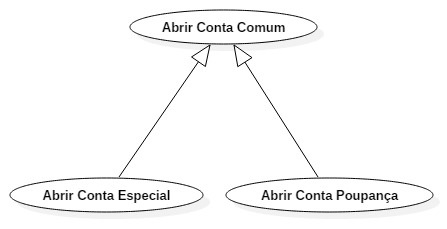


Figura 3 - Exemplo de Generalização entre Casos de Uso

Fonte: O autor (2017)

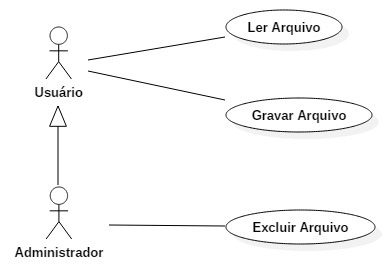


Figura 4 - Exemplo de Generalização entre Atores

Fonte: O autor (2017)

Ainda, entre casos de uso podem ocorrer os seguintes tipos de relacionamentos:

* *Extend* - Em um relacionamento entre um caso de uso A para um caso de uso B, significa que o A pode ser um complemento à funcionalidade B.
* *Include* - Em um relacionamento A para B, significa que para a execução de A será necessário executar a funcionalidade B como complemento.

A Figura 5 apresenta associações com *extend* e *include*.

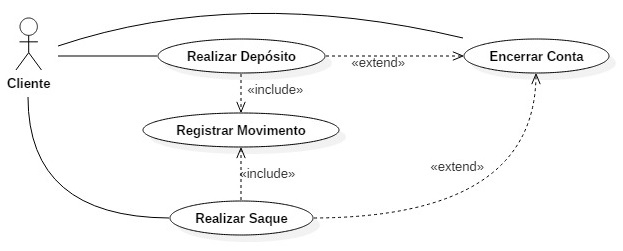


Figura 5 - Exemplo de Include e Extend

Fonte: O autor (2017)

### Detalhamento dos Casos de Uso

Consiste em descrever de forma detalhada e com uma linguagem bastante simples, informações como a função do caso de uso, quais atores interagem com ele, quais as etapas devem ser executadas pelo ator e pelo sistema para que o caso de uso execute a sua função, quais parâmetros devem ser fornecidos e quais restrições e validações devem existir no caso de uso.

O fluxo principal é a sequência ordenada de ações do usuário e respostas do sistema a essas ações que compõe o fluxo natural do caso de uso em questão. Um fluxo de exceção é um conjunto de ações do sistema e do usuário que são executadas em determinado ponto caso ocorra algum erro, utilizado para descrever tratamentos de exceção. O primeiro dígito da numeração de um fluxo de exceção indica o passo do fluxo principal ao qual o mesmo se refere. As marcações ‘[U]’ e ‘[S]’ indicam respectivamente os passos executados pelo usuário e pelo sistema.

Na Tabela 2 é apresentado o modelo de documentação para os casos de uso.

Tabela 2 - Exemplo de documentação de Caso de Uso

|  |  |
| --- | --- |
| UC0 - <Nome do Caso de Uso> | |
| **Ator principal** | <Os atores do caso de uso> |
| **Resumo** | <Descrição resumida do caso de uso> |
| **Pré-condições** | <Condições iniciais do sistema para que o caso de uso possa ser executado> |
| **Pós-condições** | <Estado final do sistema após todas as execuções do caso de uso> |
| **Fluxo principal** | |
| 01. [U] <Passo ou ação realizado pelo usuário> | |
| 02. [S] <Passo ou ação realizado pelo sistema> | |
| 03. [S] <Passo ou ação realizado pelo sistema> | |
| **Fluxo Alternativo – <Nome do Fluxo Alternativo>** | |
| 02.a.1. [S] <Passo ou ação realizado pelo sistema como alternativa ao passo 2 do fluxo principal> | |
| 02.a.2. [S] <Passo ou ação realizado pelo sistema como alternativa ao passo 2 do fluxo principal> | |
| **Fluxo de Exceção – <Nome do Fluxo de Exceção>** | |
| 03.a.1. [S] <Passo ou ação realizado pelo sistema em resposta a um erro ocorrido no passo 3 do fluxo principal>. | |

Fonte: O autor (2017)

## DIAGRAMA DE CLASSES

Um diagrama de classes é um modelo fundamental de uma especificação orientada a objetos. Produz a descrição mais próxima da estrutura do código de um programa, ou seja, mostra o conjunto de classes com seus atributos e métodos e os relacionamentos entre classes. Classes e relacionamentos constituem os elementos sintáticos básicos do diagrama de classes (SILVA, 2007).

Medeiros (2006) descreve o diagrama de classes como o estágio mais aguardado na construção de um *software* orientado a objetos, pois nesta fase a automatização do negócio começa a tomar forma.

A classe é representada por um retângulo que pode ter até três divisões, que são, respectivamente: nome, atributos e métodos. Em geral, uma classe possui atributos e métodos, mas é possível encontrar classes que contenham apenas uma dessas características.

Os atributos de uma classe, também conhecidos como propriedades, representam características da mesma, ou seja, as peculiaridades que podem variar de um objeto para outro. Cada atributo tem seu tipo de dado especificado após dois pontos, e à sua esquerda é indicado o modificador de acesso. Essa mesma notação vale para os métodos, porém estes possuem entre parênteses os parâmetros que recebem como entrada, podendo ser nenhum ou vários, e o tipo do método (após os dois pontos) reflete o tipo do dado que ele retorna. Na Tabela 3 são apresentados os níveis de acesso e na Figura 6 uma exemplificação.

Tabela 3 - Modificadores de Acesso do Diagrama de Classes

|  |  |  |
| --- | --- | --- |
| **Representação** | **Nome** | **Descrição** |
| - | Private | Indica que o atributo ou método é visível somente para a própria classe. |
| + | Public | Indica que o atributo ou método é visível para todos. |
| # | Protected | Indica que o atributo ou método é visível somente para a própria classe e para as classes que herdam dela. |
| ~ | Internal | Indica que o atributo ou método é visível somente para a para as classes do mesmo pacote ou *assembly*. |

Fonte: O autor (2017)

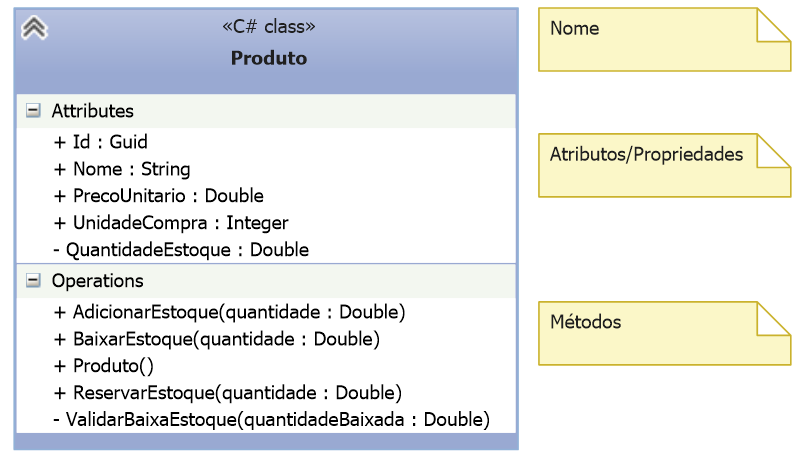


Figura 6 - Exemplo de Diagrama de Classes

Fonte: O autor (2017)

Outro elemento importante do diagrama de classes são as associações que conectam as classes. A associação entre as classes é representada por meio de linhas que indicam a relação estrutural entre duas classes. A navegabilidade indica que determinada classe possui conhecimento da outra. Nas extremidades das retas há a multiplicidades que indica o número mínimo e máximo de objetos envolvidos na associação. Na Tabela 4 são apresentadas as multiplicidades e na Figura 7 sua utilização

Tabela 4 - Multiplicidade do Diagrama de Classes

|  |  |
| --- | --- |
| **Multiplicidade** | **Significado** |
| 0..1 | Não existirá nenhuma instância da classe ou somente uma instância. |
| 1..1 | Somente existirá uma instância no determinado relacionamento. |
| 0..\* | Pode haver nenhum ou muitos objetos no determinado relacionamento. |
| 1..\* | Pode haver no mínimo um ou muitos objetos no determinado relacionamento. |
| \* | Muitos objetos da classe estão envolvidos na associação. |

Fonte: O autor (2017)

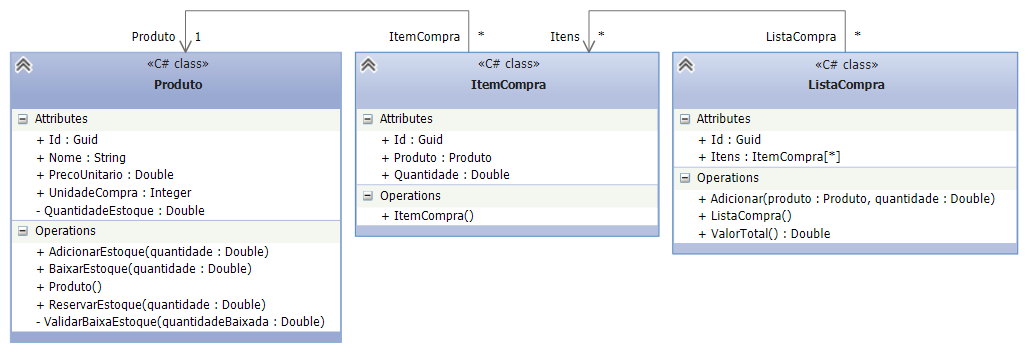


Figura 7 - Exemplo de Multiplicidade no Diagrama de Classes

Fonte: O autor (2017)

Uma Interface é uma coleção de assinatura de operações e/ou atributos que define um conjunto coeso de comportamentos.

A utilização de Interfaces possibilita a redução de acoplamentos, uma vez que ela assina a classe utilizada, somente os métodos dela estarão visíveis. Na UML, a implementação de Interfaces é representada como na Figura 8, onde há uma linha pontilhada sendo em uma das extremidades uma seta indicando a interface, enquanto na outra ponta está a classe que implementa a interface.

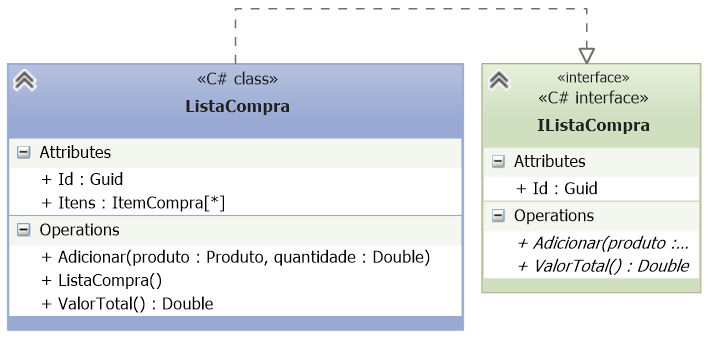


Figura 8 - Exemplo de implementação de Interface

Fonte: O autor (2017)

## DIAGRAMA DE SEQUÊNCIA

Pressman (2011) explica que o diagrama de sequência é utilizado para indicar as comunicações dinâmicas entre os objetos de um sistema. Ele mostra a ordem na qual as mensagens são enviadas de um objeto para outro ao longo do tempo, e a sequência de objetos na memória. Diagramas de sequência são geralmente utilizados para mostrar as interações de cada caso de uso do sistema.

É necessário que tenha todas as informações preenchidas de forma clara e lógica para uma melhor representação do cenário apresentado. Os métodos utilizados nos diagramas de classe também aparecem no diagrama de sequência.

Em um diagrama de sequência temos os atores, que são aqueles que irão interagir com o sistema e representados por um boneco que os representa, os objetos são representados por um retângulo vertical, a linha pontilhada vertical representa o tempo de execução, a flecha horizontal continua o envio de eventos ao objeto, já a flecha pontilhada a resposta do evento enviado ao objeto.

A Figura 9 demonstra um exemplo de um diagrama de sequência, que representa a ação de login de um usuário em algum sistema.

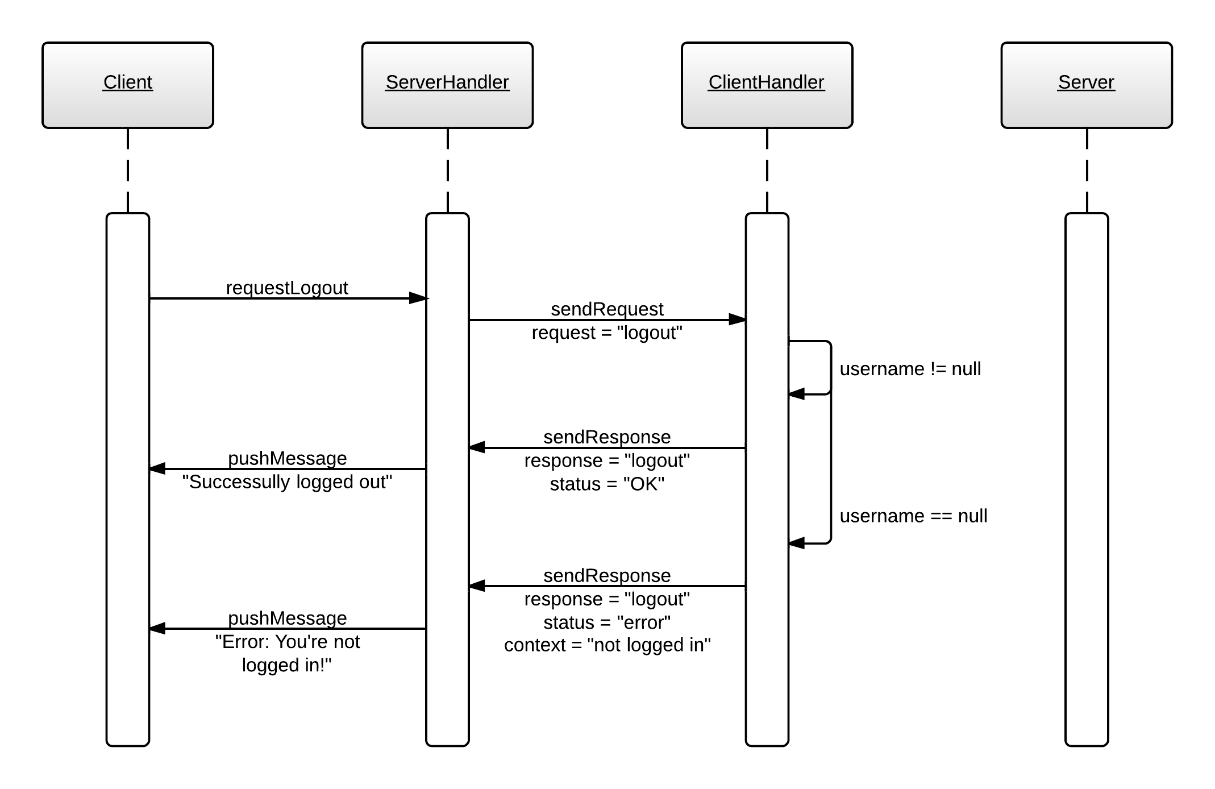


Figura 9 - Exemplo de Diagrama de Sequência de Logout

Fonte: https://github.com/larseen/KTN-ELM-CHAT/tree/master/documentation

## DIAGRAMA DE ENTIDADE-RELACIONAMENTO

O modelo-entidade relacionamento (ER) é um modelo de dados para descrever os aspectos de informação de um domínio de negócio ou seus requerimentos de processo, de uma maneira abstrata que em última análise se presta a ser implementada em um banco de dados, como um banco de dados relacional.

Como descrito por Medeiros (2004), um relacionamento entre entidades indica um vínculo entre elas, e este possui uma cardinalidade (quantos registros de uma entidade podem ser relacionados a outra). O relacionamento é definido através de chaves primárias (PK) e chaves estrangeiras (FK). Chave primária é definida por um ou mais atributos identificadores de uma tabela, este (s) atributo (s) nunca pode (m) ter seu valor nulo. Já a chave estrangeira é definida por um atributo que representa uma referência a chave primária de outra tabela.

Para este projeto foi utilizada a notação “Pé de Galinha” para representação das cardinalidades pois esta facilita o entendimento dos relacionamentos. Abaixo, a Tabela 5 mostra as representações dos diferentes tipos de cardinalidade.

Tabela 5 - Cardinalidades do Diagrama ER

|  |  |
| --- | --- |
| **Representação** | **Descrição** |
|  | Zero ou muitos |
|  | Um ou muitos |
|  | Um e somente um. |
|  | Nenhum ou um |

Fonte: O autor (2017)

Estes diagramas estão representados por meio do diagrama ER como apresentado na Figura 10.

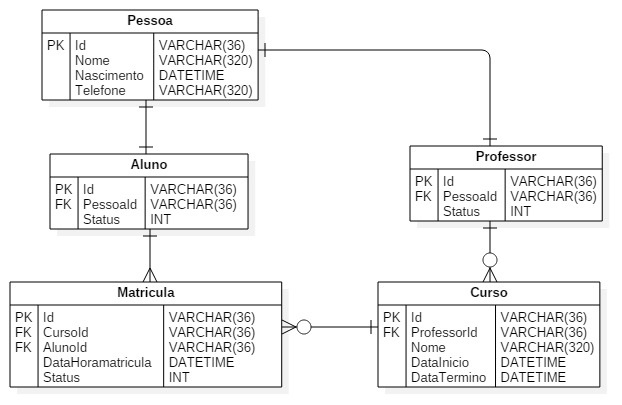


Figura 10 - Exemplo de Diagrama ER

Fonte: O autor (2017)

O dicionário de dados representa cada tabela do banco de dados da camada de persistência, são demonstradas as informações: Campo da tabela (Campo), chave primária (PK), chave estrangeira (FK), chave única (UK), tipo de dado (Tipo) e descrição da coluna (Descrição). Os atributos do campo serão preenchidos com um “X” quando possuírem o a característica descrita. O tipo de dado representa qual a informação esperada pela coluna para gravação. A descrição serve para explicar a funcionalidade do campo. A Tabela 6 apresenta um modelo de dicionário de dados da tabela ‘Curso’ do diagrama apresentado anteriormente.

Tabela 6 - Exemplo de Tabela do Dicionário de Dados

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Curso** | | | | | | |
|  | **Campo** | **PK** | **FK** | **UK** | **Tipo** | **Descrição** |
| 01 | Id | X |  | X | varchar(36) | Identificador único |
| 02 | ProfessorId |  | X |  | varchar(36) | Código de Professor |
| 03 | Nome |  |  |  | varchar(320) | Nome do curso |
| 04 | DataInicio |  |  |  | datetime | Data de início do curso |
| 05 | DataTermino |  |  |  | datetime | Data de encerramento do curso |

Fonte: O autor (2017)

## PROTOTIPAÇÃO DE INTERFACES

Os protótipos de tela servem para fornecer uma visão macro de como será a interface do sistema, para que assim seja possível desenvolver as funções necessárias que cada funcionalidade necessita. Pelo fato do sistema se dividir em duas frentes, suas interfaces foram pensadas voltadas para seu contexto de utilização.

Na Figura 11 fica representado o modelo de interface desenvolvido.

Uma imagem contendo equipamentos eletrônicos, mostrador, captura de tela

Descrição gerada com alta confiança

Figura 11 - Protótipo de interface

Fonte: O autor (2017)

# TECNOLOGIAS

A aplicação será desenvolvida utilizando o Ionic Framework juntamente com Google Firebase. Foi optado por esse modelo pois o desenvolvimento de aplicativos híbridos permite a reutilização de código, ainda, a tecnologia Typescript consequentemente menos tempo é necessário ao é a linguagem na qual o desenvolvedor tem maior facilidade, logo não deverá passar por estágios de aprendizagem. Ainda, se trata de uma linguagem robusta que possui uma grande empresa responsável, tornando difícil sua obsolescência, garantindo sempre os recursos necessários para o desenvolvimento do projeto.

A estrutura do projeto se destaca em três partes:

* Backend: local de processamento das informações utilizando banco de dados SQL Server. Este serviço irá disponibilizar uma API a qual o website e o aplicativo *mobile* irão fazer consumo dos dados.
* Website: será desenvolvido utilizando HTML5, CSS3 e Javascript.
* Aplicativo Móvel: será desenvolvido utilizando Xamarin, uma ferramenta de desenvolvimento onde aplicativos para Android ou iOS são programados em C#. O aplicativo será desenvolvido para Android 4 ou superior.

# ESPECIFICAÇÃO DOS REQUISITOS

Este capítulo da documentação tem por objetivo descrever as funções que o sistema deve implementar. O conjunto de todos os requisitos formam a base para o posterior desenvolvimento do produto de *software*.

Abaixo estão listados os requisitos funcionais:

* RF1 - Cadastro Inicial
* RF2 - Login
* RF3 - Recuperar Senha
* RF4 - Perfil
* RF5 - Alterar E-mail
* RF6 - Alterar Senha
* RF7 - Manter Equipe
* RF8 - Convidar
* RF9 - Gerenciar Membros
* RF10 - Mapa da Equipe
* RF11 - Localização do Membro
* RF12 - Chat
* RF13 - Manter Local
* RF14 - Manter Tarefa

## ESQUEMA DE REQUISITOS FUNCIONAIS

A seguir serão apresentados todos os requisitos funcionais do sistema com sua respectiva descrição, requisitos não funcionais e protótipos de tela. Os requisitos estão da Tabela 7 até a Tabela 20.

Tabela 7 - Requisito Funcional ‘Cadastro Inicial’

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Cadastro Inicial | | | |
| **Descrição:**  Esta funcionalidade permite que qualquer pessoa realize um cadastro rápido na aplicação de modo a informar somente seus dados básicos e então já utilizar a ferramenta. | | | |
| **Campo** | | **Obr.** | **Descrição** |
| Nome | | Sim | Campo de texto no qual o usuário deve informar seu nome. |
| E-mail | | Sim | Campo de texto no qual deve ser informado um e-mail válido. |
| Senha | | Sim | Campo de texto no qual deve ser informada a senha correspondente ao e-mail digitado anteriormente. |
| Criar conta | | - | Botão: ao ser clicado, dispara a solicitação. |
| **Requisitos não funcionais** | | | |
| **NF1.1** | O campo de e-mail deve permitir no máximo 320 caracteres. | | |
| **NF1.2** | O sistema deve garantir que a senha informada tenha no mínimo a complexidade de 8 caracteres. | | |
| **NF1.3** | Caso o e-mail informado seja inválido, a mensagem ***msg\_emailInvalido*** deve ser apresentada. | | |
| **NF1.4** | O sistema deve garantir que o e-mail informado seja único dentro do sistema, caso o usuário tente realizar o cadastro com um e-mail já utilizado, a mensagem ***msg\_usuario\_emaiJaCadastrado*** deve ser apresentada. | | |
| **NF1.5** | O botão “Criar conta” deve permanecer como inativo enquanto as informações não forem informadas. | | |
| **NF1.6** | O sistema deve garantir que o e-mail informado seja um endereço válido e existente. Para tal, um e-mail contendo um link para confirmação de cadastro será enviado. Somente após a confirmação o usuário poderá adentrar ao sistema. | | |
| **NF1.7** | O sistema deve garantir que a senha informada tenha no mínimo a complexidade de 8 caracteres. Caso contrário, a mensagem ***msg\_usuario\_senhaFraca*** será apresentada. | | |

Fonte: O autor (2017)

Tabela 8 - Requisito Funcional ‘Login’

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Login | | | |
| **Descrição:**  Esta funcionalidade permite que o usuário autentique na aplicação e tenha acesso as funcionalidades. O processo de autenticação é realizado confirmando os dados de acesso, que são e-mail e senha vinculados ao usuário. | | | |
| **Campo** | | **Obr.** | **Descrição** |
| E-mail | | Sim | Campo de texto de até 320 posições no qual deve ser informado um e-mail já cadastrado. |
| Senha | | Sim | Campo de texto no qual deve ser informada a senha correspondente ao e-mail digitado anteriormente. |
| Acessar | | - | Botão: ao ser clicado, dispara o evento de validação da autenticação. |
| **Requisitos não funcionais** | | | |
| **NF2.1** | O botão “Acessar” deve permanecer como inativo enquanto as informações não forem informadas. | | |
| **NF2.2** | Caso o e-mail informado seja inválido, a mensagem ***msg\_emailInvalido*** deve ser apresentada. | | |
| **NF2.3** | Caso o usuário não exista na base de dados ou a senha estiver errada, deve-se retornar ***msg\_login\_falha*.** | | |
| **NF2.4** | Após o login bem-sucedido, o usuário será redirecionado para a Página Inicial. | | |

Fonte: O autor (2017)

Tabela 9 - Requisito Funcional ‘Recuperar Senha’

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Recuperar Senha | | | |
| **Descrição:**  A funcionalidade ‘Recuperar Senha’ permite que um usuário solicite ao sistema a alteração de sua senha. Esse processo deve ocorrer de forma que seja informado o e-mail do usuário, o sistema valide a existência de um usuário com este e-mail e envie um e-mail contendo um link o qual possibilita que o usuário acesse e faça a alteração de sua senha.  O link de confirmação recebido por e-mail será aberto em um navegador de internet. | | | |
| **RF3.1 - Solicitação de Recuperação de Senha** | | | |
| **Campo** | | **Obr.** | **Descrição** |
| E-mail | | Sim | Campo de texto de até 320 posições no qual deve ser informado um e-mail válido e vinculado a um usuário existente. |
| Recuperar minha senha | | - | Botão: realiza a solicitação para envio do e-mail com o acesso para alteração de senha. |
| **RF3.2 – Alteração de senha** | | | |
| **Campo** | | **Obr.** | **Descrição** |
| Nova senha | | Sim | Nova senha. |
| Salvar | | - | Botão: confirma a alteração da senha |
| **Requisitos não funcionais** | | | |
| **NF3.1** | Quando solicitada a recuperação de senha o sistema deve validar se o e-mail informado está vinculado a um usuário, caso não esteja, a mensagem ***msg\_recuperarSenha\_usuarioNaoExiste*** deve ser disparada. | | |
| **NF3.2** | O botão “Recuperar minha senha” deve permanecer inativo enquanto um valor não for informado no campo ‘E-mail’. | | |
| **NF3.3** | Ao disparar o evento de ‘Solicitação de recuperação’ o sistema deve validar o preenchimento do campo de e-mail. A mensagem ***msg\_emailInvalido*** será disparada caso o e-mail informado seja inválido. | | |
| **NF3.4** | A validação de senha deve consistir conforme descrito no requisito RF1 - Cadastro Inicial. | | |

Fonte: O autor (2017)

Tabela 10 - Requisito Funcional ‘Perfil’

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Perfil | | | |
| **Descrição:**  Esta funcionalidade permite ao usuário alterar suas informações básicas de perfil. Nesta funcionalidade não serão alteradas as informações de autenticação do usuário sendo essas especificadas nos requisitos RF5 - Alterar E-mail e RF6 - Alterar Senha.  Deve ser possível selecionar uma imagem da galeria ou tirar uma foto afim de definir como imagem de perfil. | | | |
| **Campo** | | **Obr.** | **Descrição** |
| Nome | | Não | Campo de texto de até 60 posições, no qual deve ser informado o nome do usuário. |
| Tags | | Não | Campo no qual permite ao usuário selecionar ou inserir características que o descreve. A obrigatoriedade aplicada é de no máximo 5 itens. |
| Imagem | | Não | Permite ao usuário fazer upload de uma imagem para ser utilizada como imagem de perfil. |
| Atualizar Perfil | | - | Botão: realiza a solicitação para atualização de perfil. |
| **Requisitos não funcionais** | | | |
| **NF 4.1** | O sistema não deve permitir que tags repetidas sejam informadas a um usuário. | | |
| **NF4.2** | O sistema deve permitir ao usuário selecionar qualquer foto de seu dispositivo sendo que esta deve ser redimensionada onde sua maior dimensão seja de no máximo 500px. | | |

Fonte: O autor (2017)

Tabela 11 - Requisito Funcional ‘Alterar E-mail’

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Alterar E-mail | | | |
| **Descrição:**  Esta funcionalidade permite ao usuário alterar seu e-mail na aplicação.  Após a solicitação bem-sucedida o usuário será desconectado da aplicação sendo necessário a confirmação do e-mail para realizar um novo acesso. | | | |
| **Campo** | | **Obr.** | **Descrição** |
| Novo e-mail | | Sim | Campo de texto de até 320 posições, no qual deve ser informado o novo e-mail a ser utilizado. |
| Senha | | Sim | Campo de texto no qual o usuário deve informar sua atual senha. |
| Alterar e-mail | | - | Botão: realiza a solicitação para alteração de e-mail. |
| **Requisitos não funcionais** | | | |
| **NF5.1** | O sistema deve validar se a senha informada é referente ao usuário que está solicitando alteração de seu e-mail, caso contrário a mensagem ***msg\_senha\_invalida*** deve ser apresentada. | | |
| **NF5.2** | O sistema deve garantir que o novo e-mail informado seja único dentro do sistema, caso o usuário tente informar um e-mail já cadastrado a mensagem ***msg\_usuario\_emaiJaCadastrado*** deve ser apresentada. | | |
| **NF5.3** | O sistema deve garantir que o e-mail informado seja um endereço válido e existente. Para tal, um e-mail contendo um link para confirmação da alteração de e-mail será enviado. Somente após a confirmação o usuário poderá adentrar ao sistema. | | |

Fonte: O autor (2017)

Tabela 12 - Requisito Funcional ‘Alterar Senha’

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Alterar Senha | | | |
| **Descrição:**  Esta funcionalidade permite ao usuário alterar sua senha na aplicação. | | | |
| **Campo** | | **Obr.** | **Descrição** |
| Senha atual | | Sim | Campo de texto no qual o usuário deve informar a atual senha. |
| Nova senha | | Sim | Campo de texto no qual o usuário deve informar a nova senha. |
| Confirmar Senha | | Sim | Campo de texto no qual o usuário deve confirmar a nova senha. |
| Alterar senha | | - | Botão: realiza a solicitação para alteração de senha. |
| **Requisitos não funcionais** | | | |
| **NF6.1** | O sistema deve validar se a senha informada é referente ao usuário que está solicitando alteração de seu e-mail, caso contrário a mensagem ***msg\_senha\_invalida*** deve ser apresentada. | | |
| **NF6.2** | O botão “Alterar senha” deve permanecer inativo caso: os valores não tenham sido informados; ou o valor do campo ‘nova senha’ seja diferente do valor do campo ‘confirmar nova senha’. | | |

Fonte: O autor (2017)

Tabela 13 - Requisito Funcional ‘Manter Equipe’

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Manter Equipe | | | |
| **Descrição:**  A funcionalidade ‘Manter Equipe’ permite que o usuário crie uma nova equipe ou então altere uma equipe a qual ele é administrador.  Qualquer membro da equipe pode visualizar estas informações, porém, somente o usuário administrador pode realizar alterações. | | | |
| **Campo** | | **Obr.** | **Descrição** |
| Nome | | Sim | Campo de texto de até 60 posições, no qual deve ser informado o nome da equipe |
| Data/Hora Início | | Sim | Campo Data/Hora em formato 24 horas, no qual deve ser informado a data e hora em que a equipe começará suas ações. |
| Data/Hora Fim | | Sim | Campo Data/Hora em formato 24 Horas, no qual deve ser informado a data e hora em que a equipe finalizará suas ações. |
| Imagem | | Não | Imagem representando a equipe. |
| Salvar | | - | Botão: realiza a solicitação para criar ou alterar uma equipe. |
| **Requisitos não funcionais** | | | |
| **NF7.1** | Ao disparar o evento de gravação/alteração o sistema deve validar o preenchimento dos campos. A mensagem ***msg\_campos\_invalidos*** será disparada caso algum dos campos não esteja informado. | | |
| **NF7.2** | A data/hora de fim deve ser maior que a data/hora de início. Caso contrário a mensagem ***msg\_equipe\_datahora\_menor*** deve ser apresentada. | | |
| **NF7.3** | O sistema deve permitir ao usuário selecionar qualquer foto de seu dispositivo sendo que esta deve ser redimensionada onde sua maior dimensão seja de no máximo 500px. | | |

Fonte: O autor (2017)

Tabela 14 - Requisito Funcional ‘Convidar’

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Convidar | | | |
| **Descrição:**  A funcionalidade ‘Convidar’ permite que o usuário administrador envie convites a pessoas que ainda não estejam inseridas na equipe. O membro que receber este convite será notificado e poderá aceitar ou recusar.  Ao enviar os convites o sistema deverá buscar pelo e-mail os usuários associados no banco de dados e estes serem notificados na aplicação. Os e-mails que não estiverem vinculados a nenhum usuário devem permanecer no banco de dados visto que se um novo usuário realizar seu cadastro com um e-mail que possua convites, este será notificado.  Deve ser possível enviar convites em lote. | | | |
| **PI 6.1 – Campos de Convite** | | | |
| **Campo** | | **Obr.** | **Descrição** |
| E-mail | | Sim | Campo de texto de até 320 posições, no qual deve ser informado o e-mail da pessoa que será convidada |
| Adicionar convite | | - | Botão: adiciona na interface um novo campo para ser possível convidar mais usuários. |
|  | |  |  |
| **Requisitos não funcionais** | | | |
| **NF8.1** | Deve ser possível remover um destinatário. | | |
| **NF8.2** | O botão ‘Enviar Convites’ deve permanecer inativo enquanto houverem campos sem valor ou com e-mails inválidos. | | |

Fonte: O autor (2017)

Tabela 15 - Requisito Funcional ‘Gerenciar Membros’

|  |  |
| --- | --- |
| 1. Gerenciar Membros | |
| **Descrição:**  Esta funcionalidade possibilita que o usuário administrador da equipe remova membros.  O usuário administrador deverá remover um usuário por vez. | |
| **Requisitos não funcionais** | |
| **NF9.1** | Ao disparar o evento de remoção de membro a mensagem ***msg\_equipe\_confirmarRemocaoMembro*** deve ser apresentada solicitando confirmação. |

Fonte: O autor (2017)

Tabela 16 - Requisito Funcional ‘Mapa da Equipe’

|  |  |
| --- | --- |
| 1. Mapa da Equipe | |
| **Descrição:**  Esta funcionalidade apresenta um mapa onde nele estarão visíveis as localizações dos membros (guiado pelo requisito RF11 - Localização do Membro) e locais de interesse da equipe (guiado pelo requisito RF13 - Manter Local). | |
| **Requisitos não funcionais** | |
| **NF10.1** | Deve ser possível realizar a navegação no mapa. |
| **NF10.2** | Ao clicar sobre uns dos itens do mapa, deve ser mostrada suas informações. |

Fonte: O autor (2017)

Tabela 17 - Requisito Funcional ‘Localização do Membro’

|  |  |
| --- | --- |
| 1. Localização do Membro | |
| **Descrição:**  Esta funcionalidade permite que os membros da equipe vejam a última localização uns dos outros.  A ação deve ocorrer quando: ao clicar sobre um dos membros da equipe, o usuário seja direcionado para a tela descrita no requisito RF10 - Mapa da Equipe e em seu mapa seja centralizada a localização do membro. | |
| **Requisitos não funcionais** | |
| **NF11.1** | Ao requisitar para o sistema a ‘Visualização da Última Localização do Usuário’, caso o valor não exista, deve ser disparada a mensagem ***msg\_localizacao\_vazia***. |

Fonte: O autor (2017)

Tabela 18 - Requisito Funcional ‘Chat’

|  |  |
| --- | --- |
| 1. Chat | |
| **Descrição:**  A funcionalidade de ‘Chat’ permite que os membros da equipe se comuniquem por meio de texto. Junto ao chat serão mostradas notificações de ações realizadas dentro do sistema.  As seguintes notificações serão apresentadas:   * Alterações em informações das equipes ou seus membros (RF7 - Manter Equipe); * Alterações em Locais (RF13 - Manter Local); * Alterações em Tarefas e sua conclusão (RF14 - Manter Tarefa).   A troca de mensagens deve acontecer em tempo real.  Deve ser possível a visualização de mensagens antigas. | |
| **Requisitos não funcionais** | |
| **NF12.1** | Para o usuário, suas mensagens enviadas devem aparecer destacadas. |
| **NF12.2** | Caso ocorra falha no envio de uma mensagem, deve ser mostrada a mensagem ***msg\_chat\_erroEnvio*.** |

Fonte: O autor (2017)

Tabela 19 - Requisito Funcional ‘Manter Local’

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Manter Local | | | |
| **Descrição:**  Esta funcionalidade possibilita ao administrador da equipe que pontos geográficos sejam definidos como locais de interesse para o contexto. Estes locais devem estar visíveis no mapa descrito no requisito RF7 - Manter Equipe.  Somente o usuário administrador poderá manter os locais.  O local não poderá ser excluído caso esteja vinculado a uma tarefa conforme requisito RF14 - Manter Tarefa. | | | |
| **Campo** | | **Obr.** | **Descrição** |
| Nome | | Sim | Campo de texto de até 60 posições, no qual deve ser informado o nome do local. |
| Descrição | | Não | Campo de texto de até 150 posições, no qual pode ser informado a descrição do local |
| Marcação no mapa | | Sim | Mapa onde deve ser inserido os pontos delimitando um local. |
| Alterar/Criar | | - | Botão: realiza a solicitação para inserir ou alterar. |
| Excluir | | - | Botão: realiza a solicitação para exclusão. |
| **Requisitos não funcionais** | | | |
| **NF13.1** | Ao disparar o evento de gravação/alteração o sistema deve validar o preenchimento dos campos. A mensagem ***msg\_campos\_invalidos*** será disparada caso algum dos campos não esteja informado. | | |
| **NF13.2** | A mensagem ***msg\_local\_falha\_excluirTarefaVinculada*** deverá ser disparada caso não seja possível a exclusão do local por estar vinculado a uma Tarefa | | |

Fonte: O autor (2017)

Tabela 20 - Requisito Funcional ‘Manter Tarefa’

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Manter Tarefa | | | |
| **Descrição:**  Esta funcionalidade possibilita que o usuário administrador criar/alterar tarefas e com isso designar seus responsáveis.  Somente o usuário administrador pode criar, alterar ou cancelar tarefas.  Uma tarefa deve ter no mínimo um responsável para sua execução.  Todas as tarefas devem estar visíveis para todos os membros da equipe.  Os usuários somente podem concluir tarefas as quais forem responsáveis. | | | |
| **Campo** | | **Obr.** | **Descrição** |
| Nome | | Sim | Campo de texto de até 60 posições, no qual deve ser informado o nome da tarefa. |
| Responsáveis | | Sim | Campo de seleção onde serão informados os usuários responsáveis da tarefa. |
| Local | | Não | Campo possibilitando a seleção de um local. (Ver requisito **Erro! Fonte de referência não encontrada.** **Erro! Fonte de referência não encontrada.**) |
| Descrição | | Não | Campo de texto de até 150 posições, no qual pode ser informada uma descrição para a tarefa |
| Concluir | | - | Botão: altera a tarefa marcando-a como concluída. |
| Cancelar | | - | Botão: realiza a solicitação para cancelar a tarefa. |
| Alterar/Criar | | - | Botão: realiza a solicitação para manter a tarefa. |
| **Requisitos não funcionais** | | | |
| **NF14.1** | Ao disparar o evento de gravação/alteração o sistema deve validar o preenchimento dos campos. A mensagem ***msg\_campos\_invalidos*** será disparada caso algum dos campos não esteja informado. | | |
| **NF14.2** | O botão ‘Concluir’ somente pode estar visível para os usuários responsáveis pela tarefa.  Os botões ‘Cancelar’ e ‘Alterar/Criar’ somente estarão visíveis para o administrador da equipe. | | |
| **NF14.3** | Ao momento que o campo de marcação‘Concluir’ for marcado, deve ser solicitada a confirmação por meio da mensagem ***msg\_tarefa\_concluir***. | | |

Fonte: O autor (2017)

## MENSAGENS DA APLICAÇÃO

As mensagens da aplicação possuem finalidade de avisar ao usuário sobre o estado do sistema detalhando os acontecimentos.

Tabela 21 - Mensagens da Aplicação

|  |  |
| --- | --- |
| **Nome** | **Conteúdo** |
| ***msg\_campos\_invalidos*** | Campos obrigatórios não preenchidos |
| ***msg\_emailInvalido*** | O e-mail digitado não é válido. |
| ***msg\_login\_falha*** | Usuário e/ou senha inválido(s) |
| ***msg\_recuperarSenha\_usuarioNaoExiste*** | O usuário não foi encontrado. |
| ***msg\_solicitacaoExpirada*** | Esta solicitação expirou e por isso não é mais válida. |
| ***msg\_recuperarSenha\_sucess\_emailEnviado*** | Você recebera um e-mail com as informações sobre como alterar sua senha. |
| ***msg\_usuario\_emaiJaCadastrado*** | O e-mail digitado já está em uso. |
| ***msg\_usuario\_senhaFraca*** | Senha fraca. Por favor, informe ao menos 8 caracteres. |
| ***msg\_equipe\_datahora\_menor*** | A Data/Hora Final deve ser maior que a Data/Hora de Início. |
| ***msg\_equipe\_confirmarRemocaoMembro*** | Você tem certeza que deseja remover esse usuário da equipe? |
| ***msg\_equipe\_sair\_confirmar*** | Você tem certeza que deseja sair da equipe? |
| ***msg\_equipe\_usuarioSairConfirma*** | Você saiu da equipe {nomeDaEquipe}. |
| ***msg\_localizacao\_vazia*** | O usuário selecionado não esteve em qualquer lugar. |
| ***msg\_chat\_erroEnvio*** | Não foi possível enviar sua mensagem. Tente novamente ou então atualize a página. |
| ***msg\_tarefa\_concluir*** | Você tem certeza que deseja concluir esta tarefa? |
| ***msg\_local\_falha\_excluirTarefaVinculada*** | Este local não pode ser excluído pois está vinculado a uma tarefa. |
| ***msg\_senha\_invalida*** | A senha informada não corresponde |
| ***msg\_equipe\_conviteAceito*** | Agora você é um membro da equipe {{ nomeDaEquipe }}. |
| ***msg\_equipe\_conviteRecusado*** | Agora recusou o convite da equipe {{ nomeDaEquipe }}. |

Fonte: O autor (2017)

# PROTÓTIPOS DE INTERFACE

Este capítulo está destinado a exibir de forma mais detalhada e completa os protótipos de interface mencionados anteriormente. Cada figura representa o protótipo de uma das telas do sistema.

## PROTÓTIPOS

A Figura 12 apresenta o protótipo da tela ‘Cadastro Inicial’, a qual permite que um novo usuário se cadastre no sistema.

Uma imagem contendo captura de tela, equipamentos eletrônicos

Descrição gerada com muito alta confiança

Figura 12 - Protótipo de 'Cadastro Inicial'

Fonte: O autor (2017)

A Figura 13 apresenta o protótipo da tela ‘Login’, que permite aos usuários a autenticação no sistema. Esta é também a tela inicial.

Uma imagem contendo monitor, captura de tela, telefone, computador

Descrição gerada com muito alta confiança

Figura 13 - Protótipo de 'Login'

Fonte: O autor (2017)

A Figura 14 apresenta o protótipo ‘Menu’ onde estarão disponíveis todas as equipes (contextos) que o usuário está vinculado, seja como membro ou administrador.

Uma imagem contendo equipamentos eletrônicos, monitor

Descrição gerada com muito alta confiança

Figura 14 - Protótipo de 'Menu'

Fonte: O autor (2017)

A Figura 15 apresenta o protótipo ‘Membros da Equipe’, o qual permite que qualquer membro da equipe visualize as informações do contexto em que está inserido.

Uma imagem contendo captura de tela

Descrição gerada com muito alta confiança

Figura 15 - Protótipo de 'Membros da Equipe'

Fonte: O autor (2017)

A Figura 16 apresenta o protótipo ‘Chat’, que é a área onde irá ocorrer a troca de mensagens instantâneas entre os membros de uma equipe.

Uma imagem contendo captura de tela

Descrição gerada com muito alta confiança

Figura 16 - Protótipo de 'Chat'

Fonte: O autor (2017)

A Figura 17 apresenta o protótipo ‘Contexto de Equipe’. Local onde será apresentado um mapa contendo as informações do contexto tais como locais da equipe e a localização dos membros.

Uma imagem contendo monitor, equipamentos eletrônicos, interior, computador

Descrição gerada com muito alta confiança

Figura 17 - Protótipo de 'Contexto da Equipe'

Fonte: O autor (2017)

A Figura 18 apresenta o protótipo ‘Manter Tarefa’, o qual permite que o usuário administrador crie e designe novas tarefas aos demais membros da equipe.

Uma imagem contendo captura de tela, monitor, equipamentos eletrônicos

Descrição gerada com muito alta confiança

Figura 18 - Protótipo de 'Manter Tarefa'

Fonte: O autor (2017)

A Figura 19 apresenta o protótipo ‘Tarefas’, que permite aos membros do contexto a visualização das tarefas existentes. Ainda, é possível indicar a conclusão da tarefa.

Uma imagem contendo monitor, captura de tela

Descrição gerada com muito alta confiança

Figura 19 - Protótipo de 'Tarefas'

Fonte: O autor (2017)

A Figura 20 apresenta o protótipo ‘Manter Local’, que permite ao usuário administrador a definição de locais relevantes ao contexto.

Uma imagem contendo monitor, captura de tela, equipamentos eletrônicos, telefone

Descrição gerada com muito alta confiança

Figura 20 - Protótipo de 'Manter Local'

Fonte: O autor (2017)

A Figura 21 apresenta o protótipo ‘Locais’, que permite que os membros do contexto saibam as localizações relevantes ao mesmo.

Uma imagem contendo monitor, captura de tela, equipamentos eletrônicos

Descrição gerada com muito alta confiança

Figura 21 - Protótipo de 'Locais'

Fonte: O autor (2017)

# DIAGRAMAS DE CASOS DE USO

A seguir serão apresentados os casos de uso da aplicação. Para tal, dois diagramas macro foram criados. O primeiro na Figura 22 apresenta as funcionalidades relativas ao usuário membro da equipe. O segundo, na Figura 23 apresenta as funcionalidades relativas ao usuário administrador da equipe. Nota-se que no segundo diagrama que o usuário administrador pode realizar todos os casos de uso de um usuário membro da equipe.

Uma imagem contendo texto, mapa

Descrição gerada com muito alta confiança

Figura 22 - Diagrama de casos de uso do usuário Membro da Equipe

Fonte: O autor (2017)

Uma imagem contendo texto

Descrição gerada com muito alta confiança

Figura 23 - Diagrama de casos de uso do usuário Administrador da Equipe

Fonte: O autor (2017)

Visando o melhor entendimento, os diagramas foram divididos em casos de uso micro, onde é possível visualizar que os Casos de Uso do tipo “manter” foram expandidos para, por exemplo “inserir”, “alterar” ou “remover”.

Os casos de uso são apresentados da Tabela 22 até a Tabela 40.

## DIAGRAMAS DE CASO DE USO DE USUÁRIO

Os casos de uso aqui descritos são a partir do ponto de vista de um usuário, possibilitando que este se cadastre, autentique e altere suas informações. A Figura 24 apresenta a visão geral destes casos de uso sendo seus detalhamentos apresentados da Tabela 22 até a Tabela 25.

Uma imagem contendo texto, mapa

Descrição gerada com muito alta confiança

Figura 24 - Diagrama de casos de uso de ‘Usuário’

Fonte: O autor (2017)

Tabela 22 - Caso de uso ‘Login’

|  |  |
| --- | --- |
| 1. Login | |
| **Ator principal** | Usuário |
| **Resumo** | Esse caso de uso descreve as etapas para o usuário autenticar no sistema.  Caso de uso guiado pelo requisito RF2 - Login. |
| **Pré-condições** | * Estar na tela ‘Login’. |
| **Pós-condições** | Usuário autenticado no sistema. |
| **Fluxo principal** | |
| 01. [U] Informar e-mail e senha. | |
| 02. [U] Solicitar a autenticação: clicar no botão ‘Acessar’. | |
| 03. [S] Validar e-mail e senha informados. | |
| 04. [S] Autenticar o usuário. | |
| 05. [S] Redirecionar o usuário para a Página Inicial. | |
| **Fluxo de Exceção – Usuário e/ou senha não informado(os)** | |
| 03.a.1. [S] Informar o usuário por meio da mensagem ***msg\_campos\_invalidos*** que os dados de autenticação devem ser preenchidos | |
| 03.a.2. Retornar ao passo 1. | |
| **Fluxo de Exceção – Dados de Login inválidos** | |
| 04.a.1. [S] Informar a pessoa por meio da mensagem ***msg\_login\_falha*** que os dados de autenticação são inválidos. | |
| 04.a.2. Retornar ao passo 1. | |

Fonte: O autor (2017)

Tabela 23 - Caso de uso ‘Recuperar Senha’

|  |  |
| --- | --- |
| 1. Recuperar Senha | |
| **Ator principal** | Usuário |
| **Resumo** | Esse caso de uso descreve as etapas para o usuário alterar sua senha por meio de uma solicitação, a qual ele recebe um e-mail, acessa um link e então informa sua nova senha.  Caso de uso guiado pelo requisito RF3 - Recuperar Senha. |
| **Pré-condições** | * Estar na tela ‘Login’. |
| **Pós-condições** | Usuário com senha alterada permitindo seu acesso ao sistema. |
| **Fluxo principal** | |
| 01. [U] Clica em “Esqueceu sua Senha?”. | |
| 02. [S] Retorna a tela modal para solicitar a alteração de senha. | |
| 03. [U] Informa o e-mail vinculado ao seu usuário. | |
| 04. [U] Solicitar a alteração: clicar no botão ‘Enviar’. | |
| 05. [S] Valida as informações. | |
|  | |
| 06. [S] Enviar e-mail com link para alteração de senha do usuário | |
| 07. [S] Redireciona o usuário para a tela ‘Login’ e apresenta a mensagem ***msg\_recuperarSenha\_sucess\_emailEnviado***. | |
| 08. [U] Acessar o e-mail recebido e então clicar no link para alteração de senha. | |
| 09. [S] Sistema web valida a solicitação. | |
| 10. [S] Sistema web retorna a tela para informar a nova senha. | |
| 11. [U] Informa a senha. | |
| 12. [U] Clicar em “Confirmar”, solicitando a alteração. | |
| 13. [S] Valida as informações. | |
| 14. [S] Altera a senha do usuário | |
| **Fluxo de Exceção – E-mail inválido** | |
| 05.a.1. [S] Informar o usuário por meio da mensagem ***msg\_campos\_invalidos*** quais os campos estão com informações inválida. | |
| 05.a.2. Retorna ao passo 02. | |
| **Fluxo de Exceção – E-mail não vinculado** | |
| 05.b.1. [S] Informar ao usuário por meio da mensagem ***msg\_recuperarSenha\_usuarioNaoExiste*** que não há nenhuma conta com o e-mail informado. | |
| 05.b.2. Retornar ao passo 02. | |
| **Fluxo de Exceção – Solicitação já utilizada** | |
| 10.a.1. [S] Informar ao usuário por meio da mensagem ***msg\_solicitacaoExpirada*** que a solicitação de alteração de senha já foi utilizada e por isso não é mais válida. | |
| 10.a.2. Retornar ao passo 01. | |
| **Fluxo de Exceção – Senha inválida** | |
| 13.a.1. [S] Informar ao usuário por meio da mensagem ***msg\_campos\_invalidos*** quais os campos estão com informações inválida. | |
| 13.a.2. Retorna ao passo 11. | |
| **Fluxo de Exceção – Senha fraca** | |
| 13.b.1. [S] Informar o usuário por meio da mensagem ***msg\_usuario\_senhaFraca*** que a senha informada não atende as exigências mínimas. | |
| 13.b.2. Retornar ao passo 11. | |

Fonte: O autor (2017)

Tabela 24 - Caso de uso ‘Cadastrar Usuário’

|  |  |
| --- | --- |
| 1. Cadastrar Usuário | |
| **Ator principal** | Usuário |
| **Resumo** | Esse caso de uso descreve as etapas para um usuário realizar seu cadastro no sistema.  Caso de uso guiado pelo requisito RF1 - Cadastro Inicial. |
| **Pré-condições** | * Estar na tela ‘Cadastre-se’. |
| **Pós-condições** | Novo usuário cadastrado no sistema. |
| **Fluxo principal** | |
| 01. [U] Informar os dados mínimos obrigatórios para cadastro. | |
| 02. [U] Solicitar a gravação: clicar no botão ‘Cadastre-se’. | |
| 03. [S] Validar os dados informados pelo usuário. | |
| 04. [S] Cadastrar o usuário no sistema. | |
| 05. [S] Enviar e-mail com link para confirmação. | |
| 06. [U] Confirmar o endereço de e-mail clicando no link enviado. | |
| 07. [S] Ativar o cadastro do usuário. | |
| **Fluxo de Exceção – Informações Inválidas** | |
| 03.a.1. [S] Informar ao usuário quais os campos estão com informações inválida. Por meio da mensagem ***msg\_campos\_invalidos***. | |
| 03.a.2. Retorna ao passo 01 | |
| **Fluxo de Exceção – Informações Inválidas** | |
| 03.b.1. [S] Informar ao usuário por meio da mensagem ***msg\_usuario\_emaiJaCadastrado*** que o e-mail informado já está sendo utilizado. | |
| 03.b.2. Retorna ao passo 01 | |
| **Fluxo de Exceção – Senha fraca** | |
| 03.c.1. [S] Informar o usuário por meio da mensagem ***msg\_usuario\_senhaFraca*** que a senha informada não atende as exigências mínimas. | |
| 03.c.2. Retornar ao passo 01. | |

Fonte: O autor (2017)

Tabela 25 - Caso de uso ‘Alterar Usuário’

|  |  |
| --- | --- |
| 1. Alterar Usuário | |
| **Ator principal** | Usuário |
| **Resumo** | Esse caso de uso descreve as etapas para um usuário alterar os dados de sua conta.  Caso de uso guiado pelo requisito RF4 - Perfil. |
| **Pré-condições** | * Ter uma conta no sistema. * Estar autenticado na aplicação. * Estar na tela ‘Perfil’. |
| **Pós-condições** | Conta do usuário alterada. |
| **Fluxo principal** | |
| 01. [U] Alterar as informações da conta. | |
| 02. [U] Solicitar a gravação: clicar no botão ‘Alterar’. | |
| 03. [S] Validar os dados informados pelo usuário. | |
| 04. [S] Atualizar o usuário na aplicação. | |
| **Fluxo de Exceção – Informações Inválidas** | |
| 03.a.1. [S] Informar ao usuário quais os campos estão com informações inválida. Por meio da mensagem ***msg\_campos\_invalidos***. | |
| 03.a.2. Retorna ao passo 01 | |

Fonte: O autor (2017)

Tabela 26 - Caso de uso ‘Alterar E-mail’

|  |  |
| --- | --- |
| 1. Alterar E-mail | |
| **Ator principal** | Usuário |
| **Resumo** | Esse caso de uso descreve as etapas para um usuário alterar seu e-mail na aplicação.  Caso de uso guiado pelo requisito RF5 - Alterar E-mail. |
| **Pré-condições** | * Ter uma conta no sistema. * Estar autenticado na aplicação. * Estar na tela ‘Alterar E-mail’. |
| **Pós-condições** | E-mail do usuário alterado. |
| **Fluxo principal** | |
| 01. [U] Informar o novo email e senha atual | |
| 02. [U] Solicitar a alteração: clicar no botão ‘Alterar e-mail’. | |
| 03. [S] Validar os dados informados pelo usuário. | |
| 04. [S] Atualizar o usuário na aplicação e enviar e-mail solicitando confirmação da alteração. | |
| 05. [S] Redirecionar para a tela “Login”. | |
| 06. [U] Acessar o e-mail enviado e clicar no link para confirmação da alteração. | |
| 07. [S] Validar o link. | |
| 08. [S] Consistir a alteração de e-mail do usuário | |
| **Fluxo de Exceção – Informações Inválidas** | |
| 03.a.1. [S] Informar ao usuário quais os campos estão com informações inválida. Por meio da mensagem ***msg\_campos\_invalidos***. | |
| 03.a.2. Retorna ao passo 01 | |
| **Fluxo de Exceção – Informações Inválidas** | |
| 03.b.1. [S] Informar ao usuário por meio da mensagem ***msg\_usuario\_emaiJaCadastrado*** que o e-mail informado já está sendo utilizado. | |
| 03.b.2. Retorna ao passo 01 | |
| **Fluxo de Exceção – Solicitação já utilizada** | |
| 07.a.1. [S] Informar ao usuário por meio da mensagem ***msg\_solicitacaoExpirada*** que a solicitação de alteração de e-mail já foi utilizada e por isso não é mais válida. | |
| 07.a.2. Retornar ao passo 01. | |

Fonte: O autor (2017)

Tabela 27 - Caso de uso ‘Alterar Senha’

|  |  |
| --- | --- |
| 1. Alterar Senha | |
| **Ator principal** | Usuário |
| **Resumo** | Esse caso de uso descreve as etapas para um usuário alterar sua senha na aplicação.  Caso de uso guiado pelo requisito RF6 - Alterar Senha. |
| **Pré-condições** | * Ter uma conta no sistema. * Estar autenticado na aplicação. * Estar na tela ‘Alterar Senha. |
| **Pós-condições** | Senha do usuário alterada. |
| **Fluxo principal** | |
| 01. [U] Informar a nova senha e a senha atual. | |
| 02. [U] Solicitar a alteração: clicar no botão ‘Alterar senha’. | |
| 03. [S] Validar os dados informados pelo usuário. | |
| 04. [S] Atualizar o usuário na aplicação. | |
| 05. [S] Redirecionar para a tela “Perfil”. | |
| **Fluxo de Exceção – Informações Inválidas** | |
| 03.a.1. [S] Informar ao usuário quais os campos estão com informações inválida. Por meio da mensagem ***msg\_campos\_invalidos***. | |
| 03.a.2. Retorna ao passo 01 | |
| **Fluxo de Exceção – Senha fraca** | |
| 03.b.1. [S] Informar o usuário por meio da mensagem ***msg\_usuario\_senhaFraca*** que a nova senha informada não atende as exigências mínimas. | |
| 03.b.2. Retornar ao passo 01. | |
| **Fluxo de Exceção – Senha atual não corresponde** | |
| 03.c.1. [S] Informar o usuário por meio da mensagem ***msg\_senha\_invalida*** que a senha informada não corresponde com a senha atual. | |
| 03.c.2. Retornar ao passo 01. | |

Fonte: O autor (2017)

## DIAGRAMAS DE CASOS DE USO DE GERENCIAMENTO DE EQUIPE

Os casos de uso aqui apresentados são referentes a dois pontos de vista: o primeiro sendo de um usuário qualquer, possibilitando a ele gerenciar as equipes as quais ele está inserido; e o ponto de vista de um usuário administrador, que consiste em ser um usuário comum, porém, com poderes de administrador somente sobre as equipes as quais este mesmo usuário criou. A Figura 25 apresenta a visão geral destes casos de uso sendo seus detalhamentos apresentados da Tabela 28 até a Tabela 30.

Uma imagem contendo texto, mapa

Descrição gerada com muito alta confiança

Figura 25 - Diagrama de casos de uso de ‘Gerenciamento de Equipe’

Fonte: O autor (2017)

Tabela 28 - Caso de uso ‘Manter Equipe (Criar)’

|  |  |
| --- | --- |
| 1. Manter Equipe | |
| **Ator principal** | Usuário |
| **Resumo** | Esse caso de uso descreve as etapas para um usuário criar ou alterar uma equipe  Caso de uso guiado pelo requisito RF7 - Manter Equipe. |
| **Pré-condições** | * Estar autenticado; * Estar na tela de "Manter Equipe". |
| **Pós-condições** | Usuário passa a ser administrador da equipe criada. |
| **Fluxo principal** | |
| 01. [U] Informar os dados mínimos para criação da equipe. | |
| 02. [U] Solicitar a gravação: clicar no botão ‘Gravar’. | |
| 03. [S] Validar os dados informados pelo usuário. | |
| 04. [S] Criar a Equipe no sistema. | |
| 05. [S] Redirecionar o Usuário para o contexto da equipe criada. | |
| **Fluxo de Exceção – Informações Inválidas** | |
| 03.a.1. [S] Informar ao usuário por meio da mensagem ***msg\_campos\_invalidos*** quais os campos estão com informações inválidas. | |
| 03.a.2. Retorna ao passo 1 | |
| **Fluxo de Exceção – Data/Hora Final errada** | |
| 03.b.1. [S] Informar ao usuário por meio da mensagem ***msg\_equipe\_datahora\_menor*** que a Data/Hora Final é maior que a Data/Hora Inicial. | |
| 03.b.2. Retorna ao passo 1 | |

Fonte: O autor (2017)

Tabela 29 - Caso de Uso ‘Manter Equipe (Remover)’

|  |  |
| --- | --- |
| 1. Manter Equipe (Remover) | |
| **Ator principal** | Usuário |
| **Resumo** | Esse caso de uso descreve as etapas para um usuário administrador excluir uma equipe e todas as suas informações.  Caso de uso guiado pelo requisito RF7 - Manter Equipe. |
| **Pré-condições** | * Estar autenticado na aplicação; * Estar no contexto de uma equipe; * Ator ser administrador da equipe; * Estar na tela de ‘Manter Equipe’. |
| **Pós-condições** | Equipe removida da aplicação |
| **Fluxo principal** | |
| 01. [U] Solicitar a exclusão: clicar no botão ‘Excluir’. | |
| 02. [S] Excluir a Equipe do sistema. | |

Fonte: O autor (2017)

Tabela 30 - Caso de uso ‘Remover Usuário’

|  |  |
| --- | --- |
| 1. Remover Usuário | |
| **Ator principal** | Usuário administrador |
| **Resumo** | Esse caso de uso descreve as etapas para um usuário administrador remover um usuário membro de uma equipe. |
| **Pré-condições** | * Estar autenticado na aplicação; * Ser administrador da equipe; * Estar na tela de ‘Manter Equipe’; * Usuário que será removido ser membro da equipe. |
| **Pós-condições** | Usuário membro não ter mais acesso as funcionalidades |
| **Fluxo principal** | |
| 01. [U] Localiza o usuário na listagem de membros da equipe. | |
| 02. [U] Clica no botão ‘Remover usuário’, disparando o evento. | |
| 03. [S] Solicita confirmação da ação por meio da mensagem ***msg\_equipe\_confirmarRemocaoMembro***. | |
| 04. [U] Administrador da equipe autoriza remoção do usuário. | |
| 05. [S] Remove o usuário da equipe. | |
| **Fluxo Alternativo - Administrador não autoriza a Remoção** | |
| 04.a.1. [U] Administrador não autoriza a remoção | |
| 04.a.2. Retorna ao passo 01 | |

Fonte: O autor (2017)

Tabela 31 - Caso de uso ‘Sair da Equipe’

|  |  |
| --- | --- |
| 1. Sair da Equipe | |
| **Ator principal** | Usuário |
| **Resumo** | Esse caso de uso descreve as etapas para um usuário sair de uma equipe a qual está vinculado. |
| **Pré-condições** | * Estar autenticado; * Estar vinculado a uma equipe como membro; * Estar no contexto de uma equipe e na tela “Membros da Equipe”. |
| **Pós-condições** | Usuário deixa de fazer parte de determinada equipe. |
| **Fluxo principal** | |
| 01. [U] Clicar no botão ‘Sair’, requisitando a desvinculação. | |
| 02. [S] Solicitar confirmação da ação por meio da mensagem ***msg\_equipe\_sair\_confirmar***. | |
| 03. [U] Confirmar ação. | |
| 04. [S] Realiza a desvinculação. | |
| 05. [S] Retorna mensagem ***msg\_equipe\_usuarioSairConfirma*** confirmando a ação | |
| 06. [S] Atualiza listagem de Equipes do usuário. | |
| **Fluxo Alternativo – Cancelamento** | |
| 03.a.1. [U] Usuário cancela a ação | |
| 03.a.2. Retorna ao passo 1 | |

Fonte: O autor (2017)

Tabela 32 - Caso de uso ‘Convidar Usuário’

|  |  |
| --- | --- |
| 1. Convidar Usuário | |
| **Ator principal** | Usuário |
| **Resumo** | Esse caso de uso descreve as etapas para um usuário administrador enviar convites a outros usuários para adentrarem a uma determinada equipe.  Caso de uso guiado pelo RF8 - Convidar. |
| **Pré-condições** | * Ator estar autenticado na aplicação; * Ator ser administrador da equipe; * Estar na tela de "Manter Equipe". |
| **Pós-condições** | Convite enviado para os usuários. |
| **Fluxo principal** | |
| 01. [U] Clicar no botão "Convidar". | |
| 02. [S] Retornar a tela "Convidar". | |
| 03. [U] Informar os e-mails dos usuários que serão convidados. | |
| 04. [S] Clicar em "Convidar", disparando o evento. | |
| 05. [S] Validar e-mails informados. | |
| 06. [S] Registrar os convites enviados. | |
| **Fluxo de Exceção – E-mails informados incorretamente** | |
| 05.a.1. [S] Informar ao usuário quais e-mails são inválidos por meio da mensagens ***msg\_campos\_invalidos***. | |
| 05.a.2. Retorna ao passo 03 | |

Fonte: O autor (2017)

Tabela 33 - Caso de uso ‘Aceitar Convite’

|  |  |
| --- | --- |
| 1. Aceitar Convite | |
| **Ator principal** | Usuário normal |
| **Resumo** | Esse caso de uso descreve as etapas para um usuário normal entrar em uma equipe por meio de um convite recebido.  Caso de uso guiado pelo requisito RF8 - Convidar. |
| **Pré-condições** | * Ator ter recebido um convite conforme RF8 - Convidar. |
| **Pós-condições** | Usuário como membro de determinada equipe. |
| **Fluxo principal** | |
| 01. [U] Acessa a tela “Meus convites” | |
| 02. [S] Retorna os convites do usuário. | |
| 03. [U] Seleciona o convite | |
| 04. [U] Aceita o convite. | |
| 05. [S] Processa a solicitação e apresenta a mensagem ***msg\_equipe\_conviteAceito.*** | |
| **Fluxo de Alternativo – Recusar Convite** | |
| 04.a.1. [S] Usuário recusa o convite | |
| 04.a.2. Processa a solicitação e apresenta a mensagem ***msg\_equipe\_conviteRecusado.*** | |

Fonte: O autor (2017)

## DIAGRAMAS DE CASOS DE USO DE EQUIPE

Os casos de uso aqui descritos envolvem funcionalidades inerentes a equipe como um todo, porém, ainda há restrições quando ao nível de acesso destas funcionalidades por tipo de usuário. Os diagramas de caso de uso micro estão representados da Figura 26 até a Figura 29 enquanto seus detalhamentos são apresentados da Tabela 34 até a Tabela 40.

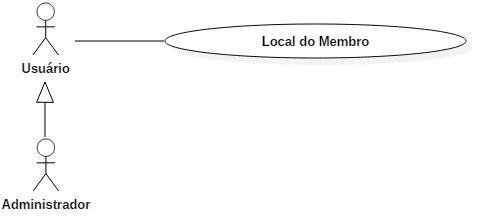


Figura 26 - Diagrama de casos de uso de ‘Membro’

Fonte: O autor (2017)

Tabela 34 - Caso de uso ‘Local do Membro’

|  |  |
| --- | --- |
| 1. Local do Membro | |
| **Ator principal** | Usuário |
| **Resumo** | Esse caso de uso descreve as etapas para que um membro da equipe visualize no Mapa do contexto a localização atual de outro membro da equipe. |
| **Pré-condições** | * Estar autenticado na aplicação; * Estar no contexto de uma equipe; * Estar na tela de ‘Membros da Equipe’. (**Erro! Fonte de referência não encontrada.**) |
| **Pós-condições** | Permite visualizar a última localização disponível de um membro da equipe. |
| **Fluxo principal** | |
| 01. [U] Localiza na listagem o membro que se deseja ver a localização. | |
| 02. [U] Clica no botão ‘Local Atual’, disparando o evento. | |
| 03. [S] Valida se há uma última localização para o usuário selecionado. | |
| 04. [S] Apresentar no Mapa do contexto a localização do usuário. | |
| **Fluxo de Exceção – Usuário não possui localização** | |
| 03.a.1. [U] Informar ao usuário por meio da mensagem ***msg\_localizacao\_vazia*** que o membro selecionado não possui nenhum registro de localizações | |
| 03.a.2. Retorna ao passo 1 | |

Fonte: O autor (2017)

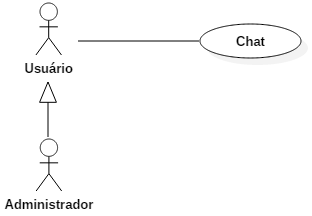


Figura 27 - Diagrama de casos de uso de ‘Chat’

Fonte: O autor (2017)

Tabela 35 - Caso de uso ‘Chat’

|  |  |
| --- | --- |
| 1. Chat | |
| **Ator principal** | Usuário |
| **Resumo** | Esse caso de uso descreve as etapas para que um membro da equipe envie mensagens no chat da equipe. |
| **Pré-condições** | * Estar autenticado na aplicação; * Estar no contexto de uma equipe; * Estar na tela de "Chat". |
| **Pós-condições** | Mensagem enviada para os demais membros da equipe. |
| **Fluxo principal** | |
| 01. [U] Informar a mensagem que será enviada. | |
| 02. [U] Solicitar envio da mensagem: clicar no botão ‘Enviar’. | |
| 03. [S] Enviar a mensagem. | |
| 04. [S] Atualizar o chat apresentando a nova mensagem. | |
| **Fluxo de Exceção – Erro no envio da mensagem** | |
| 03.a.1. [U] Informar ao usuário por meio da mensagem ***msg\_chat\_erroEnvio*** que a mensagem não pode ser enviada. | |
| 03.a.2. Retorna ao passo 04. | |

Fonte: O autor (2017)

Uma imagem contendo mapa, texto

Descrição gerada com alta confiança

Figura 28 - Diagrama de casos de uso de ‘Tarefa’

Fonte: O autor (2017)

Tabela 36 - Caso de uso ‘Manter Tarefa’

|  |  |
| --- | --- |
| 1. Manter Tarefa | |
| **Ator principal** | Usuário |
| **Resumo** | Esse caso de uso descreve as etapas para que um usuário administrador crie ou altere uma tarefa.  Caso de uso guiado pelo requisito RF14 - Manter Tarefa. |
| **Pré-condições** | * Estar autenticado na aplicação; * Estar no contexto de uma equipe; * Estar na tela de ‘Tarefas’. |
| **Pós-condições** | Nova tarefa criada no contexto. |
| **Fluxo principal** | |
| 01. [U] Usuário clica no botão ‘Adicionar Tarefa’ (+). | |
| 02. [S] Sistema retorna a tela de Manter Tarefa. | |
| 03. [U] Informar os dados mínimos responsáveis. | |
| 04. [U] Solicitar a gravação clicando no botão “Gravar”. | |
| 05. [S] Validar os dados informados pelo usuário. | |
| 06. [S] Grava a tarefa no sistema. | |
| 07. [S] Retorna para a tela ‘Tarefas’. | |
| **Fluxo de Alternativo – Editar Tarefa** | |
| 01.a.1. [U] Localiza a tarefa na lista de tarefas. | |
| 01.a.2. [U] Clica sobre a tarefa. | |
| 01.a.3. Continua no passo 2 do fluxo principal. | |
| **Fluxo de Exceção – Informações Inválidas** | |
| 05.a.1. [S] Informar ao usuário por meio da mensagem ***msg\_campos\_invalidos*** quais os campos estão com informações inválida. | |
| 05.a.2. Retorna ao passo 3. | |

Fonte: O autor (2017)

Tabela 37 - Caso de uso ‘Concluir Tarefa’

|  |  |
| --- | --- |
| 1. Concluir Tarefa | |
| **Ator principal** | Usuário |
| **Resumo** | Esse caso de uso descreve as etapas para que um usuário responsável por uma tarefa realize a conclusão da mesma. |
| **Pré-condições** | * Ter selecionado uma equipe; * Estar na tela de "Tarefas"; * Ser um usuário responsável pela tarefa. |
| **Pós-condições** | Tarefa concluída no sistema. |
| **Fluxo principal** | |
| 01. [U] Localiza a tarefa na listagem. | |
| 02. [U] Clica no campo de marcaçãorequisitando a conclusão. | |
| 03. [S] Solicita confirmação do usuário por meio da mensagem ***msg\_tarefa\_concluir***. | |
| 04. [U] Confirma conclusão. | |
| 05. [S] Grava a conclusão. | |
| 06. [S] Atualiza a tela ‘Tarefas’. | |
| **Fluxo Alternativo – Concluir Tarefa pela ‘Manutenção de Tarefa’** | |
| 01.a.1. [U] Usuário clica sobre a tarefa, requisitando a tela 'Manter Tarefa'. | |
| 01.a.2. [S] Retorna a tela 'Manter Tarefa'. | |
| 01.a.3. Continua no passo 2. | |
| **Fluxo Alternativo – Usuário cancela a conclusão** | |
| 03.a.1. [U] Usuário cancela a ação. | |
| 03.a.2. Retorna ao passo 1. | |

Fonte: O autor (2017)

Tabela 38 - Caso de uso ‘Cancelar Tarefa’

|  |  |
| --- | --- |
| 1. Cancelar Tarefa | |
| **Ator principal** | Usuário |
| **Resumo** | Esse caso de uso descreve as etapas para que um usuário administrador cancele uma tarefa. |
| **Pré-condições** | * Ter selecionado uma equipe; * Estar na tela de "Tarefas"; * Ser um usuário administrador. |
| **Pós-condições** | Tarefa cancelada no sistema. |
| **Fluxo principal** | |
| 01. [U] Localiza a tarefa na listagem. | |
| 02. [U] Clica sobre a tarefa, requisitando a tela 'Manter Tarefa'. | |
| 03. [S] Retorna a tela 'Manter Tarefa'. | |
| 04. [S] Clica no botão ‘Cancelar’. | |
| 05. [S] Solicita confirmação do usuário. | |
| 06. [U] Confirma a ação. | |
| 07. [S] Grava o cancelamento da Tarefa | |
| **Fluxo Alternativo – Usuário cancela o cancelamento** | |
| 6.a.1. [U] Usuário cancela a ação. | |
| 6.a.2. Retorna ao passo 1. | |

Fonte: O autor (2017)

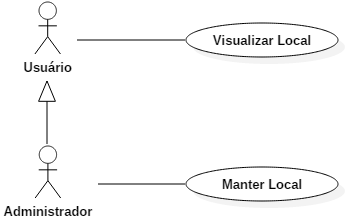


Figura 29 - Diagrama de casos de uso de ‘Local’

Fonte: O autor (2017)

Tabela 39 - Caso de uso ‘Localização do Local’

|  |  |
| --- | --- |
| 1. Localização do Local | |
| **Ator principal** | Usuário |
| **Resumo** | Esse caso de uso descreve as etapas para que um membro da equipe visualize no Mapa do contexto onde é determinado local. |
| **Pré-condições** | * Estar autenticado na aplicação; * Estar no contexto de uma equipe; * Estar na tela de "Locais". (**Erro! Fonte de referência não encontrada.**) |
| **Pós-condições** | Permite visualizar onde é o local selecionado. |
| **Fluxo principal** | |
| 01. [U] Localiza na listagem o local que se deseja ver a localização. | |
| 02. [U] Clica no botão ‘Local’, disparando o evento. | |
| 03. [S] Apresentar no Mapa do contexto a área do local. | |

Fonte: O autor (2017)

Tabela 40 - Caso de uso ‘Manter Local’

|  |  |
| --- | --- |
| 1. Manter Local | |
| **Ator principal** | Usuário |
| **Resumo** | Este caso de uso descrever o processo de criação e alteração de um Local. |
| **Pré-condições** | * Estar autenticado na aplicação; * Estar no contexto de uma equipe; * Estar na tela de "Locais". |
| **Pós-condições** | Novo local vinculado a equipe. |
| **Fluxo principal** | |
| 01. [U] Usuário clica no botão ‘Adicionar Local’ (+). | |
| 02. [S] Sistema retorna a tela ‘Manter Locais’. | |
| 03. [U] Informar os dados: nome e localização. | |
| 04. [U] Solicitar a gravação: clicar no botão ‘Alterar/Criar’. | |
| 05. [S] Validar os dados informados pelo usuário. | |
| 06. [S] Cadastrar o local no sistema. | |
| 07. [S] Retorna para a tela ‘Locais’. | |
| **Fluxo de Alternativo – Editar Tarefa** | |
| 01.a.1. [U] Localiza o local na lista de locais. | |
| 01.a.2. [U] Clica sobre o local. | |
| 01.a.3. Continua no passo 2 do fluxo principal. | |
| **Fluxo de Exceção – Informações Inválidas** | |
| 05.a.1. [S] Informar o usuário quais os campos estão com informações inválida. | |
| 05.a.2. Retorna ao passo 3. | |

Fonte: O autor (2017)

# DIAGRAMAS DE CLASSES

A seguir será apresentado o diagrama de classes do sistema. Os diagramas estão documentados com a finalidade de representar a estrutura e a relação dos objetos do sistema. Da Figura 30 até a Figura 33 estão representadas respectivamente as seguintes camadas da aplicação:

Modelos (*Models*): representa as classes essenciais para funcionamento do sistema, são elas que definem como os objetos estão relacionados e suas dependências.

Páginas (*Pages*): representa as interfaces da aplicação, nela estão as interações entre homem e máquina.

Provedores ou Serviços (*Provider* ou *Service*): representa a camada de serviços onde está a lógica do sistema e o acesso ao servidor.

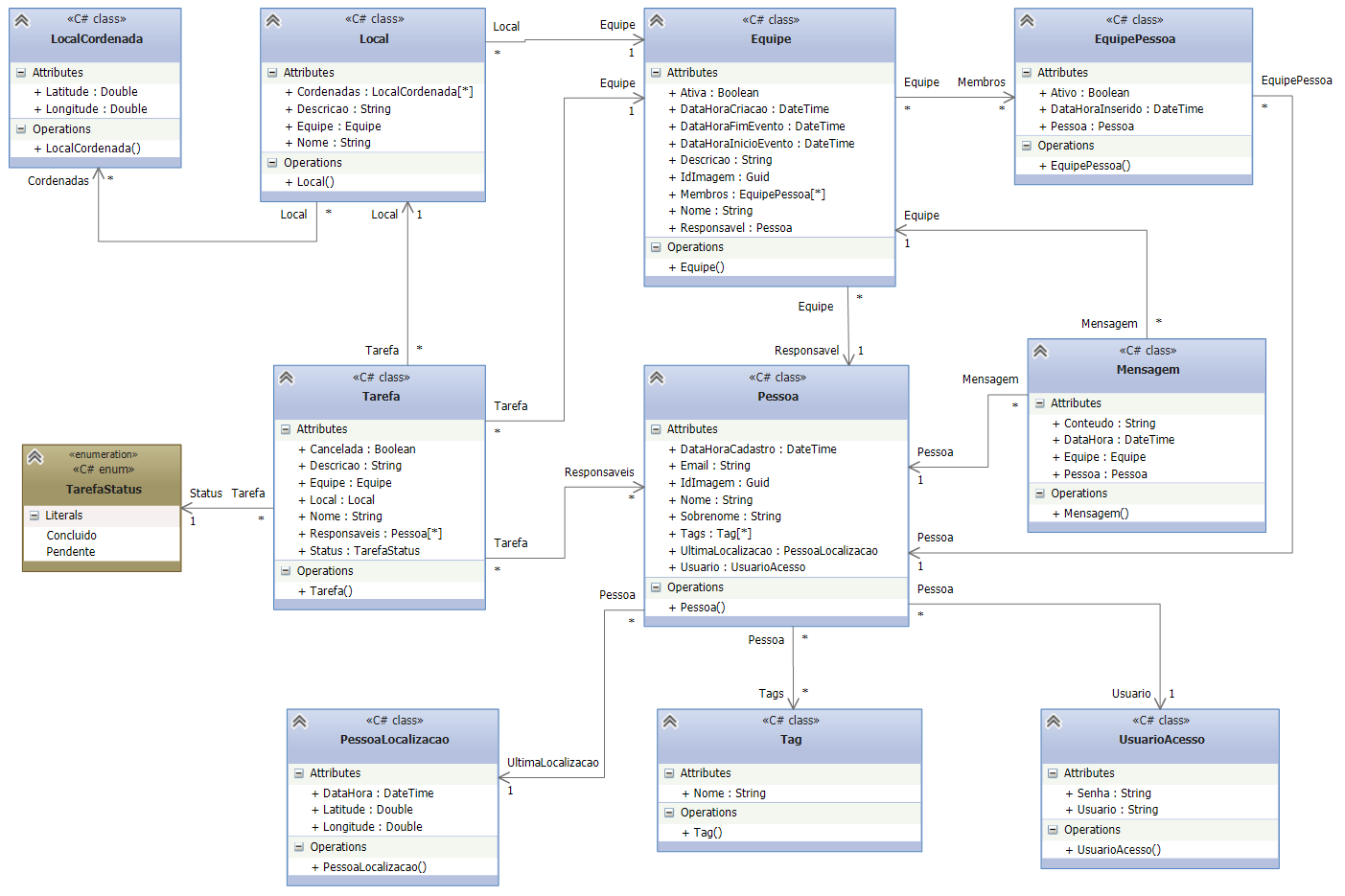


Figura 30 - Diagrama de Classe da camada de Modelos

Fonte: O autor (2017)

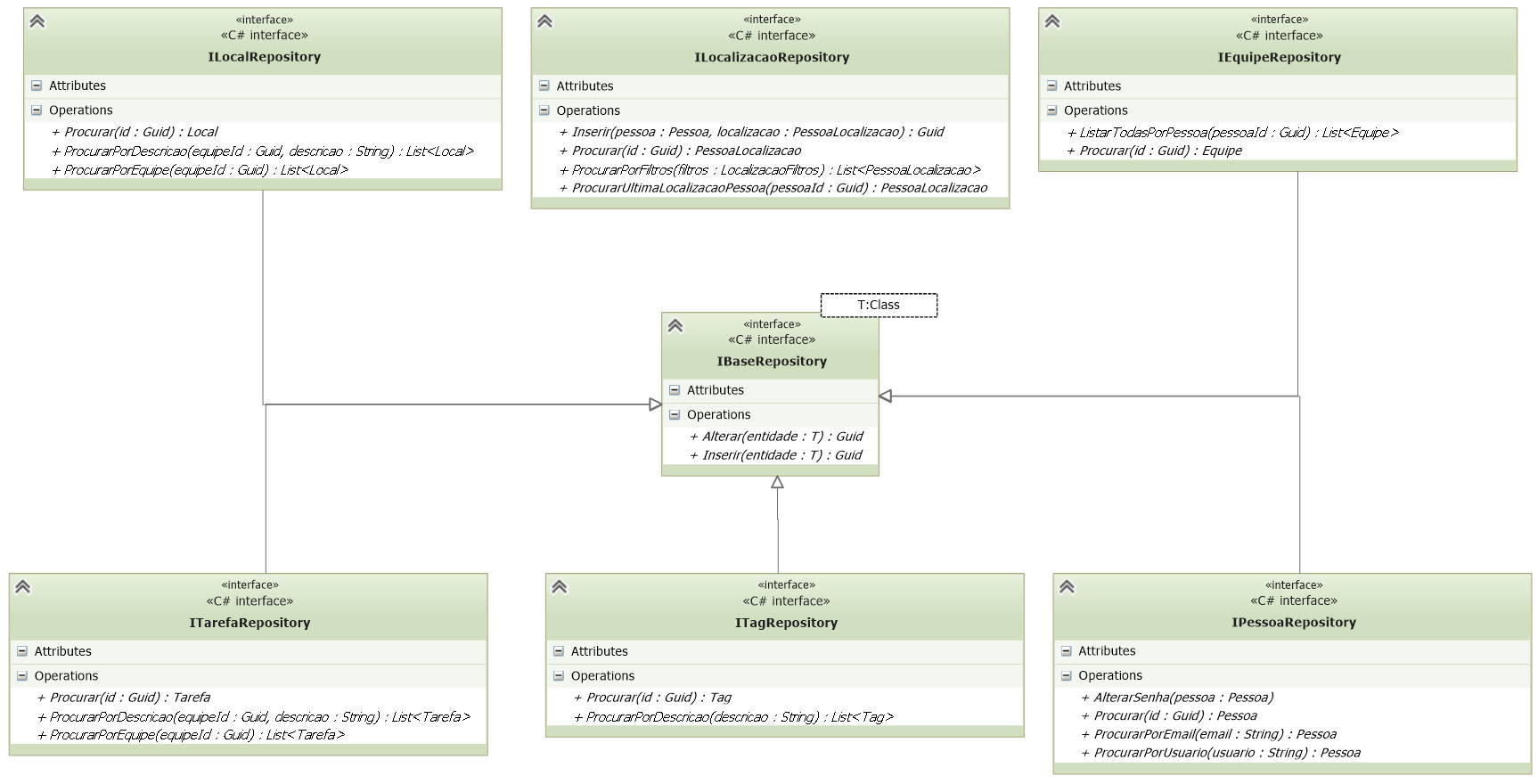


Figura 31 - Diagrama de Classe da camada de Repositório

Fonte: O autor (2017)

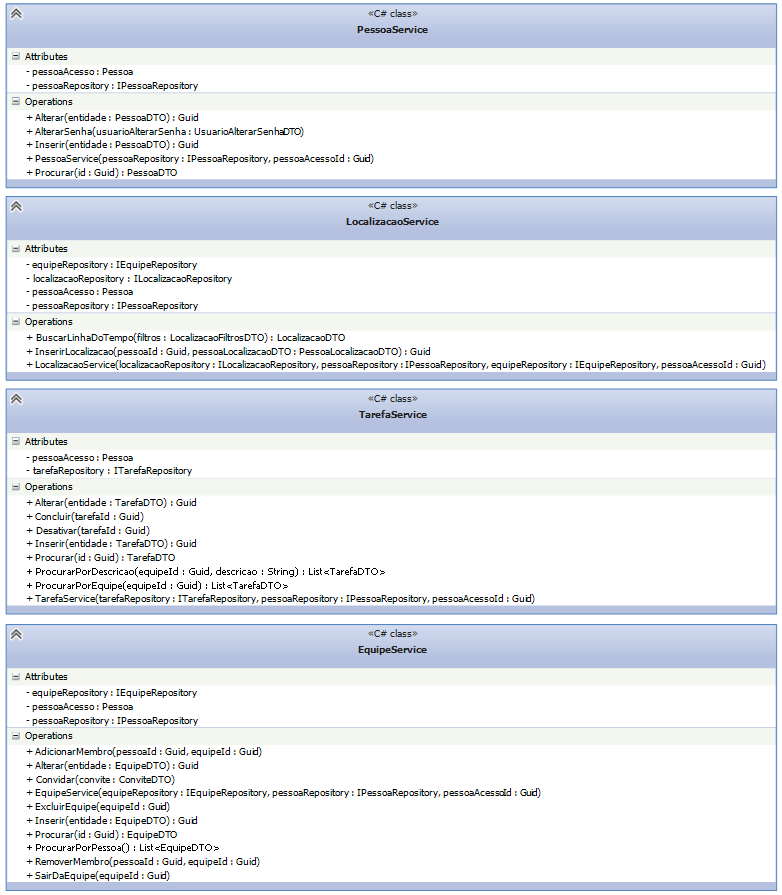


Figura 32 - Diagrama de Classe da camada de Aplicação (Parte 1)

Fonte: O autor (2017)

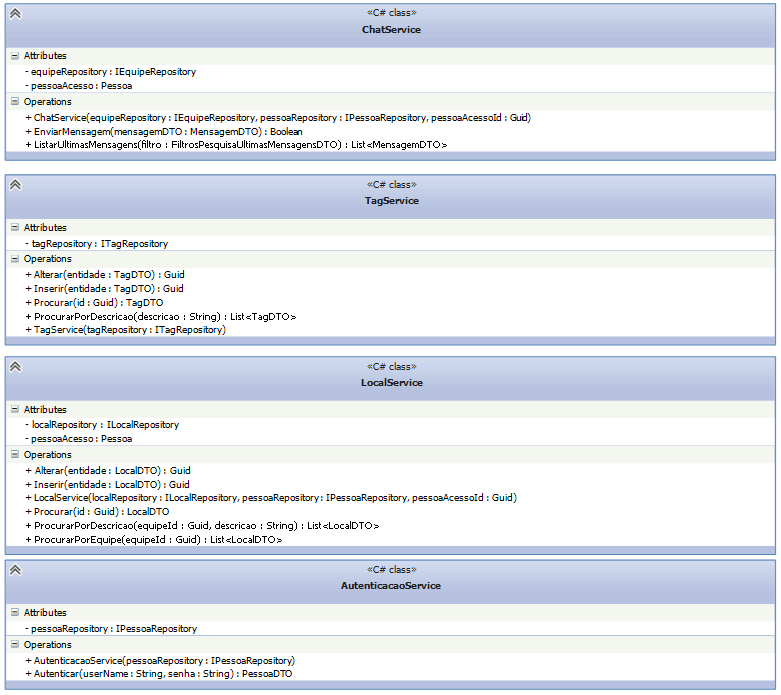


Figura 33 - Diagrama de Classe da camada de Aplicação (Parte 2)

Fonte: O autor (2017)

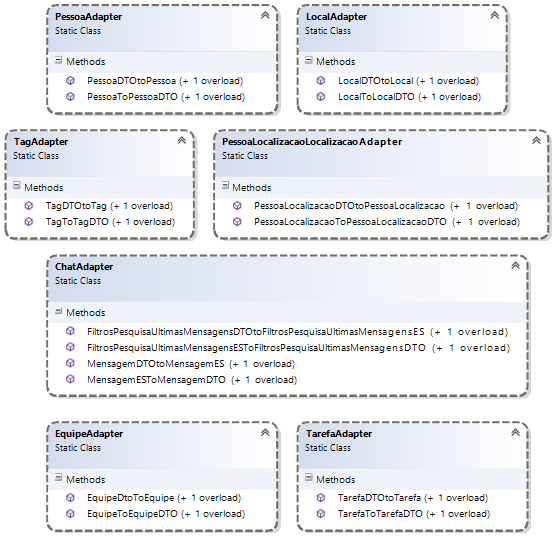


Figura 34 - Diagrama de Classe da camada de Aplicação – Adapters

Fonte: O autor (2017)

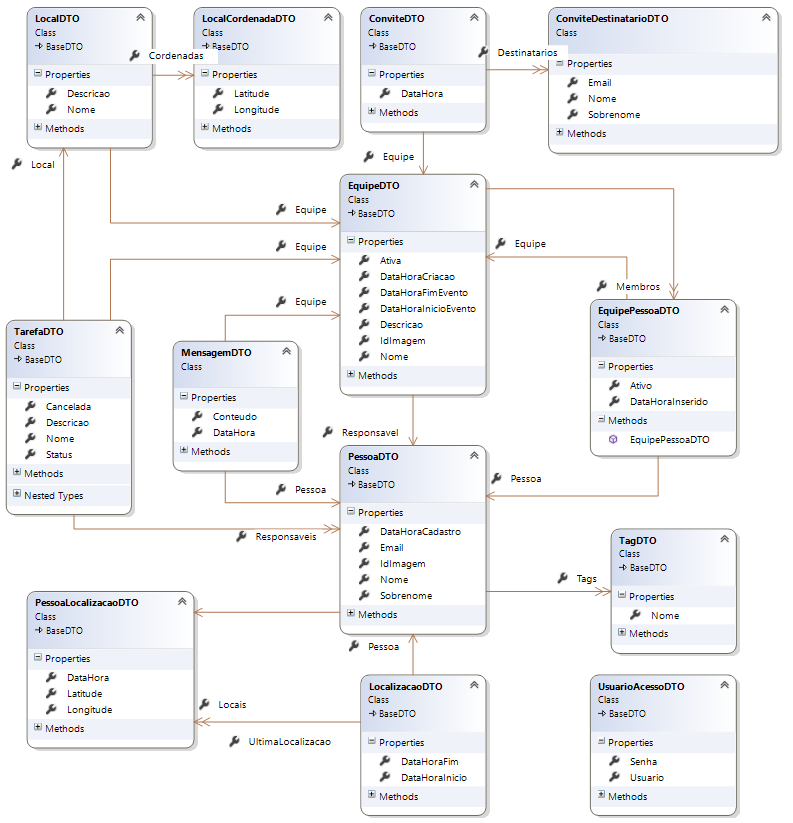


Figura 35 - Diagrama de Classe da camada de Aplicação – DTO

Fonte: O autor (2017)

# DIAGRAMAS DE SEQUÊNCIA

Os diagramas de sequência aqui apresentados são voltados ao *backend* da aplicação, especificamente a partir da camada de *application*, isto se dá pois acima da camada de aplicação existirá somente a camada de API sendo que esta é uma camada implementada unicamente para expor os métodos da camada de *application* para que então o website e o aplicativo móvel acessem. É importante notar que validações quanto a autenticação e níveis de acesso de usuário estão ocultas nestes diagramas pois esta foi uma responsabilidade atribuída a camada de API. Os diagramas são apresentados da Figura 36 até a Figura 51.

A Figura 36 apresenta o diagrama de sequência referente a autenticação do usuário no sistema, realizado a partir da tela ‘Login’.

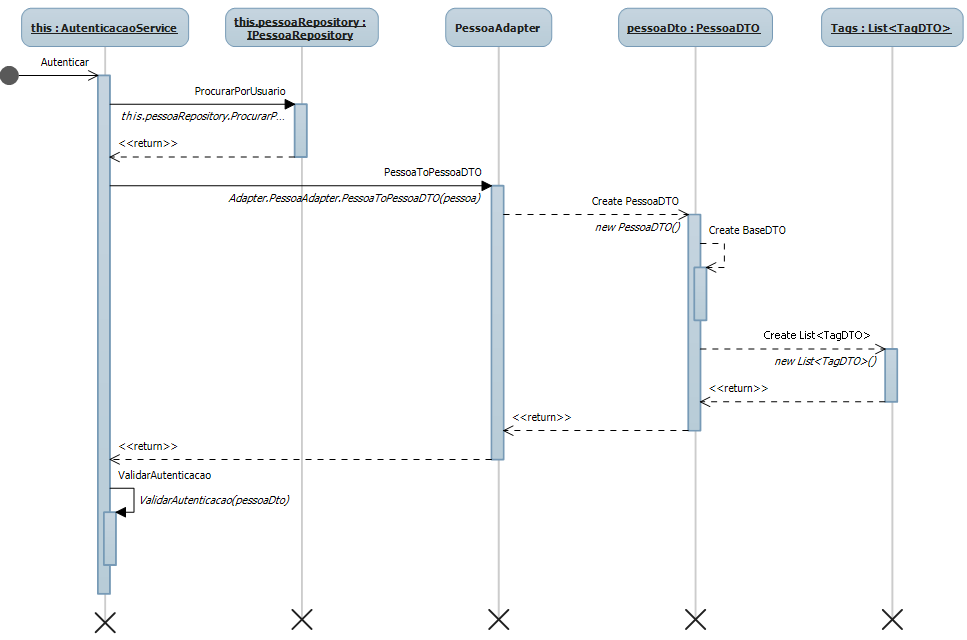


Figura 36 - Diagrama de Sequência ‘Autenticar’

Fonte: O autor (2017)

A Figura 37 apresenta o diagrama de sequência referente a tela ‘Página Inicial’, onde é apresentada uma listagem das equipes a quais o usuário está inserido.

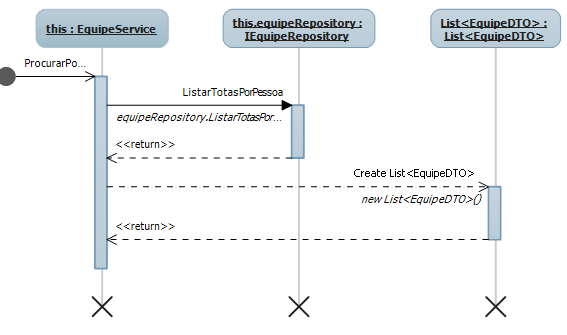


Figura 37 - Diagrama de Sequência ‘Listar Equipes’

Fonte: O autor (2017)

A Figura 38 apresenta o diagrama de sequência referente a tela ‘Página Inicial’. Na listagem de equipes fica disponível um botão permitindo que seja possível ao usuário sair de uma equipe onde ele não é administrador.

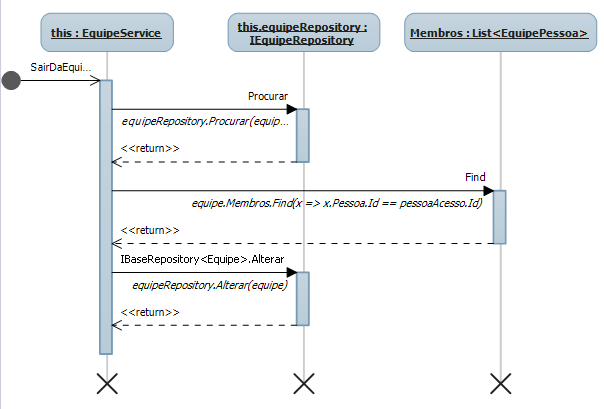


Figura 38 - Diagrama de Sequência ‘Sair da Equipe’

Fonte: O autor (2017)

A Figura 39 apresenta o diagrama de sequência referente a criação de uma equipe.

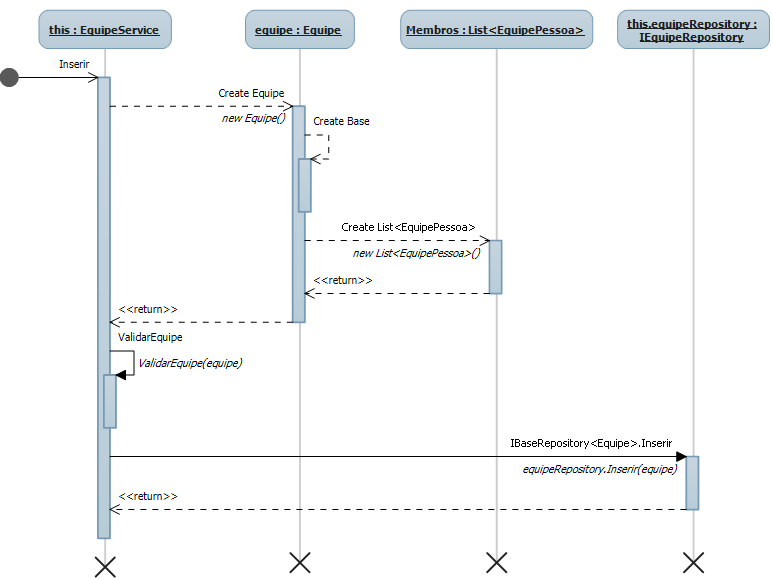


Figura 39 - Diagrama de Sequência ‘Criar Equipe’

Fonte: O autor (2017)

A Figura 40 apresenta o diagrama de sequência referente a alteração de uma equipe.

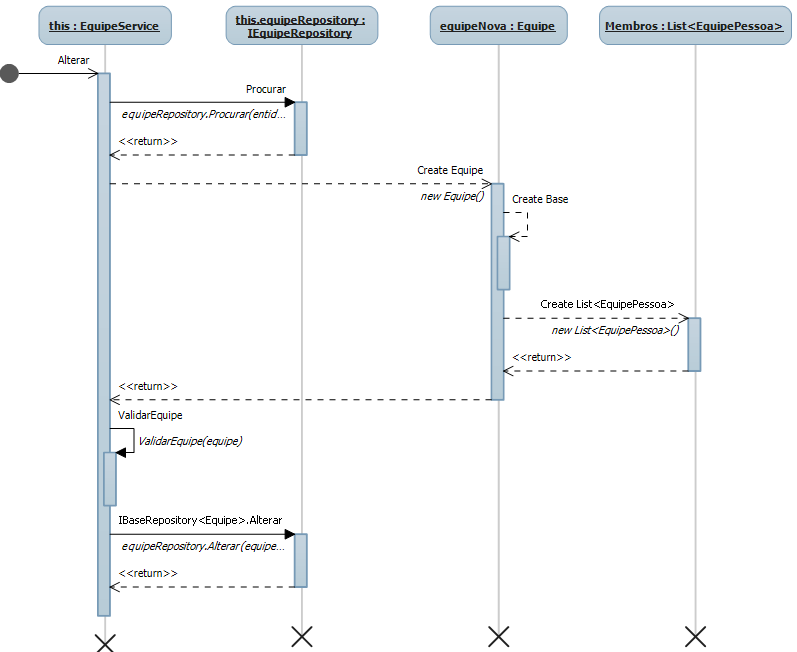


Figura 40 - Diagrama de Sequência ‘Alterar Equipe’

Fonte: O autor (2017)

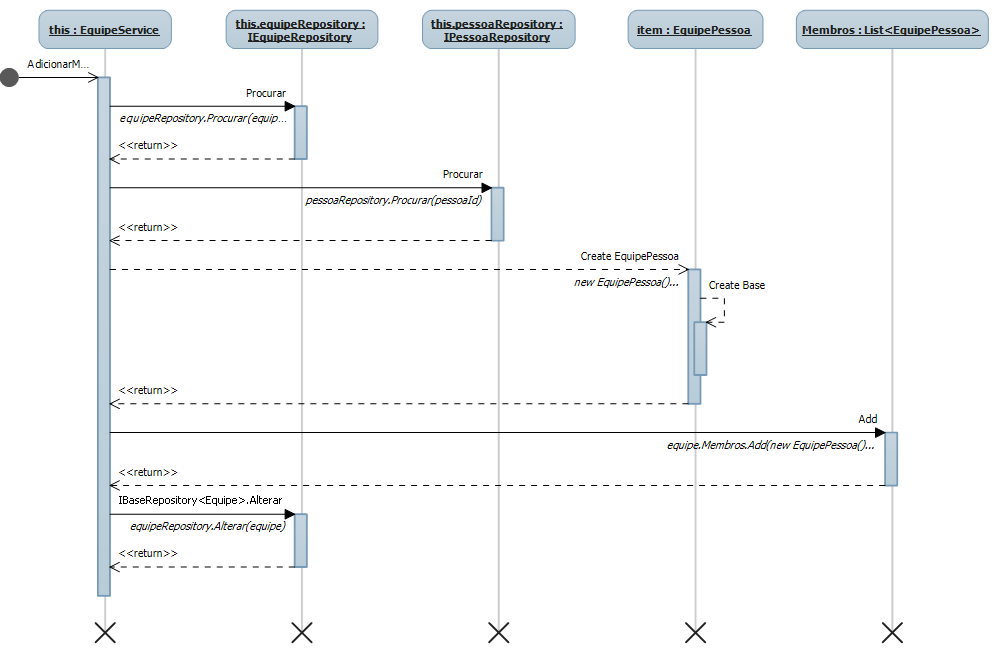


Figura 41 - Diagrama de Sequência ‘Adicionar Membro na Equipe’

Fonte: O autor (2017)

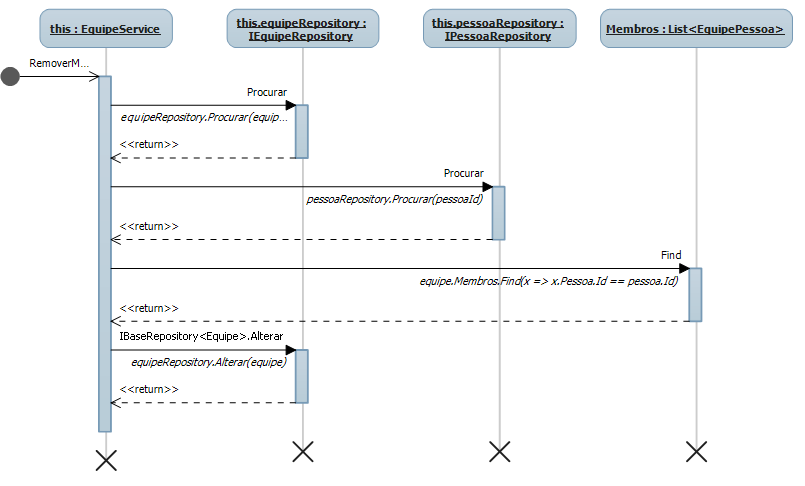


Figura 42 - Diagrama de Sequência ‘Remover Membro de Equipe’

Fonte: O autor (2017)

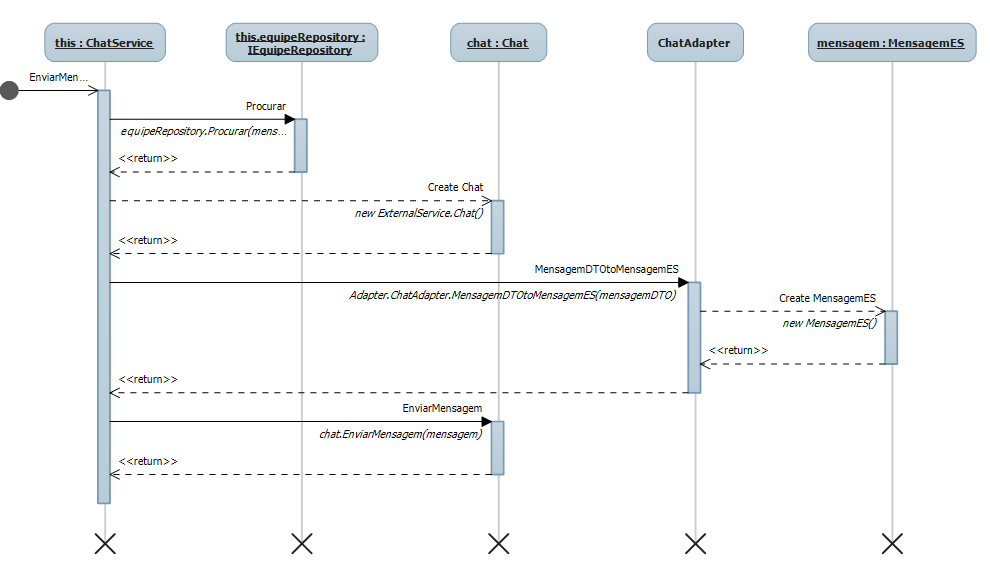


Figura 43 - Diagrama de Sequência ‘Enviar Mensagem no Chat’

Fonte: O autor (2017)

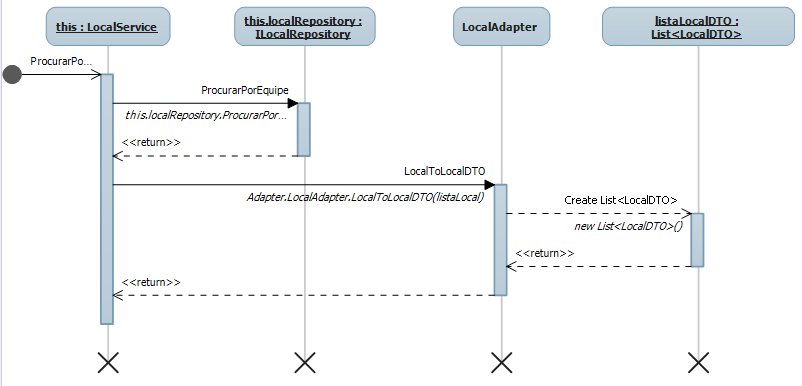


Figura 44 - Diagrama de Sequência ‘Locais’

Fonte: O autor (2017)

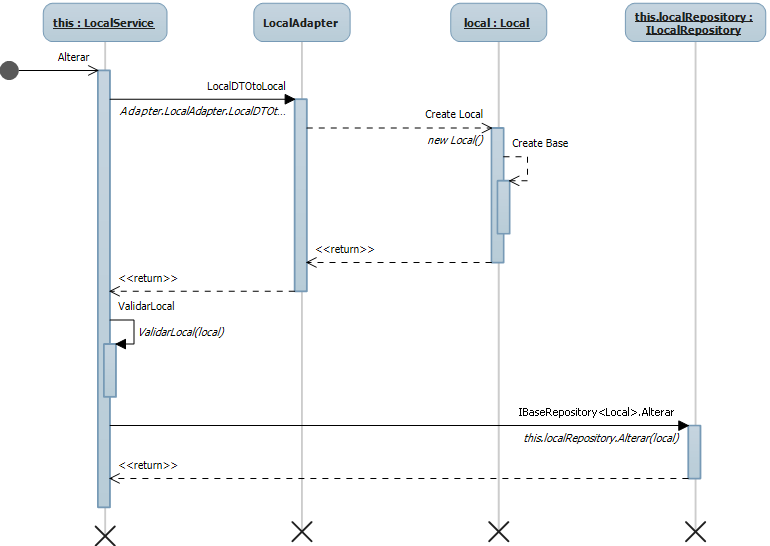


Figura 45 - Diagrama de Sequência ‘Alterar Local’

Fonte: O autor (2017)

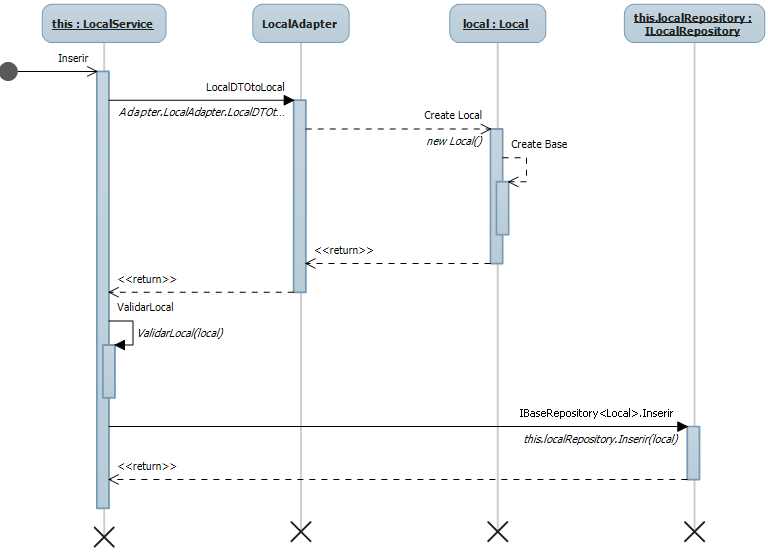


Figura 46 - Diagrama de Sequência ‘Inserir Local’

Fonte: O autor (2017)

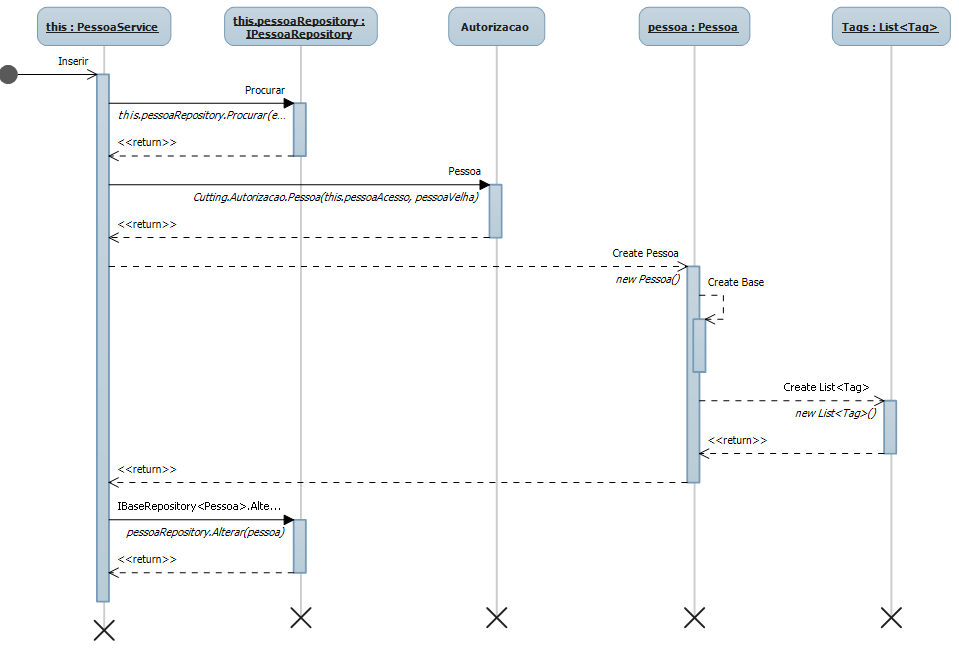


Figura 47 - Diagrama de Sequência ‘Cadastre-se’

Fonte: O autor (2017)

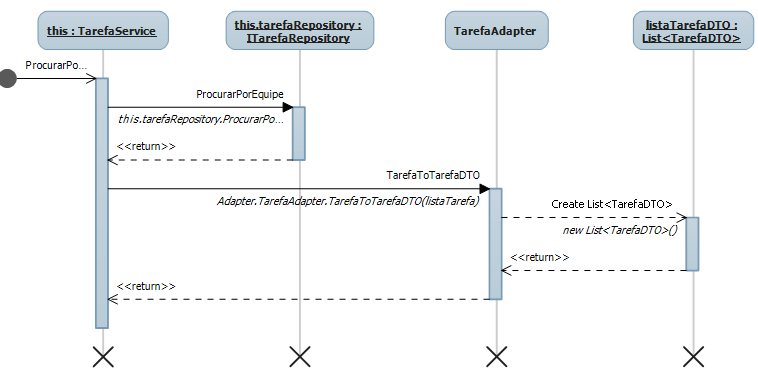


Figura 48 - Diagrama de Sequência ‘Tarefas’

Fonte: O autor (2017)

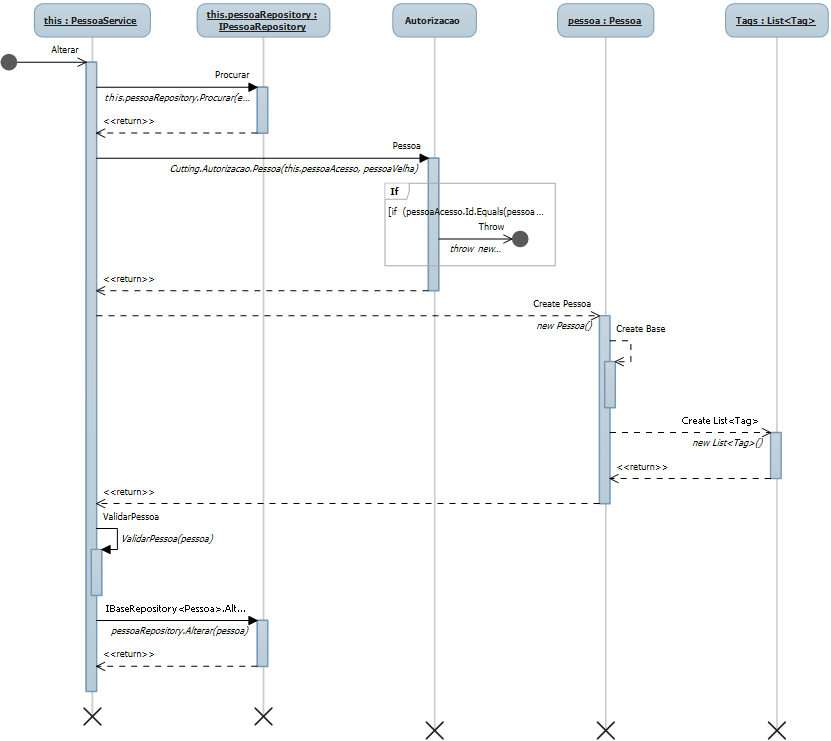


Figura 49 - Diagrama de Sequência ‘Alterar Usuário’

Fonte: O autor (2017)

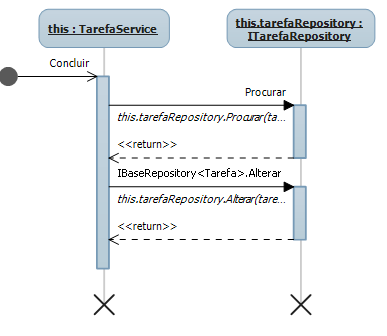


Figura 50 - Diagrama de Sequência ‘Concluir Tarefa’

Fonte: O autor (2017)

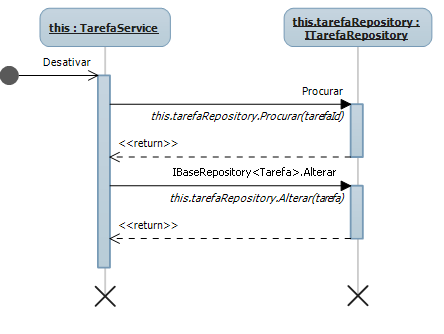


Figura 51 - Diagrama de Sequência ‘Desativar Tarefa’

Fonte: O autor (2017)

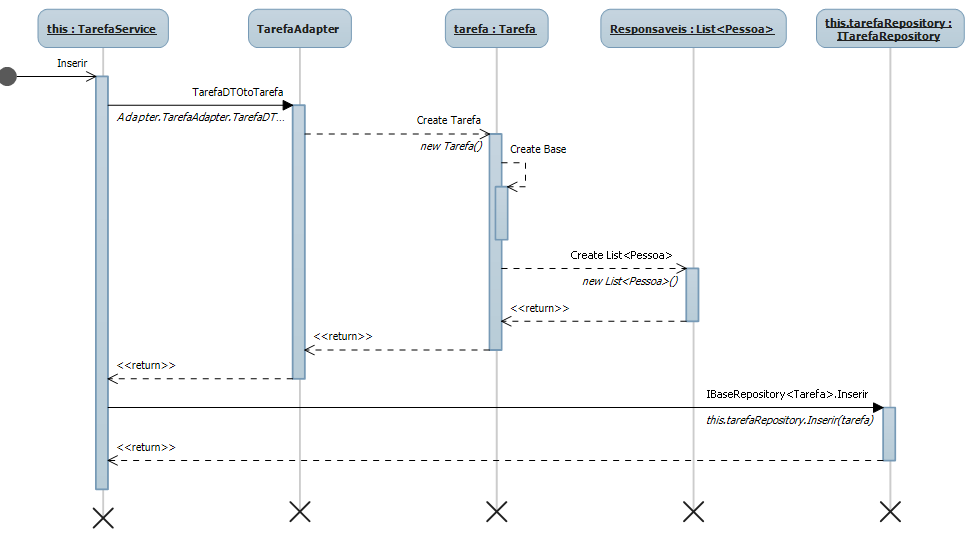


Figura 52 - Diagrama de Sequência ‘Inserir Tarefa’

Fonte: O autor (2017)

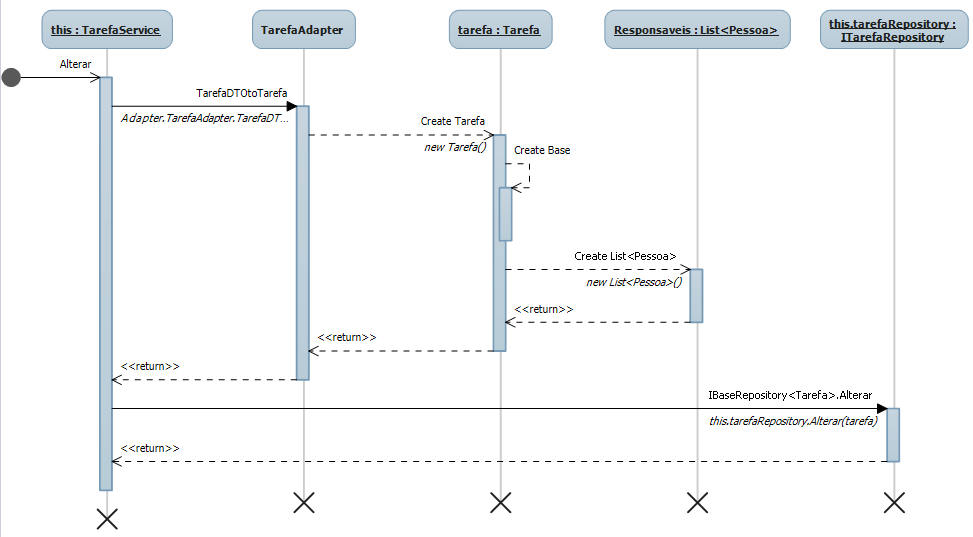


Figura 53 - Diagrama de Sequência ‘Alterar Tarefa’

Fonte: O autor (2017)

# DIAGRAMAS DE ENTIDADE-RELACIONAMENTO

No diagrama ER do sistema, será apresentado a estrutura de banco de dados da aplicação. Junto, será demonstrado o dicionário de dados, da Tabela 41 até a Tabela 46

Tabela 41 - Dicionário de dados do diretório ‘Equipe’

|  |  |  |  |
| --- | --- | --- | --- |
| **Equipe** | | | |
| **#** | **Campo** | **Tipo** | **Descrição** |
| 01 | dataFim | any | Data e hora de Fim do evento da Equipe |
| 02 | dataInicio | any | Data e hora de Início do evento da Equipe |
| 03 | fotoUrl | string | Url da imagem da equipe |
| 04 | keyResponsavel | string | Código do Usuário administrador da Equipe |
| 05 | nome | string | Nome |
| 06 | timestamp | any | Data e hora de criação da Equipe |
| 07 | keyMembros | string[] | Membros da equipe |

Fonte: O autor (2017)

Tabela 42 - Dicionário de dados do diretório ‘Convite’

|  |  |  |  |
| --- | --- | --- | --- |
| **Convite** | | | |
| **#** | **Campo** | **Tipo** | **Descrição** |
| 01 | email | string | Email do usuário convidado |
| 02 | keyEquipe | string | Chave da equipe que fez o convite |
| 03 | timestamp | any | Data e hora de envio do convite |
| 04 | keyUsuario | string | Chave do usuário convidado se existente na aplicação |

Fonte: O autor (2017)

Tabela 43 - Dicionário de dados do diretório ‘Chat’

|  |  |  |  |
| --- | --- | --- | --- |
| **Chat** | | | |
| **#** | **Campo** | **Tipo** | **Descrição** |
| 01 | keyUsuario | string | Código do usuárop que envio a mensagem |
| 02 | conteudo | string | Conteudo da mensagem |
| 03 | timestamp | any | Data e hora de envio da mensagem |

Fonte: O autor (2017)

Tabela 44 - Dicionário de dados do diretório ‘Local’

|  |  |  |  |
| --- | --- | --- | --- |
| **Local** | | | |
| **#** | **Campo** | **Tipo** | **Descrição** |
| 01 | nome | string | Descrição |
| 02 | descricao | string | Nome |
| 03 | cordenadas/lat | number | Latitude |
| 04 | cordenadas/lng | number | Longitude |

Fonte: O autor (2017)

Tabela 45 - Dicionário de dados do diretório ‘Tarefa’

|  |  |  |  |
| --- | --- | --- | --- |
| **Tarefa** | | | |
|  | **Campo** | **Tipo** | **Descrição** |
| 01 | keyEquipe | string | Código de Equipe |
| 02 | keyLocal | string | Código de Local |
| 03 | nome | string | Nome |
| 04 | descricao | string | Descrição |
| 05 | situacao | number | Situacao da tarefa |
| 06 | keyResponsaveis | string[] | Chave dos usuários responsáveis pela tarefa |

Fonte: O autor (2017)

Tabela 46 - Dicionário de dados do diretório ‘Usuario’

|  |  |  |  |
| --- | --- | --- | --- |
| **Usuario** | | | |
| **#** | **Campo** | **Tipo** | **Descrição** |
| 01 | nome | string | Nome |
| 02 | email | string | E-mail |
| 03 | tags | string[] | Tags vinculadas ao usuario |
| 04 | equipes | string[] | Equipes que o usuário é membro |
| 05 | fotoUrl | string | Url da imagem de perfil |

Fonte: O autor (2017)

# ESTRATÉGIA DE TESTES

Teste de *software* é uma das atividades do processo de desenvolvimento de sistema de *software* que visa executar um programa de modo sistemático com o objetivo de encontrar falhas afim de verificar e validar tanto as regras de negócio quanto os quesitos técnicos.,

Antes do sistema entrar em ambiente de produção, serão realizados testes em ambientes controlados para verificar o comportamento do sistema e avaliar seu desempenho e confiabilidade. Os testes seguiram o desenvolvimento do sistema, logo assim que uma etapa de desenvolvimento for concluída, a mesma será testada.

O método de testes será por meio de *Unit Testing* (Teste Unitário) onde se têm o objetivo de explorar as menores unidades do projeto, com o intuito de provocar falhas por problemas de lógica em cada módulo separado do sistema.

Na Tabela 47 são apresentados os principais itens a serem validados, ainda assim, os itens especificados nos requisitos devem ser consistidos.

Tabela 47 - Itens de validação

|  |  |
| --- | --- |
| **#** | **Descrição** |
| 01 | Criação de usuário a partir da tela de primeiro acesso. |
| 02 | Confirmação de usuário por e-mail. (Verificar validade da confirmação.) |
| 03 | Alteração de usuário. |
| 04 | Saída da aplicação. |
| 05 | Recuperação de senha. |
| 06 | Criação de equipe. |
| 07 | Envio de convites para ser inserido em uma equipe. |
| 08 | Aceitação de convite para ser inserido em uma equipe. (Verificar validade do convite.) |
| 09 | Remoção de usuário da equipe. |
| 10 | Linha do tempo e navegação correspondente as datas determinadas na equipe. |
| 11 | Locais (criação, alteração.) |
| 12 | Tarefas (criação, alteração, exclusão e conclusão.) |
| 13 | Verificar se a exclusão de um local interfere nas tarefas ao existir relacionamento. |
| 14 | Envio de mensagens no Chat |
| 15 | Notificações de ações realizadas pela equipe no chat. |
| 16 | Captura da localização atual do usuário. |
| 17 | Sair de uma equipe. |
| 18 | Consistir que as informações pertinentes a uma equipe não estão sendo mostradas em outros contextos. |

Fonte: O autor (2017)

# CONSIDERAÇÕES SOBRE A IMPLEMENTAÇÃO (REMOVER)

A aplicação será desenvolvida utilizando o Ionic Framework juntamente com Google Firebase. Foi optado por esse modelo pois o desenvolvimento de aplicativos híbridos permite a reutilização de código, ainda, a tecnologia Typescript consequentemente menos tempo é necessário ao é a linguagem na qual o desenvolvedor tem maior facilidade, logo não deverá passar por estágios de aprendizagem. Ainda, se trata de uma linguagem robusta que possui uma grande empresa responsável, tornando difícil sua obsolescência, garantindo sempre os recursos necessários para o desenvolvimento do projeto.

A estrutura do projeto se destaca em três partes:

* Backend: local de processamento das informações utilizando banco de dados SQL Server. Este serviço irá disponibilizar uma API a qual o website e o aplicativo *mobile* irão fazer consumo dos dados.
* Website: será desenvolvido utilizando HTML5, CSS3 e Javascript.
* Aplicativo Móvel: será desenvolvido utilizando Xamarin, uma ferramenta de desenvolvimento onde aplicativos para Android ou iOS são programados em C#. O aplicativo será desenvolvido para Android 4 ou superior.

# CONSIDERAÇÕES PARA TRABALHOS FUTUROS

Neste trabalho foi apresentada a aplicação Minha Equipe, desenvolvida com o objetivo de auxiliar a gestão de equipes em ambientes externos. Abaixo são listados pontos levantados para melhoria e/ou implantação que agregariam valor ao produto final:

* Opção para adicionar mais usuários administradores a uma equipe;
* Lista de amigos
  + Permitir que estes amigos sejam inseridos na equipe;
* Vinculação com Facebook a fim de criar uma lista de amigos;
* No requisito de convite, informar ao usuário por e-mail que este foi convidado a participar de uma equipe;
* Na exclusão de um local, caso este esteja vinculado a uma tarefa, deve-se perguntar se deseja excluir mesmo assim ou não.

# CONCLUSÃO

A utilização da modelagem UML foi importante pois ajudou a definir e modelar o sistema. Assim toda a documentação das aplicações, como requisitos, diagrama de classes e diagramas ER, que auxiliam no desenvolvimento e manutenção do sistema, já estão disponibilizadas para que todo o sistema possa evoluir e crescer de maneira coerente.

Mediante o que foi proposto no trabalho, foi alcançado o objetivo de demonstrar e detalhar na engenharia a proposta para o desenvolvimento do *software*. Foram muitas as dificuldades encontradas durante o desenvolvimento da engenharia, mas com persistência foi possível dar continuidade ao trabalho até o fim. Sem dúvida foi o maior e mais complexo trabalho já realizado.

Com o desenvolvimento da engenharia de *software*, foi possível visualizar de forma clara a importância da engenharia de *software* nas etapas de construção de um *software*. Com a correta utilização das metodologias, pode-se obter um ganho considerável na qualidade do *software* além de facilitar a comunicação entre a equipe de engenharia e desenvolvimento. Portanto, para que seja possível desenvolver um *software* de qualidade, competitivo no mercado, cabe a todo profissional sempre realizar cada etapa a fim de garantir a qualidade e satisfação dos clientes.

# BIBLIOGRAFIA

SOMMERVILLE, Ian. **Engenharia de Software**. Tradução de Maurício de Andrade. 6ª. ed. São Paulo: Addison Wesley, 2003.

NUNES, O'NEILL. **Fundamental de UML**. Lisboa: FCA - Editora de Informática, 2004

GUEDES, Gilleanes T. A. **UML2: uma abordagem prática**. 2. ed. São Paulo: Novatec, 2011

LEITE, Jair Cavalcanti. **Requisitos de Software**. Disponível em: <http://engenhariadesoftware.blogspot.com.br/2007/05/requisitos-de-software.html> Acesso em: 24 de jun 2017.

**O Movimento Escoteiro**. Disponível em: <http://www.escoteiros.org.br/o-movimento-escoteiro/>. Acesso em: 26 de jun. 2017.

**Relatório Anual 2016**. Disponível em <http://www.escoteirosrs.org.br/static/media/uploads/relatorio\_anual\_2016\_-\_escoteirosrs.pdf>. Acesso em: 04 de jul. 2017.

MEDEIROS, Ernani Sales de. **Desenvolvendo software com UML 2.0**. São Paulo: Pearson Makron Books, 2004.

SILVA, Ricardo Pereira e. **UML 2 em Modelagem Orientada a Objetos**. Florianópolis: Visual Books, 2007.

PRESSMAN, Roger S**. Engenharia de Software - Uma Abordagem Profissional**. Tradução de Ariovaldo Griesi e Mario Moro Fechio. 7ª. ed. Porto Alegre: AMGH Editora Ltda, 2011.