**Sistema de Gerenciamento de Requisitos**

**1 - O que é?**

Se trata de um sistema web onde uma pessoa pode criar um projeto e rastrear os seus requisitos. O sistema apresentaria de forma visual os requisitos em cartões e mostra a dependência entre eles e como algumas informações como prioridade, complexidade e estado do requisito. Várias pessoas podem contribuir em um projeto e um requisito pode conter um ou mais desenvolvedores. A ferramenta tem como objetivo apenas auxiliar na tomada de decisões e poder visualizar com maior precisão os requisitos do sistema.

**2 - Sobre requisito:**

Requisito é o principal modelo do sistema. Ele contêm as seguintes informações:

***Id***: Identificação pessoal do requisito. O primeiro requisito sempre tem o Id 0 e os próximos são de acordo com a quantidade. Esse valor não pode ser alterado depois de criado.

***Nome***: É obrigatório e não podem existir requisitos com nomes repetidos.

***Especificação***: Descrição do requisito e sobre o que deve ser feito. Não é obrigatório pois muitas vezes o próprio nome já é a descrição.

***Estado***: O sistema conta com 4 estados de requisito.

Espera: O requisto apenas foi criado mas ainda não foi validade.

Aberto: O requisito está pronto para ser implementado.

Fechado: O requisito já foi implementado.

Cancelado: O requisito foi cancelado por algum motivo.

***Tipo***: Podem ser criados vários tipos para os requisitos como feature, bug, artefato, documentação ou quaisquer que o usuário desejar. O sistema já entrega pronto os seguintes tipos: Funcionalidade, Bug, Refatoração, Documentação.

***Lista de Dependências***: Cada requisito tem a sua lista de dependências. Um requisito só pode ser Fechado caso todas as sua dependências também estejam fechadas. Se um requisito não tiver nenhuma dependência, isso significa que ele é um requisito inicial.

***Prioridade***: A prioridade é um nome, seguido de um valor. Por exemplo: Alta(5) ou Baixa(1). Todos os requisitos possuem pelo menos a prioridade 0. Essa prioridade não pode ser negativa, mas pode ser alterada. Caso o desenvolvedor não especifique uma prioridade, uma outra genérica chamada *Sem Prioridade* vai ser adicionada.

***Complexidade***: Este atributo também é um nome seguido de um número. Por exemplo: Fácil(1), Médio(2), Difícil(3), Muito difícil(6). Todo requisito tem complexidade de pelo menos 1 e não pode ser 0 ou negativo.

***Desenvolvedor***: Nome usuário que vai desenvolver o requisito. Pode ser vazio.

***Projeto***: Toda vez que um requisito é criado, ele é alocado ao projeto no qual o desenvolvedor está trabalhando. Um requisito possui apenas um projeto.

**3 – Sobre o Projeto:**

A primeira coisa que um usuário deve fazer é criar um projeto. O projeto possui um nome e uma descrição obrigatoriamente. Todo projeto possui um conjunto ilimitado de desenvolvedores e pode possuir um conjunto ilimitado de requisitos. O criador do projeto é o administrador e por sua vez ele tem o poder de criar e apagar todo tipo de coisa dentro do projeto. Ele pode adicionar novos desenvolvedores que possuam conta no sistema e delegar a esses desenvolvedores o papel de administrador. Dentro de um projeto, existem três tipos de Papel:

***Administrador:*** Tem o poder de criar e apagar todo tipo de informação, bem como apagar o próprio projeto. Um administrador só não pode apagar a conta de outros usuários. Administradores podem adicionar mais desenvolvedores ao projeto. Todo projeto tem que ter pelo menos um administrador, mas este pode delegar sua função para outro administrador. Esse papel tem todas as permissões dos desenvolvedores.

***Desenvolvedor:*** Esse papel é responsável pela criação de requisitos e administração dos mesmos. Podem criar tipos para os requisitos e definir suas prioridades. Um desenvolvedor não pode adicionar ou remover membros das equipe, nem alterar o próprio papel.

**4 – Criação de Papéis Customizados:**

Os Administradores podem criar Papéis Customizados para seus cargos como: *Programador, Design, Editor, Escritor, Engenheiro* entre outros. No início, não há qualquer papel criado para o projeto e esses papeis são vinculados à um projeto unicamente. Um desenvolvedor pode possuir quantos papeis ele quiser, e pode se auto atribuir ou remover.

**5 – Classificação dos Requisitos:**

Os sistema infere algumas informações a partir dos requisitos. Essas informações são utilizadas para analisar o impacto e o custo de se adicionar ou se remover um requisito de sistema.

***Classe:*** Devido as dependências, os requisitos se assemelham a uma estrutura de árvore. A classe é o valor da maior cadeia de nós que existe do requisito até a raiz da arvore.

***Custo:*** Corresponde à quantidade total de requisitos queum requisito tem de dependência.

***Impacto:*** Valor total de requisitos que dependem desse requisito.

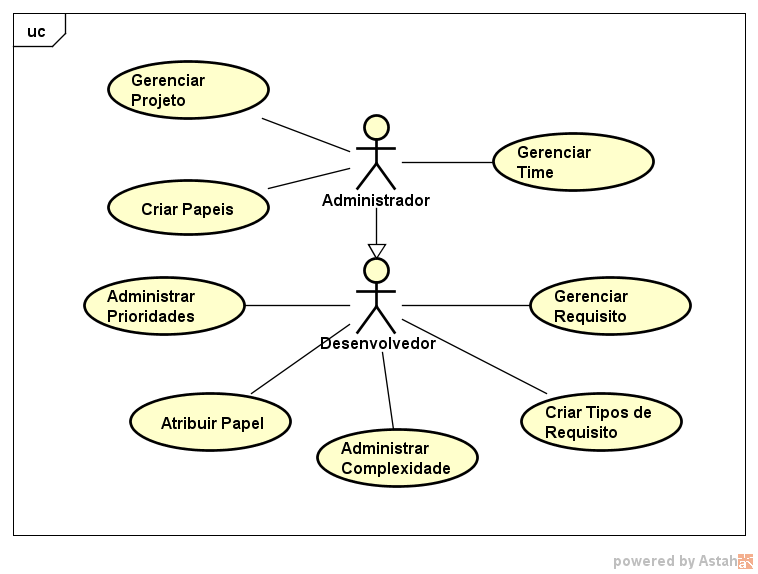
Devido a essa estrutura de árvore que esses requisitos apresentam, é necessário tomar cuidado com algumas ações:

***Apagar Requisito*:** O ato de apagar um requisito, não apagar outros requisitos que possuem dependência em cascata. Os requisitos que possuem o requisito apagado como referência, são automaticamente mudados para a classe 1, se tornado raiz da árvore.

***Alterar Dependência*:** A alteração de dependência pode mudar a classe de uma cadeia inteira de requisitos. Um requisito pode ter a sua dependência alterada para nulo o que o torna automaticamente classe 1.

Durante a Exibição, os requisitos são ordenados de acordo com a sua classe, de cima para baixo, e sua prioridade, da esquerda para a direita.

**6 – Casos de Uso**



***Gerenciar Projeto:*** Consiste em poder criar ou apagar um projeto. Também possui a permissão para alterar as características de um projeto como nome e descrição.

***Gerenciar Time:*** Pode adicionar ou remover membros do projeto, bem como adicionar o poder de administrador para outro desenvolvedor. Contudo, um administrador não pode tirar outra pessoa do papel de administrador.

***Administrar Prioridades:*** Pode criar ou apagar prioridades para requisitos, e editá-las quando necessário.

***Gerenciar Requisitos:*** Possibilidade de criar, editar, remover os requisitos.

***Criar Papeis:*** O administrador pode criar diversos tipos de papeis, editá-los e apagá-los. Ele também pode atribuir os papeis aos usuários.

***Atribuir Papeis:*** Os desenvolvedores podem se auto atribuir papeis, mas não podem atribuí-los à outros desenvolvedores. Caso um desenvolvedor não possua um papel, deverá ser adicionado um papel chamado *Indefinido*.

***Administrar Complexidade:*** Permite criar, editar, apagar e atribuir complexidade aos requisitos.

***Criar Tipos de requisito:*** Permite criar, editar, apagar e atribuir tipos ou categorias aos requisitos.