

### **Requisitos Funcionais**

O sistema consiste em uma rede social em que usuários podem postar problemas de TI e outros usuários podem postar as respectivas soluções. Caso o usuário que descreveu o problema deseje, uma solução pode ser escolhida e se transformar em uma solicitação de serviço, que ficará disponível para busca e visualização. Ou, caso já conheça o problema e as soluções, pode criar a solicitação de serviço sem necessidade de passar pela descrição de problema e solução dos outros usuários. Em todo caso, o usuário pode descrever requisitos de conhecimentos técnicos necessários para realizar tal serviço, além da quantidade de freelancers que podem atender o serviço ao mesmo tempo (formando um time) e o preço a pagar para cada freelancer que trabalhar no serviço. O usuário deve aceitar o atendimento de tais freelancers, com base em uma análise do perfil e das avaliações médias do freelancer. Ao aceitar o serviço, é criada uma espécie de contrato entre o usuário e o freelancer. Ao término do serviço, e feito o pagamento, o usuário contratante avalia o serviço em termos de eficiência, rapidez, interação e outros parâmetros. Se desejar, pode também dar avaliações diferentes para cada usuário caso tenha maior conhecimento do processo de desenvolvimento e da atuação de cada freelancer.

No cadastro, o usuário deve escolher um login e uma senha para se identificar no sistema. Além disso, deve fornecer dados pessoais, como nome completo, data de nascimento, profissão e CPF. Após logar no sistema, ele pode cadastrar seus conhecimentos técnicos (por exemplo Programação Ágil, Neo4J, Hibernate, etc) atribuindo um nível de domínio e apresentando certificados de cursos relacionados a tais conhecimentos.

Um usuário deve ser capaz de buscar por outro usuário (informando dados, como nome, profissão, conhecimentos, etc) para convidar a atender um serviço criado, ou então, com a intenção de atender algum serviço do usuário buscado. Também deve ter a opção de “seguir” outro usuário, de modo a receber notificações em uma linha do tempo sobre serviços, problemas e soluções de tal usuário.

Um usuário deve ser capaz de buscar por problemas e serviços que queira atender como freelancer, seja procurando serviços que necessitem de seus conhecimentos, serviços de usuários para quem já trabalhou ou serviços atendidos por freelancers com quem já trabalhou.

### **Uso do Banco de Dados de Grafos (Neo4J)**

A opção pelo uso do banco de dados de grafo aliado ao relacional foi devido ao formato de rede de usuários, e necessidade de buscas com profundidade entre os nós (usuários, problemas, soluções, serviços e conhecimentos). Um usuário X pode por exemplo buscar usuários que tenham trabalhado com usuário Y ou que tenham determinados conhecimentos. Um freelancer pode buscar por usuários que tenham conhecimentos em comum ou até mesmo receber recomendações de conhecimentos que tenham grande afinidade com os conhecimentos que possui no momento.

A idéia é que apenas os dados necessários para identificação e relação entre nós sejam mantidos no Neo4J, e os dados adicionais, serão mantidos no PostgreSQL. Por exemplo, para um usuário que deseja buscar outro usuário, apenas dados como nome, profissão e conhecimentos são relevantes. Login, senha, CPF, etc não são tão relevantes para esse tipo de busca. Nesse caso, não há necessidade de armazenar login, senha e CPF no banco de dados de grafos, e como estarão presentes em todo tipo de usuário, funcionam bem se armazenados no banco de dados relacional e poderão ser consultados caso algum usuário deseje ver o perfil de outro usuário.