



## Escopo do Projeto

Nosso sistema tem como objetivo, criar uma rede de apoio nas horas necessárias. Ligando interessados em serviços aos prestadores desse tipo de serviço, nesse primeiro momento nosso foco é em quem necessita de profissionais para executar serviços de tecnologia.

## Partes Interessadas

- **Profissional**, perfil de usuário que deseja ofertar um serviço no sistema.
- **Usuário**, perfil do usuário que contrata um serviço oferecido por um usuário com perfil Profissional.

## Arquitetura Usada

Para atender os requisitos que nossos stakeholders necessitam optamos por uma arquitetura MVC. Olhando para os requisitos disponíveis o estilo de arquitetura MVC é simples e suporta bem os atributos de qualidade propostos já que separa bem os componentes do sistema e por ser um padrão bastante popular acreditamos que a implementação seja facilitada também.

## Análise das Decisões Arquiteturais

Análise das decisões arquiteturais para cada requisito de qualidade de alta prioridade.

- **Ponto sensível:** Decisão arquitetural que impacta um ou mais atributos de qualidade.
- **Tradeoff:** Ponto sensível que afeta mais de um atributo de qualidade de maneira oposta.
- **Riscos:** Decisões arquiteturais potencialmente problemáticas
- **Não riscos:** Boas decisões arquiteturais

Nº Requisito: R01		R01: Desejo que o sistema seja fácil de utilizar		
Atributo (s): Usabilidade				
Decisões Arquiteturais	Sensibilidade	Tradeoff	Risco	Não Risco
Aplicação Web acessível por navegador	Usabilidade	Não encontrado		X
<p><b>Ponto Sensível:</b> O sistema vai ser web para que o sistema seja possível de ser acessado por qualquer dispositivo que tenha acesso à internet e navegador.</p> <p><b>Tradeoff:</b> Essa decisão não afeta outros atributos de qualidade diretamente.</p> <p><b>Risco:</b> Não apresenta risco para o projeto, uma vez que se mostrou a melhor decisão para satisfazer o requisito de qualidade.</p>				

Nº Requisito: R02		R02: Apresentar níveis aceitáveis de risco aos usuários		
Atributo (s): Segurança				
Decisões Arquiteturais	Sensibilidade	Tradeoff	Risco	Não Risco
Autenticação via Token e validação de dados recebidos via api REST	Segurança	Não encontrado		X

**Ponto Sensível:** Em conjunto com a camada de autenticação temos outra de validação que opera quando pode checar se a requisição recebida tem os dados e os campos válidos e seguros para podermos realizarmos as ações desejadas pelos usuários.

**Tradeoff:** Não encontrado.

**Risco:** Não apresenta risco para o projeto, uma vez que se mostrou a melhor decisão para satisfazer o requisito de qualidade.

**Nº Requisito:** R03

**R03:** Suporta vários usuários que desejam ofertar serviços no sistema

**Atributo (s):** Disponibilidade, Tolerância a falhas

Decisões Arquiteturais	Sensibilidade	Tradeoff	Risco	Não Risco
Suporta vários usuários que desejam ofertar serviços no sistema O produto de software precisa manter um nível de desempenho especificado em casos de defeitos no software ou de violação de sua interface especificada.	Disponibilidade, Tolerância a falhas	Não encontrado		X

**Ponto Sensível:**

**Disponibilidade:** Para contemplarmos este requisito vamos usar containerização o que vai facilitar para que escalemos o sistema horizontalmente com bastante facilidade e talvez futuramente usando alguns orquestradores mais complexos o que possibilitaria deixar o sistema em diversas máquinas.

**Tolerância a falhas:** Como usaremos containerização junto com um esquema de réplicas restabelecer o sistema em casos de falha será simples, pois caso um falhe outro poderá assumir e em casos que será necessário o rollback é possível o fazer sem que o sistema fique offline.

**Tradeoff:** Não encontrado.

**Risco:** Não apresenta risco para o projeto, uma vez que se mostrou a melhor decisão para satisfazer o requisito de qualidade.

**Nº Requisito:** R04

**R04:** Preciso que o sistema suporte novas categorias de serviços, que possam ser implementadas futuramente.

**Atributo (s):** Manutenibilidade

Decisões Arquiteturais	Sensibilidade	Tradeoff	Risco	Não Risco
Separar o sistema em componentes.	Manutenibilidade	Desempenho		X

**Ponto Sensível:** Como nosso sistema é dividido em partes devido a usar o padrão MVC, isso facilita a manutenção pois ao alterarmos uma camada a outra não deve ser afetada a menos que tenha adição de uma nova feature.

**Tradeoff:** Pode afetar o desempenho.

**Risco:** Não apresenta risco para o projeto, uma vez que se mostrou a melhor decisão para satisfazer o requisito de qualidade.