

# FURPS

## Classificando os Atributos de Qualidade

O documento de *Especificação Suplementar de Requisitos* captura os requisitos de sistema que não são capturados imediatamente nos casos de uso do modelo de casos de uso. Entre os requisitos estão incluídos os seguintes atributos de qualidade do sistema: usabilidade, confiabilidade, desempenho e suportabilidade. Tais requisitos fazem parte da FURPS, que é um modelo de classificação de atributos de qualidade de software, desenvolvido na Hewlett-Packard (HP) e publicado pela primeira vez por Grady e Caswell.

**FURPS** é um acrônimo que representa um conjunto de características de qualidade que são frequentemente consideradas durante o desenvolvimento de software. Cada letra do acrônimo FURPS representa uma categoria diferente de requisitos de qualidade:

- F** **Functionality (Funcionalidade):** Refere-se às funcionalidades ou recursos que o software deve oferecer. Isso inclui as capacidades específicas do software, como as operações que ele deve realizar e os comportamentos esperados.
- U** **Usability (Usabilidade):** Envolve a facilidade com que os usuários podem interagir com o software. Isso abrange aspectos como a interface do usuário, a clareza das instruções, a eficiência das operações e a experiência geral do usuário.
- R** **Reliability (Confiabilidade):** Diz respeito à capacidade do software de executar suas funções de forma consistente e livre de erros. Isso inclui a estabilidade do software, a capacidade de lidar com situações inesperadas e a prevenção de falhas.
- P** **Performance (Desempenho):** Refere-se ao desempenho do software em termos de velocidade, eficiência e uso de recursos. Isso inclui a capacidade de responder rapidamente às solicitações dos usuários e de lidar com cargas de trabalho elevadas.
- S** **Supportability (Suportabilidade):** Envolve a facilidade com que o software pode ser mantido, atualizado e corrigido. Isso inclui a documentação, a capacidade de depuração e a facilidade de implementação de correções e melhorias.

Essas categorias ajudam a definir e a priorizar os requisitos de qualidade do software durante o processo de desenvolvimento, garantindo que as expectativas dos usuários sejam atendidas e que o software seja confiável, eficiente e fácil de usar. É importante considerar esses aspectos ao planejar, projetar e testar um sistema de software para garantir que ele atenda às necessidades e expectativas dos usuários.

**Considerando um sistema qualquer de sua escolha, preencha cada uma das seções abaixo do documento de Especificação Suplementar de Requisitos referente à FURPS.**

1. Qual é o nome do sistema?

O nome do sistema é : Cockpit

## 2. Descreva resumidamente o sistema e suas principais funcionalidades

É um sistema de gerenciamento para chamados de serviços TI, sendo focado na garantia do SLA contratado com o cliente, sendo possível acompanhar em tempo real o ciclo de vida do chamado.

## 1. Funcionalidade

Descreva abaixo os requisitos funcionais do seu sistema, colocando o nome da funcionalidade e a descrição dela, por exemplo:

- **E-mail:** fornecer serviços que permitam que os usuários enviem e recebam mensagens;
- **Ajuda Online:** disponibilizar ajuda online para os usuários;
- **Segurança:** Proporcionar serviços para proteção de acesso a determinados recursos ou informações.

- Gestão de SLA; Tela onde podemos visualizar a número, status, descrição, tempo até a violação e mais algumas informações.
- Gestão de backlog: Sendo possível verificar a distribuição de chamados por clientes, grupos, status e etc.
- Registro de acompanhamento: Campo de registro do acompanhamento feito no chamado, eventos importantes, dados apartados, ações realizadas pelo ticket manager.
- A. Hyperlink para o chamado: clicando no número do chamado gera um hyperlink para outra ferramenta que o mesmo foi registrado e lá pode ser visualizado e modificado.
- B. Acompanhamento de mudanças: Mesmo esquema dos chamados, sendo focada em mudança que fazem referências os chamados. Sendo possível acompanhar as tarefas de mudanças, prazo de mudança.

## 2. Usabilidade

Descreva nesta sessão todos os requisitos de qualidade relacionados a usabilidade, tais como: facilidade de uso, facilidade de aprendizado, padrões de usabilidade e localização; Por exemplo:

- Tempo de treinamento necessário para que usuários comuns ou avançados se tornem produtivos em operações específicas no sistema;
- Especifique períodos de tempo mensuráveis para tarefas típicas no sistema;
- UI/UX Design Patterns.

Os requisitos de usabilidade podem incluir as seguintes subcategorias: fatores humanos, estética, consistência na interface com o usuário, ajuda on-line sensível ao contexto, assistentes e agentes, documentação do usuário e materiais de treinamento.

R:

- C. Interface gráfica de simples entendimento de cores básicas e bem sinalizadas.
- D. O usuário levará em média 3 a 5 dias para começar a ter um bom desempenho e entendimento da ferramenta e para ter um ótimo desempenho com a ferramenta em cerca de 20 dias.
- E. O tempo médio de resposta é de 1 segundo para a execução das tarefas.

## 3. Confiabilidade

Os requisitos de confiabilidade a serem considerados são: frequência e gravidade de falha, capacidade de recuperação, possibilidade de previsão, precisão e tempo médio entre falhas (MTBF). Por exemplo:

- A. Disponibilidade:** especifique a porcentagem de tempo disponível (xx.xx%), as horas de uso, o acesso à manutenção, as operações de modo degradado, etc.
- B. Tempo Médio entre Falhas (MTBF):** normalmente especificado em horas, mas também poderá ser especificado em termos de dias, meses ou anos.
- C. Tempo Médio para Reparo (MTTR):** quanto tempo o sistema poderá ficar sem funcionar após uma falha?
- D. Exatidão:** especifique a precisão (resolução) e exatidão (através de algum padrão conhecido) necessárias na saída do sistema.
- E. Taxa máxima de erros ou defeitos:** geralmente expressa em termos de erros / KLOC (thousands of lines of code, milhares de linhas de código) ou de erros / ponto de função.

- F. Taxa de erros ou defeitos** (categorizada em termos de erros de pouca, média ou muita importância): os requisitos devem definir o que se entende por erro “crítico”(por exemplo, perda total de dados ou total incapacidade de usar determinadas partes da funcionalidade do sistema).

Descreva abaixo os requisitos de confiabilidade para seu sistema referente aos itens **A**, **B** e **C** da lista acima:

- A) Cerca de 99,95% de disponibilidade de horas trabalhadas dentro da aplicação sem nenhuma degradação. As manutenções são realizadas nas madrugadas e nos finais de semana, por conta do fluxo.
- B) Sobre o tempo médio de falhas, temos ligado a isto, uma leve lentidão quando está tendo uma massiva de acessos simultâneos e com isso o servidor que sustenta a aplicação está tendo um alto consumo de recursos. Porém ao longo do dia, ele se sustenta sem apresentar falhas.
- C) O tempo médio para reparo onde houveram indisponibilidades da aplicação foi em cerca de 1 a 2 horas.

## 4. Desempenho

Descreva as características de desempenho do seu sistema, tais como: velocidade, eficiência, disponibilidade, precisão, produtividade, tempo de resposta, tempo de recuperação e uso de recursos. Por exemplo:

- **Tempo de resposta de uma transação:** tempos médio e máximo;
- **Taxa de transferência:** quantidade de transações por segundo;
- **Capacidade:** o número de clientes ou de transações que o sistema pode acomodar;
- **Modos de degradação:** o modo aceitável de operação quando o sistema tiver sido degradado de alguma maneira;
- **Uso de recursos:** memória, disco, comunicações, etc

- O tempo de resposta médio para executar as tarefas dentro da aplicação é de cerca de 1 a 4 segundos para a execução.
- Em relação a taxa de transferência o fluxo é bem distribuído, sendo possível realizar várias tarefas simultâneas sem que a aplicação apresente gargalos.
- Cerca de 250 usuários simultâneos.
- Referente ao uso de recursos, os servidores de aplicação e banco que hospedam o sistema utilizam escaláveis níveis de processamento de dados.

## 5. Suportabilidade

Descreva todos os requisitos que aprimorarão a suportabilidade ou manutenibilidade do seu sistema, tais como: padrões de codificação, convenções de nomeação, bibliotecas de classes, acesso à manutenção e utilitários de manutenção. Ainda, os requisitos de suportabilidade podem incluir as possibilidades de teste, adaptação, manutenção, compatibilidade, configuração, serviço, instalação e localização (internacionalização)

Utilização de testes unitários para verificar o comportamento do código;

Controle de versões: utilização do git para rastrear e gerenciar as alterações no código.

Utilização de nomes descritivos para variáveis, funções, classes e métodos.

Utilização de letras maiúsculas iniciais para nomes de classes (CamelCase) e minúsculas para variáveis e funções.

Utilização de práticas de segurança, como validação de entrada, prevenção de injeção de SQL e XSS, e autenticação adequada.

Utilização de servidores em Cloud, montados em container.