Requisitos Funcionais vs. Não-Funcionais

Definição de Requisito

Condição, capacidade ou restrição que expressa as necessidades do utilizador ou do sistema. Existem dois tipos principais: Requisitos Funcionais e Requisitos Não-Funcionais. **Requisitos são dinâmicos** e podem evoluir com acordo das partes interessadas durante o projeto.

Oito Propriedades Desejáveis dos Requisitos

Quando a separação problema-solução é mantida, os requisitos resultantes devem possuir oito propriedades:

Propriedade	Definição	Exemplo
Necessário	Serve um propósito; nada supérfluo	Cada requisito contribui para valor
Claro	Sem ambiguidade; interpretação única	"O sistema deve enviar email" é claro; "O sistema deve comunicar" é vago
Correto	Reflete realmente o que cliente necessita	Validação com cliente, não presunção
Completo	Cobre todos os aspetos; sem lacunas	Para "criar encomenda" também especificar "cancelar", "modificar", "rastrear"
Viável	Possível implementar com recursos disponíveis	Não pedir "reconhecimento de voz com 99.99% acuidade" com orçamento de €1000
Rastreável	Ligação clara a origem e implementação	Cada requisito tem ID, ligado a testes, ligado a código fonte
Verificável	Testável objetivamente; critérios de aceitação definidos	"Sistema deve ser rápido" não é verificável; "responder em <500ms" é
Negociável	Sujeito a acordo; não imutáveis	Pode ser modificado após discussão com stakeholders

Requisitos Funcionais (O quê?)

Descrevem o que o sistema deve fazer, ou seja, as funcionalidades e comportamentos esperados. **Devem ser independentes de aspetos de design e implementação** (requisitos não funcionais). Mas **requisitos funcionais são dependentes uns dos outros**, uma vez que se pretende que o sistema funcione como um todo coeso.

Exemplos incluem:

- Autenticação de utilizadores.

- Processamento de transações.
- Geração de relatórios.

O conjunto de todos os requisitos funcionais de um sistema deve possuir duas qualidades essenciais:

- **Coerente:** Não deve haver contradições entre os requisitos. Um requisito não pode exigir que uma ação seja permitida e outro exigir que a mesma ação seja proibida.
- **Completo:** Deve considerar todas as necessidades que o cliente deseja ver satisfeitas.

É também importante mencionar os **requisitos implícitos**. Estes são requisitos tão óbvios para um determinado domínio que, por vezes, não são explicitamente pedidos pelo cliente, mas a equipa de desenvolvimento inclui-os com base no seu conhecimento e experiência.

implícitos => não estão documentados explícitos => estão documentados

---

Requisitos Não-Funcionais (Como?)

Definem restrições ou atributos de qualidade do sistema, como desempenho, segurança e usabilidade. Eles **não alteram a essência das funcionalidades** (requisitos funcionais), mas são absolutamente **cruciais para a arquitetura** do sistema. São requisitos que influenciam o sistema como um todo (são emergentes).

É impossível maximizar todos os requisitos não-funcionais ao mesmo tempo. **Melhorar um, muitas vezes, implica sacrificar outro**. Por exemplo, um sistema otimizado para um desempenho extremo pode tornar-se mais complexo, reduzindo a sua manutenibilidade. No entanto, as relações nem sempre são de compromisso. A adaptabilidade de um sistema, por exemplo, contribui positivamente para a sua portabilidade.

Exemplos incluem:

- O sistema deve responder em menos de 2 segundos.
- O sistema deve estar disponível 99,9% do tempo.
- O sistema deve ser compatível com dispositivos móveis.

> Se o sistema for projetado apenas com base nos requisitos funcionais, pode existir como uma **entidade monolítica.**

Categorias Comuns de Requisitos Não-Funcionais

1. **Aparência:** Aspeto visual e estética do sistema. Ex: design da interface do utilizador.
2. **Usabilidade:** facilidade de utilização e com tudo o que contribui para uma experiência amigável. Palavras-chave: intuitivo, consistente, feedback ao utilizador, personalização, facilidade de uso, facilidade de aprendizagem, acessibilidade. Exemplo: O produto deverá ser utilizável por pessoas com deficiência visual.
3. **Desempenho:** Velocidade, capacidade de resposta e eficiência do sistema. Ex: tempo de resposta, taxa de transferência.
4. **Operacional:** Descreve as características necessárias para que o sistema funcione corretamente no ambiente técnico onde será inserido.. Ex: ambiente de hardware, sistema operativo.
5. **Manutenção e Suporte:** Abrange os atributos que permitem que o sistema seja facilmente reparado ou melhorado. A manutenção divide-se em quatro tipos: **preventiva**, **corretiva**, **perfetiva** e

adaptativa. A *Modificabilidade*, ou seja, a facilidade de localizar e alterar componentes com impacto reduzido, é um aspeto central desta categoria.

6. **Segurança:** Trata de questões relacionadas com o acesso, a confidencialidade, a proteção e a integridade dos dados e do sistema.
7. **Cultural/Político:** Considera fatores relacionados com a cultura, hábitos e normas das partes interessadas (stakeholders) e dos utilizadores finais.
8. **Legal:** Refere-se a todas as leis e regulamentos que se aplicam ao sistema. É fundamental consultar juristas e especialistas para garantir a conformidade. Ex: conformidade com o RGPD.

Requisito Não-Funcional	Trade-off Resultante	Exemplo
Performance (velocidade)	Maintainability (manutenibilidade)	Otimizar para velocidade pode resultar em código menos legível
Security (segurança)	Usability (usabilidade)	Mais camadas de segurança podem complicar a experiência do utilizador
Adaptability (adaptabilidade)	Performance	Sistemas altamente adaptáveis têm overhead de abstração que reduz performance
Reliability (fiabilidade)	Cost (custo)	Aumentar fiabilidade frequentemente aumenta custos
Generality (generalidade)	Simplicity (simplicidade)	Sistemas muito gerais são mais complexos

Requisitos do Utilizador

Definição: Representam:

1. Uma funcionalidade que o sistema é esperado fornecer aos seus utilizadores
2. Uma restrição aplicável à operação desse sistema
3. definem **NECESSIDADES** do utilizador no domínio do problema

Características:

- Relacionadas ao domínio do problema
- Expressas normalmente sem grande rigor matemático
- Utilizam linguagem natural e diagramas informais
- Permitem aos stakeholders ler, analisar e discutir

Benefícios desta Abordagem:

- Acessibilidade a não-técnicos
- Facilita discussão e consenso

- Reduz barreiras de comunicação entre negócio e tecnologia

Exemplos de Requisitos de Utilizador bem-formulados:

- "Como cliente, quero poder filtrar contactos por categoria para encontrar facilmente as pessoas que procuro"
- "Como gerente de projetos, preciso de gerar relatórios de progresso semanais para comunicar ao cliente"
- "Como utilizador de baixa visão, quero que o sistema tenha modo de alto contraste para melhor legibilidade"

Requisitos do Sistema

Definição: Constituem especificação mais detalhada de um requisito.

1. Traduzem essas necessidades em **ESPECIFICAÇÕES** técnicas.

Características:

- Constituem um modelo mais formal do sistema
- Orientadas para o domínio da solução
- Auxiliam engenheiros em design e construção
- Documentadas em linguagem técnica mais precisa
- Idealmente **independentes** de pre-decisões de design/implementação
- Frequentemente acompanhados por diagramas formais (UML, etc.)

Exemplos de Requisitos de Sistema:

- "O sistema deve validar que um email satisfaz o padrão RFC 5322 antes de armazenar um contacto"
- "O sistema deve usar uma tabela 'CONTACTS' com colunas: ID (chave primária), NAME (varchar 100), EMAIL (varchar 100), PHONE (varchar 20)"
- "O sistema deve implementar autenticação baseada em tokens JWT com expiração de 24 horas"