

## COMO DEFINIR REQUISITOS?

- **aspectos gerais:** é importante considerar no momento da redação de um requisito:
  - ✓ que a declaração de um requisito normalmente começa com uma frase semelhante a “O sistema deverá...”;
  - ✓ que a qualidade importante para uma boa declaração de requisito é que possa ser verificado. “A interface com o usuário deve ser intuitiva”, não pode ser objetivamente testada e, portanto, não é verificável;
  - ✓ que uma declaração de requisitos deve concentrar-se em um ponto. A granularidade mais fina permite um rastreamento melhor ao longo do modelo.
- **requisitos funcionais:** descrevem uma ação que o sistema deve executar, ou seja, identificam os procedimentos que o sistema pode fazer, normalmente em resposta a entrada de dados externa.

De maneira bem simples e objetiva, os requisitos funcionais correspondem à listagem de todas as coisas que o sistema deve fazer.

### **EXEMPLOS:**

O **sistema deverá** produzir um relatório resumido de todas as vendas efetuadas em um determinado dia.

O **sistema deverá** permitir ao coordenador inserir, alterar e excluir as disciplinas do professor bem como, os horários das respectivas disciplinas.

O **sistema deverá** permitir que os alunos visualizem as notas obtidas por semestre letivo.

- **requisitos não-funcionais:** são restrições que se coloca sobre como o sistema deve realizar seus requisitos funcionais.

Os requisitos não-funcionais podem ser categorizados para torna-los mais fácil de compreender. Algumas categorias, dentre outras possíveis, são:

- ✓ **usabilidade:** refere-se aos aspectos gerais da interface entre o usuário e o sistema;
- ✓ **desempenho:** descreve o desempenho da execução do sistema e normalmente estão relacionados ao tempo;
- ✓ **segurança:** tende a especificar níveis de acesso ao sistema e freqüentemente mapeiam classes de usuários do sistema. Normalmente descrito no contexto de controle de acesso, autenticação e criptografia;

- ✓ **implementação**: descreve condições específicas a serem observadas durante a implementação do software, tal como arquitetura, uso de padrões e nomenclaturas;
- ✓ **hardware/software**: define hardware e software mínimos exigidos para implementar/ implantar o sistema;
- ✓ **conformidade**: declara a observância a padrões de documento, leis, formatos vigentes.

#### EXEMPLOS:

- ✓ requisito não-funcional de **usabilidade**
  - A interface do sistema não deve usar quadros HTML.
  - O sistema deve ser compatível com browser Explorer 5.0.
  - Todas as funções relacionadas a empréstimo devem ser efetuadas em uma única janela.
  - As fitas devem ser identificadas através de código de barras.
- ✓ requisito não-funcional de **desempenho**
  - As páginas web não devem levar mais de 15 segundos para serem carregadas no navegador durante o uso normal do sistema.
  - O recibo para o cliente deve ser emitido até oito segundos após a transação.
- ✓ requisito não-funcional de **segurança**
  - Os perfis de usuário para acesso ao sistema são:
    - a) administrador: pode efetuar todas as operações;
    - b) operador: pode efetuar as operações de empréstimo, devolução, pagamento e cadastramento;
    - c) visitante: pode efetuar somente consulta.
- ✓ requisito não-funcional de **implementação**
  - A camada de persistência deve ser implementada de forma que diferentes bancos de dados possam vir a ser utilizados no futuro.
- ✓ requisito não-funcional de **hardware/software**
  - O sistema deverá utilizar banco de dados Oracle.
  - Para a implantação do sistema deverá ser disponibilizado um leitor de código de barras.
- ✓ requisito não-funcional de **conformidade**
  - O processo judicial impresso deve estar em conformidade com a legislação vigente e o formato estabelecido pelo Fórum de Blumenau.