



1

Requisitos de Softwares2

# Tópicos

Requisitos de Softwares

Estudo de Caso

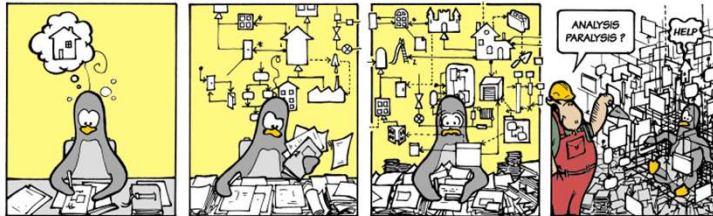
Regras de Negócio

2

1

# Requisitos de Softwares

## Introdução



*“A **parte mais árdua** na construção de um sistema de software é **decidir o que construir**. Nenhuma outra parte do trabalho compromete mais o sistema se for feito de maneira imprópria. Nenhuma outra parte é mais difícil de corrigir a posteriori.”*

F. P. Brooks Jr, “No Silver Bullet: Essence and Accidents in Software Engineering”, IEEE Computer, abril 1987.

3

# Requisitos de Softwares

## Conceitos

- Requisito:
  - “Uma **condição** ou capacidade **que deve ser satisfeita** ou possuída por um **sistema** ou **componente do sistema** para satisfazer um contrato, um padrão ou uma especificação”  
IEEE Std 610.12, 1990
- Requisitos são **propriedades desejáveis** para um sistema de software
  - Um requisito pode ser mensurável (ex., tempo médio de atendimento de requisições), ou avaliado subjetivamente (ex., qualidade da documentação)

4

## Requisitos de Softwares

### Exemplos de Requisitos

- **Ex. 1:** O sistema deve controlar os cartões de ponto dos funcionários, registrando os horários de chegada e de saída de cada funcionário;
- **Ex. 2:** O sistema deve dar baixa no estoque de cada produto vendido ao finalizar a venda;
- **Ex. 3:** O sistema deve emitir um comprovante de locação após a realização da locação de um ou mais filmes;
- **Ex. 4:** O sistema deve ser capaz de recuperar os dados de uma venda interrompida devido a uma falha de hardware ou queda de energia.

Os requisitos de software são definidos para identificar o que o software deve fazer para que as necessidades do cliente sejam satisfeitas, ou seja, para definir o enunciado do problema.



5

## Estudo de Caso

### Parte 1



Preciso de um sistema que gerencie o financeiro de minha empresa.



Depois de ter conhecido sua empresa, minha equipe compreendeu, com o auxílio da sua, que o sistema:

Deve registrar todas as operações de débito e de crédito da empresa.

Deve controlar as contas a pagar e a receber.

Deve importar extratos bancários para conciliação.

Deve gerar um gráfico comparativo com os saldos de cada conta bancária da empresa.

Deve registrar as ações dos funcionários para futuras auditorias.

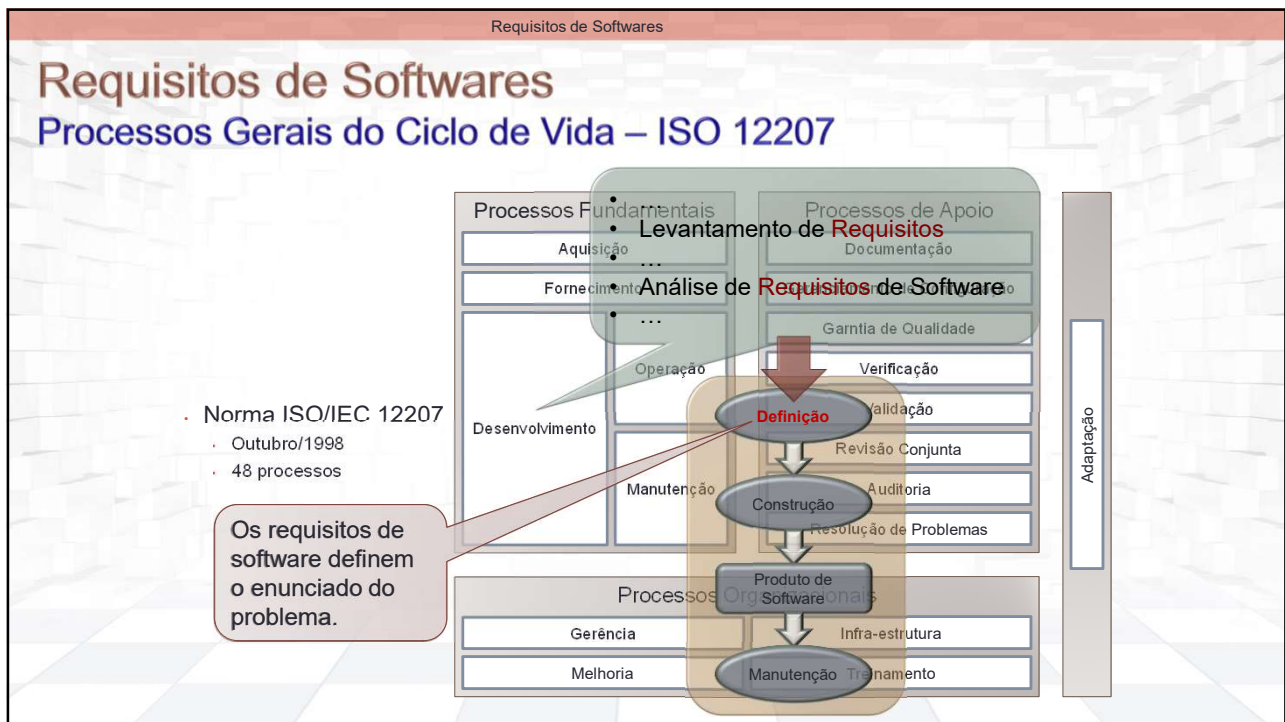
Deve possuir um visual com desenhos e animações de cédulas e de moedas.

Não deve emitir carnês de pagamento aos clientes.

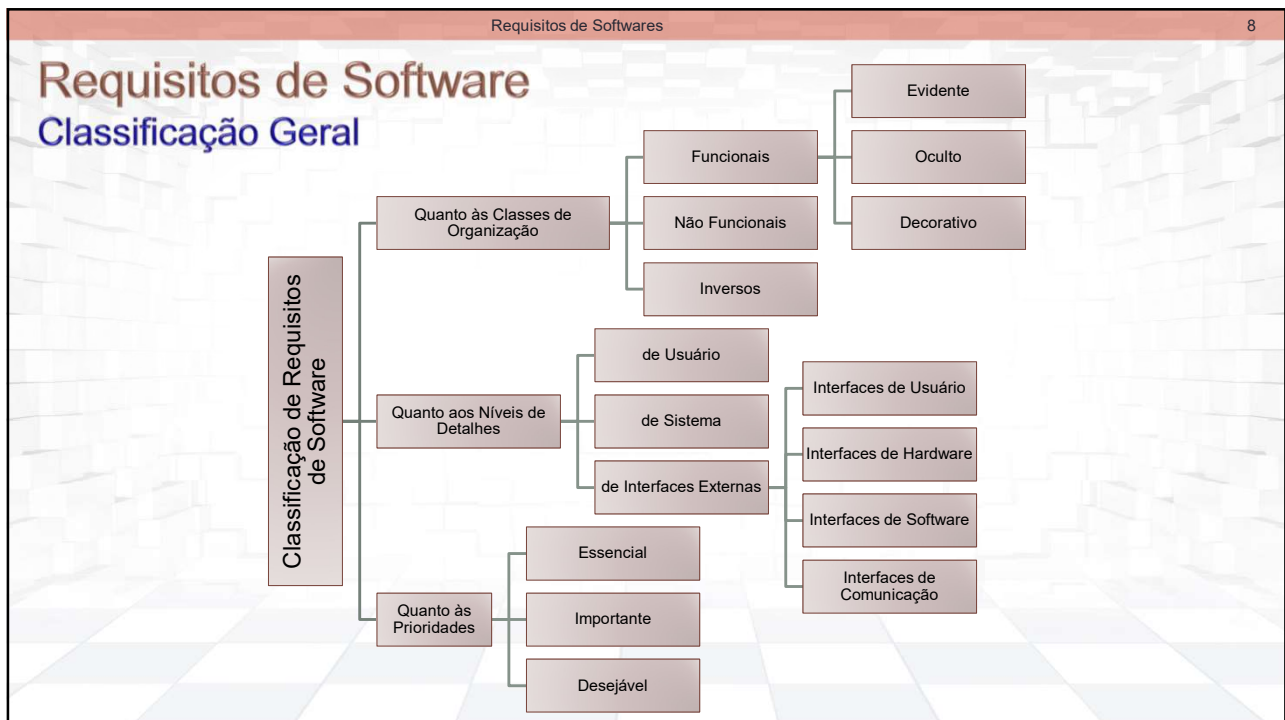
Deve funcionar em vários sistemas operacionais.

Etc.

6



7



8



## Classificações de Requisitos

### Quanto às Classes de Organização [1/9]

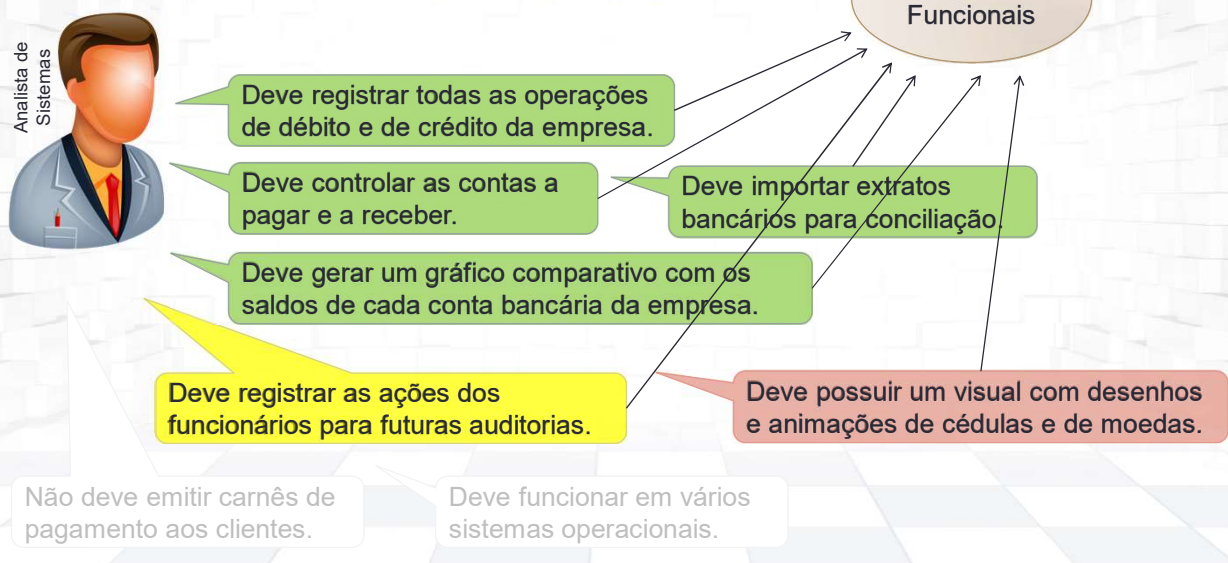
#### • Requisitos Funcionais [1/2]:

- Aqueles diretamente relacionados à **funcionalidade** do Software
- Descrevem **o que** o software deve realizar
  - serviços (funcionalidades)
- Devem ser **completos**, **precisos** e **consistentes**
- Exemplos:
  - **RF001**: O sistema deve permitir que um aluno realize sua matrícula em uma disciplina pela Internet
  - **RF002**: O sistema deve permitir que um cliente realize empréstimo de mais de um livro
  - **RF003**: A ferramenta CASE deve gerar o código fonte, em uma linguagem Orientada a Objetos, a partir de um Diagrama de Classes

9

## Classificações de Requisitos

### Quanto às Classes de Organização [2/9]



10

## Classificações de Requisitos

### Quanto às Classes de Organização [3/9]

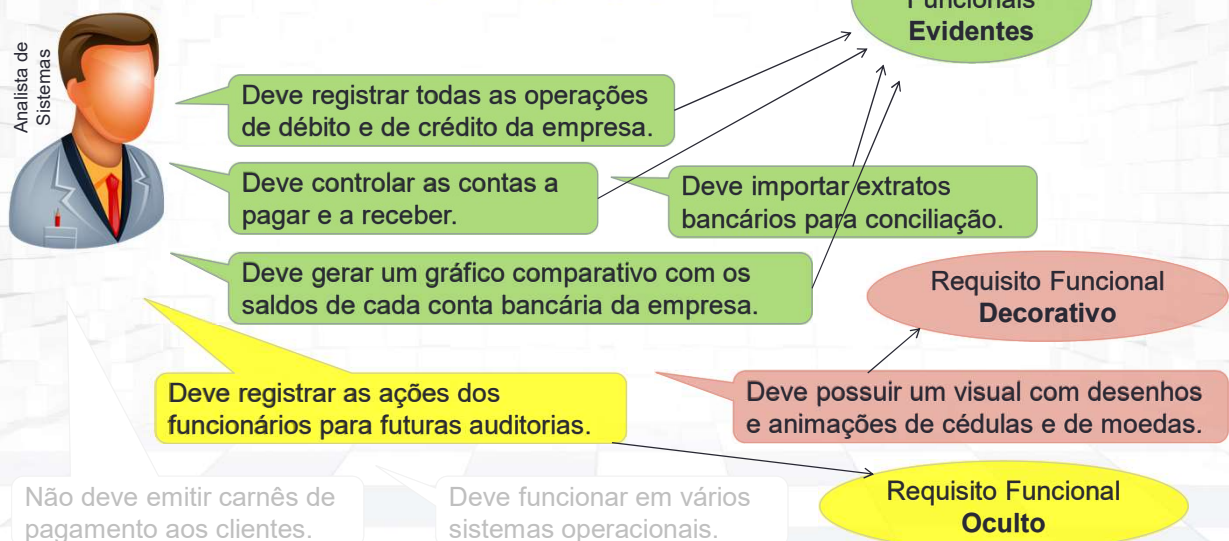
#### • Requisitos Funcionais [2/2]:

- **Evidente:** O usuário tem conhecimento de que a referida funcionalidade foi executada.
  - Ex. - **RF001:** O sistema deve permitir que um aluno realize sua matrícula em uma disciplina pela Internet
- **Oculto:** Funcionalidade que não é visível para o usuário. Vale para muitos serviços técnicos de infra-estrutura
  - Ex. – **RF004:** O sistema deve atualizar a quantidade do produto em estoque, ao vender e comprar.
  - Tais funções são frequentemente (e incorretamente) esquecidas durante a fase de especificação de requisitos
- **Decorativo:** Opcional. Sua adição pode ou não afetar significativamente o **custo** ou outras funções.
  - Ex. – **RF005:** O sistema deve exibir o logotipo da empresa em cada uma de suas telas

11

## Classificações de Requisitos

### Quanto às Classes de Organização [4/9]



12

## Classificações de Requisitos

### Quanto às Classes de Organização [5/9]

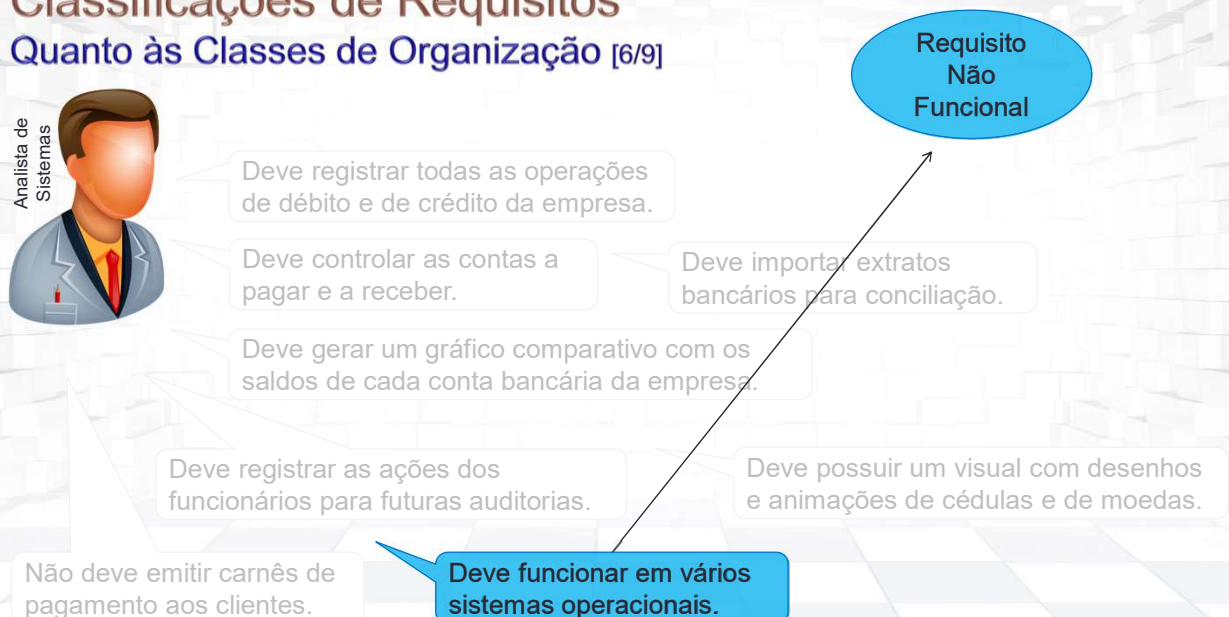
#### • Requisitos Não-Funcionais [1/2]:

- Também chamados de **requisitos suplementares**, estão relacionados aos **aspectos de qualidade** que o software deverá apresentar ou **restrições** a serem atendidas
- Descrevem as **restrições** para o software
  - São restrições sobre os serviços ou as funções oferecidos pelo sistema, tais como: restrições de tempo, restrições sobre o processo de desenvolvimento, padrões, entre outros.
- Exemplos:
  - **RNF001**: O tempo de resposta para uma consulta deve demandar no máximo 3 segundos
  - **RNF002**: O sistema deve ser executado no ambiente Windows
  - **RNF003**: O usuário será capaz de utilizar todas as funcionalidades dos sistema após 2 horas de treinamento

13

## Classificações de Requisitos

### Quanto às Classes de Organização [6/9]



14

## Classificações de Requisitos

### Quanto às Classes de Organização [7/9]

- **Requisitos Não-Funcionais [2/2]:**

- São frequentemente confundidos com funções
- Podem aplicar-se a todas as funções (o sistema como um todo) ou ser específicos de uma função particular ou grupo de funções
- Em geral, podem ser aplicados para qualquer sistema

15

## Classificações de Requisitos

### Quanto às Classes de Organização [8/9]

- **Requisitos Inversos:**

- Relacionados a **condições** que **nunca** poderão ocorrer ou que estão **fora do escopo** da solução
- São, de certa maneira, restrições de alcance geral (restrições de escopo)
- Ex.:
  - **RI001:** O sistema não deve permitir apagar os dados do cliente.
  - **RI002:** O sistema deverá ser implementado somente em idioma nacional.
  - **RI003:** O sistema não emitirá nota fiscal.
  - **RI004:** O sistema não permitirá vender a crediário.

16



## Classificações de Requisitos

### Quanto às Classes de Organização [9/9]



Analista de  
Sistemas

Deve registrar todas as operações de débito e de crédito da empresa.

Deve controlar as contas a pagar e a receber.

Deve importar extratos bancários para conciliação.

Deve gerar um gráfico comparativo com os saldos de cada conta bancária da empresa.

Deve registrar as ações dos funcionários para futuras auditorias.

Deve possuir um visual com desenhos e animações de cédulas e de moedas.

Não deve emitir carnês de pagamento aos clientes.

Deve funcionar em vários sistemas operacionais.

Requisito Inverso

17

## Classificações de Requisitos

### Quanto aos Níveis de Detalhes [1/6]

#### • Requisitos de Usuário:

- Declarações em linguagem natural (ou diagramas) sobre as funções que o sistema deve fornecer e as restrições sobre as quais deve operar (funcionais e não funcionais).
- Escritos para os clientes e usuários
- Exemplo:
  - **RF001:** O software deve oferecer um meio de representar e acessar arquivos externos criados por outras ferramentas

Sommerville, 2003

18

## Classificações de Requisitos

### Quanto aos Níveis de Detalhes [2/6]

#### • Requisitos de Sistema [1/2]:

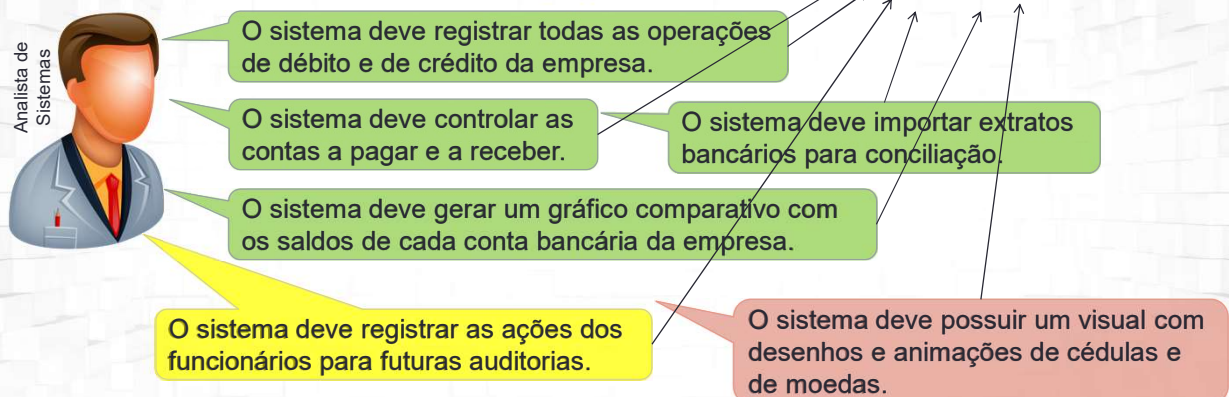
- Estabelecem detalhadamente as restrições que o sistema deverá atender (são descrições mais detalhadas dos requisitos do usuário; ponto de partida para o projeto do sistema).
- Escritos para servir como contrato
- Exemplos:
  - **RF001.1:** O usuário deve dispor de recursos para definir o tipo dos arquivos externos
  - **RF001.2:** Cada tipo de arquivo externo pode ter uma ferramenta associada
  - **RF001.3:** Cada tipo de arquivo externo pode ser representado por um ícone específico na tela do usuário
  - **RF001.4:** Quando um usuário seleciona um arquivo externo, a ferramenta associada é ativada para manipular adequadamente esse arquivo

Sommerville, 2003

19

## Classificações de Requisitos

### Quanto aos Níveis de Detalhes [3/6]



20

## Classificações de Requisitos

### Quanto aos Níveis de Detalhes [4/6]



Vamos definir alguns **requisitos de sistema** a partir destes 2 **requisitos de usuário**...

O sistema deve registrar todas as operações de débito e de crédito da empresa.

O sistema deve controlar as contas a pagar e a receber.

Para cada conta ser paga, o sistema deve gerar uma operação de débito com o status "em aberto".

O sistema deve alertar quando estiver perto da data de vencimento de uma conta a ser paga e quando uma conta não tiver sido paga.

Quando uma conta tiver sido paga, o sistema deve registrar a operação de débito correspondente como "quitada".

O sistema deve classificar cada operação de débito e de crédito por centros de custo e por contas contábeis.

O sistema deve gerar a planilha de fechamento das despesas e receitas mensais e a projeção para o mês seguinte.

O sistema deve parcelar as contas a receber, lançando cada parcela em uma operação de crédito separada.

O sistema deve marcar o cliente de negócio como "inadimplente" quando uma respectiva conta a receber estiver vencida a mais de um mês.

21

## Classificações de Requisitos

### Quanto aos Níveis de Detalhes [5/6]

#### • Requisitos de Sistema [2/2]:

- A especificação dos requisitos **funcionais** de **sistema** deve ser [Sommerville, 2003]:
  - **Completa**: significa que todas as funções requeridas pelo usuário devem estar definidas, e
  - **Consistente**: significa que os requisitos não devem ter definições contraditórias
- A dificuldade em se alcançar uma especificação de requisitos completa e consistente, em parte, está na complexidade inerente ao sistema e, em parte, nas necessidades inconsistentes de diferentes pontos de vista

22

## Classificações de Requisitos

### Quanto aos Níveis de Detalhes [6/6]

#### • Requisitos de Interfaces Externas:

- Interfaces externas estabelecem requisitos para que o sistema possa **interoperar** com **outros sistemas** e com os **usuários** (são, portanto, requisitos de sistema). O IEEE classifica as interfaces externas em:
  - **Interfaces de usuário**
    - Ex.: O sistema deve prover telas de fácil interpretação.
  - **Interfaces de hardware**
    - Ex.: O sistema deve permitir a captação da entrada do funcionário por meio de um dispositivo de ponto eletrônico.
  - **Interfaces de software**
    - Ex.: O sistema deve enviar os dados das vendas ao sistema financeiro.
  - **Interface de comunicação**
    - Ex.: O sistema deve enviar um SMS ao celular do gerente quando o estoque mínimo de algum produto for atingido.

23

## Classificações de Requisitos

### Quanto às Prioridades [1/4]

#### • Essencial:

- Um requisito pode ser classificado como **essencial** quando o mesmo deve ser implementado no sistema **impreterivelmente**.
- É um requisito **imprescindível**, compõe a essência do sistema.
  - Sem ele, o sistema não entrará em funcionamento, não será possível satisfazer as necessidades do cliente.
- Ex.: O sistema deve permitir ao aluno cadastrar uma autenticação para poder ter acesso às salas de aula.
  - Sem este requisito, o sistema não funcionaria, pois é imprescindível, pelas regras de negócio do cliente, que o aluno tenha acesso às salas de aula por meio do sistema.

24



## Classificações de Requisitos

### Quanto às Prioridades [2/4]

- **Importante:**

- Um requisito pode ser classificado como **importante** quando o mesmo deve ser implementado no sistema, mas, se não for, o sistema ainda assim poderá ser implantado e colocado em produção.
- É um requisito que, sem ele, o sistema **continua funcionando**, mas de maneira **insatisfatória**.
- Ex.: O sistema deve permitir que um professor envie mensagens para outros usuários de um curso.
  - Sem este requisito, o sistema continuará atendendo as necessidades do cliente, ainda que de maneira insatisfatória.

25

## Classificações de Requisitos

### Quanto às Prioridades [3/4]

- **Desejável:**

- Um requisito pode ser classificado como **desejável** quando sua ausência não compromete as funcionalidades básicas do sistema.
- É um requisito que, sem ele, o sistema pode continuar funcionando de maneira satisfatória.
- Tais requisitos podem ser deixados para versões posteriores do sistema caso não haja tempo hábil para implementá-los na versão que está sendo especificada.
- Ex.: O sistema deve permitir ao aluno visualizar gráficos de seu desempenho geral em suas disciplinas.
  - Sem este requisito, o sistema ainda funcionará de maneira satisfatória.

26

## Classificações de Requisitos

### Quanto às Prioridades [4/4]



O sistema deve registrar todas as operações de débito e de crédito da empresa.

Essencial

O sistema deve controlar as contas a pagar e a receber.

Essencial

O sistema deve importar extratos bancários para conciliação.

Importante

O sistema deve gerar um gráfico comparativo com os saldos de cada conta bancária da empresa.

Desejável

O sistema deve registrar as ações dos funcionários para futuras auditorias.

Importante

As prioridades podem variar de acordo com a necessidade de cada cliente.

O sistema deve possuir um visual com desenhos e animações de cédulas e de moedas.

Desejável

27

## Classificação de Requisitos

### Requisitos de Domínio [1/2] [Sommerville, 2003]

- São requisitos que se originam do domínio de aplicação do sistema e que refletem características desse domínio.
  - Podem ser requisitos funcionais ou não funcionais
- São derivados do **domínio da aplicação** do sistema (regras de negócios), em vez de serem obtidos a partir de necessidades específicas dos usuários do sistema
  - Podem ser novos requisitos, podem restringir os existentes ou estabelecer como devem ser realizados cálculos específicos
- São importantes por refletirem fundamentos do domínio da aplicação

28

## Classificação de Requisitos

### Requisitos de Domínio [2/2] [Sommerville, 2003]

- Exemplo – Para um sistema de biblioteca:
  - O sistema deve prover uma interface-padrão para o banco de dados, que terá como base o padrão Z39.50.
  - Trata-se de uma restrição sobre uma exigência funcional de sistema que especifica que a comunicação com o BD deve ser implementada de acordo com um padrão específico de bibliotecas.

29

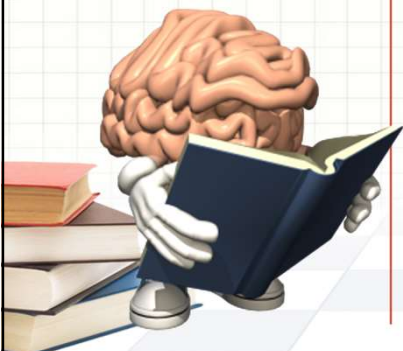
## Organização dos Requisitos de Software

### Sequência de Execução

- Devemos organizar os requisitos por sequência de execução:
  - **Funções de Entrada:** Responsáveis pelas entradas de dados no sistema. Ex.:
    - Cadastros
    - Importações
  - **Funções de Processamento:** Transformam/manipulam dos dados do sistema (**transações de negócio/movimentações**). Ex.:
    - Realizações de vendas
    - Cálculos de salários mensais
  - **Funções de Saída:** Retornam dados para os usuários finais. Ex.:
    - Consultas (gráficos, tabelas, listagens) em telas ou impressas

30

## Tópicos



### Estudo de Caso

Exemplo de requisitos de um sistema TPV

31

## Sistema TPV



- O TPV (Terminal de Ponto de Vendas) é um sistema computadorizado usado para registrar vendas e cuidar de pagamentos
- Tipicamente usado em vendas a varejo
- Inclui componentes de software e de hardware, tais como um computador e um leitor de código de barras

32



## Sistema TPV

### Definições [1/2]

- Descrição Geral
  - O propósito deste projeto é criar um terminal de ponto de vendas (TPV) para ser usado em lojas de varejo
- Clientes
  - ObjectStore, Inc. – multinacional que comercializa objetos

33

## Sistema TPV

### Definições [2/2]

- Objetivo
  - Aumentar a automatização das compras (*checkout*) para permitir serviços e processos comerciais mais rápidos, melhores e mais baratos
  - Tipicamente, isso inclui:
    - Checkout (passagem pelo caixa) mais rápido para o cliente
    - Análise rápida e precisa do crédito
    - Controle automático do estoque
- Lembrete em relação aos requisitos:
  - Devem sempre estar organizados em classificações e contextos

34

## Sistema TPV

### Requisitos Funcionais – Contexto de Vendas

Requisito  
Evidente

- **RF01** (E): O sistema deve calcular o total da venda, incluindo os cálculos de impostos e de cupons de desconto.
- **RF02** (E): O sistema deve incluir o produto na venda, usando um leitor de código de barras.
- **RF03** (E): O sistema deve exibir o nome e o preço do produto incluído na venda.
- **RF04** (o): O sistema deve reduzir a quantidade em estoque quando a venda for finalizada.

Requisito  
Oculto

35

## Sistema TPV

### Requisitos Funcionais – Contexto de Pagamento

- **RF05** (E): O sistema deve tratar os pagamentos em dinheiro: capturar a quantia recebida e informar o troco.
- **RF06** (o): O sistema deve registrar os pagamentos por crédito, no sistema de contas a receber da loja.

36

## Sistema TPV

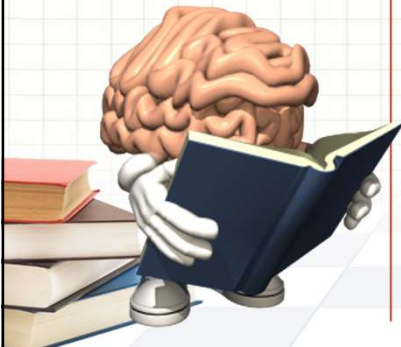
### Requisitos Não Funcionais (Qualidade)

- **RNF01:** O sistema deve garantir segurança nos dados.
- **RNF02:** O sistema deve ser tolerante a falhas.

37

## Tópicos

Regras de Negócio



38

# Regras de Negócio

## Introdução

- Uma **regra de negócio** é uma declaração que define ou restringe explicitamente um aspecto do funcionamento da empresa (a forma de fazer o negócio)
  - refletindo a política interna, o processo definido e as regras básicas de conduta.
- Sob essa ótica:
  - Nada existe se não estiver definido em uma regra de negócio;
  - Tudo é permitido se não houver uma restrição estabelecida em uma regra de negócio.
- As regras de negócio definem a **estrutura** e controlam a **operação** de uma empresa.
  - Concentram o foco nas restrições sob as quais as empresas operam

39

# Regras de Negócio

## Definições

- Uma regra de negócio é uma diretiva específica, acionável, testável, que está sobre o controle de uma organização e que suporta a política do negócio.

BABOK 2.0

40



## Regras de Negócio

### Considerações

- Regras de negócio descrevem, restringem ou controlam os dados ou as atividades de um processo de negócio.
  - O conjunto das regras de negócios também é conhecido como **domínio da aplicação do sistema** ou **domínio do negócio**.
- Expressam aspectos estáticos e dinâmicos do negócio.
  - A automatização de processo de negócios exige a **automatização das regras** que regem estes processos.
  - Mas elas tem vida totalmente independente de sistemas de software, podem ser criadas e obedecidas sem o uso deles.

41

## Regras de Negócio e Requisitos

### Exemplo

- Uma microempresa, totalmente desprovida de computadores ou de qualquer tipo de automação de processo de negócio, definiu como política que:
  - **A venda a prazo só poderá ser feita para clientes adimplentes.**
- A administração da empresa resolveu automatizar o processo de vendas. Sendo assim:
  - **O sistema deve permitir a realização da venda apenas se o cliente for adimplente.**

Regra de negócio,  
definida independente  
de sistema algum.

Requisito de domínio,  
definido para atender  
a regra de negócio.

42

## Regras de Negócio

### Exemplos

Código	Título	Descrição
RN01	Vendas por Produto	Vendas por produto é igual à quantidade vendida multiplicada pelo valor unitário
RN02	Formas de Pagamento	São aceitos no estabelecimento somente pagamentos em espécie e em cartão (débito ou crédito). Não são aceitos cheques.
RN03	Venda à Vista	Uma venda é considerada à vista quando é paga em espécie, em cartão de débito ou em cartão de crédito sem parcelamentos.
RN04	Desconto em Venda à Vista	Quando uma venda à vista é paga em espécie, é concedido um desconto de 10%; com cartão de débito, desconto de 5%.
RN05	Parcelamentos	Os pagamentos podem ser feitos à vista ou 1 entrada + uma quantidade de parcelas dependendo do valor da venda.
RN06	Registro das Vendas	Cada venda realizada deve obrigatoriamente ser registrada no sistema.

43

## Regras de Negócio

### Considerações

- Regras de negócio podem ser representadas em **métodos formais / linguagens processáveis** (UML, Linguagem Z, **Português Estruturado** etc), com a finalidade de **automatizá-las em um software**.
- Exemplo em Português Estruturado:

Código	Título	Descrição
RN04	Desconto em Venda à Vista	<b>Se</b> venda paga em espécie <b>então</b> desconto de 10% <b>Senão se</b> venda paga em cartão de débito <b>então</b> desconto de 5% <b>Senão</b> sem desconto <b>Fim se</b>

44

## Regras de Negócio e Requisitos

### Comparações

- Ambos capturam o conhecimento do negócio.
- Ambos esclarecem as necessidades e expectativas do cliente
- Ambos refletem as necessidades do negócio.
- Regras de negócio são definidas no âmbito organizacional/corporativo.
- Requisitos são definidos no âmbito do sistema em desenvolvimento.

45

## Regras de Negócio e Requisitos

### Exemplo [1/4]

Em um sistema a ser desenvolvido para lidar com cálculos trigonométricos, considere as seguintes regras de negócio:



Código	Título	Descrição
RN01	Triângulo Equilátero	Um triângulo é equilátero quando seus 3 lados são iguais.
RN02	Triângulo Isóceles	Um triângulo é isóceles quando 2 de seus lados são iguais.
RN03	Triângulo Escaleno	Um triângulo é escaleno quando nenhum de seus lados é igual.
RN04	Lados do Triângulo	Forma-se um triângulo quando qualquer um dos lados é menor que a soma dos outros 2 lados.

46

## Regras de Negócio e Requisitos

### Exemplo [2/4]

Na fase de definição, com o intuito de melhor entender as regras e definir uma futura rotina dentro do sistema, o analista de sistemas define a seguinte regra:



Código	Título	Referências Cruzadas
RN05	Tipo do Triângulo	RN01; RN02; RN03; RN04.
<b>Se</b> qualquer lado for menor que a soma dos demais <b>então</b>		
<b>Se</b> três lados forem iguais <b>então</b>		
Tipo do triângulo é equilátero		
<b>Se</b> dois lados forem iguais <b>então</b>		
Tipo do triângulo é isósceles		
<b>Se</b> nenhum dos lados forem iguais <b>então</b>		
Tipo do triângulo é escaleno		
<b>Senão</b>		
Não há triângulo		
<b>Fim se</b>		

47

## Regras de Negócio e Requisitos

### Exemplo [3/4]

Ainda na fase de definição, o analista de sistemas define os requisitos para atender tanto as necessidades do cliente quanto as regras de negócio.



Código	Descrição	Referências Cruzadas
RF01	O sistema deve validar o triângulo a partir das dimensões de seus lados.	RN04; RN05
RF02	O sistema deve definir o tipo do triângulo (equilátero, isósceles ou escaleno) a partir das dimensões de seus lados.	RN01; RN02; RN03; RN05
RF03	...	...
...	...	...

48



## Regras de Negócio e Requisitos

### Exemplo [4/4]

```
/* Valida um triângulo e verifica seu tipo.
 *
 * @param a Valor do primeiro lado do triângulo
 * @param b Valor do segundo lado do triângulo
 * @param c Valor do terceiro lado do triângulo
 * @return 3 se o triângulo é equilátero
 *         2 se o triângulo é isósceles
 *         1 se o triângulo é escaleno
 *         0 se a, b e c não formam um triângulo
 */
int tipoTriangulo(int a, int b, int c) {
    int ret;
    if (a < b + c || b < a + c || c < a + b) { //forma triângulo
        if (a == b && b == c)
            ret = 3; //triângulo equilátero
        else if (a == b || b == c || a == c)
            ret = 2; //triângulo isósceles
        else
            ret = 1; //triângulo escaleno
    }
    else
        ret = 0; //não forma triângulo
    return ret;
}
```



Na fase de construção, com base nas regras e nos requisitos, o desenvolvedor programa uma rotina semelhante a esta.