JAVASCRIPT

FAPESC – DESENVOLVEDORES PARA TECNOLOGIA DA INFORMAÇÃO

FRANCIELE PETRY
Franciele.petry@unoesc.edu.br

TRATAMENTO DE ERRROS

```
1  function soma(x,y){ //funções construtoras e tratamentos de erros
2      if(
3          typeof x !== 'number' ||
4          typeof y !== 'number'
5      ) {
6          throw new ReferenceError('mensagem: x e y precisam ser numeros')
7      }
8      return x+y;
9  }
10
11 try{
12     console.log(soma(1,2));
13     console.log(soma('1',2));
14 } catch (error){
15     console.log(error)
16 }
17
```

string

TERMINAL JUPYTER PROBLEMS OUTPUT DEBUG CONSOLE

```
ReferenceError: mensagem: x e y precisam ser numeros
    at soma (c:\ProgramacaoWeb\JS\maisjs.js:8:15)
    at Object.<anonymous> (c:\ProgramacaoWeb\JS\maisjs.js:15:17)
```

Tratamento de Erros

```
1  function soma(x,y){ //funções construtoras e tratamentos de erros
2      if(
3          typeof x !== 'number' ||
4          typeof y !== 'number'
5      ) {
6          throw new ReferenceError('mensagem: x e y precisam ser numeros')
7      }
8      return x+y;
9  }
10
11 try{
12     console.log(soma(1,2));
13     console.log(soma(1,2));
14     //console.log(soma('1',2)); //para gerar erro
15 } catch (error){
16     console.log(error)
17 } finally {
18     console.log("eu sempre sou executado com ou sem erro")
19 }
```

OO – Classes – Objetos – Atributos e Métodos

```
23
24  const pessoa = new Object();
25  pessoa.nome = "Fran";
26  pessoa.sobrenome = "Petry";
27
28  console.log(pessoa)
29
```

TERMINAL JUPYTER PROBLEMS OUTPUT DEBUG CONSOLE

```
[Running] node "c:\ProgramacaoWeb\JS\maisjs.js"
{ nome: 'Fran', sobrenome: 'Petry' }
```

```
24  const pessoa = new Object();
25  pessoa.nome = "Fran";
26  pessoa.sobrenome = "Petry";
27
28  pessoa.falarNome = function(){
29      console.log(`${this.nome} Está falando seu Nome`)
30  };
31
32  //console.log(pessoa)
33  pessoa.falarNome();
34
```

TERMINAL JUPYTER PROBLEMS OUTPUT DEBUG CONSOLE

```
[Done] exited with code=0 in 0.125 seconds

[Running] node "c:\ProgramacaoWeb\JS\maisjs.js"
Fran Está falando seu Nome
```

OO – Classes – Objetos – Atributos e Métodos

```
24  const pessoa = new Object();
25  pessoa.nome = "Fran";
26  pessoa.sobrenome = "Petry";
27  pessoa.idade = 30;
28
29  pessoa.falarNome = function(){
30      console.log(`${this.nome} Está falando seu Nome`)
31  };
32  //console.log(pessoa)
33  pessoa.falarNome();
34
35  pessoa.getDataNascimento = function() {
36      const dataAtual = new Date();
37      return dataAtual.getFullYear() - this.idade;
38  };
39
40  console.log(pessoa.getDataNascimento());
```

```
TERMINAL  JUPYTER  PROBLEMS  OUTPUT  DEBUG CONSOLE
[Running] node "c:\ProgramacaoWeb\JS\maisjs.js"
Fran Está falando seu Nome
1992
```

OO – Classes – Objetos – Atributos e Métodos

```
41
42  for (let chave in pessoa) {
43      console.log (pessoa[chave]) // chave de valor e as functions que estão dentro dos objetos chamamos de métodos
44  }
```

TERMINAL JUPYTER PROBLEMS OUTPUT DEBUG CONSOLE

```
[Running] node "c:\ProgramacaoWeb\JS\maisjs.js"
Fran Está falando seu Nome
1992
Fran
Petry
30
[Function (anonymous)]
[Function (anonymous)]
```

OO – Classes – Objetos – Atributos e Métodos

```
47  function Pessoa(nome, sobrenome, idade){
48      this.nome = nome;
49      this.sobrenome = sobrenome;
50      this.idade = idade;
51  }
52
53  const p1 = new Pessoa('Fran', 'Petry', '30'); //construtor (Pessoa), é a função que construiu o objeto
54
55  console.log(p1);
```

```
TERMINAL  JUPYTER  PROBLEMS  OUTPUT  DEBUG CONSOLE
[Running] node "c:\ProgramacaoWeb\JS\maisjs.js"
Pessoa { nome: 'Fran', sobrenome: 'Petry', idade: '30' }
```

OO – Classes – Objetos – Atributos e Métodos

```
47  function Pessoa(nome, sobrenome, idade){
48      this.nome = nome;
49      this.sobrenome = sobrenome;
50      this.idade = idade;
51
52      Object.defineProperty(this, 'idade', {
53          enumerable:true,
54          configurable: true,
55          get:function() {
56           return idade;
57      }
58  });
59  }
60
```

```
TERMINAL  JUPYTER  PROBLEMS  OUTPUT  DEBUG CONSOLE
[Done] exited with code=1 in 0.138 seconds

[Running] node "c:\ProgramacaoWeb\JS\maisjs.js"
Pessoa { nome: 'Fran', sobrenome: 'Petry', idade: [Getter] }
```

OO – Classes – Objetos – Atributos e Métodos

```
47  function Pessoa(nome, sobrenome, idade){
48      this.nome = nome;
49      this.sobrenome = sobrenome;
50      this.idade = idade;
51
52      Object.defineProperty(this, 'idade', {
53          enumerable:true,
54          configurable: true,
55          get:function() {
56           return idade;
57      },
58      set idade(valor){
59          numero = valor;
60      }
61  });
62  }
63
64  const p1 = new Pessoa('Fran', 'Petry', 30); //construtor (Pessoa), é a função que construiu o objeto
65  p1.numero = 32;
66  console.log(p1);
67
68
```

TERMINAL JUPYTER PROBLEMS OUTPUT DEBUG CONSOLE

```
[Done] exited with code=1 in 0.216 seconds

[Running] node "c:\ProgramacaoWeb\JS\maisjs.js"
Pessoa {
  nome: 'Fran',
  sobrenome: 'Petry',
  idade: [Getter],
  numero: 32
}
```

UNOESC

CENTRO DE INOVAÇÃO VIDEIRA

SANTA CATARINA

OO – Classes – Objetos – Atributos e Métodos

```
71  class Pessoa{
72      constructor(nome, sobrenome){
73          this.nome = nome;
74          this.sobrenome = sobrenome;
75      }
76
77
78  }
79
80  const pessoa1 = new Pessoa('Fran', 'Petry');
81  console.log(pessoa1);
82
83
84
```

TERMINAL JUPYTER PROBLEMS OUTPUT DEBUG CONSOLE

```
[Done] exited with code=1 in 0.154 seconds

[Running] node "c:\ProgramacaoWeb\JS\maisjs.js"
Pessoa { nome: 'Fran', sobrenome: 'Petry' }
```

```
71  const idade_ = Symbol('idade');
72  class Pessoa{
73      constructor(nome, sobrenome){
74          this.nome = nome;
75          this.sobrenome = sobrenome;
76          this[idade_] = 30;
77
78      }
79
80      falar(){
81          console.log(`${this.nome} esta falando`)
82      }
83
84      get idade(){
85          return this[idade_];
86      }
87
88      set idade(valor){
89          if(typeof valor !== 'number') return;
90      }
91  }
92
93  const pessoa1 = new Pessoa('Fran', 'Petry');
94  console.log(pessoa1);
95  pessoa1.falar();
96  pessoa1.idade = 31;
97
```

TERMINAL JUPYTER PROBLEMS OUTPUT DEBUG CONSOLE

```
[Running] node "c:\ProgramacaoWeb\JS\maisjs.js"
Pessoa { nome: 'Fran', sobrenome: 'Petry', [Symbol(idade)]: 30 }
Fran esta falando
```

UNOESC

CENTRO DE INOVAÇÃO VIDEIRA

GOVERNO DE SANTA CATARINA

REQUISITOS DE SISTEMAS

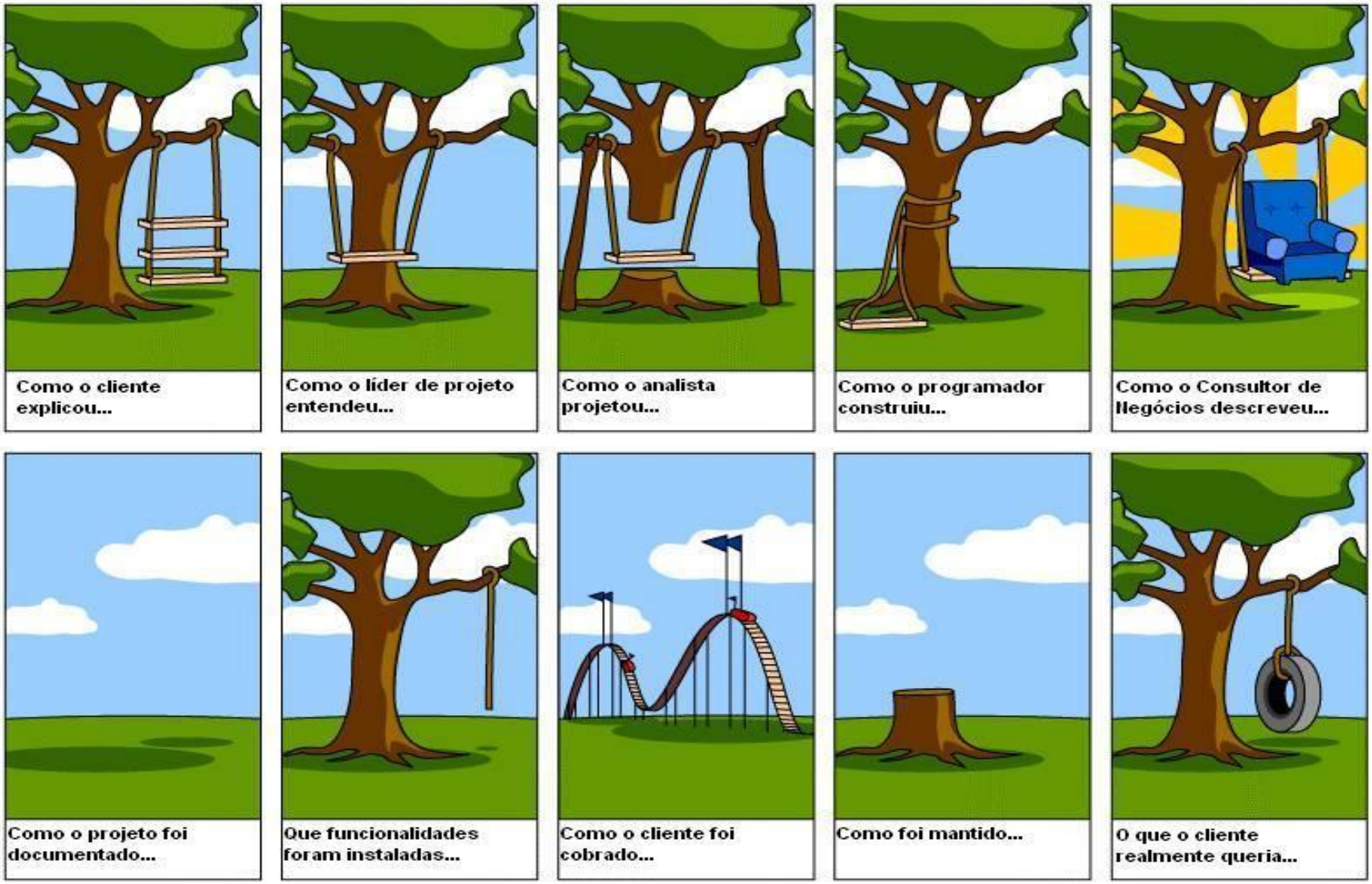
ANÁLISE DE REQUISITOS

Processo de descobrir, analisar, documentar e verificar serviços requeridos para um sistema e suas restrições operacionais.

=

REQUERIMENTOS

Requisitos de Sistemas



Requisitos de Sistemas

Pode variar de uma declaração abstrata de alto nível de um serviço ou de uma restrição de sistema para uma especificação matemática;

Isto é inevitável quando os requisitos podem ter estas características:

- Ser a base para uma proposta de um contrato – portanto deve ser aberta para interpretação;
- Ser a base para o contrato em si – portanto deve ser definido em detalhe;
- Ambas as declarações podem ser chamadas requisitos.

Requisitos de Sistemas - Tipos

Requisitos de usuário

Declarações em linguagem natural + diagramas de serviços que o sistema fornece e suas restrições operacionais. Escritos para os usuários;

Requisitos de sistema

Um documento estruturado estabelecendo descrições detalhadas das funções, serviços e restrições operacionais do sistema. Define o que deve ser implementado e assim, pode ser parte de um contrato entre o cliente e o desenvolvedor.

Requisitos de Sistemas - Classificação

REQUISITOS FUNCIONAIS

Declarações de serviços que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como o sistema deve se comportar em determinadas situações.

REQUISITOS NÃO FUNCIONAIS

Restrições sobre serviços ou funções oferecidos pelo sistema tais como restrições de timing, desempenho, restrições sobre o processo de desenvolvimento, padrões, interface com o usuário, qualidade, etc.

REQUISITOS DE DOMÍNIO

Requisitos que vêm do domínio de aplicação do sistema e que refletem as características desse domínio.

Requisitos Funcionais

Descrevem a funcionalidade ou serviços de sistema;

Dependem do tipo de software, dos usuários esperados e o tipo de sistema onde o software é usado;

Requisitos funcionais de usuário podem ser declarações de alto nível do que o sistema deve fazer mas os requisitos funcionais de sistema devem descrever os serviços de sistema em detalhe.

Requisitos Funcionais

RF001 - O usuário deve ser capaz de pesquisar em todo o conjunto inicial de banco de dados ou selecionar um subconjunto a partir dele;
Rf002 - O sistema deve fornecer telas apropriadas para o usuário ler os documentos no repositório de documentos;
RF003 - Para todo pedido deve ser alocado um identificador único (ORDER_ID) no qual o usuário deve ser capaz de copiar para a área de armazenamento permanente da sua conta;
RF004 - Incluir cliente pessoa física;
RF005 - Excluir cliente pessoa jurídica;
RF005 - Sistema deve emitir um relatório de clientes.

Requisitos Funcionais

Problemas surgem quando os requisitos não são precisamente definidos;

Requisitos ambíguos podem ser interpretados de maneiras diferentes pelos desenvolvedores e usuários;

Considere o termo ‘telas apropriadas’:

Intenção do usuário – tela de propósito especial para cada tipo diferente de documento;

Interpretação do desenvolvedor – fornece uma tela de texto que mostra o conteúdo do documento.

REQUISITOS FUNCIONAIS

RF006 - O sistema deve ser capaz de armazenar todas as informações sobre seus clientes(RG, CPF, Nome, data de nascimento e endereço) no banco de dados;
RF007 - O sistema deverá atribuir um identificador único (código) para cada pedido de produtos;
RF008 - O sistema deverá cancelar automaticamente um orçamento que tenha sido feito há mais de 30 dias e não tenha sido transformado em venda.

Requisitos Não Funcionais

- Estes definem propriedades e restrições de sistema, por exemplo, confiabilidade, tempo de resposta e requisitos de armazenamento. Restrições são capacidade de dispositivos de E/S, representações de sistema, etc;
- Requisitos de qualidade(ver ISO/IEC 9126);
- Podem ainda estar relacionados a portabilidade, de SO, de BD, etc;
- Requisitos de processo podem também ser especificados impondo uma ferramenta CASE particular, linguagem de programação ou método de desenvolvimento;
- Requisitos não funcionais podem ser muitas vezes mais críticos do que os requisitos funcionais. Se estes não forem atendidos, o sistema é inútil.

REQUISITOS NÃO FUNCIONAIS

REQUISITOS DE PRODUTO

Requisitos que especificam que o produto entregue deve se comportar de uma maneira particular, por exemplo, velocidade de execução, confiabilidade, etc.

REQUISITOS ORGANIZACIONAIS

São uma consequência de políticas e procedimentos da organização, por exemplo, padrões de processo usados, requisitos de implementação, etc.

REQUISITOS EXTERNOS

Surgem a partir de fatores externos ao sistema e seu processo de desenvolvimento, por exemplo, requisitos de interoperabilidade, requisitos legais, etc.

Requisitos Não Funcionais

RNF001 - O sistema deve imprimir o relatório em até 5 segundos;
RNF002 - Todos os relatórios devem seguir o padrão de relatórios especificado pelo setor XYZ;
RNF004 - O sistema deve ser implementado em Java;
RNF005 - A base de dados deve ser protegida para acesso apenas de usuários autorizados;
RNF006 - O software deve ser operacionalizado no sistema Linux;
RNF007 - O tempo de resposta do sistema não deve ultrapassar 30 segundos.

Requisitos De Domínio

Derivados do domínio de aplicação e descrevem características de sistema que refletem o domínio;

Podem restringir os requisitos funcionais existentes ou estabelecer como cálculos específicos devem ser realizados;

Se os requisitos de domínio não forem satisfeitos, o sistema pode não funcionar.

Requisitos De Domínio

RD001 - O calculo da média final de cada aluno é dado pela fórmula: (Nota1 * 2 + Nota2 * 3)/5;
RD002 - Um aluno pode se matricular em uma disciplina desde que ele tenha sido aprovado nas disciplinas consideradas pré-requisitos;
RD003 - Livros identificados como exemplar 2 não podem ser reservados ou emprestados;
RD004 - O arredondamento das notas precisa seguir os padrões da Norma ABNT NBR 5891 utilizada em todo território nacional.