**Universidade Comunitária da Região de Chapecó – Unochapecó**

**Escola Politécnica – Ciência da Computação/Sistemas de Informação**

**Engenharia de Software I**

**Prof. Radamés Pereira**

**Profa. Monica Tissiani De Toni Pereira**

**Roteiro para especificação de requisitos de software**

1. **Atributos de uma boa especificação de requisitos de projeto de software:**

* Clareza
* Não Ambígua
* Completa
* Simples
* Bem escrita

Referências:

IEEE Std. 830 – 1993. IEEE Recommended Practice for Software Requirements Specifications.

IEEE ISO/IEC/IEEE 29148 – 2011. IEEE Systems and software engineering — Life cycle processes — Requirements engineering.

**Obs.:** Os itens deste modelo de especificação, recomendado pela IEEE, poderão ser complementados com novos itens, caso sejam justificáveis.

**Universidade Comunitária da Região de Chapecó Unochapecó - Escola Politécnica**

**Ciência da Computação/Sistemas de Informação**

Título do projeto de software

...

Equipe (engenheiro e desenvolvedores): **...**

**...**

**…**

1. Introdução
   1. Resumo do Projeto - Descrição textual

Descreve-se aqui o sistema a ser desenvolvido.

* 1. Plataforma de desenvolvimento

Hardware e software necessários para o desenvolvimento do projeto.

Descreve-se aqui uma primeira visão das tecnologias para desenvolvimento do projeto de software.

* 1. Plataforma de operação

É o equipamento do cliente/usuário do sistema.

Descreve-se aqui uma primeira visão das tecnologias para operacionalização do sistema, no contexto e ambiente do cliente.

* 1. Definições e siglas

Quaisquer siglas utilizadas no domínio do usuário e do vocabulário do usuário. Descreve-se aqui a definição de todas as siglas, abreviações e termos usados.

* 1. Perspectiva do produto
     1. Modos de operação

Meios de acesso ao sistema e arquitetura do sistema. Identificam-se aqui os modos requeridos de operação, tais como: Back-end, Front-end, Móvel, Stand-alone etc.

* + 1. Requisitos de adaptação ao ambiente

Aspectos legais para aderência a legislação, ambiente de operação crítica como indústria, automação, protocolos de comunicação específicos.

Definem-se aqui possíveis requisitos de adaptação do produto aos ambientes particulares onde ele será implantado. Por exemplo, parâmetros e métodos de configuração requeridos para ambientes específicos devem ser descritos aqui.

|  |  |  |
| --- | --- | --- |
| **Número de ordem** | **Requisito** | **Detalhes** |
| 1 | Configuração de ticket de venda e da Nota Fiscal Eletrônica | Configuração dos campos de formulário com interface responsiva. |

* 1. Funções do produto

Funções básicas: R1.1 , R1.2, R1.3, ...

Identificam-se aqui as principais funções que o produto desempenhará, descrevendo de forma sintética o objetivo de cada uma.

* 1. Características dos usuários

Identificação da formação dos usuários para suas especialidades no uso do sistema, tal como ensino-médio, graduação, especialista, engenheiro, doutor.

Descrevem-se aqui as principais características dos grupos de usuários esperados para o produto, tais como cargo ou função, permissão de acesso, frequência de uso, nível de instrução, proficiência no processo de negócio e proficiência em informática.

* 1. Restrições

Como Lei Geral de proteção de dados, por exemplo.

Descrevem-se aqui aspectos técnicos e gerenciais que possam limitar as opções dos desenvolvedores, tais como restrições legais.

* 1. Hipóteses de trabalho

Por exemplo: Sistema operacional, versão de ferramentas de software, licenças de bibliotecas e de subsistemas. Descrevem-se aqui fatores que não são restrições limitativas do desempenho, como na subseção anterior, mas fatores cuja alteração requer modificações na ER, como, por exemplo, versão a ser utilizada do ambiente operacional ou plataforma de desenvolvimento.

1. Requisitos específicos
   1. Interfaces externas

### Visão geral

Tipos de interface, relatórios, gráficos, dashboards.

Descreve-se aqui, de forma detalhada, todas as entradas e saídas do produto.

### Requisitos para interfaces gráficas de usuário

Mockups e wireframes.

Sugere-se, no caso de interfaces gráficas, a inclusão dos seguintes elementos:

1. Um esboço do layout gráfico sugerido para a interface.
2. Uma descrição dos relacionamentos com outras interfaces.
3. Um diagrama de estados/atividades, caso necessário para melhor entender-se o comportamento requerido da interface.
4. Uma lista dos campos de dados da interface.
5. Uma lista dos comandos da interface.
6. BPM.
   1. Requisitos funcionais

### Diagramas de casos de uso

Modelo UML de Casos de Uso.

Incluir todos os casos de uso que se pretende implementar em uma liberação. Pode-se incluir ainda um certo caso de uso e seus relacionamentos, todos os casos de uso para um certo ator.

### Fluxos dos casos de uso

Casos de Uso Expandidos e Diagramas de Atividades.

1. Pré-condições para a realização do caso de uso.
2. Fluxo principal do caso de uso (sucesso), descrito na forma de uma sequência de passos.
3. Fluxos alternativos do caso de uso.
4. Descrições mais formais, como diagramas de estado ou de atividade, se a complexidade do caso de uso exigir.
5. Observações.
   1. Requisitos não-funcionais

### Requisitos de desempenho

Velocidade de banda, tempo de resposta das interfaces e/ou impressão de relatórios. Requisitos de desempenho devem ser especificados de forma quantitativa e mensurável.

### Requisitos de dados persistentes

Sistemas de gerenciamento de banco de dados e modelo de persistência.

Descrevem-se aqui estruturas lógicas de dados persistentes, que mantém seu valor após a execução do programa e que sejam usadas pelo produto. Cada estrutura de dados pode ser, por exemplo, um arquivo convencional ou uma tabela em um banco de dados.

Obs.: Incluir aqui o modelo de banco de dados.

### Restrições ao desenho

Restrições de projeto impostas por padrões externos, com influência da legislação, tipo legislação fiscal municipal, estadual e federal.

### Atributos de Qualidade

Indica os atributos de qualidade, seguindo as características e sub características recomendadas pela norma [ISO-9126](https://drive.google.com/open?id=14-OfxrvT5pD4sblFt234r8NPBNwt2HmG) ou mais atual.

* 1. Objetos/Classes

### Modelo Conceitual

Ou Classes de Análise, ou Modelo de Domínio. São sinônimos para: Classes, Associações, nomes das associações, Multiplicidades e Atributos, mas sem métodos.

### DSS – Diagramas de Sequência do Sistema

Eventos e Operações de Casos de Uso, denotando as mensagens entre os objetos do domínio para atender ao Caso de Uso).

### Contratos (das Operações do DSS)

### Classes de Implementação

Ou comumente denominado genericamente de Diagrama de Classes, com: Classes, Associações, nomes das associações, Multiplicidades, Atributos e com Métodos. Atribuição de responsabilidades com GRASP (General Responsibility Assignment Software Patterns) que são um conjunto de princípios e diretrizes para atribuição de responsabilidades em projetos de software orientados a objetos.

1. Análise de UCP

As tabelas de escopo de valor do produto e tempo de desenvolvimento com Use Case Points - UCP.

1. Referências

IEEE Std. 830 – 1993. IEEE Recommended Practice for Software Requirements Specifications.

IEEE ISO/IEC/IEEE 29148 – 2011. IEEE Systems and software engineering — Life cycle processes — Requirements engineering.

**Obs.:** Os itens deste modelo de especificação, recomendado pela IEEE, poderão ser complementados com novos itens, caso sejam justificáveis.