

# ABEX III - PROJETO DE SOFTWARE- 1030784

2023-01

<b>Professor(a)(es)</b> )	RADAMÉS PEREIRA VIVIANE DUARTE BONFIM
------------------------------	---

Atributos de uma boa especificação de Projeto de Software:

- Clareza
- Não Ambígua
- Completa
- Simples
- Bem escrita

**UNIVERSIDADE COMUNITÁRIA REGIONAL DE  
CHAPECÓ - UNOCHAPECÓ  
ÁREA DE CIÊNCIAS EXATAS E AMBIENTAIS  
SISTEMAS DE INFORMAÇÃO/CIÊNCIA DA COMPUTAÇÃO**

**Informática básica para idosos**

**Equipe: Gabriel Da Silva Machado  
Bruno Batisti  
Bruno Luiz Rigotto  
Eduardo Zankett  
Endrick Martini  
Carlos Eduardo de Matos  
Julio César Ferreira  
Leonardo Feil  
Matheus Padilha de Oliveira**

# **Introdução**

## **1.1 Resumo do Projeto**

Descreve-se aqui o sistema a ser desenvolvido.

Nosso objetivo é disponibilizar conteúdos para que as pessoas da 3ª idade possam se habituar com a tecnologia do mundo atual, ensinando-as a lidar com aparelhos eletrônicos, como: celulares, tablets, notebooks, computadores e entre outros. Além de auxiliar no gerenciamento de redes sociais, criar e-mail, ensinar a mexer nas plataformas de streaming.

## **1.2 Plataforma de desenvolvimento**

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

Exemplo: Tecnologias, Sistema de edição; Máquina virtual...

Algumas das principais tecnologias que podem ser usados é para front-end: HTML, CSS e JavaScript. Algumas das possíveis para o back-end: C++, C, C#, python, java, node.js, etc

## **1.3 Plataforma de operação**

Descreve-se aqui uma primeira visão das tecnologias para operacionalização.

Exemplo: Onde sistema será executado: Cliente-servidor...

Será um software voltado para o windows, mas futuramente atingir linux e mac.

## **1.4 Definições e siglas (em andamento)**

Descreve-se aqui a definição de todas as siglas, abreviações e termos usados.

Conceitos do domínio: Siglas de conceitos específicos...

## **1.5 Perspectiva do produto (em andamento)**

### **1.5.1 Modos de operação**

Identificam-se aqui os modos requeridos de operação, tais como:  
Back-End-Front-End, Móvel, Stand-Alone, ...

Back-end e front-end.

### **1.5.2 Requisitos de adaptação ao ambiente**

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.

## **1.6 Funções do produto (em andamento)**

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

Exemplo: Caso de uso: função que será realizada.

### **1.7 Características dos usuários (em andamento)**

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.

Exemplos: usuários do nível gerencial; operacionais;

### **1.8 Restrições (em andamento)**

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

Exemplo: sistemas de vendas se trabalho sistema fiscal de vendas.

### **1.9 Hipóteses de trabalho (em andamento)**

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.

Exemplos: Linguagem de programação que depende de apis e hardware...

Fatores que podem impactar na tecnologia utilizada para desenvolver.

## **2 Requisitos específicos**

### **2.1 Interfaces externas (em andamento)**

#### **2.1.1 Visão geral**

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

#### **2.1.2 Requisitos para interfaces gráficas de usuário**

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

- Um esboço do layout gráfico sugerido para a interface;
- Uma descrição dos relacionamentos com outras interfaces;
- Um diagrama de estados/atividades, caso necessário para melhor entender-se o comportamento requerido da interface;
- Uma lista dos campos de dados da interface;
- Uma lista dos comandos da interface;
- BPM;

### **2.2 Requisitos funcionais (em andamento)**

#### **2.2.1 Diagramas 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.

#### **2.2.2 Fluxos dos casos de uso**

- Pré-condições para a realização do caso de uso;
- Fluxo principal do caso de uso (sucesso), descrito na forma de uma sequência de passos;
- Fluxos alternativos do caso de uso;

- Descrições mais formais, como diagramas de estado ou de atividade, se a complexidade do caso de uso exigir;

## **2.3 Requisitos não-funcionais (em andamento)**

### **2.3.1 Requisitos de desempenho**

Requisitos de desempenho devem ser especificados de forma quantitativa e mensurável.

Exemplo: Capacidade do servidor; largura de banda; capacidade de armazenamento.

### **2.3.2 Requisitos de dados persistentes**

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

INCLUIR AQUI O MODELO DE BANCO DE DADOS

### **2.3.3 Restrições ao desenho**

Restrições de projeto impostas por padrões externos, com influência da legislação..

### **2.3.4 Atributos de Qualidade**

Indica os atributos de qualidade, seguindo as características e sub-características recomendadas pela norma ISO-9126.

## **3 Análise de UCP (em andamento)**

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

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*

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