

## **1 – INTRODUÇÃO**

### **1.1 Objetivo**

*<Delinear o propósito da ERS e especificar o público-alvo.>*

### **1.2 Escopo**

*<*

- *Identificar o(s) produto(s) de software a ser produzido pelo nome.*
- *Explicar o **quê** o(s) produto(s) de software fará(ão) e, se necessário, o que não fará(ão).*
- *Descrever a aplicação do software a ser especificado, incluindo benefícios relevantes, objetivos e metas.*
- *Ser consistente com as especificações de mais alto nível (tal como a especificação de requisitos do sistema), se ele existir.>*

### **1.3 Definições, Siglas e Abreviações**

*<Fornecer as definições de todos os termos, acrônimos e abreviações necessárias para interpretar de modo apropriado a ERS. >*

### **1.4 Referências**

*<*

- *Fornecer uma lista completa de todos os documentos referenciados na ERS.*
- *Identificar cada documento pelo título, número do relatório (se aplicável), data e organização que publicou.*
- *Especificar a(s) origem(s) das referências, ou seja, onde e/ou com quem podem ser obtidas.>*

### **1.5 Informações Adicionais**

#### **1.5.1 Dados da Instituição**

Universidade do Oeste Paulista (Unoeste)

Faculdade de Informática de Presidente Prudente (FIPP)

José Bongiovani, 700 - Cidade Universitária - Bloco H - 1º andar

Fone: (18) 3229-1060

Email Coordenação Estágio: fippcoordestagios@fipp.unoeste.br

#### **1.5.2 Dados da Empresa**

*<Descrever a Empresa inserindo um breve histórico, seu ramo de atividade e setor de Informática da empresa, se existir. Apresentar um organograma da empresa destacando o setor de informática, se existir. >*

#### **1.5.3 Legislação de Software**

*<opcional, dependendo do acordo estipulado entre a empresa e o aluno, consultar coordenação.>*

### **1.6 Visão Geral**

*<Descrever o **restante** do conteúdo da ERS; explicar como a ERS está organizada.>*

## 2 – DESCRIÇÃO GERAL DO PRODUTO

*<Tem por objetivo descrever fatores gerais do produto e seus requisitos, fornecendo um contexto para esses requisitos os quais são definidos em detalhes no capítulo 3 da ERS.>*

### 2.1 Estudo de Viabilidade

<

*Apresentar a alternativa de informatização escolhida, considerando as viabilidades técnica, econômica e legal, destacando as vantagens/desvantagens do sistema com base na relação custo X benefício do sistema. >*

#### 2.1.1 Justificativa para a alternativa selecionada

*<Explicar o porquê da escolha da alternativa apresentada anteriormente em relação a(s) alternativa(s) apresentada(s) no Apêndice 1.>*

### 2.2 Funções do Produto

Tem por objetivo descrever as funções do sistema classificando-as em:

- **Funções Básicas:** referem-se às operações CRUD (*create, read, update, delete*) necessárias para a execução das funções fundamentais. Esse conjunto de operações deve ser denominado “Gerenciar”.
- **Funções Fundamentais:** referem-se às transações de negócio (movimentações);
- **Funções de Saída:** referem-se às funções que geram informações de saída relevantes para atender às necessidades do cliente (consultas/relatórios com cruzamento de informações). Nesse caso, devem ser descritos não só os itens de entrada (filtros), mas também os itens de saída (informação) pertinentes.

Para essa descrição utiliza-se da tabela abaixo, onde as colunas identificam:

- **Referência:** é importante que cada função tenha um identificador, a fim de facilitar a rastreabilidade desse requisito nesse documento. Sugere-se que seja utilizada o identificador RF (requisito funcional) seguido de um underline, uma letra indicando se é função básica, fundamental ou saída externa (B, F, S) e um número sequencial. Ex: RF\_B1. e RF\_B2. para funções básicas, RF\_F1., RF\_F2. para funções fundamentais e RF\_S1., RF\_S2. para funções de saída externa).
- **Função:** nome que identifica a função, com alguns detalhes de seu funcionamento.

- **Visibilidade:** deve ser utilizada a seguinte classificação: Evidente (Visível ao usuário) ou Oculta.
- **Atributo:** são as características que representam os requisitos não funcionais do sistema, que podem afetar todas as funções ou ser específico de uma função. Ex: tolerância a falhas, tempo de resposta, portabilidade, segurança, dentre outras.
  - **Detalhes e Restrições:** Descreve o atributo em detalhes.
- **Categoria:** classifica o atributo quanto a obrigatório ou desejável.

Referência	Função	Visibilidade	Atributo	Detalhes e Restrições	Categoria
RF_B1	[Exemplo] Gerenciar	[Exemplo] Evidente	[CONSULTAR MATERIAL LIVRO LARMAN CAP. 5]	[CONSULTAR MATERIAL LIVRO LARMAN CAP. 5]	[CONSULTAR MATERIAL LIVRO LARMAN CAP. 5]
RF_F1	[Exemplo] Efetuar	[Exemplo] Evidente			
RF_F2	[Exemplo] Atualizar	[Exemplo] Oculta			
...	...	...			

### 2.3 Características do Usuário

<

*Descrever o nível educacional dos usuários do sistema, bem como a sua experiência e o conhecimento sobre informática para que seja diagnosticada a necessidade de treinamento específico.>*

### 2.4 Limites, Suposições e Dependências

<

*Descrever itens que limitem as opções do desenvolvedor, tais como: Normas regulamentadoras, Limitações do hardware, considerações de segurança, requisitos de confiabilidade, Linguagem de programação.*

*Com relação às suposições e dependências, descrever qualquer fator que afeta os requisitos expressos na ERS. Por exemplo: A não aquisição do ponto eletrônico fará com que o sistema não tenha o seu total desempenho, pois a entrada de dados será feita manualmente, inserindo somente as exceções do ponto diário, ou seja, a falta dos funcionários.*

*Limites de Memória: especificar os limites mínimos de memória primária e secundária.>*

### 2.5 Requisitos Adiados

<

---

*Identificar os requisitos, que foram levantados, mas que por alguma razão serão adiados para versões futuras do sistema.>*

### **3 – REQUISITOS ESPECÍFICOS**

#### **3.1 Diagrama de Casos de Uso**

*< inserir aqui o diagrama de casos de uso do sistema. Caso haja necessidade, podem ser feitos pacotes de casos de uso.*

*>*

##### **3.1.1 Especificações de Casos de Uso**

Modelo disponível no RequisitePró – Ivar Jacobson

##### **3.1.1.1 Diagrama(s) de Atividades para Casos de Uso**

*< (opcional – é interessante fazê-lo quando se tem atividades executadas em paralelo no sistema).*

#### **3.2 Requisitos de Interface Externa**

*<Esses requisitos referem-se aos requisitos não funcionais. Segundo o padrão IEEE, se o produto a ser desenvolvido for parte de um sistema maior a ERS deverá identificar quais as interfaces existentes entre esse sistema e o produto a ser desenvolvido. >*

##### **3.2.1 Interfaces do Usuário**

*<Deve ser descrito como o usuário vai interagir com o sistema, sem mostrar graficamente as telas, pois existe uma seção específica para isso. Descrever como será o formato padrão das telas e relatórios, quais os procedimentos a serem adotados em caso de erros, para que servem e como serão apresentadas as mensagens do sistema para o usuário (por exemplo, não serão exibidas mensagens em caixas de diálogo, mas através de um label em um determinado local da tela. As mensagens exibidas em caixas de diálogos serão somente para erros do sistema que só podem ser resolvidas pelo desenvolvedor).*

*Descrever as operações: rotinas de inicialização (definir níveis de acesso, porém o detalhamento das funcionalidades a serem acessadas por cada nível é representado no item 4.4.1 - Mapa de Navegação/Diagrama Estrutural das Telas), processamento, backup e restauração do sistema.>*

##### **3.2.2 Interfaces de Software**

*<descrever os detalhes dos softwares necessários para o desenvolvimento e execução do software em questão (banco de dados, sistemas operacionais, software para capturar imagem, ou outros softwares aplicativos de mesma natureza). Isso inclui nome, mnemônico, especificação numérica, versão e fonte>*

##### **3.2.3 Interfaces do Sistema**

*<caso o software em questão tenha que ser integrado com algum já existente. Normalmente um software faz parte de um sistema (sistema administrativo) maior existente dentro de uma empresa. A ERS deve listar as interfaces do sistema para com o produto, identificando as funcionalidades do software que irão realizar essas interfaces.*

### 3.2.4 Interfaces de Hardware

<Se o produto interage com dispositivos de hardware, estes devem ser especificados (por exemplo, scanners, relógios de ponto, catracas eletrônicas ou outros dispositivos eletrônicos com o qual o produto irá comunicar-se).

A ERS deverá detalhar como será realizada a interface em questão. Se for um relógio de ponto, por exemplo, como será o layout do arquivo que será emitido pelo equipamento ao sistema. Por exemplo: O equipamento para a leitura do ponto dos funcionários será da marca XXXX, modelo YYYY, etc. O relógio ponto gera um arquivo com a extensão txt, o qual possui a seguinte estrutura em cada uma das linhas:

CÓDIGO\_FUNCIONÁRIO: 4 caracteres

DATA: 10 caracteres

HORÁRIO: 5 caracteres

Ex.:

CODFUN	DATA	HORÁRIO
43219	05/09/2003	7:30
43219	05/09/2003	11:30
	...	

>

### 3.2.5 Interfaces de Comunicação

<Especificar os tipos de comunicação utilizados para integração com outros periféricos e tecnologias (par-trançado, protocolos de comunicação, etc).

Por exemplo, o software que é responsável pelo gerenciamento da catraca precisa comunicar-se com as mesmas por meio de um par-trançado. Portanto, é necessário descrever como deverá ser implementada essa comunicação>

### 3.3. Outros Requisitos

<Descrever aqui restrições do projeto que podem ser impostas por conformidade a padrões, limitações de hardware, e por outros requisitos não funcionais: manutenibilidade, portabilidade, confiabilidade, requisitos éticos, requisitos de entrega, etc.>

<Exemplos:

Especificar os requisitos numéricos estáticos e dinâmicos sobre o software ou uma interação humana com o software. Os requisitos numéricos estáticos podem incluir: o número de terminais suportado, o número de usuários simultâneos suportado, a quantidade e o tipo de informação a ser manipulado. Os requisitos numéricos dinâmicos podem incluir, por exemplo, o número de transações e tarefas e a quantidade de dados a ser processado dentro de determinado período de tempo e condições de pico de sobrecarga. Todos esses requisitos devem ser declarados em termos mensuráveis. Por ex: “95 % das transações devem ser processadas em menos de 1 segundo”.>

Especificar situações em que o software deve ser adaptado antes da instalação, por exemplo: em um sistema que necessite a conexão com a internet. Se no momento da instalação o computador não estiver conectado, o sistema identifica e grava os dados em um arquivo temporário e, quando restabelecer a conexão, os dados são recuperados deste arquivo temporário e a instalação pode concluída. Outro exemplo refere-se às adaptações necessárias para a instalação do software em outra versão do S.O.>

### **3.4 Modelo Conceitual**

*<no modelo conceitual utiliza-se da representação do diagrama de classes da UML, entretanto, são colocados somente o nome do conceito, os seus atributos mais relevantes e as multiplicidades. Um conceito, não necessariamente, será uma classe de implementação.>*

## **4 – PROJETO DE SOFTWARE**

### **4.1 Diagramas de Interação**

*< diagrama de seqüência ou diagrama de colaboração (que na versão UML 2.0 recebeu o nome de diagrama de comunicação)>*

### **4.2 Diagrama de Classes**

*<diagrama de classes de implementação>*

### **4.3 Mapeamento OO-Relacional**

*<O diagrama de classes é elaborado pensando-se no desenvolvimento orientado a objetos, entretanto, os bancos de dados normalmente utilizados são relacionais. Sendo assim, é necessário apresentar como as classes serão mapeadas para tabelas usadas em um banco de dados relacional.>*

## **5 – DOCUMENTOS ESPECÍFICOS PARA SISTEMAS WEB**

### **5.1 Projeto de Interface**

*<Devem ser descritos os elementos que compõem a interface final do software, padrões de cores, esquematização do layout, disposição dos elementos na tela (menus, textos, formulários e objetos) e mensagens ao usuário.>*

### **5.2 Mapa de Navegação**

*<mostra os possíveis caminhos de navegação entre as páginas do sistema. O mapa de navegação pode ser desenhado por níveis de acesso. Pode ser usada a notação WAE.>*

## **APÊNDICE 1 – ESTUDO DE VIABILIDADE**

*< Deverão estar descritas as alternativas que não foram escolhidas na seção 2.2>*

## **APÊNDICE 2 – PROTÓTIPO e RELATÓRIO DE ANÁLISE**

*< O protótipo tem, nesse caso, a função de auxiliar no levantamento e análise dos requisitos do sistema. É preciso que o cliente esteja ciente da importância em discutir o protótipo, pois este dará subsídio para que seja construído um sistema que atenda às suas necessidades. Todas as análises feitas junto ao cliente devem ser documentadas em um relatório, inserido no Manual do Sistema logo após o protótipo.>*

## **APÊNDICE 3 – DICIONÁRIO DE DADOS**

*<o Dicionário de Dados (DD) refere-se a um repositório de informações sobre os elementos do modelo do software. Sugere-se que seja colocado nessa seção o DD relacionado aos elementos que constituem o Modelo Físico de Dados: nome da tabela, campos, tipos dos campos, opção de nulo, chaves primária e estrangeira e também deve incluir os valores padrão dos campos, triggers e procedures. A maioria das ferramentas de modelagem de dados disponibiliza a geração automática do DD >*



---

**APÊNDICE 4 – PROCEDIMENTOS BÁSICOS DE INSTALAÇÃO E  
CONFIGURAÇÃO INICIAL DO SOFTWARE**

**ANEXO 1 – REFERÊNCIAS**