<Projeto HATUMUS >

Especificação de Requisitos de Software

Para <HATUMUS>

Versão <1.0>

Histórico da Revisão

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| <14/03/23> | <1.0> | <Revisão das politicas do sistema> | <Lucas Kohagura> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Índice

1. Introdução 4

1.1 Objetivo 4

1.2 Escopo 4

1.3 Definições, Acrônimos e Abreviações 4

1.4 Referências 4

1.5 Visão Geral 4

2. Descrição Geral 4

3. Requisitos Específicos 5

3.1 Funcionalidade 5

3.1.1 <Requisito Funcional Um> 5

3.2 Utilidade 5

3.2.1 <Requisito de Utilidade Um> 6

3.3 Confiabilidade 6

3.3.1 <Requisito de Confiabilidade Um> 6

3.4 Desempenho 6

3.4.1 <Requisito de Desempenho Um> 6

3.5 Suportabilidade 7

3.5.1 <Requisito de Suportabilidade Um> 7

3.6 Restrições de Design 7

3.6.1 <Restrição de Design Um> 7

3.7 Documentação do Usuário On-line e Requisitos do Sistema de Ajuda 7

3.8 Componentes Comprados 7

3.9 Interfaces 7

3.9.1 Interfaces com o Usuário 7

3.9.2 Interfaces de Hardware 7

3.9.3 Interfaces de Software 7

3.9.4 Interfaces de Comunicações 8

3.10 Requisitos de Licença 8

3.11 Observações Legais, sobre Direitos Autorais e Outras Observações 8

3.12 Padrões Aplicáveis 8

4. Informações de Suporte 8

Especificação de Requisitos de Software

# Introdução

A plataforma da hatumus tem como objetivo fornecer cursos de desenvolvimento de software para iniciantes

## Escopo

Será disponibilizado vídeo-aulas, livros em pdf, com professores especializados a disposição

## Definições, Acrônimos e Abreviações

Hts

## Referências

….wwww

## Visão Geral

…

# Descrição Geral

• Desenlvolvimento do estudante

• Facilmente se encaixa na rotina

• Prepara o aluno para o mercado de trabalha.

• Tudo disponibilizado aos que se matricularem no curso

# Requisitos Específicos

Focado para a área de desenvolvimento

## Funcionalidade

Area de login que redireciona para a área do curso de cada estudante

### <Requisito Funcional Um>

[A descrição do requisito.]

## Utilidade

[Esta seção descreve todos os requisitos que afetam a usabilidade do sistema. Isso inclui, mas não se limita a:

• especificar a experiência necessária do usuário para operar o sistema de maneira eficaz e eficiente;

• definir tempos de tarefas mensuráveis para tarefas típicas;

• estabelecer requisitos de conformidade com padrões de usabilidade comuns.]

O sistema deve ser projetado de forma a permitir que usuários com diferentes níveis de experiência possam usá-lo de maneira eficaz e eficiente. Para usuários iniciantes, o tempo necessário para se tornarem proficientes em operações básicas do sistema não deve exceder 1 hora. Já para usuários experientes, o tempo de treinamento não deve exceder 30 minutos.

Os tempos de tarefa mensuráveis para tarefas típicas devem ser definidos como segue:

• Cadastrar uma nova pessoa: 1 minuto

• Atualizar informações de uma pessoa: 1 minuto

• Listar todas as pessoas cadastradas: 2 minutos

• Excluir uma pessoa cadastrada: 1 minuto

O sistema deve ser projetado em conformidade com os padrões de usabilidade comuns, como as Diretrizes de Design da Interface do Usuário da Microsoft e as Diretrizes de Usabilidade da IBM. Isso garantirá que a interface do usuário seja consistente, intuitiva e fácil de usar para os usuários.<Requisito de Utilidade Um>

Requisito 1

## Confiabilidade

[Os requisitos de confiabilidade do sistema devem ser especificados aqui. Algumas sugestões são:

* Disponibilidade—sistema 99% disponivel para acesso..
* MTBF (Mean Time Between Failures) — este é, geralmente, especificado em horas, mas pode também ser especificado em dias, meses ou anos.
* MTTR (Mean Time To Repair)—por quanto tempo o sistema tem permissão para ficar fora de operação após ter falhado?
* Exatidão—especifica a precisão (resolução) e a exatidão (por algum padrão conhecido) requeridas na saída do sistema.
* Taxa Máxima de Erros ou Defeitos—geralmente expressa em termos de erros por mil linhas de código (erros/KLOC) ou erros por ponto de função (erros/ponto de função).
* Taxa de Erros ou Defeitos—categorizada em termos de erros menores, significativos e críticos: o(s) requisito(s) deve(m) definir o que quer dizer um erro “crítico”; por exemplo, perda completa de dados ou uma inabilidade completa para utilizar determinadas partes da funcionalidade do sistema.]

### <Requisito de Confiabilidade Um>

[A descrição do requisito.]

## Desempenho

* Nesta seção devem ser especificados os requisitos de confiabilidade do sistema, garantindo sua estabilidade e funcionamento adequado. Algumas sugestões para inclusão são:
* • Disponibilidade: especificar a porcentagem de tempo em que o sistema deve estar disponível para acesso, por exemplo, 99% do tempo.
* • MTBF (Mean Time Between Failures): especificar o tempo médio entre falhas do sistema, que pode ser em horas, dias, meses ou anos.
* • MTTR (Mean Time To Repair): determinar por quanto tempo o sistema pode ficar fora de operação após ter falhado, antes de ser reparado.
* • Exatidão: definir a precisão e exatidão necessárias na saída do sistema, baseadas em padrões conhecidos.
* • Taxa Máxima de Erros ou Defeitos: estabelecer a quantidade máxima de erros ou defeitos aceitáveis, geralmente expressa em termos de erros por mil linhas de código ou erros por ponto de função.
* • Taxa de Erros ou Defeitos: classificar os erros em menores, significativos e críticos, definindo claramente o que é considerado um erro "crítico", como perda total de dados ou incapacidade de utilizar determinadas funcionalidades do sistema..

### <Requisito de Desempenho Um>

[A descrição do requisito aparece aqui.]

## Suportabilidade

### Nesta seção, serão apresentados os requisitos para melhorar a suportabilidade e a manutenibilidade do sistema em construção, tais como:

### • Padrões de codificação - definir um conjunto de padrões de codificação que devem ser seguidos durante todo o processo de desenvolvimento do sistema, visando uma maior consistência e facilidade de manutenção do código;

### • Convenções de nomenclatura - definir um conjunto de regras para a nomenclatura de variáveis, funções, classes e outros elementos do código fonte, visando facilitar a compreensão do código e a sua manutenção;

### • Bibliotecas de classe - especificar as bibliotecas de classes que serão utilizadas no desenvolvimento do sistema, garantindo a sua compatibilidade e estabilidade;

### • Acesso de manutenção - estabelecer um sistema de controle de acesso às áreas de manutenção do sistema, permitindo que apenas pessoas autorizadas possam realizar alterações;

## Restrições de Design

### Nesta seção devem ser descritas as limitações de design que devem ser consideradas durante o desenvolvimento do sistema. Essas limitações representam decisões de design que são obrigatórias e devem ser seguidas. Exemplos de limitações de design incluem a utilização de linguagens de programação específicas, requisitos de processo de desenvolvimento de software, uso prescrito de ferramentas de desenvolvimento, restrições de arquitetura e design, componentes adquiridos, bibliotecas de classes e assim por diante. É importante que todas as limitações de design sejam claramente definidas e documentadas para que o desenvolvimento possa ser executado de acordo com as especificações.

## Documentação do Usuário On-line e Requisitos do Sistema de Ajuda

Necessários que os usuários tenham um computador com acesso à internet

## Componentes Comprados

Servidores, Licenças de BD,

## Interfaces

Liberações HTML, STMP, FTP

### Interfaces com o Usuário

Interface simples intuitiva divida em área :Pendentes, Ao Vivo, Feitos

### Interfaces de Comunicações

## Requisitos de Licença

[Define os requisitos de reforço de licença ou outros requisitos de restrição de uso que devem ser exibidos pelo software.]

## Observações Legais, sobre Direitos Autorais e Outras Observações

[Esta seção descreve as isenções legais necessárias, garantias, observações sobre direitos autorais, observações sobre patente, wordmark, marca registrada ou problemas de conformidade de logotipo para o software.]

## Padrões Aplicáveis

[Esta seção descreve, por referência, o padrão aplicável e as seções específicas desses padrões que se aplicam ao sistema que está sendo descrito. Por exemplo, isso pode incluir padrões jurídicos, de qualidade e reguladores, padrões de mercado para utilidade, interoperabilidade, internacionalização, conformidade com o sistema operacional e assim por diante.]

# Informações de Suporte

[As informações de suporte facilitam a utilização da **SRS**. Elas incluem:

* Índice
* Índice
* Apêndices

Estes podem incluir storyboards de caso de uso ou protótipos de interface com o usuário. Quando os apêndices são incluídos, a **SRS** deve determinar explicitamente se os apêndices devem ou não ser considerados parte dos requisitos.]