<Nome do Projeto>

Especificação de Requisitos de Software

Para <Nome do Sistema>

Versão <1.0>

Autor 1

Autor 2

*[Observação: O template a seguir é fornecido para uso com o Rational Unified Process (RUP). O texto em azul exibido entre colchetes e em itálico (style=InfoBlue) foi incluído para orientar o autor e deve ser excluído antes da publicação do documento. Qualquer parágrafo inserido após esse estilo será definido automaticamente como normal (estilo=BodyText).]*

Histórico da Revisão

| **Data** | **Versão** | **Descrição** | **Autor** |
| --- | --- | --- | --- |
| <dd/mm/yyyy/1> | <x.x> | <detalhes> | <nome> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Tabela de Conteúdo

[**1. Introdução 4**](#_gjdgxs)

[1.1 Finalidade 4](#_30j0zll)

[1.2 Referências 4](#_2et92p0)

[1.3 Visão Geral 4](#_tyjcwt)

[**2. Descrição Geral 4**](#_3dy6vkm)

[**3. Cronograma 4**](#_jjkfxlwonei7)

[**4. Requisitos Específicos 5**](#_1t3h5sf)

[4.1 Requisitos Funcionais 5](#_4d34og8)

[4.1.1 <Requisito funcional um> 5](#_2s8eyo1)

[4.2 Requisitos Não Funcionais 5](#_17dp8vu)

[4.2.1 <Requisito de Desempenho Um> 5](#_3rdcrjn)

[4.3 Atributos do Sistema de Software 6](#_26in1rg)

[4.3.1 Restrições do Design 6](#_lnxbz9)

[4.3.2 Interfaces 6](#_35nkun2)

[4.3.3 Componentes Comprados 6](#_1ksv4uv)

[4.3.4 Observações Legais, de Direitos Autorais etc 6](#_44sinio)

[4.3.5 Padrões Aplicáveis 6](#_2jxsxqh)

[4.4 Outros Requisitos 7](#_z337ya)

[4.4.1 Requisitos de Usabilidade 7](#_3j2qqm3)

[4.4.2 Requisitos de Confiabilidade 7](#_1y810tw)

[4.4.3 Suportabilidade 7](#_4i7ojhp)

[4.4.4 Requisitos de Sistema de Ajuda e de Documentação de Usuário On-line 7](#_2xcytpi)

[4.4.5 Requisitos de Licenciamento 8](#_1ci93xb)

[**5. Riscos 8**](#_2bn6wsx)

[5.1.1 Riscos Identificados 8](#_qsh70q)

[5.1.2 Estratégias para Gerenciamento de Riscos 8](#_3as4poj)

[**6. Fluxogramas 8**](#_ro7go5qcozv)

[**7. Print de protótipos, Print de tela com descrição 8**](#_r93w9nqvl4vv)

[**8. Considerações finais (conclusão) 9**](#_n2bjtkf56sv1)

[**9. REFERÊNCIAS BIBLIOGRÁFICAS 9**](#_w72jwe6ccstx)

Especificação de Requisitos de Software

# Introdução

*[A introdução da Especificação de Requisitos de Software (SRS) deve fornecer uma visão geral de toda a SRS. Ela deve incluir a finalidade, o escopo, as definições, os acrônimos, as abreviações, as referências e a visão geral da SRS.]*

*[Observação: A Especificação de Requisitos de Software (SRS) captura todos os requisitos de software do sistema ou de uma parte do sistema..  A seguir, há um esquema de uma SRS típica para um projeto que utiliza somente requisitos em estilo de linguagem natural tradicional - sem modelagem de casos de uso. Essa SRS captura todos os requisitos em um único documento, com seções aplicáveis inseridas a partir das Especificações Suplementares (que não serão mais necessárias). Para ter acesso a um template de uma SRS que utilize a modelagem de casos de uso, que consiste em um pacote contendo Casos de Uso do modelo de casos de uso e Especificações Suplementares aplicáveis, assim como outras informações de suporte, consulte o arquivo rup\_SRS-uc.dot.]*

*[É possível organizar uma SRS de várias maneiras diferentes. Consulte o padrão [IEEE830-1998] para obter explicações mais detalhadas, assim como outras opções de organização da SRS.]*

## Finalidade

*[Especifique a finalidade desta SRS. A SRS deverá descrever totalmente o comportamento externo do aplicativo ou do subsistema identificado. Ela também deverá descrever requisitos não funcionais, restrições de design e outros fatores necessários para fornecer uma visão completa e abrangente dos requisitos do software.]*

## Referências

*[Esta subseção deve fornecer uma lista completa de todos os documentos mencionados em qualquer outra parte da SRS. Cada documento deverá ser identificado por título, número do relatório (se aplicável), data e organização de publicação. Especifique as fontes a partir das quais as referências podem ser obtidas. Essas informações podem ser fornecidas por um anexo ou outro documento.]*

## Visão Geral

*[Esta subseção descreve o que o restante da SRS contém e explica como o documento está organizado.]*

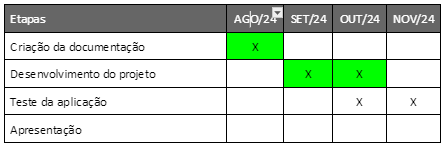
# Descrição Geral

*Esta seção da SRS deve descrever os fatores gerais que afetam o produto e seus requisitos. Ela não deve especificar requisitos específicos. Em vez disso, deve fornecer uma base para esses requisitos, que serão definidos detalhadamente na Seção 3, e facilitar sua compreensão. Inclua itens como:*

* *perspectiva do produto*
* *funções do produto*
* *características do usuário*
* *restrições*
* *suposições e dependências*
* *subconjuntos de requisitos*

# Cronograma

O Cronograma de um projeto de desenvolvimento é um planejamento detalhado das atividades e etapas que serão realizadas ao longo do projeto, distribuídas ao longo do tempo. Ele define prazos para a conclusão de cada fase, como levantamento de requisitos, design, desenvolvimento, testes e implementação, garantindo que o projeto siga uma sequência organizada e que os marcos sejam cumpridos. Um cronograma bem estruturado é essencial para o gerenciamento eficiente de tempo e recursos, além de facilitar o acompanhamento do progresso.



# Requisitos Específicos

*[Esta seção da SRS deve conter todos os requisitos de software em um nível de detalhamento suficiente para possibilitar que os designers projetem um sistema que satisfaça esses requisitos e que os testadores verifiquem se o sistema satisfaz esses requisitos.   Quando for utilizada a modelagem de casos de uso, esses requisitos serão capturados nos Casos de Uso e nas especificações suplementares aplicáveis. Se a modelagem de casos de uso não for utilizada, o esquema das especificações suplementares poderá ser inserido diretamente nesta seção, conforme mostrado a seguir.]*

## Requisitos Funcionais

*[Esta seção descreve os requisitos funcionais do sistema que são expressos no estilo de linguagem natural. Para muitos aplicativos, este poderá ser o volume do Pacote da SRS. É necessário refletir muito para organizar esta seção. Normalmente, esta seção é organizada por recurso, mas outros métodos de organização também poderão ser adequados como, por exemplo, a organização por usuário ou a organização por subsistema. Entre os requisitos funcionais poderão estar incluídos conjuntos de recursos, capacidades e segurança.*

*Quando as ferramentas de desenvolvimento de aplicativos, como ferramentas de requisitos, ferramentas de modelagem, entre outras, forem utilizadas para capturar a funcionalidade, esta seção fará referência à disponibilidade desses dados, indicando o local e o nome da ferramenta usada para capturar os dados.]*

### <Requisito funcional um>

*[A descrição do requisito deve ser feita aqui.]*

## Requisitos Não Funcionais

*[As características de desempenho do sistema devem ser descritas nesta seção. Inclua tempos de resposta específicos. Quando aplicável, faça referência, por nome, aos Casos de Uso relacionados.*

* *tempo de resposta de uma transação (médio, máximo)*
* *taxa de transferência como, por exemplo, transações por segundo*
* *capacidade como, por exemplo, o número de clientes ou de transações que o sistema pode acomodar*
* *modos de degradação (o modo aceitável de operação quando o sistema tiver sido degradado de alguma maneira)*
* *a utilização de recursos como, por exemplo, memória, disco, comunicações, etc.]*

### <Requisito de Desempenho Um>

*[A descrição do requisite deve ser feita aqui]*

## Atributos do Sistema de Software

### Restrições do Design

*[Esta seção deve indicar todas as restrições de design referentes ao sistema que está sendo criado. As restrições de design representam decisões de design que foram impostas e devem ser respeitadas. Entre os exemplos desse tipo de restrição estão linguagens de software, requisitos de processo de software, uso prescrito de ferramentas de desenvolvimento, restrições de design e de arquitetura, componentes comprados, bibliotecas de classes, etc]*

#### <Restrição de Design Um>

*[A descrição do requisite deve ser feita aqui]*

### Interfaces

*[Esta seção define as interfaces que devem ser suportadas pelo aplicativo. Ele deve conter especificidades, protocolos, porta e endereços lógicos adequados, entre outros, para que o software possa ser desenvolvido e verificado em relação aos requisitos de interface.]*

#### Interface de Usuário

*[Descreva as interfaces de usuário que deverão ser implementadas pelo software.]*

#### Interfaces de Hardware

*[Esta seção define todas as interfaces de hardware que devem ser suportadas pelo software, incluindo a estrutura lógica, os endereços físicos, o comportamento esperado, etc. ]*

#### Interfaces de Software

*[Esta seção descreve as interfaces de software com outros componentes do sistema de software. Poderão ser componentes comprados, componentes reutilizados de outro aplicativo ou componentes que estejam sendo desenvolvidos para subsistemas fora do escopo desta SRS, mas com os quais esse aplicativo de software deve interagir.]*

#### Interfaces de Comunicação

*[Descreva todas as interfaces de comunicação com outros sistemas ou dispositivos como, por exemplo, redes locais, dispositivos seriais remotos, etc.]*

### Componentes Comprados

*[Esta seção descreve todos os documentos comprados para serem usados com o sistema, quaisquer restrições de utilização ou de licenciamento aplicáveis, e quaisquer padrões associados de compatibilidade e de interoperabilidade ou de interface.]*

### Observações Legais, de Direitos Autorais etc

*[Esta seção descreve todos os avisos legais necessários, garantias, observações sobre direitos autorais, observações sobre patentes, logomarcas, marcas comerciais ou problemas de conformidade com logotipos referentes ao software.]*

### Padrões Aplicáveis

*[Esta seção descreve, por meio de referências, todos os padrões aplicáveis e as seções específicas desses padrões que se aplicam ao sistema que está sendo descrito. Entre esses padrões estão incluídos, por exemplo, padrões reguladores, de qualidade e legais, padrões de indústria referentes à usabilidade, interoperabilidade, internacionalização, compatibilidade com o sistema operacional, etc.]*

## Outros Requisitos

### Requisitos de Usabilidade

*[Esta seção deve incluir todos os requisitos que afetam a usabilidade. Por exemplo,*

* *especifique o tempo de treinamento necessário para que usuários normais e usuários com conhecimentos avançados se tornem produtivos em operações específicas*
* *especifique períodos de tempo mensuráveis para tarefas típicas ou baseie os requisitos de usabilidade do novo sistema em outros sistemas que os usuários conheçam e gostem*
* *especifique requisitos de forma que estejam em conformidade com padrões comuns de usabilidade como, por exemplo, os padrões CUA da IBM ou os padrões GUI da Microsoft]*

#### <Requisito de Usabilidade Um>

*[A descrição do requisito deve ser feita aqui]*

### Requisitos de Confiabilidade

*[Os requisitos de confiabilidade do sistema devem ser especificados aqui. A seguir, há algumas sugestões:*

* *Disponibilidade - especifique a porcentagem de tempo disponível ( xx.xx%), as horas de uso, o acesso à manutenção, as operações de modo degradado, etc.*
* *Tempo Médio entre Falhas (MTBF) - normalmente especificado em horas, mas também poderá ser especificado em termos de dias, meses ou anos.*
* *Tempo Médio para Reparo (MTTR) - quanto tempo o sistema poderá ficar sem funcionar após uma falha?*
* *Exatidão - especifique a precisão (resolução) e a exatidão (através de algum padrão conhecido) necessárias na saída do sistema.*
* *Taxa Máxima de Erros ou Defeitos - geralmente expressa em termos de erros por milhares de linhas de código (erros/KLOC) ou de erros por ponto de função (erros/ponto de função).*
* *Taxa de Erros ou Defeitos - categorizados em termos de erros pouco importantes, importantes e críticos: o(s) requisito(s) devem definir o que se entende por um erro "crítico"; por exemplo, a perda total de dados ou uma total incapacidade de usar determinadas partes da funcionalidade do sistema.]*

#### <Requisito de Confiabilidade Um>

*[A descrição do requisite deve ser feita aqui.]*

### Suportabilidade

*[Esta seção indica todos os requisitos que aprimorarão a suportabilidade ou manutenibilidade do sistema que está sendo criado, incluindo padrões de codificação, convenções de nomeação, bibliotecas de classes, acesso à manutenção e utilitários de manutenção.]*

#### <Requisito de Suportabilidade Um>

*[A descrição do requisite deve ser feita aqui.]*

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

*[Descreva os requisitos, se houver, de documentação de usuário on-line, sistemas de ajuda, observações sobre ajuda, etc.]*

### Requisitos de Licenciamento

*[Esta seção define todos os requisitos de imposição de licenciamento ou outros requisitos de restrição de utilização que devem ser exibidos pelo software.]*

# Riscos

### Riscos Identificados

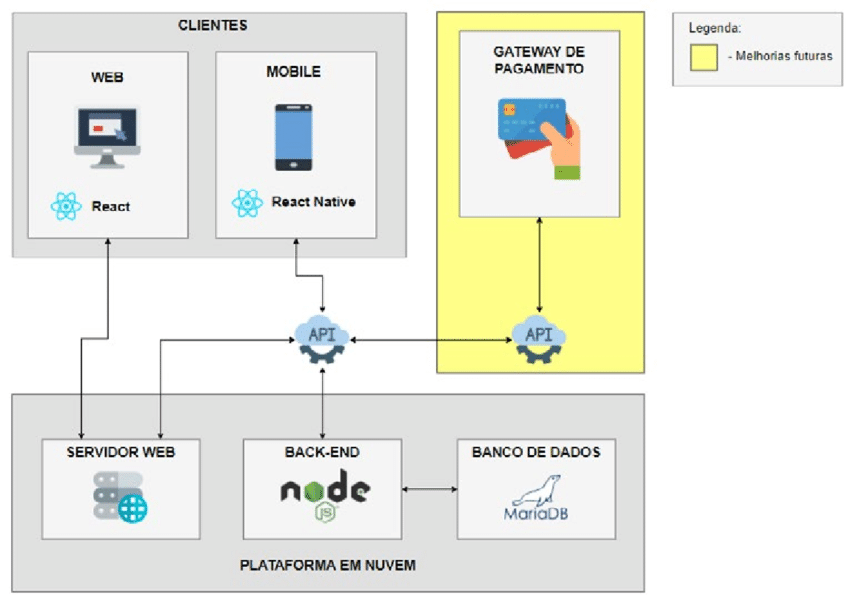
*[Esta seção deve incluir todos os riscos que foram identificados na etapa de levantamento de riscos.]*

### Estratégias para Gerenciamento de Riscos

*[Esta seção deve incluir todas as estratégias definidas para o gerenciamento dos riscos.]*

# Fluxogramas

* 1. Listar fluxogramas, UMLs, casos de uso utilizados e criados durante o desenvolvimento do projeto.
  2. Arquitetura do Sistema



# Print de protótipos, Print de tela com descrição

O que cada coisa faz ou Análise de resultados

Descreve detalhadamente cada tela e o que ela executa dentro do sistema.

Descrição dos componentes principais, como front-end, back-end, banco de dados, API, etc.

# Considerações finais (conclusão)

As Considerações Finais de um projeto de desenvolvimento são a última parte do relatório, onde se faz uma análise dos resultados alcançados e uma reflexão sobre o processo como um todo. Nessa seção, o autor avalia se os objetivos gerais e específicos foram atingidos, destacando os principais desafios e as soluções encontradas. Além disso, podem ser apontadas possíveis melhorias, limitações do sistema desenvolvido e sugestões para trabalhos futuros ou expansões do projeto.

Exemplo:

"Concluímos que o projeto atingiu seus objetivos ao desenvolver um sistema de gerenciamento de clientes eficiente e funcional, melhorando a organização das informações e a comunicação com os usuários. Durante o processo, enfrentamos desafios na integração de tecnologias, mas conseguimos superá-los com soluções adaptativas. Futuramente, recomendamos a implementação de novas funcionalidades, como um módulo de análise preditiva, para potencializar o uso dos dados. O sistema já se mostra como uma solução eficaz, mas sua evolução pode continuar com base nas demandas emergentes do mercado."

# REFERÊNCIAS BIBLIOGRÁFICAS

CALLADO, Antônio André Cunha. Agronegócio. 2. ed. São Paulo: Atlas, 2009.

CUNHA, HERBET DE SOUZA. 2007. Uso de estratégias orientadas a metas para modelagem de requisitos de segurança. Pontifica Universidade Católica do Rio de Janeiro.

Documentos eletrônicos:

BARBI, Fernando C. Conceitos Importantes. S.D. http://www.gestaodeprojeto.info/introducao

BEDANI, Janaína. Engenharia de Software 2 - Técnicas para levantamento de Requisitos. 2009. http://www.devmedia.com.br/engenharia-de-software-2-tecnicas-para-levantamento-de-requisitos/9151