

Tabela de conteúdos

Histórico de Alterações	1.1
Introdução	1.2
Descrição Geral do Sistema	1.3
Requisitos do Sistema	1.4
Artefatos de Análise	1.5
Protótipos	1.6
Referências	1.7

- [1. Histórico de Alterações](#)

1. Histórico de Alterações

Data	Versão	Descrição	Autor(es)
dd/mm/aaaa	1.0	Primeira versão da documentação com descrição e diagramas	
.			
.			
.			
.			.

Copyright © 2022 Herbert Barreto all right reserved , powered by
GitbookArquivo revisado pela última vez em 04/02/2022

- **1. Introdução**
 - **1.1. Visão Geral do Documento**
 - **1.2. Convenções, termos e abreviações**

1. Introdução

O objetivo do documento de projeto é oferecer uma listagem detalhada, por meio de textos e diagramas, do **[Editar variável name no arquivo 'book.json' com o nome do projeto]**, a fim de esclarecer todos os pontos do projeto para os stakeholders — um indivíduo, grupo ou organização que possa afetar, ser afetado, ou sentir-se afetado por uma decisão, atividade, ou resultado de um projeto, programa ou portfólio.

1.1. Visão Geral do Documento

Este documento apresenta uma visão abrangente dos requisitos do **[Editar variável name no arquivo 'book.json' com o nome do projeto]** e utiliza diferentes visões para ilustrar os diversos aspectos do sistema. Sua intenção é capturar e transmitir as decisões significativas que foram tomadas em relação ao sistema.

Nas seções seguintes estão dispostos: Descrição Geral do Sistema; Requisitos do Sistema; Artefatos de análise, visualizados pela Arquitetura 4+1 e seus respectivos diagramas; os protótipos e, por fim, as referências. Assim, funcionando como base para apresentação do **[Editar variável name no arquivo 'book.json' com o nome do projeto]** para os stakeholders.

1.2. Convenções, termos e abreviações

A correta interpretação deste documento exige o conhecimento de algumas convenções e termos específicos, assim como, abreviações e glossários, que são descritos a seguir.

Siglas/Termos	Significado
Loren	Sequência de palavras, geralmente latinas ou alatinadas, utilizada para preencher o espaço destinado a texto numa publicação, página de Internet, interface gráfico, etc.
Ipsum	e testar as opções de formatação e edição e o arranjo dos elementos gráficos antes da inserção do conteúdo.

Copyright © 2022 Herbert Barreto all right reserved , powered by
GitbookArquivo revisado pela última vez em 04/02/2022

- **1. Descrição Geral do Sistema**
 - **1.1.** Descrição do problema
 - **1.2.** Objetivo do projeto
 - **1.3.** Método de trabalho
 - **1.4.** Principais stakeholders e usuários
 - **1.5.** Regras de negócio
 - **1.6.** Estórias dos usuários
 - **1.7.** Documento de Visão
 - **1.8.** Prospecção tecnológica e busca de anterioridade

1. Descrição Geral do Sistema

Esta seção tem como objetivo descrever de forma geral o **[Editar variável name no arquivo 'book.json' com o nome do projeto]**, o escopo e as suas principais funções.

1.1. Descrição do problema

Descrever em um texto:

- Quem é afetado pelo sistema?
- Qual é o impacto do sistema?
- Qual seria uma boa solução para o problema?

1.2. Objetivo do projeto

O projeto **[Editar variável name no arquivo 'book.json' com o nome do projeto]** consiste no *[Complete aqui com seu texto]*.

1.3. Método de trabalho

Para chegar ao objetivo serão utilizadas algumas metodologias que melhoram a rotina de trabalho e auxiliam no ganho de produtividade, entre elas *[Complete aqui com as utilizadas no projeto, sempre com uma brevíssima descrição do para quê foi usada]*.

1.4. Principais stakeholders e usuários

Papel	Responsabilidades	Stakeholders
Contratante	Comunicar os requisitos, bem como mudanças dos mesmos; Validação e acompanhamento do andamento do projeto	SEMAST
Parceiros	Desenvolvimento da aplicação.	SEPLOG; DITIN
Usuários	Alimentar o sistema, testar os protótipos e contribuir com sugestões de melhorias e/ou construção do ponto de vista do usuário.	Gerente; Gestor A; Gestor B; Técnico.

1.5. Regras de negócio

Neste item devem ser descritas as regras de negócio relevantes para o sistema, como por exemplo: restrições de negócio, restrições de desempenho, tolerância à falha, volume de informações a serem armazenadas, estimativa de crescimento de volume, ferramentas de apoio, entre outras.

1.6. Estórias dos usuários

Eu, enquanto (Quem)	Quero (o que quero)	Para (porque)
Loren	Sequência de palavras, geralmente latinas ou alatinadas, utilizada para preencher o espaço destinado a texto numa publicação, página de Internet, interface gráfico, etc.	Latim
Ipsum	e testar as opções de formatação e edição e o arranjo dos elementos gráficos antes da inserção do conteúdo.	Inglês
Loren	Sequência de palavras, geralmente latinas ou alatinadas, utilizada para preencher o espaço destinado a texto numa publicação, página de Internet, interface gráfico, etc.	Latim
Ipsum	e testar as opções de formatação e edição e o arranjo dos elementos gráficos antes da inserção do conteúdo.	Inglês

1.7. Documento de Visão

Nesta seção se apresenta diagramas que resumem as necessidades do dono do produto e as características esperadas do produto a ser desenvolvido. Assim como também pode ser feita uma análise interna e externa acerca do projeto.

1.8. Prospecção tecnológica e busca de anterioridade

Neste item devem ser identificadas e descritas soluções de software que resolvem problemas semelhantes à solução proposta neste projeto. Importante identificar os aspectos da solução e os diferenciais.

Copyright © 2022 Herbert Barreto all right reserved , powered by
GitbookArquivo revisado pela última vez em 04/02/2022

- **1. Requisitos do Sistema**
 - **1.1. Requisitos Funcionais (RF)**
 - **1.2. Inputs dos Usuários**
 - **1.2.1. RF001 Login no sistema**
 - **1.3. Requisitos Não-Funcionais (RNF)**
 - **1.3.1. RF001 Login no sistema**

1. Requisitos do Sistema

1.1. Requisitos Funcionais (RF)

1.2. Inputs dos Usuários

Explicar prioridade.....

1.2.1. RF001 Login no sistema

Explicar a descrição aqui

- Prioridade: Essencial

1.3. Requisitos Não-Funcionais (RNF)

Explicar prioridade.....

1.3.1. RF001 Login no sistema

Explicar a descrição aqui

- Prioridade: Essencial

Copyright © 2022 Herbert Barreto all right reserved , powered by
GitbookArquivo revisado pela última vez em 04/02/2022

- **1. Artefatos de Análise**
 - **1.1. Modelo de visão da arquitetura de software escolhida**
 - **1.1.1. Visão Lógica**
 - **1.1.2. Visão de desenvolvimento**
 - **1.1.3. Visão de processo**
 - **1.1.4. Visão física**
 - **1.1.5. Visão de caso de uso**

1. Artefatos de Análise

Nesta seção será analisado o conjunto de atividades proveniente das tarefas envolvidas no processo de desenvolvimento do projeto. *[Cite aqui o que foi utilizado no projeto]*.

1.1. Modelo de visão da arquitetura de software escolhida

A arquitetura visão-modelo 4+1 foi desenvolvida por Philippe Cruchten com o objetivo de descrever o funcionamento de sistemas de software e é baseado no uso de múltiplas visões concorrentes. Ela foi utilizada neste projeto pois suas visões são usadas para mostrar o sistema sob várias perspectivas, como usuário final, desenvolvedores e gerentes de projetos.

As 4 visões de modelo são: visão lógica, visão de desenvolvimento, visão de processo e visão física. A visão de caso de uso é usada para ilustrar a arquitetura e representa a visão +1.

1.1.1. Visão Lógica

Ilustra a funcionalidade que o sistema disponibiliza para o usuário final. Para esta finalidade foi utilizado os Diagramas de classes e o Diagrama de Entidade Relacionamento Estendido (EER).

Diagrama de Classes de Domínio

É uma representação visual de classes conceituais ou objetos do mundo real em um domínio de interesse. Usando a notação UML, o modelo de domínio é ilustrado com um conjunto de diagramas de classes nos quais não são definidas operações.

Diagrama de Classes de Projeto

É utilizado para representar os objetos que o sistema irá manipular, suas operações ou serviços (métodos). Os métodos são responsáveis por manipular os atributos, efetuar ações que alterem o estado dos atributos, manipular as associações e relacionamentos entre as classes.

Diagrama Entidade Relacionamento Estendido (EER)

Modelos entidade-relacionamento estendidos são diagramas avançados de banco de dados bastante semelhantes aos diagramas ER regulares. Diagramas ER estendidos são modelos de alto nível que representam os requisitos e complexidades de bancos de dados complexos.

1.1.2. Visão de desenvolvimento

Ilustra o sistema do ponto de vista do programador e se preocupa com o gerenciamento de projeto. Para isto foi utilizado o Diagrama de Componentes e Diagrama de Pacotes.

Diagrama de Componentes

Este diagrama permite a modelagem física de um sistema, através da visão dos seus componentes e relacionamentos entre os mesmos. Similar a um pacote, mas com um enfoque do empacotamento físico de código. Basicamente, um diagrama de componentes é composto por: componente, interface e relacionamentos.

Diagrama de Pacotes

Diagramas de pacotes são diagrama estruturais comumente usados para simplificar os diagramas de classe complexos e organizar as classes em pacotes. Um pacote é uma coleção de elementos relacionados, incluindo diagramas, documentos, classes e pacotes de eventos.

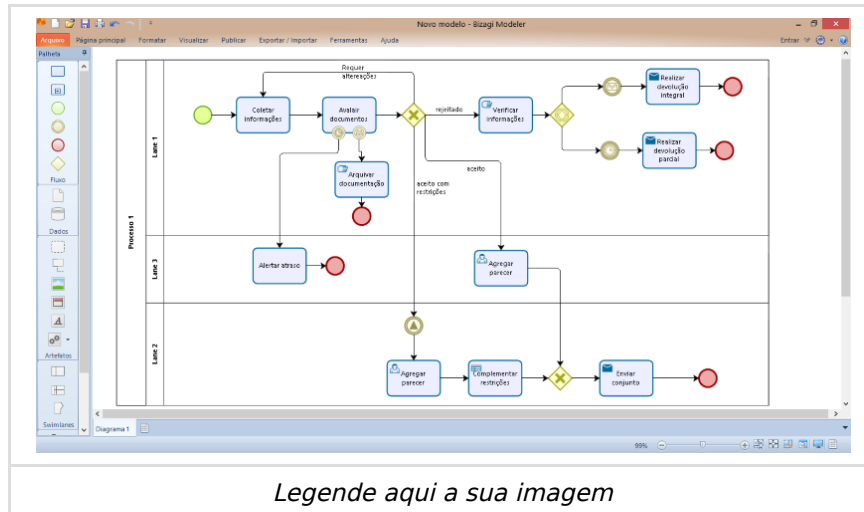
1.1.3. Visão de processo

Permite visualizar as partes dinâmicas do sistema, explicar os processos e como eles se comunicam, focando no comportamento do sistema. Para este processo normalmente é utilizado o Diagrama de Atividades, mas ele pode (e foi nesse projeto) ser substituído pelo Diagrama Business Process Model and Notation (BPMN) pois este também serve para modelar processos.

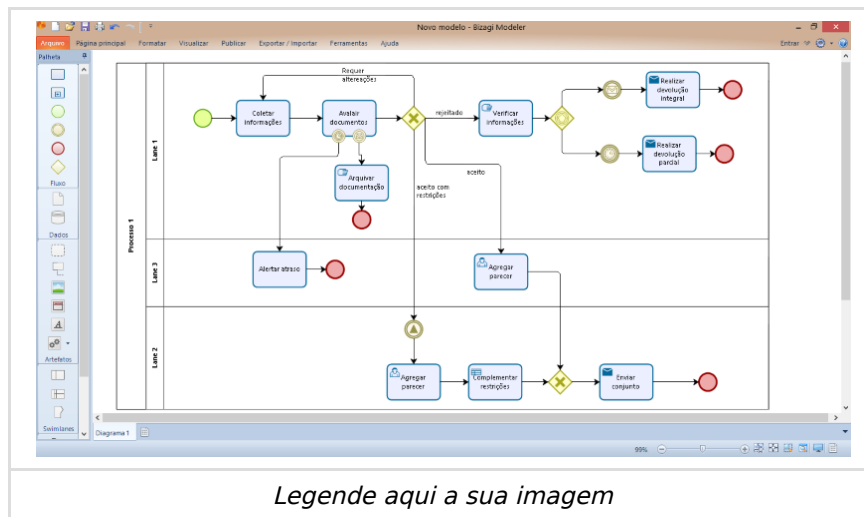
Diagrama Business Process Model and Notation (BPMN)

Um diagrama BPMN é uma representação gráfica feita a partir de ícones que simbolizam o fluxo de processo. Ou seja, a partir dessa notação é possível fazer o mapeamento dos processos. Portanto, cada ícone representa uma etapa do processo de produção.

- Como é:



- Como será:



1.1.4. Visão física

Mostra como os executáveis e componentes são mapeados para plataformas e nós físicos, ou seja, a distribuição física de elementos do sistema.

Diagrama de Implantação

Este diagrama foca a questão da organização da arquitetura física sobre a qual o software irá ser implantado e executado em termos de hardware, ou seja, as máquinas (computadores pessoais, servidores etc.) que suportam o sistema, além de definir como estas máquinas serão conectadas e por meio de quais protocolos se comunicarão e transmitirão as informações.

1.1.5. Visão de caso de uso

Descreve a arquitetura do sistema através do uso de Diagramas de casos de uso pois cada diagrama descreve sequências de interações entre os objetos e processos.

Diagramas de caso de uso

Um modelo de caso de uso é um modelo que descreve como diferentes tipos de usuários interagem com o sistema para resolver um problema. Como tal, ele descreve as metas dos usuários, as interações entre os usuários e o sistema, bem como o comportamento necessário do sistema para satisfazer estas metas.

Descrição de casos de uso

UC001 - Logar no sistema

Objetivo	Requisitos	Atores	Gatilho	Pré-condição
Acessar o sistema.	RF001.	Técnico, Gestor B, Gestor A ou Gerente.	Acessar o sítio eletrônico do sistema.	O ator deve estar cadastrado no sistema.

- Fluxo principal:

Sistema	Atores
-	1. Insere os dados de acesso.
-	2. Confirma os dados no sistema.
3. Verifica os dados.	-
4. Concede acesso ao sistema.	-

- Fluxo Alternativo 1:

Sistema	Atores
-	2. Envia dados inconsistentes.
3. Não concede acesso ao sistema.	-
4. Exibe mensagem de erro.	-

- Fluxo Alternativo 2:

Histórico de Alterações

Sistema	Atores
4. Identifica que o usuário tentou fazer o login na página errada.	-
5. Redireciona para a página de login correta.	-

Copyright © 2022 Herbert Barreto all right reserved , powered by
GitbookArquivo revisado pela última vez em 04/02/2022

- [1. Protótipos](#)

1. Protótipos

Nesta seção são apresentados os protótipos validados pelo cliente (product owner).



Copyright © 2022 Herbert Barreto all right reserved , powered by
GitbookArquivo revisado pela última vez em 04/02/2022

- **1. Referências bibliográficas**
 - **1.1.** Arquitetura de software escolhida
 - **1.2.** Artefatos de Análise
 - **1.3.** Diagrama de Classes de Domínio
 - **1.4.** Diagrama de Classes de Projeto
 - **1.5.** Diagrama Entidade Relacionamento Estendido (EER)
 - **1.6.** Diagrama de Componentes
 - **1.7.** Diagrama de Pacotes
 - **1.8.** Diagrama Business Process Model and Notation (BPMN)
 - **1.9.** Diagrama de Implantação

1. Referências bibliográficas

Esta seção apresenta o conjunto das fontes efetivamente utilizadas na construção desta ferramenta. Cada referência dessa bibliografia está organizada pelo tópico que a mesma ajudou a construir — para facilitar uma busca mais aprofundada do conteúdo.

1.1. Arquitetura de software escolhida

- Arquitetura Visão-Modelo 4+1. **Basef**. Disponível em:
<http://www.basef.com.br/old/uml/204-arquitetura-visao-modelo-41>.
Acesso em: 27 de jan. 2022.

1.2. Artefatos de Análise

- Arquitetura de desenho de software. **PUC Goiás**. Disponível em:
<http://professor.pucgoias.edu.br/SiteDocente/admin/arquivosUpload/17676/material/Parte%201%20-%20Artefatos%20de%20Software.pdf>. Acesso em: 30 de jan. 2022.

1.3. Diagrama de Classes de Domínio

- Modelo de Domínio: Visualizando Conceitos. **Instituto de Computação, Universidade Estadual de Campinas**. Disponível em:
<https://www.ic.unicamp.br/~ariadne/mc436/1s2017/Lar10ModDom.pdf>. Acesso em: 27 de jan. 2022.

1.4. Diagrama de Classes de Projeto

- Diagrama de Classe de Projeto. **Treinamento WAEI/MSE**. Disponível em: <https://treinamentowaei.wordpress.com/diagrama-de-classe-de-projeto/>. Acesso em: 27 de jan. 2022.

1.5. Diagrama Entidade Relacionamento Estendido (EER)

- O que é um diagrama entidade relacionamento estendido?. **Lucidchart**. Disponível em: <https://www.lucidchart.com/pages/pt/diagrama-entidade-relacionamento-estendido>. Acesso em: 27 de jan. 2022.

1.6. Diagrama de Componentes

- Diagrama de Componentes. **Departamento de Ciência da Computação - UFMG**. Disponível em: https://homepages.dcc.ufmg.br/~amendes/GlossarioUML/glossario/conteudo/componentes/diagrama_de_componentes.htm. Acesso em: 27 de jan. 2022.

1.7. Diagrama de Pacotes

- Diagrama de Pacotes: Definição, Componentes e Exemplos. **GitMind**. Disponível em: <https://gitmind.com/pt/diagrama-de-pacotes.html>. Acesso em: 27 de jan. 2022.

1.8. Diagrama Business Process Model and Notation (BPMN)

- Como funciona e quais as vantagens da notação BPMN?. **TOTVS**. Disponível em: <https://www.totvs.com/blog/gestao-industrial/bpmn/>. Acesso em: 27 de jan. 2022.

1.9. Diagrama de Implantação

- Artigo SQL Magazine 68 - Utilizando UML: Diagramas de Implantação, Comunicação e Tempo. Artigo SQL Magazine 68 - Utilizando UML: Diagramas de Implantação, Comunicação e Tempo. **DEVMEDIA**. Disponível em: <https://www.devmedia.com.br/artigo-sql-magazine-68-utilizando-uml-diagramas-de-implantacao-comunicacao-e-tempoartigo-sql-magazine-68-utilizando-uml-diagramas-de-implantacao-comunicacao-e-tempo/16353>. Acesso em: 27 de jan. 2022.

Copyright © 2022 Herbert Barreto all right reserved , powered by
GitbookArquivo revisado pela última vez em 04/02/2022