**CENTRO UNIVERSITÁRIO BELAS ARTES DE SÃO PAULO**

**GRADUAÇÃO: ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - EAD**

**FELIPE SCHAITEL - 24112424**

**DOCUMENTAÇÃO DE SOFTWARE - ATIVIDADE 01**

**FLORIANÓPOLIS, SC**

**2025**

**1 INTRODUÇÃO**

Este trabalho contém o relatório da pesquisa e análise de seis documentos sendo, dois documentos de visão, três documentos de arquitetura de *software* e um de regras de negócios. Para a disciplina Documentação de *Software* ministrada pela professora Elaine B. Figueiredo. Essa atividade está dividida em três partes. Todos os *templates* e modelos dos documentos escolhidos estão em anexo em cada capítulo. Essa forma de anexar os documentos foi escolhida com o intuito de facilitar a leitura por conta de serem documentações extensas.

**2 TEMPLATES DE DOCUMENTO DE VISÃO DO *RUP* (UML)**

**2.1 O que é e documento de visão de *Software*? Como é utilizado? Acesso a dois *templates* de documento de visão**

O Documento de Visão de Software é o registro que define o escopo, propósito, descreve as características, utilidades, define os objetivos de negócios da aplicação. Além de listar os recursos, restrições, requisitos do produto. Contextualiza o problema que a solução tecnológica visa resolver/automatizar. Também lista as ferramentas. Funciona como um guia para a equipe e para o cliente ter uma visão estruturada do projeto. Outras características desse documento são:

* definições
* acrônimos, abreviações
* referências
* visão geral de todo o documento

Esse tipo de documentação é considerado processo orientado a documentos e é uma metodologia tradicional de desenvolvimento de Software. As metodologias ágeis também são utilizadas para desenvolvimento, mas é recomendado para equipes de 20 até 40 pessoas. De acordo com Wislayne Aires Moreira no texto Processos Tradicionais de Desenvolvimento de Software capítulo 01:

Estes processos apresentam as seguintes características: adotam uma abordagem que é voltada à documentação detalhada da execução das atividades, possuem fases seqüenciais com um conjunto de artefatos para cada fase e dão ênfase à definição rigorosa de papéis para a execução do trabalho.

(MOREIRA, p. 01)

Selecionei dois templates de documento de visão a partir do material didático postado no Moodle e do conteúdo visto durante as aulas.

**ANEXOS:**

**Template 01**

[**Documento-de-visão-de-sistema.docx**](https://docs.google.com/document/u/0/d/1D0sNsj4V_kfLSsdNZ3gysUkZHLewaS_r/edit)

[**https://docs.google.com/document/u/0/d/1D0sNsj4V\_kfLSsdNZ3gysUkZHLewaS\_r/edit**](https://docs.google.com/document/u/0/d/1D0sNsj4V_kfLSsdNZ3gysUkZHLewaS_r/edit)

**Template 02**

[**SiglaProjeto\_DocumentoVisao.docx**](https://docs.google.com/document/u/0/d/15eG7hGrrFzUTNzx68z3jR42I87MJIV-4/edit)

[**https://ps.mctic.gov.br/MCTI-PS/guidances/examples/resources/SiglaProjeto\_DocumentoVisao.docx**](https://ps.mctic.gov.br/MCTI-PS/guidances/examples/resources/SiglaProjeto_DocumentoVisao.docx)

**3 DOCUMENTOS DE ARQUITETURA DE *SOFTWARE* E TRÊS MODELOS**

**3.1 Características, diferenças do documento de arquitetura de *software* e tecnologias abordadas na documentação de Arquitetura;.**

Um documento de arquitetura de software irá descrever de forma estruturada e detalhada a arquitetura de um sistema. Planeja e documenta a estrutura, design, padrões, diretrizes, componentes e interações complexas de um sistema para desenvolvimento.

Esse documento é essencial para guiar equipes nas etapas do processo de desenvolvimento de um software, por exemplo, a metodologia ADIT (Análise, Design, Implementação e Testes). A arquitetura nesse contexto, é utilizada como base para elaborar um design e para a codificação, além de ser um dos guias para o tester depurar o software posteriormente.

O documento de visão, portanto, define o escopo, requisitos, restrições etc do projeto que funciona como um guia para todos os envolvidos (desde equipe a stakeholders) enquanto o documento de arquitetura fornece uma visão geral do sistema e da sua estrutura para a equipe de desenvolvedores.

**3.2 Tecnologias abordadas na Documentação de Arquitetura**

As documentações tem como finalidade fornecer uma visão arquitetural dos sistemas. Geralmente os tipos de tecnologia que aparecem nos documentos de arquitetura são:

* Linguagens de programação
* Frameworks e Bibliotecas
* banco de dados
* Arquitetura e padrões
* Protocolo e comunicação
* Serviços em Nuvem e Infraestrutura
* Ferramentas de DevOps
* Autenticação e Segurança
* Testes e Qualidade do código

Os três modelos escolhidos neste trabalho para análise são:

**ANEXOS:**

**Modelo 01** - SysArq

Link de acesso ao Modelo: <https://fga-eps-mds.github.io/2021-2-SysArq-Doc/doc_arquitetura/>

**Modelo 02** -

[SiglaProjeto\_DocumentoArquiteturaSoftware.docx](https://docs.google.com/document/u/0/d/1HlBeuEmh-hAnIvEocdETj7Zf2AYnQlAR/edit)

<https://pdp.mctic.gov.br/MCTI-PDP/guidances/examples/resources/SiglaProjeto_DocumentoArquiteturaSoftware.docx>

**Modelo 03** -

[Modelo-solução-arquitetural.docx](https://docs.google.com/document/u/0/d/1pbgX2Zq9wBfZB6JjYtZJU1uvMCDlMjBI/edit)

<https://datasus.saude.gov.br/wp-content/uploads/2020/01/Modelo-solu%C3%A7%C3%A3o-arquitetural.docx>

A documentação 01 é referente a arquitetura do sistema *SysArq* que é uma plataforma *Web*. A arquitetura desse projeto é baseada em microsserviços. Segundo o documento

Microsserviço é uma abordagem para desenvolver uma única aplicação como um conjunto de serviços, cada um rodando em seu próprio processo e se comunicando através de mecanismos leves, geralmente através de uma API HTTP.

(VITOR, 2021)

No modelo 01 notei que as tecnologias presentes são: *Framework DjangoREST, Framework React*, Banco de Dados *PostgreeSQL*, Requisições HTTP com *token* de autenticação. Especifica a linguagem *Python* e *JavaScript.* Também notei que na parte “*Visão de implementação”* tem diagrama de caso, Entidade Relacionamento e diagrama lógico. Além dos elementos: finalidade, escopo, definições, acrônimos, abreviações, qualidade, histórico de revisão, tamanho e desempenho.

Esse foi o único modelo que tem a representação visual das relações e da lógica através dos diagramas.

A documentação 02 é um modelo de exemplo. Possui e ensina tópicos essenciais que esse tipo de documento precisa conter.

É possível notar a partir da análise e comparação entre o modelo 01, 02 e 03 que mesmo sendo documentos de softwares diferentes, a estrutura da escrita, os tópicos/ capítulos, e registro é praticamente a mesma. Ambos os modelos têm em comum: Capa, índice, Identificação do projeto, Introdução, Representação da arquitetural, Restrições, Visão geral, histórico de revisão, modelagem etc. Tanto o modelo 02 e 03 têm em comum no início o tópico de Identificação de Projeto como mostra as imagens 01 e 02:

Imagem 01: Modelo 02

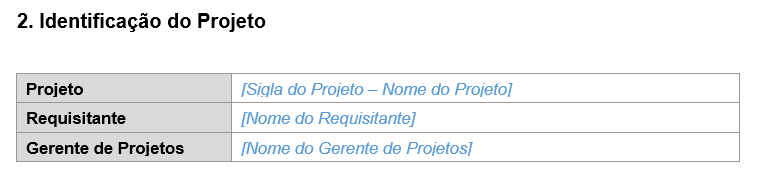
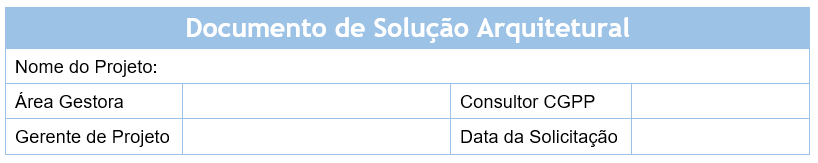


Imagem 02: Modelo 03



A documentação 03 é o último modelo escolhido e também é do DataSUS, assim como o *template* de documento de visão 01.

O modelo 03 tem como base a arquitetura de multicamadas aderente a arquitetura orientada a serviço. As tecnologias que o modelo 03 abordam são: Execução (*Android* 2.3, *IOS* 6 e *Windows* phone 8). *Framework* Sencha Touch versão 2.4.1, Cordova versão 4.3.0, HTML 5 etc.

**4 DOCUMENTO DE REQUISITOS - CASE E ANEXO DO TEMPLATE DE DOCUMENTO DE REQUISITOS**

Todo desenvolvimento de software começa com a comunicação. Principalmente, para o levantamento e análise de requisitos. Os requisitos são fundamentais para definir as funcionalidades do sistema. Os requisitos funcionais (RF) referem-se às especificações que definem o que o software deve fazer. Enquanto os requisitos não funcionais (RNF) refere-se às qualidades e restrições do sistema. A especificação dos requisitos serve como base para o desenvolvimento e testes do sistema. Regras de negócio (RN).

O case escolhido é: (*Case* 01) Sistema de Gestão de comércio eletrônico *E-commerce.*

Escolhi essa temática pois vou analisar o sistema do Mercado Livre porque desenvolvi sobre a empresa e suas aplicações em trabalhos anteriores do curso como na matéria Análise de Requisitos.

**ANEXOS:**

[**ATV 02 Felipe LEVANTAMENTO E ANÁLISE DE REQUISITOS.pdf**](https://drive.google.com/open?id=17YHOCx47KXOBscCtlsLVHrqLgrC5v3yP)

[**TemplateDocumentoRequistosSoftware.docx**](https://docs.google.com/document/u/0/d/1NQGUijnHddynGDTxeGLhYhUZNU-2Q7fo/edit)

[**https://www.facom.ufu.br/~william/Disciplinas%202016-2/BSI-GSI030-EngenhariaSoftware/Templates/TemplateDocumentoRequistosSoftware.docx**](https://www.facom.ufu.br/~william/Disciplinas%202016-2/BSI-GSI030-EngenhariaSoftware/Templates/TemplateDocumentoRequistosSoftware.docx)

**Case: E-commerce**

**1 Contexto:** Uma plataforma de e-commerce que *vende produtos* de diferentes *fornecedores* e precisa gerenciar diversas funcionalidades, como: catálogo de produtos, carrinho de compras, pedidos, pagamento, entrega, e feedback dos clientes.

**2 Cliente**

Mercado Livre LTDA

Ramo: Produtos variados

**3 Descrição dos Requisitos (RF, RNF e RN)**

**3.1 Requisitos Funcionais (RF)**

Refere-se às especificações que definem o que o software deve fazer:

**RF1 -** O sistema deve permitir que o usuário faça o login utilizando nome de usuário e senha;

**RF2 -** O software precisa deixar que o usuário pesquise e olhe as opções disponíveis dos produtos do seu interesse;

**RF3 -** O software deve permitir que o usuário coloque e visualize o produto no carrinho de compras;

**RF4 -** O sistema deve possuir um espaço para o usuário poder gerenciar sua conta, poder acessar e editar seus dados pessoais como: foto de perfil, endereço e método de pagamento.

**RF5 -** Acompanhar pedidos: o usuário deve conseguir ver seus pedidos, acompanhar o envio, cancelamentos e solicitar suporte) .

**3.2 Requisitos não funcionais (RNF)**

Refere-se às qualidades e restrições do sistema:

**RNF 1 -** O sistema deve criptografar com segurança os dados de pagamento como: número de cartão de crédito, débito, CPF etc.

**RNF 2 -** O sistema deve enviar um e-mail de confirmação da compra e a nota fiscal;

**RNF 3 -** O sistema deve ser acessível em dispositivos móveis (celular, tablet etc)

**RNF 4 -** O tempo de carregamento da página deve ser inferior a 1 minuto;

**RNF 5 -** O sistema deve ser capaz de processar mais de 30.000 transações por segundo.

**3.3 Regras de negócio (RN)**

**RN1 - Cadastro de Vendedores**

* Apenas usuários previamente cadastrados e aprovados pela plataforma podem vender produtos.
* Os vendedores devem fornecer documentação válida para comprovação da identidade e conta bancária.

**RN2 - Política de Preços e Promoções**

* O preço dos produtos é definido pelos vendedores, mas a plataforma pode oferecer cupons de desconto e promoções especiais.
* Promoções ativas não podem ser alteradas ou canceladas antes do prazo estipulado.

**RN3 - Processamento de Pagamentos**

* O pagamento só será repassado ao vendedor após a confirmação da entrega do pedido.
* A plataforma cobra uma taxa de comissão sobre cada venda realizada.

**RN4 - Política de Devolução e Reembolso**

* O cliente pode solicitar devolução do produto dentro de um prazo de 7 dias após o recebimento, conforme o Código de Defesa do Consumidor.
* O reembolso será realizado apenas após a devolução do produto ao vendedor e verificação de integridade.

**RN5 - Avaliações e Feedbacks**

* Apenas clientes que compraram um produto podem avaliá-lo e deixar um comentário.
* O vendedor pode responder às avaliações, mas não pode excluí-las ou modificá-las.

**RN6 - Política de Entrega**

* O prazo de entrega informado ao cliente deve considerar o tempo de processamento do pedido e o transporte.
* O cliente deve ser notificado por e-mail ou SMS sobre o status da entrega.

**REFERÊNCIAS BIBLIOGRÁFICAS**

TEMPLATE 01

BRASIL. Ministério da Saúde. Documento de visão de sistema. Brasília, DF, 2019. Disponível em: <https://datasus.saude.gov.br/wp-content/uploads/2019/12/Documento-de-vis%C3%A3o-de-sistema.docx> . Acesso em: 14 mar. 2025.

TEMPLATE 02

UNIVERSIDADE FEDERAL DE UBERLÂNDIA. Template Documento de Requisitos de Software. Uberlândia, MG, 2016. Disponível em: <https://www.facom.ufu.br/~william/Disciplinas%202016-2/BSI-GSI030-EngenhariaSoftware/Templates/TemplateDocumentoRequistosSoftware.docx> . Acesso em: 14 mar. 2025.

MODELO 01 - SysArq

UNIVERSIDADE DE BRASÍLIA. Documento de Arquitetura de Software – SysArq. Brasília, DF, 2021. Disponível em: <https://fga-eps-mds.github.io/2021-2-SysArq-Doc/doc_arquitetura/> . Acesso em: 15 mar. 2025.

MODELO 02 - Modelo de Documento de Arquitetura de Software

BRASIL. Ministério da Ciência, Tecnologia e Inovações. Modelo de Documento de Arquitetura de Software. Brasília, DF, 2025. Disponível em: <https://pdp.mctic.gov.br/MCTI-PDP/guidances/examples/resources/SiglaProjeto_DocumentoArquiteturaSoftware.docx> . Acesso em: 15 mar. 2025.

MODELO 03 - DataSUS

BRASIL. Ministério da Saúde. Modelo de solução arquitetural. Brasília, DF, 2020. Disponível em: <https://datasus.saude.gov.br/wp-content/uploads/2020/01/Modelo-solu%C3%A7%C3%A3o-arquitetural.docx> . Acesso em: 15 mar. 2025.

**BRASIL. Ministério da Saúde. DATASUS.** Regras de Negócio: SIGLA PROJETO – SIGLA MÓDULO DRE. Disponível em:<https://datasus.saude.gov.br/wp-content/uploads/2019/08/8.SIGLAPROJETO_SIGLAMODULO_DRE.docx>. Acesso em: 16 mar. 2025.