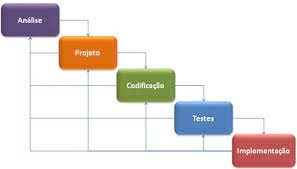
**ENGENHARIA DE SOFTWARE**

**(Material de apoio para 1º Bimestre)**

CICLO DE VIDA CLASSICO



1- ANÁLISE

**Levantamento de Requisitos**

**Nesta fase, identificamos e analisamos as necessidades e expectativas do cliente.**

É crucial compreender completamente os requisitos do projeto para evitar retrabalhos e a insatisfação do cliente.

**Análise**

A equipe de desenvolvimento avalia os requisitos identificados e cria uma especificação detalhada do software.

2. PROJETO

Nesta fase, os detalhes técnicos são elaborados com base nas especificações da análise.

A arquitetura de software é projetada, e os algoritmos e estruturas de dados são definidos.

**As principais atividades envolvidas na etapa de Projeto incluem:**

Definição da Arquitetura do Sistema: Nesta atividade, é determinada a estrutura geral do software, incluindo a divisão em módulos, subsistemas e componentes. Decisões sobre como os diferentes elementos do sistema se relacionarão entre si são tomadas nesta fase.

Escolha de Tecnologias: Decisões sobre as ferramentas, linguagens de programação, bibliotecas e frameworks que serão utilizados para implementar o software são tomadas nesta etapa. É importante escolher tecnologias que sejam adequadas aos requisitos do projeto e que possibilitem um desenvolvimento eficiente.

Revisão e Validação: O projeto deve ser revisado e validado por outras partes interessadas, como a equipe de desenvolvimento, arquitetos e clientes. Isso ajuda a garantir que o design atenda aos requisitos do cliente e esteja alinhado com a visão do produto final.

3. CODIFICAÇÃO

A equipe de desenvolvimento começa a **escrever o código-fonte** com base nos projetos e nas especificações.

As melhores práticas de programação são seguidas para garantir a qualidade e a manutenibilidade do código.

4. TESTE

**O software é submetido a teste**s rigorosos para identificar erros e defeitos.

Testes de unidade, integração e sistema são realizados para garantir que o software atenda aos requisitos definidos.

5. ENTREGA

O software é lançado em ambiente de produção após os testes serem concluídos com sucesso.

Os usuários finais podem começar a usar o produto e fornecer feedback.

**Manutenção**

Após a implantação, a equipe monitora o software em produção para corrigir problemas e fazer melhorias.

As atualizações e correções de bugs são lançadas conforme necessário para garantir o bom funcionamento contínuo do software.

# REQUISISTO X ESPECIFICAÇÃO DE REQUISITO

## REQUISITO

**O requisito é uma declaração que descreve uma necessidade ou condição que o sistema deve atender para satisfazer as expectativas dos stakeholders** (clientes, usuários finais, equipe de desenvolvimento, etc.). Os requisitos são geralmente expressos de forma ampla, não técnica e sem detalhes específicos sobre como o sistema atenderá a essas necessidades. Eles definem o "o quê" deve ser feito, não o "como" será feito.

ESPECIFICAÇÃO DE REQUISITO

**A especificação de requisito é uma documentação detalhada que descreve em profundidade cada requisito funcional e não funcional do sistema.** Ela transforma as declarações de alto nível dos requisitos em descrições detalhadas e precisas de como o sistema deve se comportar para atender a esses requisitos.

A especificação de requisito inclui informações técnicas, diagramas, fluxos de trabalho, requisitos de desempenho, interfaces do usuário, critérios de aceitação, entre outros detalhes relevantes. Ela serve como uma referência clara e detalhada para a equipe de desenvolvimento e para as partes interessadas envolvidas no projeto, garantindo uma compreensão comum dos requisitos do sistema.

# **O QUE É E COMO FAZER A DEFINIÇÃO DE REQUISITO FUNCIONAL**

Um requisito funcional é uma declaração que descreve uma função específica ou uma capacidade que um sistema de software deve ter para atender às necessidades dos seus usuários

Exemplo:

**RF-0001**: O sistema deve enviar um e-mail de confirmação após a conclusão de um pedido.

[O e-mail deve informar: número do pedido, itens que compõem o pedido e valor total do pedido]

# **O QUE QUE É E COMO DEFINIÇÃO DE REQUISITO NÃO FUNCIONAL**

São os requisitos relacionados ao uso da aplicação em termos de **desempenho**, **usabilidade**, **confiabilidade**, **segurança**, **disponibilidade**, **manutenibilidade** e **tecnologias envolvidas**.

Exemplos:

[**RNF-0001**] O sistema deverá estar de acordo com a LGPD.   
(Requisitos decorrentes de fatores externos ao sistema e ao processo de desenvolvimento, interoperabilidade, legislação, localização geográfica etc.)

[**RNF-0001**] Os usuários deverão poder operar o sistema após 1 semana de treinamento.  
(Requisitos de facilidade de uso)

[**RNF-0002**] O sistema deverá processar 200 requisições por minuto   
(*Requisitos de eficiência)*

[**RNF-0003**] O sistema deverá rodar em celulares Android  
(Requisitos de portabilidade)

[**RNF-0004**] O sistema deverá ser desenvolvido na linguagem Java.  
(Requisitos de implementação)

[**RNF-0005**] O sistema deverá se comunicar com o SQL Server  
(Requisitos de interoperabilidade)

PARA QUE SERVE A DOCUMENTAÇÃO DE UM SOFTWARE

**Compreensão do Sistema:** A documentação detalhada permite que os membros da equipe de desenvolvimento, futuros desenvolvedores e outras partes interessadas compreendam a estrutura, a lógica e o funcionamento do sistema. Isso facilita a colaboração entre as equipes e garante que todos tenham uma visão clara do software.

**Manutenção e Evolução:** Com o passar do tempo, o software requer manutenção e atualizações para corrigir erros, adicionar novas funcionalidades ou melhorar o desempenho. A documentação fornece informações essenciais para facilitar a manutenção do software, permitindo que a equipe identifique e corrija problemas com mais facilidade.

**Transferência de Conhecimento:** A documentação ajuda a transferir conhecimentos críticos dentro da equipe ou entre diferentes projetos. Isso é especialmente útil quando há mudanças de membros da equipe ou quando um projeto é repassado para outra equipe.

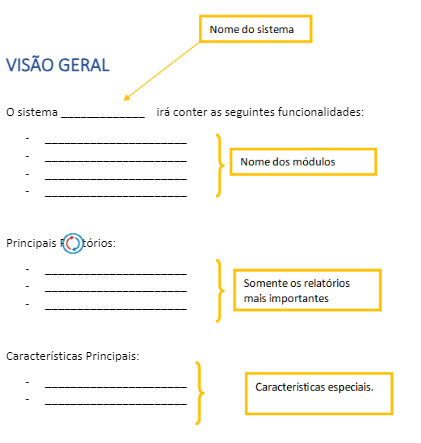
**Reutilização de Código:** A documentação detalhada do código-fonte permite que os desenvolvedores entendam como os componentes podem ser reutilizados em outros projetos, economizando tempo e recursos.

ESTRUTURA BÁSICA DO DOCUMENTO DE ESPECIFICAÇÃO DE REQUISITOS

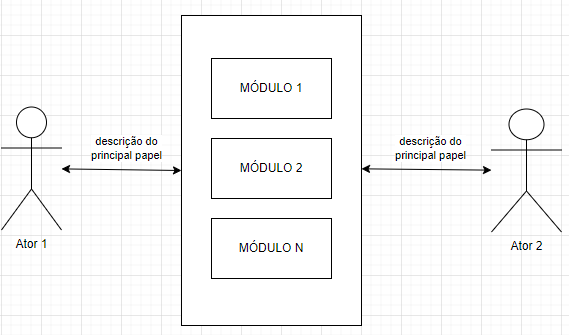
O documento de especificação de requisitos possui a seguinte estrutura:

|  |
| --- |
| **VISÃO GERAL *(****Breve descrição das funcionalidades). (nível macro) (“ouvir o sistema”)*  **DIAGRAMA DE CONTEXO *(****Breve descrição das funcionalidades. (nível macro) (“ver o sistema”)*  **REQUISITOS *(****Especificações dos requisitos funcionais e não funcionais).*  **GLOSSÁRIO *(****Descrição dos termos técnicos).* |

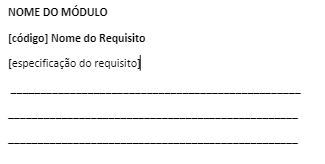
FÓRMULA PARA ESCREVER A VISÃO GERAL



FÓRMULA PARA ESCREVER O DIAGRAMA DE CONTEXTO



# **FÓRMULA BÁSICA PARA ESPECIFICAÇÃO DOS REQUISITOS**



LISTA DE QUESTÕES

BATERIA 1

**1) Qual é a segunda fase do Ciclo de Vida Clássico de Software?**

a) Análise

b) Projeto

c) Codificação

d) Teste

**2) O que é a especificação de requisito?**

a) Documento que descreve as necessidades e expectativas do cliente.

b) Documento que contém a arquitetura do sistema.

c) Documento que detalha cada requisito funcional e não funcional do sistema.

d) Documento que descreve as atividades de codificação do software.

**3) Qual é a atividade realizada na etapa de PROJETO do Ciclo de Vida Clássico?**

a) Levantamento de Requisitos

b) Definição da Arquitetura do Sistema

c) Teste

d) Manutenção

**4) Para que serve a documentação de um software?**

a) Apenas para treinamento dos desenvolvedores.

b) Apenas para garantir a conformidade regulatória.

c) Para facilitar a colaboração entre as equipes e a compreensão do sistema.

d) Apenas para reutilização de código em outros projetos.

**5) Qual é a fase do Ciclo de Vida Clássico de Software onde são definidas as tecnologias que serão utilizadas no sistema.?**

a) Codificação

b) Análise

c) Teste

d) Projeto

**Respostas:**

1. **B**
2. **C**
3. **B**
4. **C**
5. **D**

BATERIA 2

**1) Sobre os objetivos da Engenharia de software marque certo ou errado.**

( ) A engenharia de software visa criar e manter software de alta qualidade de forma eficiente, seguindo métodos e práticas disciplinadas, garantindo segurança e conformidade, e atendendo às necessidades dos usuários e das partes interessada

( ) Um dos objetivos da engenharia de software é desenvolver software de alta qualidade que atenda às necessidades do usuário.

( ) A engenharia de software não se preocupa com o gerenciamento de projetos e prazos de entrega.

( ) A engenharia de software visa minimizar os custos de desenvolvimento, manutenção e operação do software.

( ) A engenharia de software não envolve a documentação adequada do software desenvolvido.

( ) A engenharia de software não se preocupa com a otimização do desempenho do software.

( ) A engenharia de software não se preocupa com a evolução do software ao longo do tempo.

( ) A engenharia de software visa garantir que o software desenvolvido atenda aos padrões éticos e legais aplicáveis.

**Respostas:**

**C ,C , E, C, E, E, E, C**

**2) Qual das seguintes opções define corretamente a Fase de ANÁLISE dentro do ciclo de vida clássico de software?**

a) A fase do desenvolvimento de software após a implementação.

b) O processo de gerenciar projetos de engenharia.

c) A fase do ciclo de vida do software que lida com a coleta e documentação de requisitos.

d) A etapa de teste do ciclo de desenvolvimento ágil.

**Resposta C**

**3) Qual das seguintes opções define corretamente a Fase de PROJETO dentro do ciclo de vida clássico de software?**

A. O objetivo principal da fase de projeto é escrever todo o código-fonte do software.

B. A fase de projeto visa definir a arquitetura do software, a estrutura de dados e a interface do usuário, antes da implementação.

C. A fase de projeto não é relevante no ciclo de vida clássico de um software.

D. A fase de projeto tem como objetivo principal a implantação do software nos sistemas dos usuários finais.

**Resposta B.**

**4) Qual das seguintes opções define corretamente a Fase de CODIFICAÇÃO dentro do ciclo de vida clássico de software?**

A. O objetivo principal da fase de codificação é criar a documentação técnica detalhada do software.

B. A fase de codificação tem como objetivo principal definir a arquitetura do software e a estrutura de dados.

C. A fase de codificação visa escrever o código-fonte do software com base no projeto definido, transformando o projeto em um produto de software executável.

D. A fase de codificação não é relevante no ciclo de vida clássico de um software.

**Resposta C.**

**5) Qual das seguintes opções define corretamente a Fase de TESTES dentro do ciclo de vida clássico de software?**

A. O objetivo principal da fase de testes é definir a arquitetura do software.

B. A fase de testes tem como objetivo identificar e corrigir defeitos e problemas no software, garantindo que ele atenda aos requisitos e funcione corretamente.

C. A fase de testes visa criar a documentação técnica detalhada do software.

D. A fase de testes não é relevante no ciclo de vida clássico de um software.

**Resposta B.**

**6) Qual o conceito de REQUISITOS FUNCIONAIS**

Um requisito funcional é uma declaração que descreve as funções ou funcionalidades específicas que um sistema de software deve executar.

**7) Qual o conceito de REQUISITOS NÃO FUNCIONAIS**

Requisitos não-funcionais são os requisitos relacionados ao uso da aplicação em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenibilidade e tecnologias envolvidas.

**8) Requisitos funcionais referem-se a:**

a) Restrições de desempenho e confiabilidade do sistema.

b) Necessidades e expectativas dos stakeholders.

c) Características técnicas do hardware e software.

d) Processos de validação e verificação.

**Resposta: B**

**9) O que é o escopo de um projeto de Engenharia de Software?**

a) Uma lista de riscos identificados.

b) O tamanho total da equipe de desenvolvimento.

c) As metas e objetivos do projeto, incluindo os limites do que será e não será desenvolvido.

d) A lista final de funcionalidades após o desenvolvimento.

**Resposta C**

**10) O que é a especificação de requisitos?**

a) Uma lista de erros encontrados durante a fase de testes.

b) Um resumo do projeto em uma única página.

c) Um documento que detalha os requisitos do sistema, suas características e restrições.

d) Uma apresentação para os stakeholders sobre o andamento do projeto.

**Resposta C**