



Análise e Desenvolvimento de Sistemas

## **ENGENHARIA DE SOFTWARE ÁGIL**

### **RELATÓRIO DE AULAS PRÁTICAS**

Nome: \_\_\_\_\_

RA: \_\_\_\_\_

Polo de matrícula: \_\_\_\_\_

Local da realização da Aula Prática: \_\_\_\_\_

Ano da postagem: \_\_\_\_\_

<b>Instituto de Ciências Exatas e Tecnologia</b>	<b>Disciplina:</b> Análise e Projeto de Sistemas <b>Título da Aula:</b> Engenharia de Software	<b>Relatório 1</b>
--	---	--------------------

## 1. Resumo Teórico

Escreva aqui um texto de 8 a 10 linhas abordando:

- O que é engenharia de software?
- Diferenças entre as análises de um software e de um sistema de software.
- Exemplifique com situações reais o ciclo evolutivo de um determinado software ou sistema de software popularmente conhecido no mercado.

---

---

---

---

---

---

---

---

---

---

## 2. Estudo de Caso

Sistema escolhido: (exemplo: Aplicativo do pacote Office, Adobe, Corel e outros, sistemas de software da IBM, TOTVS, SAP ou de um sistema operacional).

Descrição:

- Breve resumo do sistema.
- Qual o ciclo evolutivo do software, como as principais mudanças que ocorreram desde sua criação até a atualidade?

---

---

---

---

---

- Explique os campos do código de versionamento, quais as versões e releases que mais se destacaram e o porquê.

### 3. Diagrama do versionamento

- Breve descritivo (de 3 a 5 linhas) da ferramenta escolhida para criar o mapa de versões.

---

---

---

---

---

(Inserir aqui a imagem do diagrama com um mapa das versões e releases criadas na ferramenta escolhida – GitHub, Draw.io, Astah ou Lucidchart).

#### 4. Reflexões Finais

- O quanto é importante na engenharia de software no contexto atual?
- Quais as maiores reclamações de usuário que indicam que o software precisa melhorar?
- O que você aprendeu nesta prática?

---

---

---

---

---

---

---

#### 5. Referências

(Inclua pelo menos 1 referência do livro-texto e outras fontes utilizadas.)

<b>Instituto de Ciências Exatas e Tecnologia</b>	<b>Disciplina:</b> Análise e Projeto de Sistemas <b>Título da Aula:</b> Engenharia de Requisitos	<b>Relatório 2</b>
--	---	--------------------

## 1. Resumo Teórico

Escreva aqui um texto de 8 a 10 linhas abordando:

- O que é engenharia de requisitos?
- O conceito dos requisitos RU, RNF, RF e RS.
- O negócio relacionado ao sistema de software ERP escolhido.
- Principais operações dos usuários.

---

---

---

---

---

---

---

---

---

---

---

---

## 2. Estudo de Caso

Sistema escolhido: (exemplo: sistemas de software ERP – IBM, TOTVS, SAP e outros).

Descrição:

- Preparação para a coleta de requisitos (reuniões, questionamentos, ferramentas e técnicas aplicadas na análise).
- Breve resumo sobre o negócio e o sistema ERP, destacando os principais casos de uso.

---

---

---

---

---

---

- O objetivo da validação dos requisitos.

---

---

---

---

---

---

---

---

### 3. Diagrama de casos de uso

- Breve descritivo (de 3 a 5 linhas) da ferramenta escolhida para criar o diagrama de casos de uso.

---

---

---

---

---

(Inserir aqui a imagem do diagrama de casos de uso criada na ferramenta escolhida – Astah, Draw.io ou Lucidchart).

#### 4. Diagrama dos requisitos com mapa mental

- Breve descritivo (de 3 a 5 linhas) da ferramenta escolhida para criar o mapa mental.

---

---

---

---

---

---

---

(Inserir aqui a imagem do diagrama dos requisitos, criada na ferramenta escolhida – Freemind ou Draw.io).

#### 5. Reflexões Finais

- A necessidade do documento de requisitos de software.
- A importância do caso de uso e mapas mentais na visualização e controle dos requisitos.
- Como deve ocorrer a validação dos requisitos por parte dos stakeholders e desenvolvedores.

---

---

---

---

---

---

---

---

## 6. Referências

(Inclua pelo menos 1 referência do livro-texto e outras fontes utilizadas.)



<b>Instituto de Ciências Exatas e Tecnologia</b>	<b>Disciplina:</b> Análise e Projeto de Sistemas <b>Título da Aula:</b> Processos de Software	<b>Relatório 3</b>
--	--	--------------------

## 1. Resumo Teórico

Escreva aqui um texto de 8 a 10 linhas abordando:

- O que é o processo de software e o modelo de processo de software.
- Principais modelos de processos.
- O negócio relacionado à funcionalidade escolhida do sistema de software ERP.
- Principais operações dos usuários.

---

---

---

---

---

---

---

---

---

---

## 2. Estudo de Caso

Funcionalidade escolhida: (exemplo: Funcionalidade Financeira de alguns dos sistemas de software ERP – IBM, TOTVS, SAP e outros).

Descrição:

- Especificar o modelo de processo de software adotado (Incremental, RAD ou Espiral).

- Breve resumo sobre o que precisa ser feito para criar a funcionalidade e alinhá-la com as atividades do desenvolvimento ditadas pelo modelo de processo adotado.

---

---

---

---

---

---

---

---

### 3. Mapa de navegação da funcionalidade

- Breve descritivo (de 3 a 5 linhas) da ferramenta escolhida para criar o mapa de navegação como protótipo da funcionalidade (mesmas características do mapa mental).

---

---

---

---

---

(Inserir aqui a imagem do mapa de navegação da funcionalidade criada na ferramenta escolhida – Freemind ou Draw.io).

#### 4. Diagrama de atividades

- Breve descritivo (de 3 a 5 linhas) da ferramenta escolhida para criar o diagrama de atividades).

---

---

---

---

---

---

---

(Inserir aqui a imagem do diagrama de atividades que mostre as fases do modelo de processo de software adotado com a construção da funcionalidade).

Escolha um dos modelos de processo de software e alinhe suas fases com a construção da funcionalidade. Apresente a sequência de tarefas do desenvolvedor por meio de um diagrama de atividades criada na ferramenta escolhida – Astah, Draw.io ou Lucidchart).

#### 5. Reflexões Finais

- Como se procede, o que precisa ser feito para construir uma determinada funcionalidade.
- A função de um mapa de navegação do software, que seja software empresarial, da web ou algum outro.
- O que determinou e o porquê da escolha de um modelo de processo de software.

- O porquê do uso de um diagrama de atividades.

---

---

---

---

---

---

---

## 6. Referências

(Inclua pelo menos 1 referência do livro-texto e outras fontes utilizadas.)

<b>Instituto de Ciências Exatas e Tecnologia</b>	<b>Disciplina:</b> Análise e Projeto de Sistemas <b>Título da Aula:</b> Planejamento do Processo de Software	<b>Relatório 4</b>
--	---	--------------------

## 1. Resumo Teórico

Escreva aqui um texto de 8 a 10 linhas abordando:

- O que é componentização do software.
- Principais diagramas a serem utilizados.
- Como funciona um sistema distribuído.
- Principais operações dos usuários.

---

---

---

---

---

---

---

---

---

---

## 2. Estudo de Caso

Descrição dos componentes de um sistema de software de um portal web.

Descrição:

- Breve resumo sobre sistemas distribuídos.

- Definir o que são módulos, componentes, estereótipos e nós. Quantos nós tem o sistema escolhido?

---

---

---

---

---

---

---

---

### 3. Requisitos do sistema (RS)

- Breve descritivo (de 3 a 5 linhas) sobre o portal web adotado.

---

---

---

---

---

(Inserir aqui a tabela de requisitos do sistema (pelo menos 12)).

#### 4. Diagrama de componentes

- Breve descritivo (de 3 a 5 linhas) sobre os componentes do sistema.

---

---

---

---

---

---

---

(Inserir aqui a imagem de um repertório de componentes pela construção de um diagrama de componentes), criada na ferramenta escolhida – Astah, Draw.io ou Lucidchart).

#### 5. Diagrama de implantação

- Breve descritivo (de 3 a 5 linhas) sobre os módulos de implantação do sistema.

---

---

---

---

---

(Inserir aqui a imagem da arquitetura da infraestrutura de TI, que deverá atender a um sistema distribuído, típico dos ambientes web, formados pela ligação de computadores servidores/clientes, sistemas operacionais, linguagens de programação e estereótipos das conexões (normalmente protocolos de rede). Esses componentes, estereótipos, módulos e nós de ligação deverão estar especificados na tabela de requisitos do sistema (RS), pela construção de um diagrama de implantação, criada na ferramenta escolhida – Astah, Draw.io ou Lucidchart).

## 6. Reflexões Finais

- Como se procede na componentização de um sistema de software?
- A forma de se fazer, especificar e modelar a arquitetura de um sistema de software.
- O que determina um componente do sistema?
- A melhoria da visualização, acompanhamento e suporte que oferece a arquitetura da infraestrutura de TI pelo emprego dos diagramas de componentes e de implantação da UML.

---

---

---

---

---

---

---



---

## 7. Referências

(Inclua pelo menos 1 referência do livro-texto e outras fontes utilizadas.)

<b>Instituto de Ciências Exatas e Tecnologia</b>	<b>Disciplina:</b> Análise e Projeto de Sistemas <b>Título da Aula:</b> Fusão do Produto e do Processo de Software	<b>Relatório 5</b>
--	---	--------------------

## 1. Resumo Teórico

Escreva aqui um texto de 8 a 10 linhas abordando:

- O que é uma matriz de responsabilidades.
- Principais características da MR RACI.
- Como funciona o alinhamento do processo, produto e pessoas.
- Principais medidas a serem feitas no controle e esforços de serviços pelos stakeholders.

---

---

---

---

---

---

---

---

---

---

## 2. Estudo de Caso

Descrição dos principais perfis que compõem uma equipe de desenvolvimento de software.

Descrição:

- Breve resumo sobre o modelo de processo adotado (Incremental, RAD ou Espiral), JAD e distribuição das tarefas pela matriz de responsabilidades.

---

---

---

---

---

- Definir os artefatos de software que serão produzidos.

---

---

---

---

---

---

---

---

### **3. Modelo de processo de software**

- Breve descritivo (de 3 a 5 linhas) sobre o modelo de processo de software adotado.

---

---

---

---

---

(Inserir aqui a imagem do modelo.)

#### 4. Definição dos cargos e funções dos stakeholders

- Breve descritivo (de 3 a 5 linhas) sobre stakeholders.

---

---

---

---

---

(Inserir aqui a lista dos cargos e funções de cada participante da equipe.)

#### 5. Matriz de responsabilidades

- Breve descritivo (de 3 a 5 linhas) sobre a MR RACI.

---

---

---

---

---

(Inserir aqui a matriz de responsabilidades.)

- Determinar o esforço de cada participante da equipe.

---

---

---

---

---

## 6. Diagrama de atividades

- Breve descritivo (de 3 a 5 linhas) da ferramenta escolhida para criar o diagrama de atividades.

---

---

---

---

---

(Inserir aqui a imagem do diagrama de atividades que mostre as atividades do desenvolvimento do software), criado pela ferramenta escolhida – Astah, Draw.io ou Lucidchart).

## 7. Reflexões Finais

- Como se determina a divisão de tarefas pela equipe.
- Como construir uma matriz de responsabilidades de acordo com os padrões do guia de projetos PMBOK®.
- Quais as melhorias que podem ser adotadas na distribuição de tarefas dos desenvolvedores.

---

---

---

---

---

---

---

---

## 8. Referências

(Inclua pelo menos 1 referência do livro-texto e outras fontes utilizadas.)

<b>Instituto de Ciências Exatas e Tecnologia</b>	<b>Disciplina:</b> Análise e Projeto de Sistemas <b>Título da Aula:</b> Processo Unificado	<b>Relatório 6</b>
--	---	--------------------

## 1. Resumo Teórico

Escreva aqui um texto de 8 a 10 linhas abordando:

- O que é o processo unificado, método PERT e Kanban.
- Principais características do framework RUP.
- Como funciona a integração das atividades do RUP com a distribuição de tempos e dependências entre as tarefas, com o método PERT.
- Principais medidas e acompanhamentos a serem feitos no cronograma e rede de cartões.

---

---

---

---

---

---

---

---

---

---

---

## 2. RUP

Descrição das estruturas estáticas e dinâmicas, bem como as fases do processo unificado.

Descrição:

- Breve descritivo sobre as estruturas estáticas e dinâmicas.

---

---

---

---

---

- (Inserir aqui a imagem do framework RUP.)

- Definir as tarefas, dependências entre as tarefas e tempos distribuídos em um ciclo de 30 dias do processo unificado.

---

---

---

---

---

---

---

---

### 3. Tabela PERT

Descrição sobre a aplicação do método PERT.

---

---

---

---



---

Descrição:

- Breve descritivo sobre o PERT.

---

---

---

---

---

- (Inserir aqui a imagem do framework RUP.)

- Construir uma tabela com as tarefas, dependências entre as tarefas e tempos distribuídos em um ciclo de 30 dias do processo unificado.

#### **4. Construção do cronograma e rede de cartões (Kanban)**

- Breve descritivo (de 3 a 5 linhas) sobre cronograma e kanban.

---

---

---

---

---

- Fazer o input dos dados da tabela PERT por meio da aplicação para construção do cronograma e rede de cartões (GanttProject, Trello).

---

---

---

---

---

(Inserir aqui duas imagens: 1. Cronograma do projeto “Gantt”; e 2. Mapa da rede kanban pelo “Gráfico PERT”).

## 5. Reflexões Finais

- Como funciona a distribuição de atividades e períodos do desenvolvimento de software com o RUP?
- Como funciona a construção da tabela PERT?
- A necessidade do cronograma e rede de cartões no acompanhamento das atividades do desenvolvimento.
- As principais funcionalidades de uma aplicação específica (p. ex.: GantProject) para a construção de cronogramas e rede de cartões.
- Quais as melhorias que podem ser adotadas na distribuição de tarefas e períodos para sua realização?

---

---

---

---

---

---

---

---

## 6. Referências

(Inclua pelo menos 1 referência do livro-texto e outras fontes utilizadas.)

<b>Instituto de Ciências Exatas e Tecnologia</b>	<b>Disciplina:</b> Análise e Projeto de Sistemas <b>Título da Aula:</b> Metodologias Ágeis I	<b>Relatório 7</b>
--	---	--------------------

## 1. Resumo Teórico

Escreva aqui um texto de 8 a 10 linhas abordando:

- O que são as metodologias ágeis como o XP, Scrum e FDD?
- Principais características da metodologia ágil XP.
- Como funciona a integração das atividades do XP em um quadro kanban.
- Principais medidas e acompanhamentos a serem feitos no quadro kanban.

---

---

---

---

---

---

---

---

---

---

## 2. Metodologia ágil XP

Descrição do framework XP.

- Breve descritivo sobre as atividades em pares e a distribuição das atividades-chave entre os programadores e (ou) analistas da XP.
- Breve descritivo sobre as atividades-chave da XP.

---

---

---

---

---

---

---

---

### 3. Quadro kanban

Descrição sobre a aplicação do quadro kanban.

Descrição:

- Definir o modelo de cartão, que deverá ter as informações da tarefa, datas de início e fim da tarefa e se possível o(s) responsável (ou responsáveis) pela tarefa.
  - Construir um quadro kanban (ferramentas: Draw.io, GanttProject, Trello) dimensionado para entrega da funcionalidade em 15 dias.
- 
- 
- 
- 
- 

### 4. Reflexões Finais

- Como funciona a metodologia ágil XP alinhada a um quadro kanban?
  - O porque existe um ganho no desempenho da equipe com o uso de um quadro kanban?
  - Quais as melhorias que podem ser adotadas com o uso da rede de cartões e o dimensionamento do caminho crítico (análise da rede) para a execução de uma atividade?
- 
- 
- 
- 
- 
- 
-

---

## 5. Referências

(Inclua pelo menos 1 referência do livro-texto e outras fontes utilizadas.)

<b>Instituto de Ciências Exatas e Tecnologia</b>	<b>Disciplina:</b> Análise e Projeto de Sistemas <b>Título da Aula:</b> Metodologias Ágeis II	<b>Relatório 8</b>
--	--	--------------------

## 1. Resumo Teórico

(Escreva aqui um texto de 8 a 10 linhas abordando:)

- O que é a ferramenta Trello?
- Principais características da metodologia ágil Scrum.
- Como funciona a integração das atividades do Scrum em um quadro kanban.
- Principais medidas e acompanhamentos a serem feitas com o uso da ferramenta Trello.

---

---

---

---

---

---

---

---

---

---

---

---

## 2. Metodologia ágil Scrum

Descrição do framework XP.

- Breve descritivo sobre as atividades em pares e a distribuição das atividades-chave entre os programadores e (ou) analistas da XP.
- Breve descritivo sobre as atividades-chave da XP.

---

---

---

---

---

---

---

---

### 3. Quadro kanban com a ferramenta Trello

Descrição sobre a aplicação Trello e construção do quadro kanban.

Descrição:

- Definir o modelo de cartão, que deverá ter as informações de descrição do backlog e das sprints.

---

---

---

---

- Construir um quadro kanban (ferramentas: Trello) dimensionado para entrega da funcionalidade em 15 dias.

- (Inserir aqui imagens de tela do Trello do quadro de cartões e do painel de acompanhamento.)



#### 4. Reflexões Finais

- Como funciona a metodologia ágil Scrum e os parâmetros associados da ferramenta Trello?
- Por que existe uma flexibilidade maior na gestão da equipe de desenvolvimento com o uso da ferramenta Trello em relação a outras ferramentas?
- Quais as melhorias associadas ao Trello que podem ser adotadas com o uso da função Jira?

---

---

---

---

---

---

---

#### 5. Referências

(Inclua pelo menos 1 referência do livro-texto e outras fontes utilizadas.)