



**UNIP**  
**UNIVERSIDADE PAULISTA**

# **Roteiros**

**Engenharia de Software Ágil**



**Instituto de Ciências  
Exatas e Tecnologia**

**Disciplina:** Engenharia de Software Ágil

**Título da aula:** Engenharia de Software

**ROTEIRO 1**

## 1. Objetivos da Aula

- Compreender as características do software e sua evolução no ciclo de desenvolvimento.
- Diferenciar os ciclos de vida do hardware e do software e as mudanças que ocorrem.
- Desenvolver um método baseado em códigos (labels) para o controle de versões e releases.
- Relacionar conceitos teóricos com um exemplo prático como o uso da aplicação "GitHub".

## 2. Recursos Necessários

- Computadores com acesso à internet.
- Ferramentas gratuitas para controle de versionamento (GitHub).
- Material de apoio: Capítulo 1 do livro-texto.
- Editor de texto para o relatório final.

### 3. Estrutura da Aula

1. Abertura (10 minutos): Apresentar conceitos básicos e discutir exemplos.
2. Revisão Conceitual (20 minutos): Curvas de falhas do hardware e software, e versões.
3. Demonstração (20 minutos): Apresentar os recursos do GitHub.
4. Atividade Prática (40 minutos): Cada aluno escolhe uma aplicação para o controle de versões, incluindo modelo de código, tabelas e fluxo de evolução das versões e releases.
5. Encerramento (20 minutos): Discussão das dificuldades e orientações sobre o relatório.

### 4. Relatório Final

O relatório deve conter:

- Resumo teórico (definição da engenharia de software e como ocorre sua evolução).
- Estudo de caso do sistema escolhido (pesquisa sobre a prática de versionamento).
- Fluxo da evolução de um determinado software ou sistema de software.
- Reflexões finais e referências.

### 5. Critérios de Avaliação

- Clareza do resumo teórico (2,0).
- Qualidade das imagens (3,0).
- Conexão entre teoria e prática (3,0).
- Criatividade e melhorias (2,0).

## 6. Conclusão

Ao final desta prática, o estudante deverá ser capaz de diferenciar as etapas de evolução de um software e por que elas ocorrem, representação do fluxo de versões e releases e formas de análises do desempenho e falhas do software.



**Instituto de Ciências  
Exatas e Tecnologia**

**Disciplina:** Engenharia de Software Ágil

**Título da aula:** Engenharia de Requisitos

**ROTEIRO 2**

## 1. Objetivos da Aula

- Entender os requisitos do software, seus processos e atividades.
- Diferenciar os principais tipos de requisitos, do usuário, não funcionais, funcionais e do Sistema.
- Desenvolver métodos de reuniões, questionamentos, preparação, bem como modelagem ágil com mapas mentais.
- Construir o diagrama de casos de uso do sistema ERP.
- Relacionar conceitos teóricos com um exemplo prático para criação de um modelo de relatório, para a apresentação aos stakeholders dos requisitos, pelo analista, em busca de validação, com construção de tabelas, construção de diagrama de caso de uso e criação de mapa mental para a análise dos principais requisitos: RU, RNF, RF e RS.

## 2. Recursos Necessários

- Computadores com acesso à internet.
- Ferramentas gratuitas para construção de mapas mentais (Freemind).
- Ferramentas gratuitas para construção de diagramas de casos de uso (Astar, Draw.io ou Lucidchart).
- Material de apoio: Capítulo 2 do livro-texto.
- Editor de texto para o relatório final.

### 3. Estrutura da Aula

1. Abertura (10 minutos): Apresentar conceitos básicos e discutir exemplos.
2. Revisão Conceitual (20 minutos): Construção das tabelas de requisitos (Word).
3. Demonstração (20 minutos): Principais requisitos a serem levantados e construção do mapa mental dos requisitos: RU, RNF, RF e RS.
4. Atividade Prática (40 minutos): Cada aluno deve escolher um determinado sistema de software ERP (para gestão logística, produção, financeira ou administrativa). Após a escolha fazer o levantamento dos requisitos para o sistema escolhido, construir o respectivo diagrama de casos de uso (use o Astah, Draw.io ou Lucidchart), construir no Word as respectivas tabelas de requisitos com: RU (pelo menos 5), RNF (pelo menos 5), RF (pelo menos 10) e RS (pelo menos 10), construir um mapa mental (use o Freemind) que permita visualizar e controlar mudanças de requisitos, versões, releases e outros.
5. Encerramento (20 minutos): Discussão das dificuldades e orientações sobre o relatório.

### 4. Relatório Final

O relatório deve conter:

- Resumo teórico (Necessidade de alinhamento do negócio com a TI).
- Estudo de caso do sistema escolhido (pesquisa sobre o sistema ERP e as principais ferramentas e técnicas de análise).
- Geração de um modelo de documento para levantamento e apresentação dos requisitos.
- Reflexões finais e referências.

## 5. Critérios de Avaliação

- Clareza do resumo teórico (2,0).
- Qualidade das imagens (3,0).
- Conexão entre teoria e prática (3,0).
- Criatividade e melhorias (2,0).

## 6. Conclusão

Ao final desta prática, o estudante deverá ser capaz de fazer o levantamento de requisitos, bem como técnicas de brainstorm para elicitar requisitos, criar modelos de apresentação de tabelas, casos de uso e mapas mentais interativos que permitem navegar pela história do sistema em análise.



**Instituto de Ciências  
Exatas e Tecnologia**

**Disciplina:** Engenharia de Software Ágil

**Título da aula:** Processos de Software

**ROTEIRO 3**

## 1. Objetivos da Aula

- Entender o que é o processo de software e o modelo de processo de software.
- Diferenciar os principais modelos de processos de software e suas aplicações no desenvolvimento.
- Desenvolver habilidades para trabalhos em equipe com base em modelos de processos de software.
- Construir o mapa de navegação como protótipo de uma determinada funcionalidade de um sistema de software ERP.
- Construir o diagrama de atividades que apresente a sequência dos eventos que ocorrem no ciclo de desenvolvimento, com base específica na escolha de alguns dos modelos de processo: Incremental, RAD ou Espiral. Considere no diagrama desvios, decisões e iterações.
- Relacionar conceitos teóricos com um exemplo prático para criação de um modelo de relatório para as atividades da equipe de desenvolvimento, modeladas por um processo de software na construção de funcionalidades.



## 2. Recursos Necessários

- Computadores com acesso à internet.
- Ferramentas gratuitas para construção do mapa de navegação (Freemind).
- Ferramentas gratuitas para construção de diagramas de atividades (Astar, Draw.io ou Lucidchart).
- Material de apoio: Capítulo 3 do livro-texto.
- Editor de texto para o relatório final.

## 3. Estrutura da Aula

1. Abertura (10 minutos): Apresentar conceitos básicos e discutir exemplos.
2. Revisão Conceitual (20 minutos): Amostragem do processo e principais modelos de processos de software.
3. Demonstração (20 minutos): Como funcionam os modelos de processos de software Incremental, RAD e Espiral.
4. Atividade Prática (40 minutos): Cada aluno deve escolher uma determinada funcionalidade de um sistema de software ERP (para gestão logística, produção, financeira ou administrativa). Após a escolha da funcionalidade, determine as funções que irão compor a funcionalidade (p. ex.: 1. Funcionalidade "Financeiro" – Funções: Contas a pagar, Contas a receber e Caixa); determine as tarefas das funções (p. ex. função "Contas a pagar" – tarefas: construir tela de registro dos dados com Data de vencimento, Lista de recebedores, seleção do recebedor, disponibilidade para verificar Caixa, pagamento e comprovante do pagamento). Construa um mapa de navegação como protótipo da funcionalidade com mapa mental. 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.

## 4. Relatório Final

O relatório deve conter:



- Resumo teórico (sobre o processo e os modelos de processos).
- Estudo de caso do sistema escolhido (pesquisa sobre o sistema ERP e as principais ferramentas e técnicas de análise usadas para implementar uma determinada funcionalidade).
- Geração de um modelo de documento para as atividades de construção de uma determinada funcionalidade.
- Reflexões finais e referências.

## 5. Critérios de Avaliação

- Clareza do resumo teórico (2,0).
- Qualidade das imagens (3,0).
- Conexão entre teoria e prática (3,0).
- Criatividade e melhorias (2,0).

## 6. Conclusão

Ao final desta prática, o estudante deverá adquirir habilidades práticas para o trabalho em equipe de uma determinada funcionalidade, seguindo o padrão de um modelo de processo de software.

  <p><b>Instituto de Ciências Exatas e Tecnologia</b></p>	<p><b>Disciplina:</b> Engenharia de Software Ágil</p> <p><b>Título da aula:</b> Planejamento do Processo de Software</p>	<p><b>ROTEIRO 4</b></p>
---	--	-------------------------

## 1. Objetivos da Aula

- Entender como são produzidos artefatos de sistemas de software.
- Diferenciar os elementos da qualidade, do processo, métodos e ferramentas.
- Desenvolver conhecimentos sobre a estrutura do processo e resultados obtidos.
- Construir a matriz de responsabilidades para uma equipe de desenvolvimento de aplicação conjunta (JAD).
- Relacionar conceitos teóricos com um exemplo prático de formação da equipe de desenvolvimento, determinando seus cargos e funções com base no método JAD, atribuir responsabilidades aos stakeholders associadas a cada artefato a ser produzido construindo em uma determinada atividade reproduzindo os requisitos do sistema (RS) na modelagem da arquitetura da infraestrutura de TI.

## 2. Recursos Necessários

- Computadores com acesso à internet.
- Ferramentas gratuitas para construção de diagramas de atividades (Astar, Draw.io ou Lucidchart).
- Material de apoio: Capítulo 4 do livro-texto.
- Editor de texto para o relatório final.

### 3. Estrutura da Aula

1. Abertura (10 minutos): Apresentar conceitos básicos e discutir exemplos.

2. Revisão Conceitual (20 minutos): Diferenciar a arquitetura da lógica de processamento da arquitetura do sistema que é a integração do software, hardware, pessoas, banco de dados e rede de computadores.

3. Demonstração (20 minutos): A partir da tabela de requisitos do sistema, construir as "peças" necessárias (componentes) com o aplicativo, alguma ferramenta de modelagem (Astar, Draw.io ou Lucidchart).

4. Atividade Prática (40 minutos): Cada aluno deve escolher uma plataforma web que abrigue um portal e projetar a infraestrutura de TI que dá suporte a este sistema. A infraestrutura de TI 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). A partir daí, construir a arquitetura da infraestrutura de TI usando os diagramas de componentes e de implantação.

### 4. Relatório Final

O relatório deve conter:



- Resumo teórico (sobre a componentização do software).
- Estudo de caso do sistema escolhido (escolher um portal web e fazer o levantamento dos requisitos apropriados para este sistema (RS)).
- Geração de um modelo de documento para as atividades de análise dos requisitos de sistema (RS).
- Reflexões finais e referências.

## 5. Critérios de Avaliação

- Clareza do resumo teórico (2,0).
- Qualidade das imagens (3,0).
- Conexão entre teoria e prática (3,0).
- Criatividade e melhorias (2,0).

## 6. Conclusão

Ao final desta prática, o estudante deverá adquirir habilidades práticas para o trabalho de levantamento dos requisitos do sistema "portal web" e apresentação do projeto de arquitetura da infraestrutura de TI.

  <p><b>Instituto de Ciências Exatas e Tecnologia</b></p>	<p><b>Disciplina:</b> Engenharia de Software Ágil</p> <p><b>Título da aula:</b> Fusão do Produto e do Processo de Software</p>	<p><b>ROTEIRO 5</b></p>
---	--	-------------------------

## 1. Objetivos da Aula

- Entender como são produzidos artefatos de sistemas de software.
- Diferenciar os elementos: qualidade, processo, métodos e ferramentas.
- Desenvolver conhecimentos sobre a estrutura do processo e resultados obtidos.
- Construir a matriz de responsabilidades para uma equipe de desenvolvimento de aplicação conjunta (JAD).
- Relacionar conceitos teóricos com um exemplo prático para formação da equipe de desenvolvimento, determinando seus cargos e funções com base no método JAD, atribuir responsabilidades aos stakeholders (basicamente Cliente, Gerente do projeto, Gerente do sistema, Analista do sistema, Programador e Administrador do banco de dados (DBA)), determinar as atividades associadas a cada artefato a ser produzido e suas devidas responsabilidades, por meio da distribuição dos atributos da MR RACI, determinar por meio de pontuações o esforço em serviços de cada participante do projeto e construir o diagrama de atividades da equipe de desenvolvimento.

## 2. Recursos Necessários

- Computadores com acesso à internet.
- Ferramentas de uso: editor de texto que construa tabelas (Word p. ex.).
- Ferramentas gratuitas para construção de diagramas de atividades (Astah, Draw.io ou Lucidchart).
- Material de apoio: Capítulo 5 do livro-texto.
- Editor de texto para o relatório final.

## 3. Estrutura da Aula

1. Abertura (10 minutos): Apresentar conceitos básicos e discutir exemplos.
2. Revisão Conceitual (20 minutos): Definição de cargos e funções da equipe de desenvolvimento, estrutura da equipe JAD e o papel de cada desenvolvedor.
3. Demonstração (20 minutos): Apresentar uma matriz de responsabilidades (MR RACI, pode ser a mesma do capítulo 5), como foi feita a distribuição dos atributos RACI, identificação dos cargos mais envolvidos no projeto e mostrar o diagrama de atividades da equipe.
4. Atividade Prática (40 minutos): Cada aluno deve escolher um determinado modelo de processo (p. ex.: Modelo de processo Incremental, RAD ou Espiral) e determinar as atividades e tarefas deste processo (pelo menos 10) que irão compor as linhas da matriz, distribuir nas colunas os stakeholders (considere aqui cliente e desenvolvedores), distribuir os atributos RACI, medir a capacidade de esforço de cada stakeholder e construir o diagrama de atividades da equipe.

## 4. Relatório Final

O relatório deve conter:

- Resumo teórico (sobre a matriz de responsabilidades).
- Estudo de caso sobre alinhamento do processo, produto e pessoas.
- Geração de um modelo de documento para o controle das atividades da equipe.
- Reflexões finais e referências.



## 5. Critérios de Avaliação

- Clareza do resumo teórico (2,0).
- Qualidade das imagens (3,0).
- Conexão entre teoria e prática (3,0).
- Criatividade e melhorias (2,0).

## 6. Conclusão

Ao final desta prática, o estudante deverá adquirir habilidades técnicas para a gestão de trabalhos em equipe, bem como a otimização na distribuição das tarefas.



  <p><b>Instituto de Ciências Exatas e Tecnologia</b></p>	<p><b>Disciplina:</b> Engenharia de Software Ágil</p> <p><b>Título da aula:</b> Processo Unificado</p>	<p><b>ROTEIRO 6</b></p>
---	--	-------------------------

## 1. Objetivos da Aula

- Entender como funciona o processo unificado, a tabela PERT e a rede de cartões.
- Interpretar o framework do RUP, suas disciplinas e fases.
- Desenvolver conhecimentos para estimar tempos e dependências entre as tarefas do RUP com base em seu framework.
- Construir uma matriz de tarefas considerando as atividades (disciplinas) do RUP, o tempo esperado (te) em duração de dias para realizar a tarefa e a relação com as tarefas anteriores. Se refere à construção de uma tabela PERT (ver exemplo na seção 5.1.4 Ferramentas).
- Relacionar conceitos teóricos com um exemplo prático na construção de cronogramas associando as atividades do RUP com o método PERT. O cronograma pode ser construído com uma ferramenta (GanttProject, Trello).

## 2. Recursos Necessários

- Computadores com acesso à internet.
- Ferramentas gratuitas para construção de cronogramas (GanttProject, Trello).
- Material de apoio: Capítulos 5 e 6 do livro-texto.
- Editor de texto para o relatório final.

### 3. Estrutura da Aula

1. Abertura (10 minutos): Apresentar conceitos básicos e discutir exemplos.
2. Revisão Conceitual (20 minutos): Interpretação do framework do RUP.
3. Demonstração (20 minutos): Como determinar as atividades do RUP, e apresentar a tabela PERT (rever a seção 5.1.4 Ferramentas).
4. Atividade Prática (40 minutos): Cada aluno deve interpretar e determinar as atividades e fases do RUP que irão compor a matriz, na distribuição dos períodos e tarefas anteriores (construção da tabela PERT), se basear nas fases do RUP para produzir um software em um ciclo de 30 dias. Fazer o input dos dados na matriz para a ferramenta (GanttProject, Trello). Por meio da aplicação, obter o cronograma, pela função "Gantt" e o gráfico da rede de cartões (kanban) pela função "Gráfico PERT".

### 4. Relatório Final

O relatório deve conter:

- Resumo teórico (sobre o processo unificado e método PERT).
- Estudo de caso na distribuição de tarefas em um período a partir do processo unificado.
- Geração de um modelo de documento para o controle de períodos e tempos do desenvolvimento com RUP.
- Reflexões finais e referências.

## 5. Critérios de Avaliação

- Clareza do resumo teórico (2,0).
- Qualidade das imagens (3,0).
- Conexão entre teoria e prática (3,0).
- Criatividade e melhorias (2,0).

## 6. Conclusão

Ao final desta prática, o estudante deverá adquirir habilidades na análise e interpretação do framework RUP, para extrair dados sobre as atividades e fases do RUP, bem como a obtenção de cronogramas e rede de cartões (kanban).



**Instituto de Ciências  
Exatas e Tecnologia**

**Disciplina:** Engenharia de Software Ágil

**Título da aula:** Metodologias Ágeis I

**ROTEIRO 7**

## 1. Objetivos da Aula

- Entender como funcionam os modelos prescritivos e ágeis.
- Diferenciar as metodologias ágeis (XP, Scrum e FDD), com base nas características específicas de cada metodologia.
- Desenvolver conhecimentos sobre o que foi o Manifesto Ágil e objetivos.
- Construir um quadro kanban com as tarefas da metodologia ágil XP.
- Relacionar conceitos teóricos com um exemplo prático na construção de um quadro kanban.

## 2. Recursos Necessários

- Computadores com acesso à internet.
- Ferramentas gratuitas para construção do quadro kanban (Draw.io, GanttProject, Trello).
- Outras ferramentas (uso do próprio Word).
- Material de apoio: Capítulo 7 do livro-texto.
- Editor de texto para o relatório final.

### **3. Estrutura da Aula**

1. Abertura (10 minutos): Apresentar conceitos básicos sobre as metodologias ágeis, com destaque para a XP.
2. Revisão Conceitual (20 minutos): Interpretação do framework da XP.
3. Demonstração (20 minutos): Como determinar as tarefas, o artefato a ser construído nas etapas do XP e como construir a rede de cartões.
4. Atividade Prática (40 minutos): Cada aluno deve interpretar e determinar as atividades da metodologia ágil XP, para cada atividade deverá ser determinado o período (data de início e fim) da tarefa, no modelo de cartão deverá ter as informações da tarefa, datas de início e fim da tarefa e se possível o(s) responsável (ou responsáveis) pela tarefa. O período de construção da funcionalidade deverá ser de 15 dias.

### **4. Relatório Final**

O relatório deve conter:

- Resumo teórico (sobre as metodologias ágeis, com destaque para a XP e kanban).
- Estudo de caso na distribuição de tarefas por cartões.
- Reflexões finais e referências.

### **5. Critérios de Avaliação**

- Clareza do resumo teórico (2,0).
- Qualidade das imagens (3,0).
- Conexão entre teoria e prática (3,0).
- Criatividade e melhorias (2,0).

## 6. Conclusão

Ao final desta prática, o estudante deverá adquirir conhecimentos práticos para a construção de quadros kanban.



**Instituto de Ciências  
Exatas e Tecnologia**

**Disciplina:** Engenharia de Software Ágil

**Título da aula:** Metodologias Ágeis II

**ROTEIRO 8**

## 1. Objetivos da Aula

- Entender os parâmetros básicos para a escolha da ferramenta certa para trabalhos com metodologias ágeis.
- Demonstrar os principais recursos da ferramenta Trello.
- Desenvolver conhecimentos sobre a diversidade de ferramentas para o acompanhamento do método ágil como Trello, Jira e Project.
- Construir um quadro kanban com as tarefas da metodologia ágil Scrum com a ferramenta Trello.
- Relacionar conceitos teóricos com um exemplo prático na construção de um quadro kanban com uso do Trello.

## 2. Recursos Necessários

- Computadores com acesso à internet.
- Ferramentas gratuitas para construção do quadro kanban (Trello).
- Material de apoio: Capítulo 8 do livro-texto.
- Editor de texto para o relatório final.

### 3. Estrutura da Aula

1. Abertura (10 minutos): Apresentar conceitos básicos sobre as metodologias ágeis, com destaque para a XP.

2. Revisão Conceitual (20 minutos): Apresentar o exemplo de aplicação na seção "8.5 Métodos, Ferramentas e Técnicas (MF&T): Aplicativos para operações com metodologias ágeis", do livro-texto.

3. Demonstração (20 minutos): Como funciona o Trello e suas principais funcionalidades.

4. Atividade Prática (40 minutos): Cada aluno deverá definir as principais tarefas da metodologia Scrum e usar o Trello para construir o quadro kanban (ver o exemplo na seção "8.5 Métodos, Ferramentas e Técnicas (MF&T)"). No quadro kanban deverá constar os backlogs e sprints do Scrum. Configurar as funções do Trello necessárias para o Quadro e Painel do Trello.

### 4. Relatório Final

O relatório deve conter:

- Resumo teórico (sobre a metodologia Scrum e sobre a aplicação Trello).
- Estudo de caso na distribuição de tarefas por cartões.
- Reflexões finais e referências.

### 5. Critérios de Avaliação

- Clareza do resumo teórico (2,0).
- Qualidade das imagens (3,0).
- Conexão entre teoria e prática (3,0).
- Criatividade e melhorias (2,0).



## 6. Conclusão

Ao final desta prática, o estudante deverá adquirir conhecimentos práticos para a construção de quadros kanban com a ferramenta Trello.