



Análise e Modelagem de Sistemas

Prof. Me. Junior Freitas de Amaral
Junior.amaral@faculdadeimpacta.com.br

++ | Apresentações

Nome?

Hobby?

Uma curiosidade?

O que espera da disciplina?

O que precisa ter na sala de aula?

O que você traz de conhecimento/ experiência?

...

Apresentação do Plano de Ensino

Plano de Ensino da Disciplina

 <p>FACULDADE IMPACTA TECNOLOGIA</p>	<p><i>Plano de Aula</i></p>
--	-----------------------------

Curso: CST em Análise e Desenvolvimento de Sistemas	Ano Letivo: 2026/1
Nome da Disciplina: Análise e Modelagem de Sistemas	Turma: Diurno
Carga Horária: 80 horas	Aulas: Teóricas-50%; Práticas-50%
Série: 4º	
Docente: Junior Freitas de Amaral	
Coordenação: Prof. Renato de Souza Santos	
<p style="text-align: center;">Competências</p> <ul style="list-style-type: none"> • Capacidade de especificar integralmente o comportamento do sistema de software a ser desenvolvido através do documento denominado SRS (<i>Software Requirements Specification</i>); • Capacidade de avaliar documentos SRS para garantir a qualidade da especificação e a aderência ao <i>software</i> implementado; • Capacidade de desenvolver o projeto de conclusão de curso. 	
<p style="text-align: center;">Habilidades</p> <ul style="list-style-type: none"> • Definir o que é a UML e sua aplicação na modelagem de sistemas orientados a objetos; • Derivar casos de uso a partir dos requisitos de sistema; • Identificar os cenários que compõem um caso de uso; • Detalhar os cenários através da descrição de diálogos completos e coerentes entre o ator e sistema; • A partir do contexto fornecido pela descrição dos cenários dos casos de uso, derivar e manter a rastreabilidade entre os seguintes artefatos: <i>Wireframes</i>, Dicionários de Dados, Requisitos de Software, Mensagens, Algoritmos e Modelos Conceituais; • Utilizar os 5 principais diagramas da UML (Casos de Uso, Classes, Sequência, Atividades e Máquina de Estados), demonstrando aspectos do sistema por diferentes perspectivas. 	

Apresentação da Disciplina

Análise e Modelagem de Sistemas

- **Nossos “combinados”**
 - Chamada sempre ao final de cada aula (não negociável)
 - Participação nas aulas e nas atividades, exercícios e leitura!
 - Uso de Ferramentas: MIRO, Trello, Astah, draw.io
 - Compromissos:
 - Responsabilidade, pontualidade nos prazos e respeito com todos
 - Qualidade nas entregas
 - Protagonismo no processo de aprendizagem
 - Cuidado com “plágios”...



Apresentação da Disciplina

Análise e Modelagem de Sistemas

- **Qual a necessidade desta Disciplina?**
 - Instrumento para viabilizar Software
 - Conhecer e exercitar a Modelagem de Software por meio da UML
 - Aplicação em um Projeto REAL!
- **Modelos de Software**
 - Modelos e Abstrações
 - Objetivo: capturar aspectos da realidade que sejam do interesse da situação
- **Motivadores da Modelagem**
 - Redução de erros e ambiguidades no projeto
 - Temos que considerar diversos fatores: requisitos, protótipos, tamanho do projeto, complexidade, prazos, custos, documentação, manutenção, reusabilidade...
 - Fornece uma base para o design e implementação
 - Sistema deve suportar o aumento de complexidade e abrangência (são “vivos”, dinâmicos):
Clientes solicitam modificações e/ou melhorias; mercado adota novas estratégias; Governo altera e cria leis, impostos e alíquotas...

Apresentação da Disciplina

Alguns Benefícios da Modelagem de Sistemas

- **Clareza**
 - Tradução de requisitos em representações visuais claras.
- **Economia de Custos e Tempo**
 - Identificação de problemas antes da implementação.
- **Colaboração**
 - Comunicação eficiente entre equipes multidisciplinares.
- **Escalabilidade**
 - Maior facilidade para evoluir e adaptar o Produto de Software



Apresentação da Disciplina

Análise e Modelagem de Sistemas

- **Modelos de Software: “Este modelo é aderente o suficiente para esta aplicação específica?”**
 - Há uma enorme lacuna entre o “requisito” e o “código fonte”;
 - Nossos modelos: mais detalhados que requisitos e menos complexos que o código;
 - Não são formalismos matemáticos, mas representações gráficas para um sistema de software!
- **Modelar é também Documentar!**
- **Utilizamos Modelos de Software para tomada de decisões!**



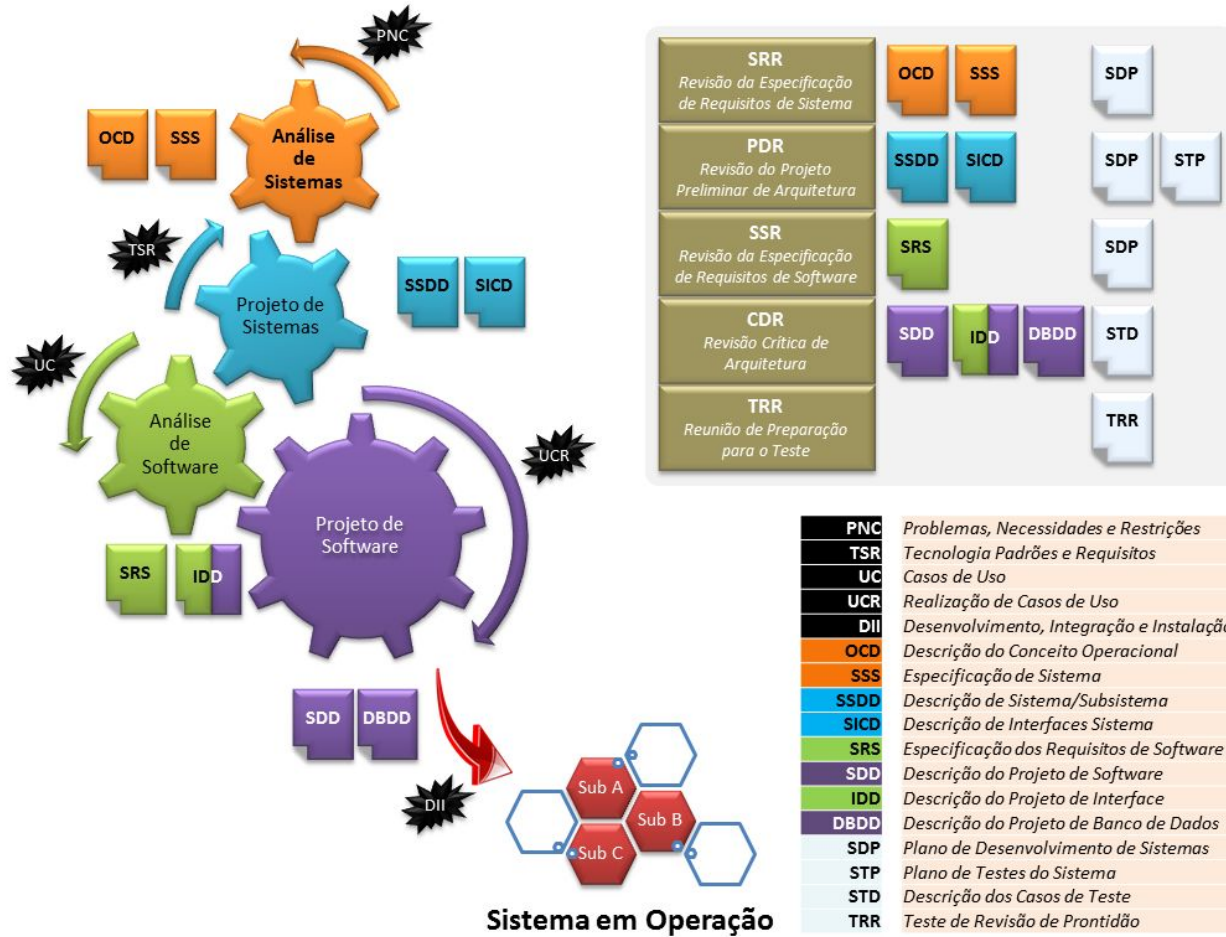
Apresentação da Disciplina

Análise e Modelagem de Sistemas e a IA

- **Auxílio na Geração de Diagramas**
 - IA pode analisar requisitos textuais e gerar diagramas UML automaticamente, como diagramas de caso de uso, de classes e de sequência, economizando tempo e esforço manual.
- **Validação de Modelos**
 - Pode validar modelos UML, verificando inconsistências, erros lógicos e garantindo que o modelo segue as melhores práticas e padrões.
- **Sugestões de Melhoria**
 - Com base em padrões de design e boas práticas, a IA pode sugerir melhorias nos diagramas, tornando o sistema mais eficiente e robusto.
- **Apoio na Documentação**
 - Ajudar a criar documentação detalhada e clara a partir dos diagramas UML, facilitando a compreensão do sistema por parte de todos os envolvidos no projeto.
- **Simulações e Análises**
 - Simular o comportamento do sistema com base nos diagramas UML e analisar possíveis falhas ou pontos de melhoria antes mesmo da implementação.

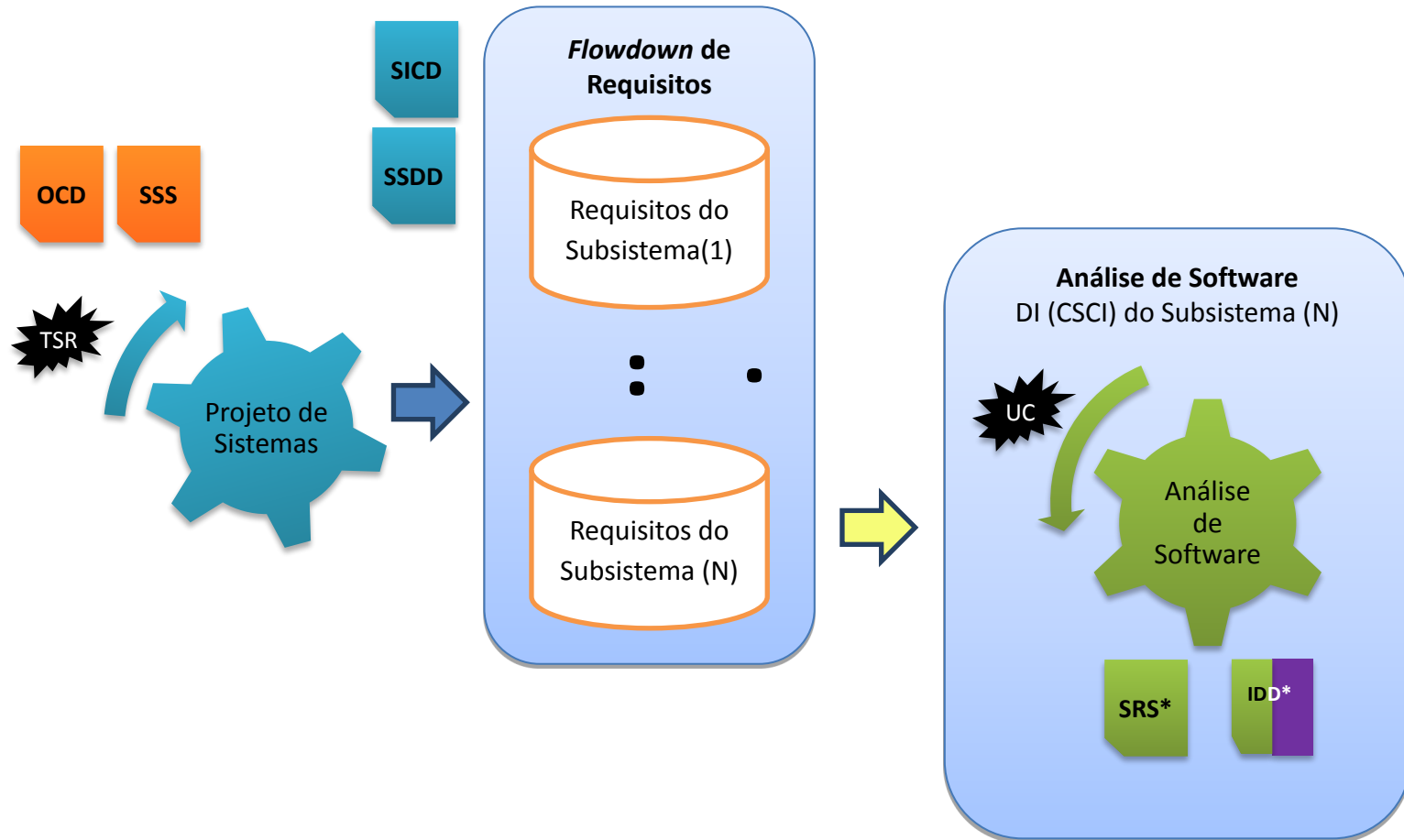
Da Modelagem de Negócio à Especificação

Processo de Desenvolvimento



Da Modelagem de Negócio à Especificação

Artefatos de Entrada



Da Modelagem de Negócio à Especificação

Onde estamos?

	Métodos	Artefatos	Resultados
Análise de Sistemas	Análise do Problema	Declaração do Escopo do Projeto	OCD
		Declaração do Problema	
		Análise das Causas Raízes	
		Usuários e Stakeholders	
		Lista de Restrições	
		Fronteira Sistêmica	
	Workshop de Características	Lista de Características (Prioridade - Esforço - Risco)	BRS
		Definição de <i>Baselines</i>	
	Modelagem dos Processos de Negócio	Arquitetura de Negócio	
		DFD Essencial	
		Análise dos Eventos	
		Análise do Ciclo de Vida dos Conceitos	
		Regras de Negócio	
		Modelo Conceitual	
		Descrição dos Processos de Negócio	
		Descrição dos Cenários de Negócio (BPMN)	
		Eventos de Negócio X Processos de Negócio	
		Processos de Negócio X Regras de Negócio	
	Definição dos Requisitos de Sistema	Requisitos do Sistema (SSS)	SSS
		Processos de Negócio X SSS	
		Características X SSS	
		Estimativa do Esforço X Escopo	
Projeto de Sistema	Definição da Arquitetura do Sistema	Arquitetura do Sistema	SSDD
		Investigação de Concorrência e Paralelismo	
		Definição de HWCI, CSCI e COTS	
		Definição de DI e NDI	
		Descrição dos Subsistemas	
		SSS X Subsistemas (<i>Flowdown</i> de Requisitos)	
		Descrição das Interfaces Internas	IDD
		Descrição das Interfaces Externas (Atores)	SICD

Da Modelagem de Negócio à Especificação

Onde estamos?

	Métodos	Artefatos	Resultados
Análise de Software	Storyboarding	Storyboard para cada Caso de Uso	SRS
	Modelagem de Casos de Uso	Diagrama de Casos de Uso do Subsistema	
		Realização dos Casos de Uso do Subsistema	
		Requisitos de Software do Subsistema (SRS)	
		Dicionário de Dados do Subsistema	
		Mensagens	
		Algoritmos	
		SSS X Casos de Uso	
		Casos de Uso X Dicionário de Dados	
		Casos de Uso X Mensagens	
		Casos de Uso X Algoritmos	
		Casos de Uso X SRS	
Projeto de Software	Projeto de Banco de Dados	Projeto Conceitual do BD	DBDD
		Projeto Lógico do BD	
		Projeto Físico do BD	
	Projeto de IHM	Definição dos Padrões de Interface	SDD
		Design das Interfaces	
	Projeto de Componentes	Padrões Aplicados ao Projeto com Estudo de Casos	
	Projeto Orientado a Objetos		
Projeto de Persistência			

Requisitos de Sistema, de SW e UCs

Entendendo as diferenças

- **Requisitos de Sistema**
 - São descrições claras e detalhadas das necessidades, funcionalidades, restrições e comportamentos que um sistema deve possuir para atender aos objetivos dos stakeholders
 - Esta definição pode variar de acordo com o autor (Sommerville, Pressman, Fairley...)
- **Requisitos de Software**
 - É muito semelhante à anterior, mas com o foco mais específico no Software em si
 - São divididos em:
 - Requisitos Funcionais: descrevem o que o software deve fazer (ligados às funcionalidades do sistema);
 - Requisitos Não-Funcionais: descrevem como o software deve fazer (desempenho, segurança, usabilidade, etc...)
 - Restrições: limitações, imposições ou condições que o software tem que atender.
- **Casos de Uso**
 - Técnica da UML para identificar e detalhar as funcionalidades de um sistema do ponto de vista de quem irá interagir com este sistema.

O artefato SRS

Software Requirements Specification (SRS)

- **Especificação de Requisitos de Software**

- Documenta de forma clara, detalhada e estruturada os requisitos do sistema, servindo como um contrato entre os stakeholders (clientes, usuários, desenvolvedores, etc.) e a equipe de desenvolvimento
- Ele deve:
 - Especificar (o que o sistema deve fazer) - **Requisitos Funcionais**
 - Definir (restrições e propriedades) - **Requisitos Não-Funcionais**
 - Ser completo, consistente e verificável
 - Ser modular e rastreável

- Exemplo:

SSS01 - O sistema **DEVE** permitir ao cliente cadastrar, alterar e consultar os dados, como nome, CPF, email, endereço, telefone. Para os clientes não frequentes, as informações ficarão armazenadas por 1 ano em nosso banco de dados.

SSS02 - O sistema **DEVE** permitir o cadastro, alteração e visualização das informações do cachorro, como nome, raça, peso, idade, preferência alimentar, exigências e a inclusão dos dados de vacinação.

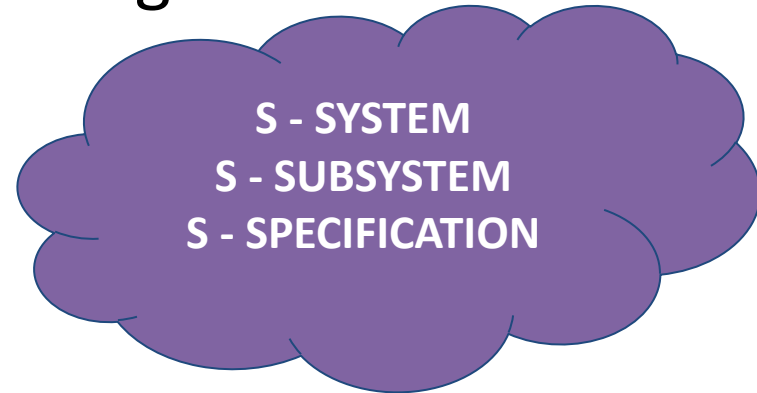
SSS03 - O sistema **DEVE** ser acessado exclusivamente pelo proprietário e por outros usuários autorizados, mediante login e senha.

SSS04 - O sistema **DEVE** permitir a visualização dos horários disponíveis para reserva, tanto para serviços de banho e tosa quanto para hospedagem, realizando assim o agendamento dos mesmos, com a possibilidade de realizar alterações quando necessário.

O artefato Software Requirements Specification - SRS

Com base no artefato 12 de Engenharia de Software:

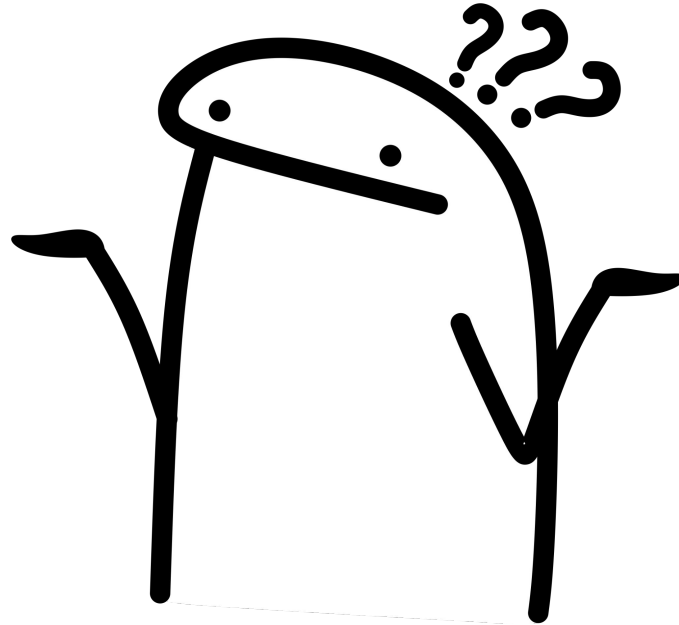
- C01 - Cadastrar Produtos
- C02 - Consultar Produtos
- C03 - Alterar Produtos



> SSS01 - O sistema DEVE permitir o cadastro, a consulta e a alteração de produtos, deve permitir o cadastro de código, nome, descrição e o código deve ser um campo obrigatório.

> SSS02 - O sistema DEVE ...

Engenharia de Requisitos



Dúvidas?

Análise e Modelagem de Sistemas

Para próxima aula:

- **Leitura: Capítulo 1 (Introdução a UML) do Livro:** **GUEDES, G. T. A. UML 2: uma abordagem prática. 1.ed. São Paulo: Novatec, 2009.**
 - **Link do capítulo:**
<https://books.google.com.br/books?hl=pt-BR&lr=&id=mJxMDwAAQBAJ&oi=fnd&pg=PA10&dq=GUEDES,+G.+T.+A.+UML+2:+uma+abordagem+pr%C3%A1tica.+1.ed.+S%C3%A3o+Paulo:+Novatec,+2009&ots=xauXUgSLr9&sig=c9BJnwP8vbFwhEGIDNgQKGGcPts#v=onepage&q&f=false>
- **Vídeos**
 - <https://www.youtube.com/watch?v=5K7OGSsWlZU> (O Analista de Sistemas)
- **Preparação para entrega dos artefatos (ATENÇÃO: entrega na próxima aula)**
 - Declaração do Escopo
 - Integrantes do Projeto
 - SRS
 - (todos do projeto feito pelos grupos no semestre passado)

Exercício em Sala

Dinâmica: Software Requirements Specification (SRS)

- **Construindo o SRS de um Sistema de Biblioteca**
 - Objetivos:
 - Praticar a análise e documentação de requisitos de software
 - Simular a colaboração entre stakeholders e equipe de desenvolvimento
 - Construir um SRS
 - Board MIRO: <https://miro.com/app/board/uXiVLkhaIRk=/>
 - Tempo Estimado: ~60minutos
- **Parte 1: Preparação (~5min)**
 - Sala dividida em grupos de 5 a 6 pessoas
 - Atribuição de Papéis
 - **Stakeholders:** Representam os usuários do sistema (bibliotecários, estudantes, professores)
 - **Analistas de Requisitos:** Responsáveis por elicitar e documentar os requisitos
 - **Facilitador:** Coordena a discussão e garante que todos participem
 - Propósito: cada grupo deve desenvolver um “Sistema de Gerenciamento de Biblioteca” que permita: *a) Cadastrar livros e usuários; b) Realizar empréstimos e devoluções; c) Gerenciar multas por atraso e d) Gerar relatórios de uso*

Exercício em Sala

Dinâmica: Software Requirements Specification (SRS)

- **Parte 2: Levantamento de Requisitos (~20min)**
 - Os “stakeholders” devem listar suas necessidades e expectativas para o sistema. Exemplos:
 - "Como bibliotecário, quero poder cadastrar novos livros rapidamente."
 - "Como estudante, quero poder renovar um empréstimo online."
 - Os “analistas de requisitos” devem fazer perguntas para esclarecer as necessidades:
 - "Quais informações são necessárias para cadastrar um livro?"
 - "Como o sistema deve lidar com multas por atraso?"
 - Use post-its para anotar cada requisito identificado.
- **Parte 3: Classificação e Priorização (~10min)**
 - Classifique os requisitos em “funcionais” e “não funcionais”:
 - Exemplo de requisito funcional: "O sistema deve permitir a renovação de empréstimos."
 - Exemplo de requisito não funcional: "O sistema deve processar uma renovação em menos de 2 segundos."
 - Priorize os requisitos com base na importância para os stakeholders.
 - Use uma escala simples (alta, média, baixa prioridade)

Exercício em Sala

Dinâmica: Software Requirements Specification (SRS)

- **Parte 4: Documentação no SRS (~10min)**
 - Cada grupo deve organizar os requisitos no template simplificado (no MIRO)
- **Parte 5: Apresentação e Feedback (~15min)**
 - Cada grupo apresenta seu SRS para turma
 - Professor e demais grupos fornecem feedback

Extra - Checklist de Requisitos

Critério	Descrição	
Consistente	<p>O requisito pode ser atendido sem entrar em conflito com outros requisitos do sistema?</p> <p>Se não, o requisito deve ser revisado ou removido.</p> <p>O requisito contém erros?</p> <p>Se sim, então o requisito deve ser corrigido.</p>	<p><u>Contraexemplo 1:</u></p> <p>1) "O Sistema DEVE processar no máximo 2000 registros por hora."</p> <p>2) "O Sistema DEVE processar no mínimo 48000 registros por dia."</p> <p>Os requisitos acima são inconsistentes entre si, uma vez que o requisito 1) permite processar menos de 2000 registros por hora. Assim, o requisito 2) nem sempre poderá ser atendido.</p> <p><u>Contraexemplo 2:</u></p> <p>"O Sistema DEVE utilizar a função: $F(id) = \text{RESTO_DA_DIVISÃO_INTEIRA}(id, 2) = 1$, para verificar se um número inteiro é par."</p> <p>Um número inteiro é par se o resto de sua divisão por dois for zero!</p>

Checklist de Requisitos

Critério	Descrição	
Rastreável	A fonte ou origem do requisito é conhecida e o requisito pode ser referenciado ou localizado através do sistema?	<p><u>Contraexemplo:</u></p> <p>Caso algum requisito não tenha rastreabilidade com Característica, Processo de Negócio, Proposta Técnica, Atas ou quaisquer outros documentos que indiquem a sua origem, então o requisito deve ser revisado.</p>
Conciso	O requisito é simples e claro?	<p><u>Contraexemplo:</u></p> <p>"O Sistema DEVE permitir a liberação de notas sempre que as condições propícias tenham sido satisfeitas."</p> <p>Se os conceitos de "liberação de notas" e "condições propícias" não puderem ser esclarecidos mesmo após consultas ao glossário e definições de projeto, o requisito acima deve ser considerado não conciso.</p>

Checklist de Requisitos

Critério	Descrição	
Construtor	O requisito está na forma imperativa utilizando a palavra “ DEVE ”.	<u>Contraexemplo:</u>
Padrão	Requisitos indicando objetivos ou intenções não são imperativos.	<p>“O Sistema NÃO DEVE permitir acessos não autorizados.”</p> <p>O requisito negativo acima pode ser corrigido como:</p> <p>“O Sistema DEVE impedir acessos não autorizados.”</p>

Checklist de Requisitos

Critério	Descrição	
Construtor	O requisito está na forma imperativa utilizando a palavra “ DEVE ”.	<u>Exemplo 1:</u>
Padrão	Requisitos indicando objetivos ou intenções não são imperativos.	<p>"O Sistema, na recepção de mensagens, DEVE comutar em 10 milissegundos, mensagens, em formato ACP128 com indicador de rota válida, recebidas na porta de entrada para a porta de saída indicada pela rota da mensagem."</p> <p>Ator: O Sistema, Condições da Ação: na recepção de mensagens, Ação: DEVE comutar Restrições da Ação: em 10 milissegundos, Objeto da Ação: mensagens Refinamento do Objeto: em formato ACP128 com indicador de rota válida, Fonte do Objeto: recebidas pela porta de entrada Destino da Ação: para a porta de saída (outro) Refinamento da Ação: indicada pela rota da mensagem.</p>

Checklist de Requisitos

Critério	Descrição	
Construtor	O requisito está na forma imperativa utilizando a palavra “ DEVE ”.	<u>Exemplo 2:</u>
Padrão	Requisitos indicando objetivos ou intenções não são imperativos.	<p>"O Sistema, quando solicitado pelo Operador de Fontes de Inteligência, DEVE registrar as Solicitações por Demandas de Monitoramento realizadas por Solicitantes Autorizados."</p> <p>Ator: O Sistema,</p> <p>Condições da Ação: quando solicitado pelo Operador de Fontes de Inteligência,</p> <p>Ação: DEVE registrar</p> <p>Restrições da Ação:</p> <p>Objeto da Ação: as Solicitações por Demandas de Monitoramento</p> <p>Refinamento do Objeto:</p> <p>Fonte do Objeto: realizadas por Solicitantes Autorizados.</p> <p>Destino da Ação:</p> <p>(outro) Refinamento da Ação:</p>

Checklist de Requisitos

Critério	Descrição	
Construtor	O requisito está na forma imperativa	<u>Exemplo 3:</u>
Padrão	utilizando a palavra “ DEVE ”.	"O Sistema DEVE encriptar, segundo o algoritmo AES (Advanced Encryption Standard) definido em Federal Information Processing Standards (FIPS) Publication 197, "Announcing the Advanced Encryption Standard (AES), mensagens enviadas para os pontos de venda."
	Requisitos indicando objetivos ou intenções não são imperativos.	<p>Ator: O Sistema</p> <p>Condições da Ação:</p> <p>Ação: DEVE encriptar,</p> <p>Restrições da Ação: segundo o algoritmo AES (Advanced Encryption Standard) definido em Federal Information Processing Standards (FIPS) Publication 197, "Announcing the Advanced Encryption Standard (AES),</p> <p>Objeto da Ação: mensagens</p> <p>Refinamento do Objeto:</p> <p>Fonte do Objeto:</p> <p>Destino da Ação: enviadas para os pontos de venda.</p> <p>(outro) Refinamento da Ação:</p>



Obrigado!

Prof. Me. Junior Freitas de Amaral
Junior.amaral@faculdadeimpacta.com.br