

Universidade Federal de Sergipe  
Disciplina de Engenharia de  
Software I

# Projektorium

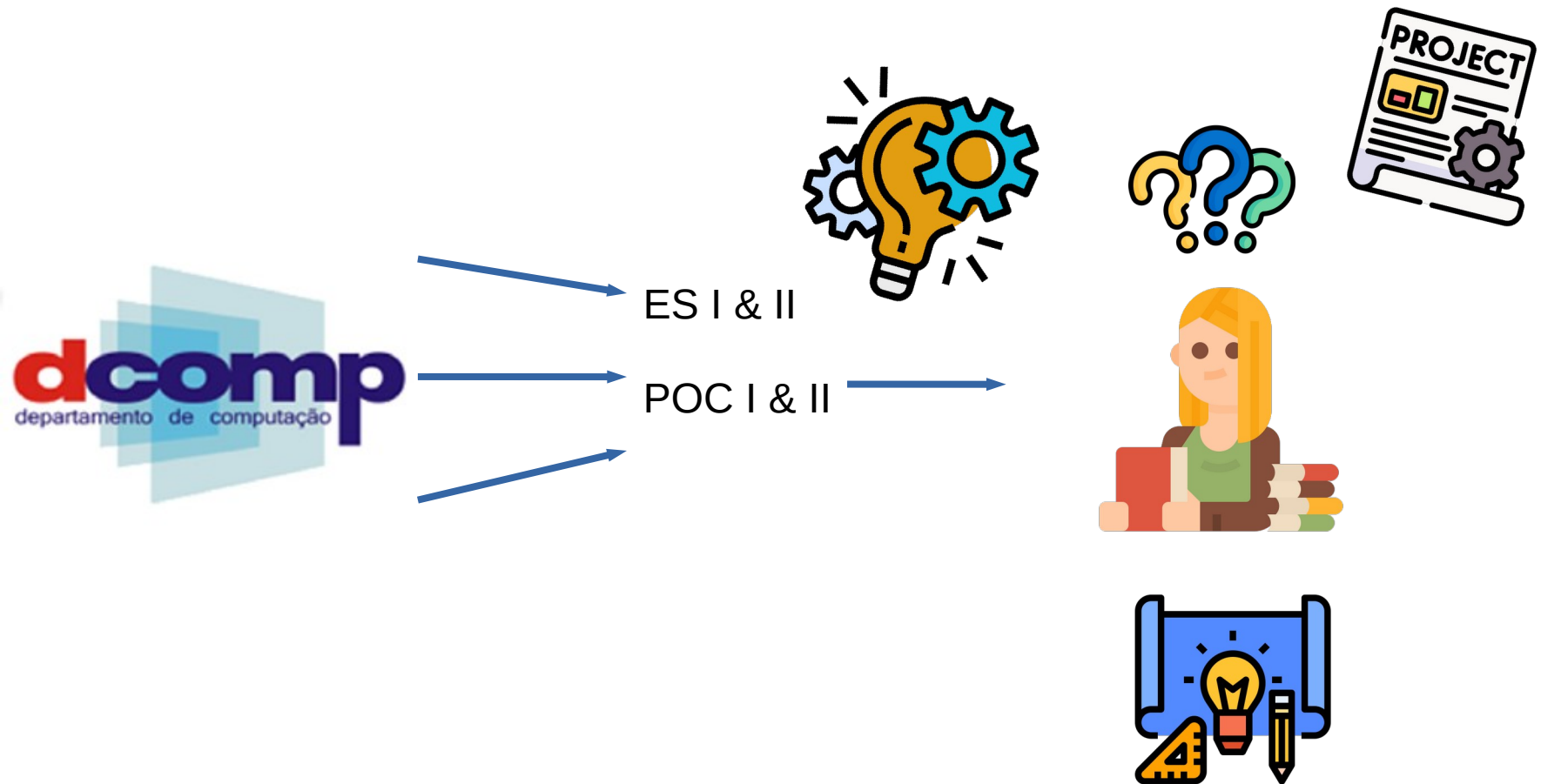
Max Antônio Lima Barreto  
Pedro Vinícius de Araújo Barreto  
Jonas Gabriel dos Santos Ribeiro  
Matheus Victor Fontes Santos

Profa. Dra. Adicinéia A. de Oliveira  
2023/1

# Roteiro da apresentação

- Introdução e contextualização.
- Objetivo do software a ser desenvolvido.
- Histórias dos usuários.
- RF, RNF e RIN.
- Diagramas.
- Protótipo das telas e relatórios.
- Planejamento do projeto.
- Conclusões.

# Introdução e contextualização



# Objetivos

Permitir a busca por projetos feitos

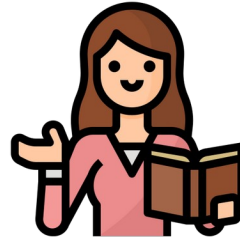
Facilitar organização dos professores

Projetorium

Documentar o que é feito no Dcomp nas disciplinas práticas

Conveniência para os alunos

# Histórias dos Usuários



Quadro 5 – Histórias de Usuários

ID de História de usuário	Como um...	Quero...	Para que eu possa...
1	Aluno	Escolher e participar de um projeto	Realizar as atividades práticas das disciplinas de Engenharia de Software
2	Aluno	Propor um projeto	Trabalhar num projeto cuja temática eu tenha auxiliado a elaborar
3	Pessoa	Consultar os projetos já feitos no contexto da disciplina	Me inspirar pessoalmente, acompanhar o trabalho dos alunos do Dcomp/UFS
4	Professor	Organizar os projetos práticos das disciplinas de Engenharia de Software	Manter um registro documentado e moderno dos projetos já foram ou estão sendo feitos

ID de História de usuário	Como um...	Quero...	Para que eu possa...
5	Professor	Propor projetos para os alunos das turmas de Engenharia de Software	Garantir que eles tenham um método rápido e fácil de escolher os projetos práticos
6	Professor	Aprovar propostas de projetos de aluno	Verificar se estão aptos a serem desenvolvidos ao longo da disciplina
7	Pessoa	Buscar projetos baseados em critérios definidos por mim	Encontrar somente os projetos que sigam os critérios definidos, para que eu possa embasar alguma análise, por ex
8	Gestor de Sistema	Adicionar e gerenciar professores,	Garantir o bom funcionamento do sistema

## Matriz de rastreabilidade

[illegible]

# Requisitos destacados

## Funcionais

- [RFMU01-06] Manter principais atores (alunos, professores, gestores, projetos)
- [RFPP01] Alunos podem propor projetos
- [RFPP02] Professores e gestores podem aprovar
- [RFPP04-05] Permitir consulta de projetos



## Não-Funcionais

- [NFIM02] Banco PostgreSQL 15
- [NFIM01, NFPA01] Sistema Web, implementado com TypeScript
- [NFUS01-03] Usabilidade, como:
  - Cor uniforme (Dcomp)
  - Formulários contidos (max. 10 campos)
  - Telas responsivas

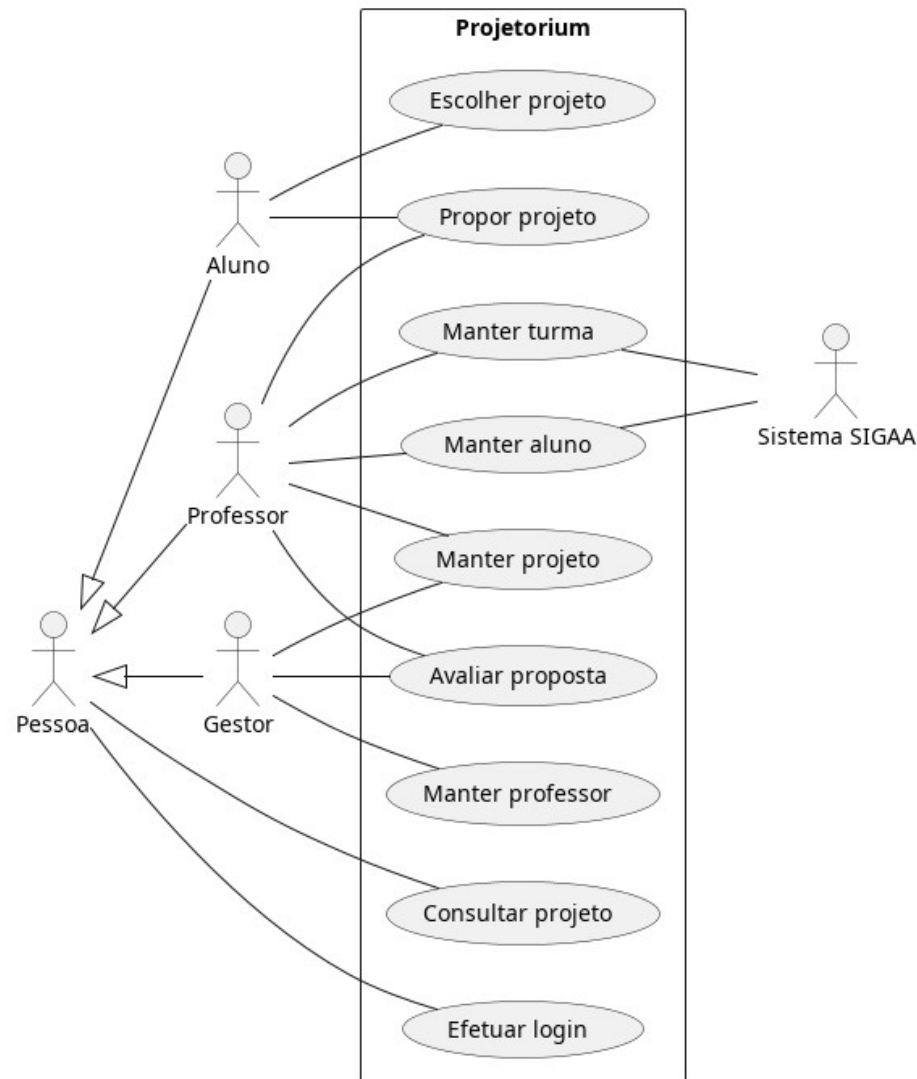


## Inversos

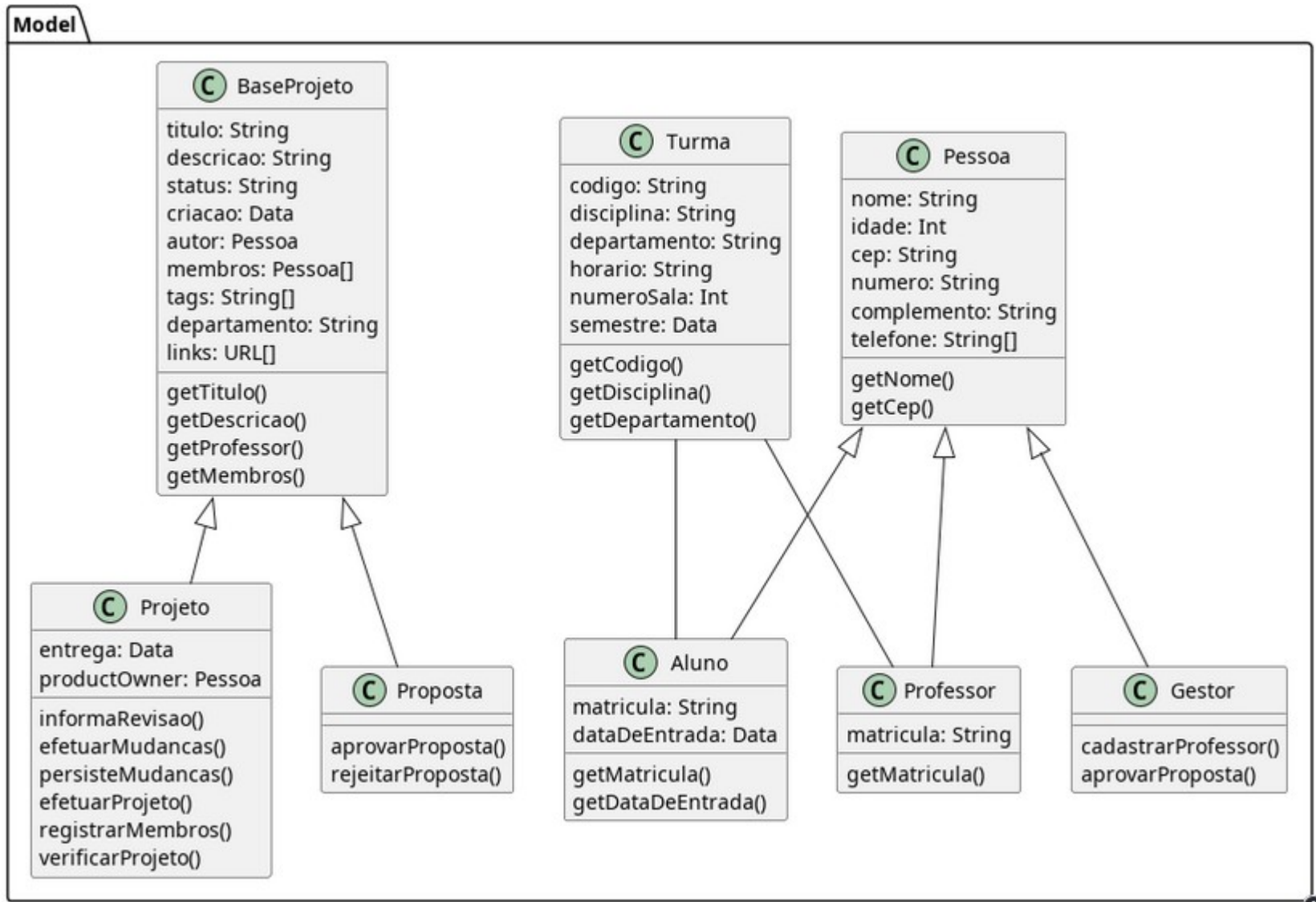
- [RIN01] Propostas nem projetos serão deletados permanentemente



# Diagramas de Análise: Use Case



# Diagrama de Classes de Projeto e



## Control

### C Buscador

processarBusca()  
inserirProjetoNoResultado()  
descartarProjetos()  
gerarResultados()

### C VerificadorLogin

verificarCadastro()

### C ControladorProjeto

tratarRevisao()  
encontrarProjeto()

### C GerenciadorContas

enviarDados()

### C ControladorTurma

registrarProposta()

## View

### C TelaLogin

inserirDados()  
informarValidadez()

### C TelaInicial

verificarNotificações()  
acessarProjeto()  
notificarAluno()  
acessarTurma()  
criarProjeto()  
verProjeto()

### C TelaTurma

verPropostas()  
revisarProposta()  
escolherProjeto()  
proporProjeto()

### C TelaRevisao

enviarRevisao()

### C TelaBusca

informarTermoBusca()  
selecionarFiltros()  
confirmarBusca()  
informarResultados()

### C TelaEscolhaProjeto

informarMembros()  
selecionarProjeto()

### C TelaCadastro

inserirDados()

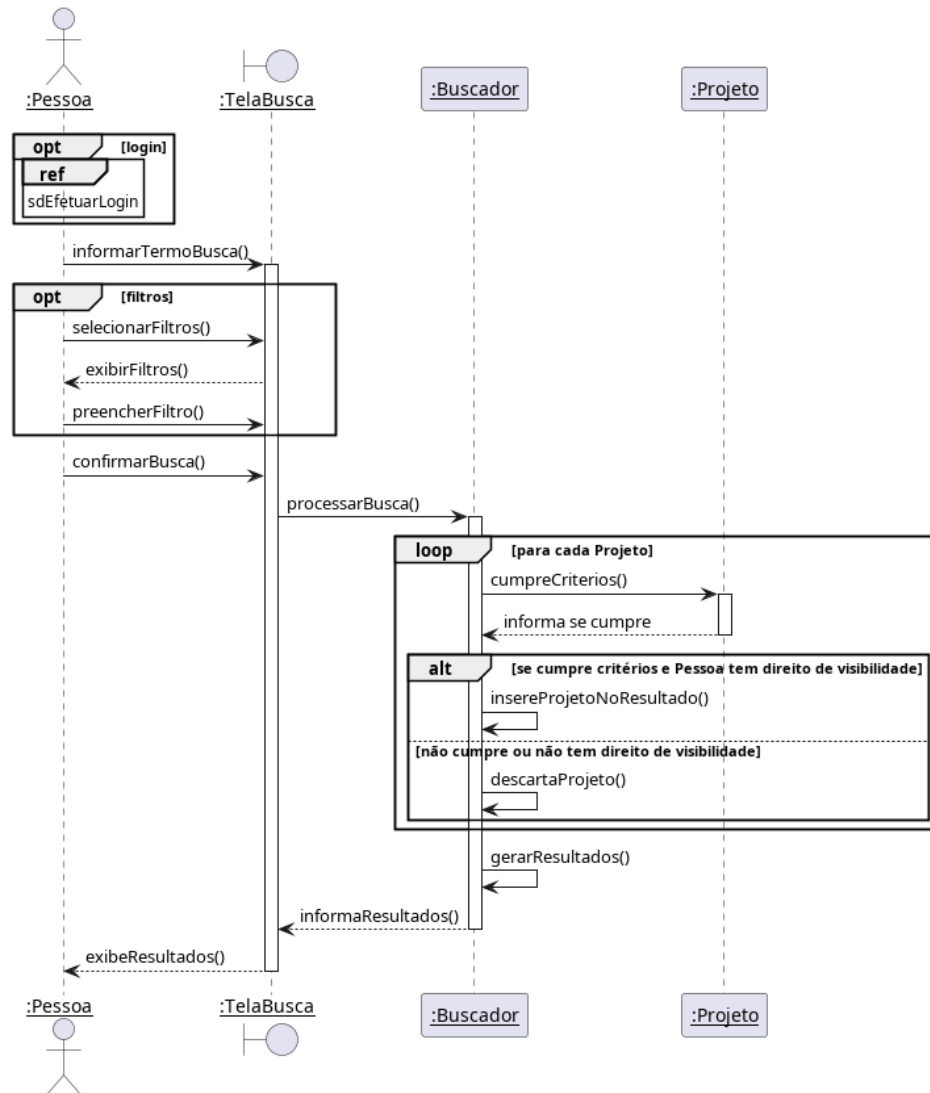
### C TelaProjeto

mudarProjeto()  
informarDados()

### C TelaProposta

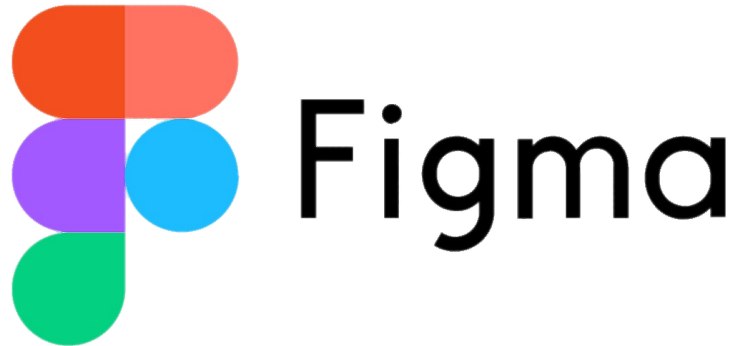
informarDados()

# Diagrama de Sequência: Consultando projetos



# Protótipo de Telas

- Criadas com o editor gráfico de vetor e prototipagem de projetos **Figma**



- Logo do software em todas as telas

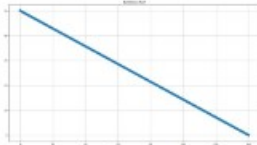
- Telas interativas

- Esquema de cores do DCOMP

Projetorium

# Planejamento Backlog do Produto

Quadro 2 – Backlog do Produto 1.

BACKLOG DO PRODUTO		BURNDOWN	
Produto:Projektorium	Release: 1		
Requisitos:	(+) prioridade ← REQUISITOS prioridade → (-)	Não Planejado	
RFMU01 - Manter Gestor		Impedimentos	
RFMU03 - Manter Aluno.			
RFMU04 - Manter Turma.			
RFPP05 - Filtro de Consultas.			

# Planejamento

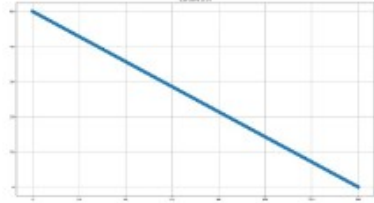
## Plano da Sprint

Plano da Sprint	
Meta da Sprint: Realizar todos os requisitos selecionados	
Duração: 4 semanas	Integrantes de Equipe: Matheus Victor, Pedro Vinícius, Max Antônio, Jonas Gabriel
Data de Início: 13/11/2023	
Data de Encerramento: 12/12/2023	
Data de Revisão/Demo: 12/12/2023	
Definição de Pronto: 14/12/2023	Requisitos Selecionados: <ul style="list-style-type: none"><li>– Manter Aluno</li><li>– Manter Professor</li><li>– Manter Gestor</li><li>– Manter Turma</li><li>– Proposição de Projeto</li><li>– Aprovação de Propostas</li></ul>
Local e Horário das Reuniões Diárias: <ul style="list-style-type: none"><li>– Discord</li><li>– As 21:00hrs</li></ul>	
ScrumMaster: Matheus Victor	
Dono do Produto: Adicinéia A. de Oliveira	

# Planejamento

## Detalhes da Sprint

Quadro 5 – Detalhes da Sprint 1.

Sprint: 1		Meta: Entrega de Sprint sem impedimentos		
BACKLOG	EM ANDAMENTO	CONCLUÍDO	BURNDOWN	
RFMU01 – Manter Gestor				
RFMU02 – Manter Professor.				
RFMU03 – Manter Aluno.				
RFMU04 – Manter Turma.				
			IMPEDIMENTOS	ITENS NÃO PLANEJADOS
RFPP01 – Proposição de Projeto.				
RFPP02 – Aprovação de Propostas				



# Conclusões

## Resultados alcançados

- Compreensão dos requisitos do cliente após entrevistas e modelagem de análise
- Prototipagem das telas e construção de diagramas de interação ajudaram a esclarecer pontos duvidosos
- Apresentar protótipos de telas executáveis/interativos ao cliente para extrair ainda mais feedback



## Pontos a melhorar

- Alinhar modelagem de classes e de sequência
- Gerenciamento de tempo



# Referências bibliográficas

- PRESSMAN, Roger S. & MAXIM, Bruce. Software Engineering: A Practitioner's Approach. 9ª Edição. Nova York: McGraw Hill, 2019.
- WINTERS, Titus, MANSHRECK, Tom & WRIGHT, Hyrum. Software Engineering at Google: Lessons Learned from Programming Over Time. Sebastopol: O'Reilly, 2020.