

---

UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE  
LABORATÓRIO DE ENGENHARIA E EXPLORAÇÃO DE PETRÓLEO  
CENTRO DE CIÊNCIA E TECNOLOGIA

PRÉ-PROJETO DE ENGENHARIA  
DESENVOLVIMENTO DO SOFTWARE

"PetroPlanner"

Github: `SimuladorDePlanejamentoDasAtividadesDoEstudanteEngenharia`

DISCIPLINA LEP - LEP01348 : Introdução ao Projeto de Engenharia

Setor de Modelagem Matemática e Computacional

Versão 1:

AUTORES

Abigail Ribeiro Pinto Silva Guimarães

Maria Luiza Cornelio Ramos

Prof. André Duarte Bueno

MACAÉ - RJ

Março - 2025

# Sumário

<b>1</b>	<b>Introdução</b>	<b>3</b>
1.1	Identificação da Proposta . . . . .	3
1.2	Resumo . . . . .	5
1.3	Escopo do Problema . . . . .	5
1.4	Objetivos . . . . .	6
1.5	Metodologia . . . . .	7
<b>2</b>	<b>Informações Adicionais Específicas</b>	<b>8</b>
2.1	Requisitos Funcionais . . . . .	8
2.2	Requisitos Não Funcionais . . . . .	8
2.3	Arquitetura do Software . . . . .	9
2.4	Design da Interface Gráfica . . . . .	10
2.5	Cálculos e Algoritmos . . . . .	12
2.6	Casos de Uso . . . . .	15
2.7	Tecnologias . . . . .	17
2.8	Referências . . . . .	17
<b>3</b>	<b>Etapas, Cronograma e Orçamento Preliminar</b>	<b>18</b>
3.1	Etapas . . . . .	18
3.2	Cronograma . . . . .	20
3.3	Orçamento Preliminar . . . . .	20

# Capítulo 1

## Introdução

Segundo o CREA/CONFEA um dos quesitos fundamentais que diferenciam a atuação de um técnico da atuação de um engenheiro é a capacidade de desenvolver um projeto de engenharia; Neste trabalho, apresenta-se a pré-proposta de um projeto em engenharia de software aplicado a solução de um problema específico da engenharia de petróleo.

Apresenta-se aqui a proposta de desenvolvimento do software SimuladorDePlanejamentoDasAtividadesDoEstudanteEngenharia, software "PetroPlanner".

- Um software aplicado que visa auxiliar estudantes do curso de engenharia de petróleo na organização da sua grade curricular, levando em consideração o nível de dificuldade das disciplinas obrigatórias, horas gastas semanais para o estudo de cada uma delas, baseado em pesquisas com ex-alunos, optativas a serem feitas, atividades complementares - AAC, atividades de extensão, estágio e TCC - Trabalho de Conclusão de Curso.
- O "PetroPlanner" será um software de apoio ao estudante de Engenharia de Petróleo, desenvolvido em C++23 com Qt 6.6 (ou superior), utilizando Git/GitHub para controle de versão, CMake para gerenciamento de compilação e bibliotecas utilitárias para gráficos, agenda e estatística. A metodologia de desenvolvimento será Scrum, com dois ciclos: ciclo 1 (funcionalidades principais sem interface gráfica) e ciclo 2 (implementação da interface gráfica).

### 1.1 Identificação da Proposta

**Tipo de investimento /divulgação:**

- PROJETO DE PESQUISA E DESENVOLVIMENTO / DESENVOLVIMENTO DE SOFTWARE  
- Versão 1

**Coordenador:**

- André Duarte Bueno

**Gerência técnica:**

- Nome Empresa/Instituição: Coordenação do Curso de Graduação em Engenharia de Petróleo
- Nome Gerente: André Duarte Bueno
- E-mail: bueno@lenep.uenf.br
- Telefone: 22-999542635

**Equipe:**

- Abigail Ribeiro Pinto Silva Guimarães [abigailsilva@lenep.uenf.br/22-997886502]
- Maria Luiza Cornelio Ramos [mariaramos@lenep.uenf.br/22-988356947]

**Vínculo institucional do processo:**

- UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE DARCY RIBEIRO/UENF
- CENTRO DE CIÊNCIA E TECNOLOGIA - CCT
- DEPARTAMENTO DE ENGENHARIA DE PETRÓLEO - LENEP
- SETOR DE MODELAGEM MATEMÁTICA E COMPUTACIONAL

**Duração:**

- 04 meses

**Data de submissão:**

- 17/03/2025

**Número da proposta:**

- LDSC-2025-1-P61
- Link no github: <https://github.com/ldsc/ProjetoEngenharia-SimuladorDePlanejamentoDasAtividadesDoEstudanteEngenharia>

**Tipo de instrumento contratual:**

- Trabalho de disciplina: Projeto de Software Aplicado à Engenharia

**Título do projeto:**

- “ProjetoEngenharia-SimuladorDePlanejamentoDasAtividadesDoEstudanteEngenharia”

**Palavras-chave:**

- Simulador de Engenharia
- Acompanhamento curricular
- Agenda do aluno
- Análise de desenvolvimento acadêmico
- Dados estatísticos do estudante

## 1.2 Resumo

Durante a graduação em Engenharia de Petróleo, muitos estudantes enfrentam dificuldades para organizar sua trajetória acadêmica de forma eficiente. Entre desafios como a escolha de disciplinas, carga horária semanal, estágio, atividades complementares e Trabalho de Conclusão de Curso (TCC), a falta de um planejamento estruturado pode impactar diretamente no desempenho acadêmico.

Diante dessa necessidade, este projeto propõe o desenvolvimento do PetroPlanner, um software interativo que auxilia estudantes no planejamento de sua grade curricular e na gestão do tempo dedicado aos estudos. A ferramenta permitirá a simulação de diferentes cenários acadêmicos, considerando fatores como taxa de aprovação das disciplinas, impacto do adiamento de matérias e carga horária semanal recomendada. Além disso, o software incluirá funcionalidades para acompanhamento de estágios, atividades complementares e progressão do coeficiente de rendimento (CR), oferecendo uma visão mais clara da jornada acadêmica do estudante.

O PetroPlanner será desenvolvido em C++23 com a interface gráfica baseada no Qt 6.6 (ou superior), utilizando SQLite para armazenamento de dados e bibliotecas específicas para gráficos e estatísticas. O diferencial desse projeto está na personalização do planejamento acadêmico para estudantes de engenharia, levando em conta a complexidade e carga horária das disciplinas, o impacto de reprovações e a necessidade de conciliar estudos com a vida pessoal e profissional. Com isso, espera-se que o PetroPlanner se torne uma ferramenta essencial para ajudar universitários a tomar decisões estratégicas ao longo do curso, reduzindo incertezas e contribuindo para uma trajetória acadêmica mais organizada e eficiente.

## 1.3 Escopo do Problema

O PetroPlanner é um software acadêmico projetado para auxiliar estudantes de Engenharia de Petróleo na organização eficiente da sua trajetória universitária. O sistema visa reduzir a insegurança na escolha de disciplinas, prevenir sobrecargas e melhorar a tomada de decisão com base em dados personalizados do aluno.

Diferentemente de sistemas genéricos de gestão acadêmica, o PetroPlanner oferece funcionalidades direcionadas à realidade de cursos de engenharia, como:

- Simulação de planejamento semestral com checagem automática de pré-requisitos;
- Estimativa de notas necessárias para aprovação com base em pesos e avaliações;
- Visualização de impacto do adiamento de disciplinas na grade curricular;
- Análise de carga horária semanal, considerando aulas e atividades extracurriculares;
- Distribuição equilibrada de disciplinas por dificuldade para evitar períodos sobrecarregados;
- Controle visual de faltas e acompanhamento de progresso acadêmico.

O escopo do sistema abrange, portanto, o planejamento, a simulação, o acompanhamento e a visualização do progresso curricular, oferecendo ao aluno autonomia e clareza em cada etapa de sua formação.

## 1.4 Objetivos

Os objetivos deste projeto de engenharia são:

- Objetivo geral:
  - Desenvolver um software interativo de apoio acadêmico para estudantes de Engenharia de Petróleo, com foco em planejamento de disciplinas, análise de desempenho e organização da trajetória curricular de forma eficiente, visual e personalizada.
- Objetivos específicos:
  - Representar matematicamente as regras e restrições curriculares do curso (pré-requisitos, carga horária, CR mínimo, etc.);
  - Desenvolver diagrama de casos de uso, diagrama de pacotes e diagrama de classes para estruturar o sistema em camadas lógicas;
  - Implementar algoritmos para: Simulação de CR final com base em notas parciais; Cálculo da nota mínima necessária para aprovação e Análise de impacto do adiamento de disciplinas;
  - Criar um módulo de planejamento que: Filtra automaticamente disciplinas elegíveis com base nos pré-requisitos e Auxilia na escolha estratégica de disciplinas para equilíbrio entre dificuldade e carga horária;
  - Desenvolver um módulo para controle de faltas, exibindo o número de faltas registradas e limite permitido;
  - Construir um quadro de horários dinâmico, baseado nas disciplinas em andamento e atividades extras;
  - Garantir o cálculo da carga horária semanal, alertando o aluno caso esteja abaixo do mínimo exigido (16h/semana);
  - Criar interface gráfica intuitiva e responsiva utilizando Qt Framework;
  - Documentar e estruturar um manual simplificado de uso do software, com foco didático.

## 1.5 Metodologia

A Figura 1.1 apresenta a metodologia a ser utilizada no desenvolvimento do sistema.

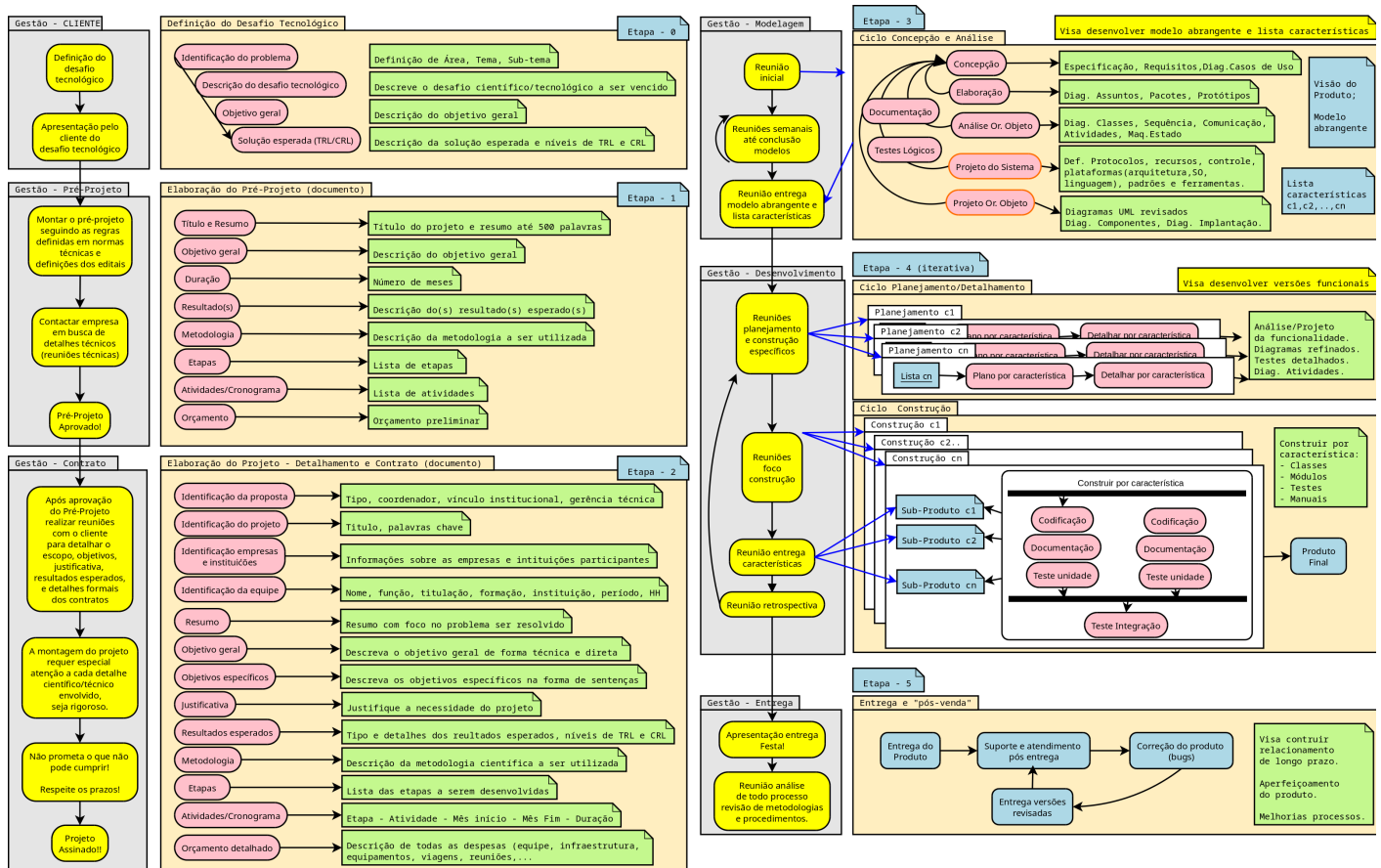


Figura 1.1: Metodologia utilizada no desenvolvimento do sistema

# Capítulo 2

## Informações Adicionais Específicas

### 2.1 Requisitos Funcionais

- RF01 – O sistema deve ler os dados do aluno a partir de um arquivo texto (InformacoesAluno.txt) contendo nome, matrícula, curso, período, CRA e histórico de disciplinas.
- RF02 – O sistema deve classificar as disciplinas em três categorias: Aprovadas, Em Curso e Não Cursadas.
- RF03 – O sistema deve permitir a visualização de uma grade curricular completa, destacando as disciplinas já cursadas ou em andamento.
- RF04 – O sistema deve possibilitar o planejamento de semestres futuros, respeitando os pré-requisitos entre disciplinas.
- RF05 – O sistema deve exibir um quadro de horários interativo, onde o aluno possa organizar suas disciplinas em andamento e atividades extracurriculares.
- RF06 – O sistema deve permitir o acompanhamento individual de disciplinas, exibindo faltas, dias e horários, trabalhos e provas.
- RF07 – O sistema deve alertar visualmente sobre o limite de faltas permitido em cada disciplina.
- RF08 – O sistema deve permitir que o aluno edite as disciplinas em curso, marcando quais está cursando no momento.

### 2.2 Requisitos Não Funcionais

- RNF01 – Usabilidade
- O sistema deve possuir uma interface gráfica intuitiva, amigável e acessível a usuários com conhecimentos básicos de informática, utilizando botões autoexplicativos, cores diferenciadas e organização clara por seções.



- RNF02 – Portabilidade
- O sistema deve ser compatível com sistemas operacionais que suportem aplicações Qt5 ou Qt6, como Windows e distribuições Linux.
- RNF03 – Persistência de dados
- Todas as informações acadêmicas do aluno (disciplinas cursadas, em andamento, atividades, faltas, horários) devem ser armazenadas e lidas a partir de um arquivo externo (InformacoesAluno.txt), garantindo persistência entre diferentes execuções do programa.
- RNF04 – Performance
- O sistema deve realizar a leitura do arquivo de entrada e o carregamento da interface principal em menos de 2 segundos, mesmo com uma grade curricular extensa.
- RNF05 – Manutenibilidade
- A estrutura do código-fonte deve estar organizada em módulos e classes (ex: CAluno, CDisciplinas, TelaInicial), facilitando a manutenção, testes e futura expansão do sistema.
- RNF06 – Visualização clara
- Os dados devem ser apresentados de forma visualmente compreensível, com destaque para disciplinas em andamento, alertas de faltas e agrupamento por período letivo ou por tipo de atividade.
- RNF07 – Confiabilidade
- O sistema deve garantir que informações críticas como pré-requisitos, horários e número de faltas não sejam perdidas ou corrompidas durante o uso, mesmo após alterações múltiplas feitas pelo usuário.
- RNF08 – Responsividade visual
- A interface deve se ajustar a diferentes tamanhos de tela, permitindo boa experiência de uso em monitores com diferentes resoluções.

## 2.3 Arquitetura do Software

- **Linguagem de Programação:** C++23
- **Interface Gráfica:** Qt 6.10.0
- **Banco de Dados:** SQLite
- **Controle de Versão:** Git/GitHub
- **Gerenciamento de Compilação:** CMake

- **Metodologia de Desenvolvimento:** Scrum (2 ciclos)
- **Módulos:**
  - Módulo de Grade Curricular
  - Módulo de Planejamento Semestral
  - Módulo de Planejamento de Estudos
  - Módulo de Atividades Complementares
  - Módulo de Estágios
  - Módulo de TCC
  - Módulo de Vida Pessoal

## 2.4 Design da Interface Gráfica

O sistema PetroPlanner foi desenvolvido com uma interface gráfica organizada em módulos interativos que buscam facilitar a experiência do usuário, organizando as informações acadêmicas de forma intuitiva e funcional. O layout foi construído usando Qt Framework, priorizando simplicidade, clareza e acessibilidade.

A seguir, são descritos os principais componentes da interface:

- **Tela Inicial:**
  - Exibe as informações gerais do aluno: nome, matrícula, curso, período atual e CRA.
  - Mostra a barra de progresso de carga horária semanal, com destaque para o cumprimento do mínimo exigido (16 horas).
  - Lista as disciplinas em andamento, organizadas em botões clicáveis.
  - Apresenta o menu principal com acesso aos demais módulos: Grade Curricular, Quadro de Horários, Simulador de Períodos e Acompanhamento de Disciplinas.
  - Ao lado de cada disciplina, mostra indicativos de faltas.
  - O fundo azul foi escolhido para facilitar a leitura e dar identidade ao sistema.
- **Módulo de Horários:**
  - Exibe todas as disciplinas do curso, organizadas por período.
  - Cada botão representa uma disciplina, com cor que indica a área temática (eixo curricular).
  - O sistema apresenta três abas principais:
    - Grade Geral: mostra todas as disciplinas previstas no curso.
    - Progresso: usa cores para indicar o status da disciplina (verde = aprovada, amarelo = em curso, vermelho = reprovada, cinza = não cursada).

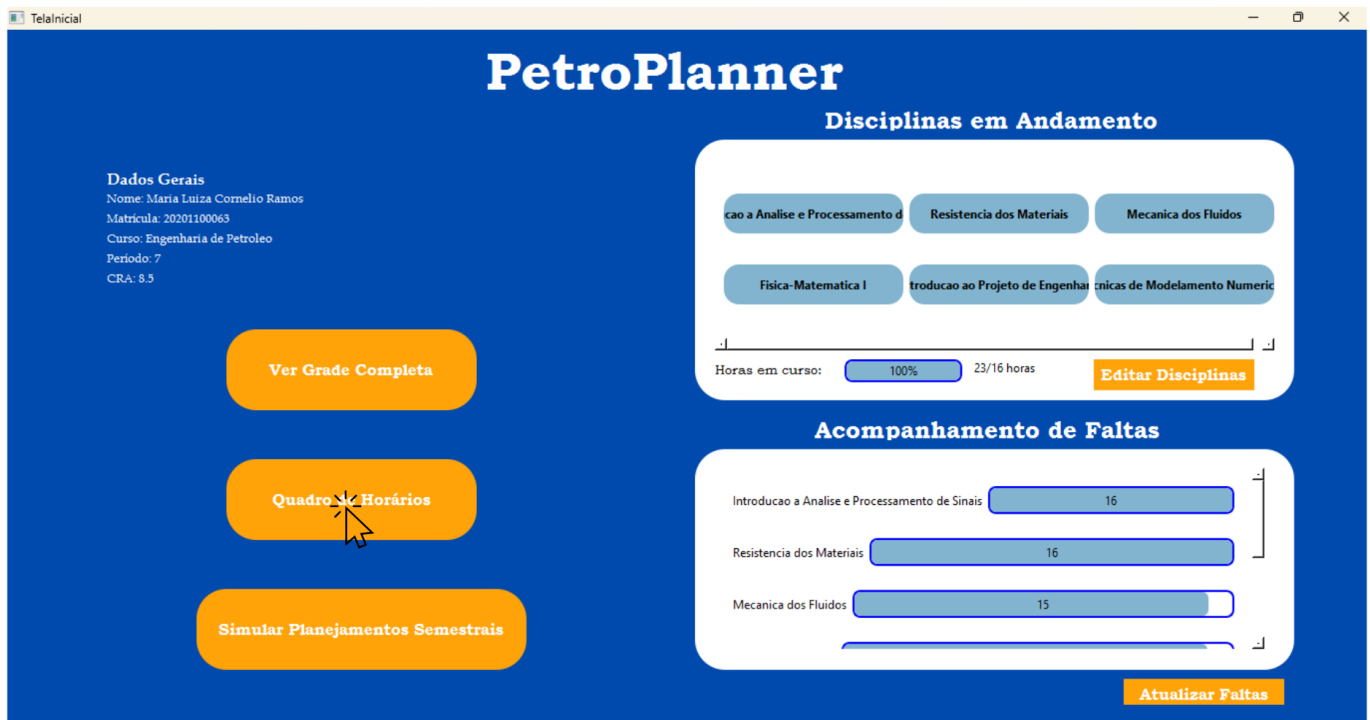


Figura 2.1: Tela inicial da interface gráfica.

- Dificuldade: usa um gradiente de cores para indicar o nível de dificuldade da disciplina, conforme informado em formulário por estudantes.
- Uma barra de progresso inferior mostra o percentual de carga horária cursada em relação ao total do curso.

#### • Módulo de Acompanhamento de Disciplinas:

- Permite abrir uma janela específica para cada disciplina.
- Nela, o aluno pode:
- Inserir notas e pesos de trabalhos e provas.
- Marcar avaliações como simuladas, e o sistema calcula automaticamente a nota necessária para aprovação.
- Visualizar a média final estimada, com coloração automática (verde =  $\geq 60$ , vermelho =  $< 60$ , cinza = simulado).
- Alternar entre modo de edição e modo de visualização.
- As alterações são salvas automaticamente no arquivo de dados do aluno.

#### • Módulo de Simulação de Planejamento Semestral

- Simula o avanço do aluno nos próximos períodos.
- Lista as disciplinas elegíveis para o próximo semestre, de acordo com os pré-requisitos já cumpridos.



Figura 2.2: Módulo de grade curricular.

- Permite ao aluno selecionar as disciplinas desejadas e visualizar sua distribuição por semestre.
- Exibe os nomes das disciplinas escolhidas organizadas por período simulado.
- Cada nova simulação é adicionada como um novo "bloco visual" no layout.

## 2.5 Cálculos e Algoritmos

O sistema PetroPlanner integra diversos algoritmos e cálculos automatizados para apoiar o aluno na organização da sua jornada acadêmica. Essas funcionalidades foram projetadas para oferecer não apenas dados, mas informações processadas e relevantes, facilitando a tomada de decisão baseada em evidências.

A seguir, são descritos os principais cálculos utilizados no sistema:

### Taxa de Aprovação por Área

- O sistema organiza as disciplinas por área temática (como Química, Física, Cálculo, Computação, etc.) com base no atributo `eixoTematico`.
- É possível, futuramente, calcular a porcentagem de aprovação em cada área, com base nos dados históricos do aluno e da turma.
- A visualização pode ser feita por meio de gráficos comparativos, gerados com a biblioteca `QCustomPlot` (biblioteca gráfica compatível com Qt, utilizada para visualizações estatísticas).

### Nível de Dificuldade por Disciplina

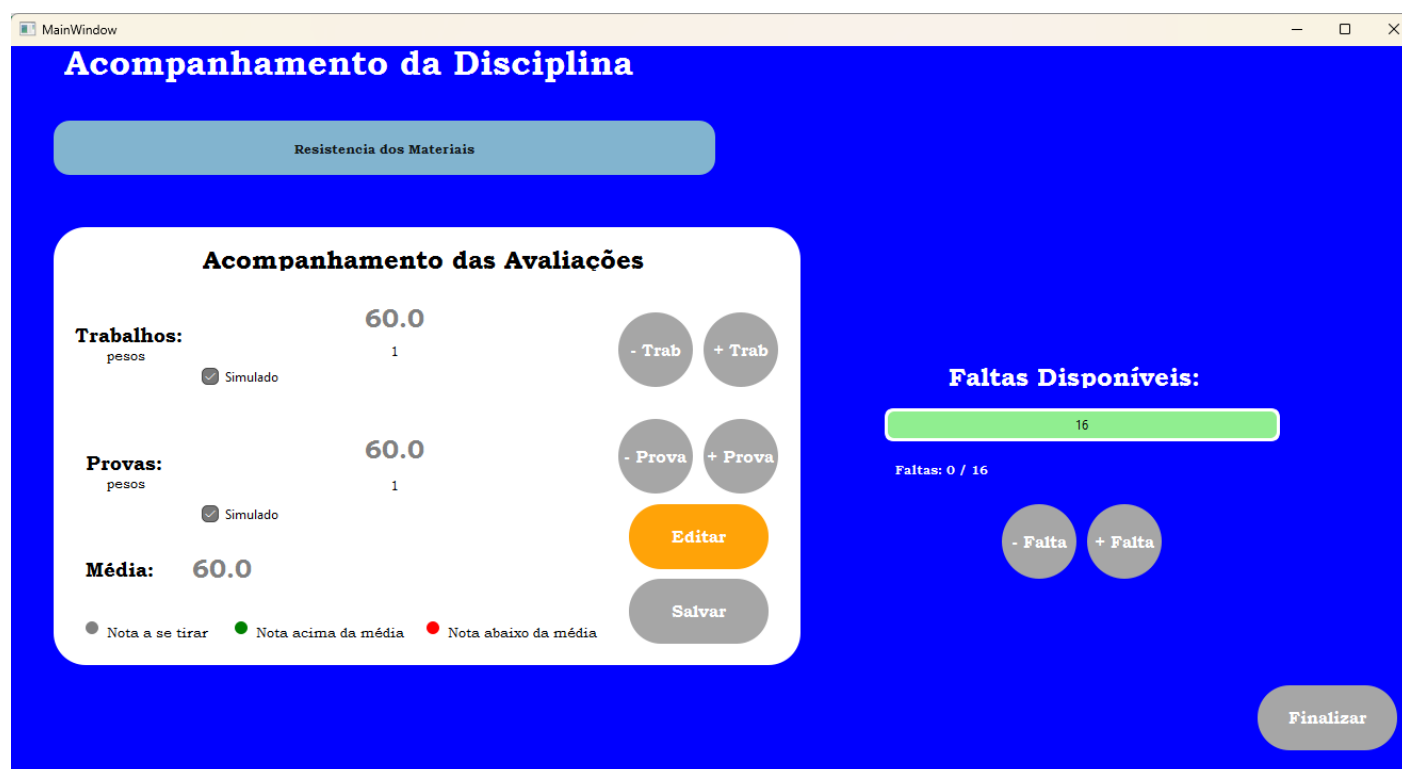


Figura 2.3: Módulo de planejamento semestral.

- Cada disciplina possui um nível de dificuldade definido numericamente (de 1 a 5), com base em uma pesquisa feita com alunos.
- Esse nível é representado visualmente por cores no módulo "Grade por Dificuldade", auxiliando o aluno a equilibrar sua carga de disciplinas por semestre.
- O algoritmo realiza uma classificação qualitativa:
- Nível 1–2: Fácil (Verde)
- Nível 3: Médio (Amarelo)
- Nível 4–5: Difícil (Laranja a Vermelho)
- Os valores também podem ser atualizados com base em estatísticas de:
- Notas médias dos alunos
- Taxa de reprovação
- Carga horária semanal

### Ajuste Automático de Disciplinas

- O módulo de simulação de planejamento semestral realiza a verificação de pré-requisitos automaticamente.



Figura 2.4: Criação de cenários.

- O algoritmo filtra e exibe apenas as disciplinas que podem ser cursadas, de acordo com o histórico do aluno.
- É possível selecionar disciplinas para cada semestre futuro, priorizando:
  - Disciplinas com pré-requisitos já cumpridos
  - Disciplinas obrigatórias
  - Distribuição equilibrada entre fáceis e difíceis
- Esse algoritmo funciona de forma reativa: ao remover uma disciplina, os semestres seguintes são reatualizados automaticamente.

### Ajuste Dinâmico do Tempo de Estudo

- O sistema permite o cálculo da carga horária semanal real (disciplinas + atividades extracurriculares).
- O quadro de horários é preenchido automaticamente com base em:
  - Horários das disciplinas em andamento
  - Atividades extras cadastradas (iniciação científica, estágio, estudos, etc.)

- Com base na quantidade de disciplinas difíceis e no tempo disponível, o aluno pode reorganizar melhor seus blocos de estudo.
- Essa visualização evita sobrecarga ou sobreposição de horários.

### Geração de Gráficos Interativos

- Para futura integração, o sistema pode utilizar a biblioteca QCustomPlot para:
- Mostrar gráficos de CR por semestre
- Apresentar a evolução da carga horária
- Comparar áreas temáticas com maior reprovação
- Essa funcionalidade será útil em uma versão avançada para análise mais visual dos dados.

## 2.6 Casos de Uso

O sistema PetroPlanner foi projetado para atender situações acadêmicas reais enfrentadas por estudantes de Engenharia de Petróleo. A seguir, estão descritos alguns casos de uso representativos, que demonstram como o sistema pode ser utilizado para apoiar o aluno em decisões importantes ao longo da graduação.

### Planejar o Semestre com Restrições Pessoais

**Cenário:** Um aluno com restrições de tempo (ex: estágio, IC, extensão ou questões pessoais) deseja montar uma grade sem se sobrecarregar.

#### Fluxo:

- O aluno acessa o módulo de simulação de planejamento semestral.
- O sistema exibe apenas as disciplinas elegíveis, verificando automaticamente os pré-requisitos cumpridos.
- O aluno seleciona as disciplinas desejadas e visualiza quais seriam cursadas em cada período futuro.
- O sistema sinaliza sobrecargas com base em carga horária total e dificuldade.
- O aluno pode testar diferentes combinações até encontrar um cenário equilibrado.

Resultado: Planejamento personalizado e estratégico, com visualização antecipada do impacto de cada decisão.

### Criar Quadro de Horários com Tempo de Estudo Ajustado

**Cenário:** O aluno quer organizar suas semanas considerando aulas, estudos e atividades extracurriculares.

#### Fluxo:

- O sistema carrega automaticamente as disciplinas em andamento e suas respectivas faixas de horário.
- O aluno pode adicionar atividades extras como IC, estágio ou horário de estudos.
- O sistema gera um quadro de horários completo, com cores distintas para cada atividade.
- O aluno pode editar os horários manualmente e salvar as alterações.
- O sistema exibe alertas visuais em caso de conflito de horários ou carga horária insuficiente.

Resultado: Organização visual da rotina semanal, otimizando o tempo de estudo.

### **Acompanhar Taxa de Aprovação e Desempenho Acadêmico**

**Cenário:** O aluno quer acompanhar seu desempenho, CR, notas e médias por disciplina ao longo do curso.

#### **Fluxo:**

- O aluno acessa o módulo de acompanhamento de disciplinas.
- Ao clicar em uma disciplina, uma janela se abre com campos para inserir notas de provas e trabalhos, com pesos.
- O sistema calcula automaticamente a média final estimada e indica a nota mínima necessária para aprovação.
- A média é destacada com cores (verde, vermelho ou cinza) para facilitar a leitura.
- Futuramente, o sistema pode apresentar gráficos da evolução do CR ou desempenho por área.

Resultado: Monitoramento preciso do progresso acadêmico, com base em simulações e previsões.

### **Visualizar Agenda Semanal de Atividades**

**Cenário:** O aluno quer ter uma visão clara de todos os seus compromissos acadêmicos semanais.

#### **Fluxo:**

- O aluno acessa o módulo de quadro de horários.
- Todas as aulas, trabalhos e atividades extras são organizadas automaticamente por dia e hora.
- As células do calendário são preenchidas com os nomes das atividades, usando cores distintas.
- O aluno pode alternar entre modo de visualização e modo de edição.

Resultado: Agenda semanal organizada, com destaque para horários ocupados e espaços livres.



## 2.7 Tecnologias

- C++23
- Qt 6.10.0
- SQLite
- Git/GitHub
- CMake

## 2.8 Referências

- UML: [Blaha and Rumbaugh, 2006, Rumbaugh et al., 1994].
  - Projetos: [autores, 2017, Inc, 2021, Pires, 2012, Woiler, 1996].
  - Gestão de Projetos: [Abrantes, 2020, de Logística e Tecnologia da Informação, 2011, Heldman, 2000, de Moura Menezes, 2018, Pahl, 2005, Valeriano, 2015, Rosa, 2007]
  - Produtos: [Abrantes, 2020].
  - C++: [Bueno, 2003]

# Capítulo 3

## Etapas, Cronograma e Orçamento Preliminar

Este capítulo apresenta as etapas, o cronograma de execução e uma previsão de orçamento preliminar para o desenvolvimento do sistema PetroPlanner, um software de apoio acadêmico para estudantes de engenharia.

### 3.1 Etapas

Etapa 0 – Definição do desafio tecnológico

- Identificação do problema enfrentado por estudantes na organização de suas disciplinas, faltas, tempo de estudo e planejamento semestral.
- Definição da solução esperada: um software com simulações e ferramentas visuais que facilitassem a tomada de decisões acadêmicas.
- Estimativa do grau de maturidade tecnológica (TRL): TRL 2 ao iniciar o projeto (conceito formulado).

Etapa 1 – Elaboração do pré-projeto

- Produção de uma versão inicial do documento, contendo:
- Definição do escopo
- Levantamento de requisitos
- Estudo de viabilidade técnica
- Modelos iniciais e estrutura básica da interface.

Etapa 2 – Elaboração do projeto (detalhamento e contrato)

- Expansão do pré-projeto com:
- Definição das funcionalidades completas

- Detalhamento das classes (ex: CAluno, CDisciplinas, ModuloGradeCompleta)
- Estrutura de dados e lógica de simulação de CR, faltas e tempo de estudo.
- Contrato informal entre os membros da equipe, dividindo responsabilidades de backend, frontend e documentação.

#### Etapa 3 – Modelagem de engenharia

- Modelagem orientada a objeto com os seguintes diagramas:
- Casos de uso
- Diagrama de pacotes
- Diagrama de classes
- Criação da estrutura modular do software com separação em camadas de lógica, dados e interface.
- Validação dos diagramas por meio de testes unitários em código.

#### Etapa 4 – Ciclos de desenvolvimento e testes

- O projeto foi dividido em ciclos incrementais:
- v0.1 – Núcleo de dados e leitura de arquivos do aluno
- v0.3 – Integração com interface gráfica inicial (Qt)
- v0.7 – Simulador de planejamento, acompanhamento de disciplinas, quadro de horários.
- A cada versão, foram realizados:
- Testes unitários
- Testes manuais da interface
- Validação entre módulos (integração)

#### Etapa 5 – Entrega final

- Testes finais de uso com diferentes perfis de aluno.
- Geração dos manuais de uso.
- Entrega do executável e dos arquivos-fonte com instruções de compilação no GitHub.okk

## 3.2 Cronograma

Mês	Atividades
Mês 1	Definição do desafio, levantamento de requisitos, protótipos de interface
Mês 2	Implementação das classes e leitura de arquivos; testes de leitura e escrita
Mês 3	Desenvolvimento da interface gráfica e integração entre módulos
Mês 4	Implementação dos simuladores, testes finais e entrega

## 3.3 Orçamento Preliminar

O projeto foi desenvolvido utilizando ferramentas gratuitas e de código aberto, com baixo custo financeiro direto, porém com alta demanda de horas de trabalho da equipe. A seguir, apresenta-se uma estimativa simplificada:

Recurso	Custo Estimado
Ferramentas (Qt, PlantUML, GitHub)	Gratuito
Computadores e ambiente de desenvolvimento	Próprio
Horas de desenvolvimento (voluntárias)	Não monetizadas
Estimativa de esforço total	~200 horas



# Referências Bibliográficas

- [Abrantes, 2020] Abrantes, J. (2020). *Projeto e Engenharia de Produtos*. Ciencia Moderna. ISBN-13 : 978-8539910847. 17
- [autores, 2017] autores, V. (2017). *Projetos de engenharia - uma introdução*. LTC. ISBN-13 : 978-8521634454. 17
- [Blaha and Rumbaugh, 2006] Blaha, M. and Rumbaugh, J. (2006). *Modelagem e Projetos Baseados em Objetos com UML 2*. Campus, Rio de Janeiro. 17
- [Bueno, 2003] Bueno, A. D. (2003). *Programação Orientada a Objeto com C++ - Aprenda a Programar em Ambiente Multiplataforma com Software Livre*. Novatec, São Paulo, 1 edition. 17
- [de Logística e Tecnologia da Informação, 2011] de Logística e Tecnologia da Informação, S. (2011). *Fundamentos em Gestão de Projetos - Construindo Competências para Gerenciar Projetos BRASIL*. Ministério do Planejamento, Orçamento e Gestão (MPOG). 17
- [de Moura Menezes, 2018] de Moura Menezes, L. C. (2018). *Gestão de Projetos*. Atlas. 17
- [Heldman, 2005] Heldman, K. (2005). *Gerência de projetos*. Elsevier. ISBN 13 : 978-8535216844, Rio de Janeiro. 17
- [Inc, 2021] Inc, P. M. I. (2021). *A Guide to the Project Management Body of Knowledge and the Standard for Project Management*. PMI Project Management Institute. ISBN 13: 978-1628256642. 17
- [Pahl, 2005] Pahl, G. (2005). *Projeto na Engenharia: Fundamentos do Desenvolvimento Eficaz de Produtos - Métodos e Aplicações*. Blucher. ISBN-13: 978-8521203636. 17
- [Pires, 2012] Pires, A. M. S. (2012). *Projeto de Instalações Elétricas e Telecomunicações*. Instituto Superior de Engenharia de Coimbra. 17
- [Rosa, 2007] Rosa, M. O. (2007). *Gerenciamento de projetos de governo*. PMI-DF - PMInforma. 17
- [Rumbaugh et al., 1994] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W. (1994). *Modelagem e Projetos Baseados em Objetos*. Edit. Campus, Rio de Janeiro. 17
- [Valeriano, 2015] Valeriano, D. (2015). *Moderno Gerenciamento de Projetos*. Pearson. 17
- [Woiler, 1996] Woiler, S. (1996). *Projetos: planejamento, elaboração, análise*. Atlas. 17

# Índice Remissivo