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

PROJETO DE ENGENHARIA
DESENVOLVIMENTO DO SOFTWARE
“PetroPlanner”

DISCIPLINA LEP - 01348: Introdução ao Projeto de Engenharia
Setor de Modelagem Matemática Computacional

Versão 2:
Abigail Ribeiro Pinto Silva Guimarães

Maria Luiza Cornélio Ramos
Prof. André Duarte Bueno

MACAÉ - RJ
Abril - 2025

Sumário

1	Introdução	1
1.1	Identificação da Proposta	1
1.2	Identificação do Projeto	2
1.3	Identificação da Universidade, Instituições e Empresas Participantes	2
1.3.1	Universidade	2
1.3.2	Equipe	2
1.4	Resumo	3
1.5	Escopo do Problema	3
1.6	Objetivos	4
1.7	Justificativas	5
1.8	Resultados Esperados	5
1.9	Metodologia	6
1.10	Mecanismos de Acompanhamento da Execução	8
1.11	Referências	9
2	Etapas, Cronograma e Orçamento	10
2.1	Etapas	10
2.2	Cronograma	12
2.3	Orçamento	12

Capítulo 1

Introdução

Segundo o CREA/CONFEA, uma das principais competências que distinguem o engenheiro do tecnólogo é a capacidade de desenvolver projetos de engenharia. Neste contexto, apresenta-se a proposta de um projeto de engenharia de software aplicado à solução de um problema específico da Engenharia de Petróleo.

Este projeto propõe o desenvolvimento do PetroPlanner, um software interativo e personalizável voltado para o planejamento da vida acadêmica de estudantes do curso de Engenharia de Petróleo. A ferramenta visa facilitar a organização da grade curricular, considerando fatores como a dificuldade das disciplinas obrigatórias, tempo de estudo semanal recomendado, atividades complementares, extensão, estágio e TCC.

Com base em entrevistas e pesquisas com ex-alunos, o PetroPlanner busca simular diferentes trajetórias acadêmicas, prever impactos de reprovações e adiamentos de matérias, e sugerir estratégias mais eficientes de combinação de disciplinas. A proposta é que esse planejamento contribua para a melhoria do desempenho acadêmico dos estudantes, reduzindo incertezas e aumentando a eficiência na gestão do tempo.

O software será desenvolvido em C++²³ com Qt 6.6 (ou superior), utilizando SQLite para persistência dos dados e bibliotecas gráficas e estatísticas para visualizações. O gerenciamento do projeto será feito com GitHub, CMake, e a metodologia de desenvolvimento será o Scrum, com dois ciclos principais: um ciclo inicial sem interface gráfica (backend e lógica) e outro com a implementação da interface gráfica e usabilidade.

1.1 Identificação da Proposta

Número da proposta

- LDSC-2023-1-P50

Tipo de investimento /divulgação

- PROJETO DE PESQUISA E DESENVOLVIMENTO / DESENVOLVIMENTO DE SOFTWARE
- Versão 2

Tipo de instrumento contratual

- Trabalho de disciplina

1.2 Identificação do Projeto

Título do projeto

- “ProjetoEngenharia-SimuladorDePlanejamentoDasAtividadesDoEstudanteEngenharia”

Palavras-chave

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

1.3 Identificação da Universidade, Instituições e Empresas Participantes

1.3.1 Universidade

- UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE DARCY RIBEIRO/UENF
- CENTRO DE CIÊNCIA E TECNOLOGIA - CCT
- DEPARTAMENTO DE ENGENHARIA DE PETRÓLEO - LENEP
- SETOR DE MODELAGEM MATEMÁTICA COMPUTACIONAL
- Representante pela universidade:
 - Professor(a): André Duarte Bueno/bueno@lenep.uenf.br/22-999542635

1.3.2 Equipe

- Abigail Ribeiro Pinto Silva Guimarães/abigailsilva@lenep.uenf.br/22-997886502.
- Maria Luiza Cornelio Ramos/mariaramos@lenep.uenf.br/22-988356947.
- Representante pela equipe:
 - Estudante: Abigail Ribeiro Pinto Silva Guimarães/abigailsilva@lenep.uenf.br/22-997886502.
 - Maria Luiza Cornelio Ramos/mariaramos@lenep.uenf.br/22-988356947.

1.4 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 6.10.0, 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.

1.5 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.

1.6 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.

1.7 Justificativas

O PetroPlanner foi idealizado para responder à crescente demanda de apoio inteligente ao planejamento acadêmico por parte de estudantes de Engenharia de Petróleo, uma área caracterizada por uma matriz curricular extensa e interdependente, com forte carga horária e diversas exigências extracurriculares (como extensão, IC, estágios e TCC).

Atualmente, muitos alunos utilizam métodos manuais (planilhas, cadernos, blocos de notas) ou aplicativos genéricos, que não contemplam a complexidade de pré-requisitos, a distribuição semanal de carga horária, nem a dinâmica de progressão acadêmica exigida pela formação em engenharia.

O PetroPlanner surge como uma solução de engenharia de software especializada, integrando:

- Simuladores de trajetória acadêmica, considerando atrasos, reprovações e cenários alternativos;
- Cálculo de carga horária semanal;
- Estimativa do CRA futuro, com sugestões de notas-alvo;
- Monitoramento de faltas e atividades complementares.

Além de ser didaticamente útil para estudantes, o sistema tem aplicação potencial para coordenadores e núcleos pedagógicos de cursos de engenharia, como ferramenta de visualização e acompanhamento.

A metodologia utilizada combina:

- Modelagem matemática e simulação computacional;
- Programação orientada a objetos (C++ com Qt);
- Engenharia de requisitos e arquitetura em camadas.

Essa abordagem é transferível para diversas áreas da engenharia, como:

- Simulação de sistemas térmicos,
- Planejamento de projetos logísticos,
- Gerenciamento de recursos e produção.

1.8 Resultados Esperados

O projeto espera entregar:

- Um simulador completo e funcional, com interface gráfica desenvolvida em C++/Qt, capaz de prever e planejar a trajetória acadêmica de alunos da Engenharia de Petróleo.
- Manuais de usuário e desenvolvedor, com detalhamento técnico da arquitetura, algoritmos implementados e fluxos de uso.

- Sistema testado com dados reais, por meio da utilização prática de alunos voluntários e análise de resultados em situações reais e simuladas.
- Código-fonte documentado e publicado em repositório GitHub, com guia de instalação, execução e compilação.
- Apresentação pública do projeto, demonstrando a interface, os cálculos internos e os diferenciais em relação a ferramentas existentes.

1.9 Metodologia

Modelagem Matemática

- O problema foi estruturado como um sistema sujeito a restrições, como pré-requisitos, carga mínima semanal e exigências complementares (extensão, TCC, estágio etc.).
- Foram definidos parâmetros de entrada, regras de validação e limitações operacionais.

Prototipagem e MVPs

- Cada versão liberada (v0.1, v0.3, v0.7...) corresponde a um produto mínimo viável (MVP), permitindo entregas parciais e incrementais.
- A arquitetura do sistema foi pensada em módulos independentes: grade curricular, acompanhamento de disciplinas, simulador de planejamento, quadro de horários, entre outros.

Programação Orientada a Objetos

- Implementada em C++ utilizando o framework Qt para interface gráfica.
- A estrutura modular utilizou classes como CAluno, CDisciplinas, TelaInicial, ModuloGradeCompleta, entre outras.

UML e Engenharia de Requisitos

Foram elaborados diagramas UML para suportar a estruturação e organização do software:

- Casos de uso
- Pacotes e camadas
- Classes e relações

Simulação Computacional

Simulação de diferentes trajetórias acadêmicas com algoritmos que:

- Avaliam impacto de adiamento de disciplinas,
- Calculam CR projetado,

- Sugere combinações eficientes de matérias por semestre.

Validação com Usuários Reais

- Foram realizados testes com estudantes reais utilizando seus próprios históricos acadêmicos ou perfis simulados.
- Os resultados foram analisados qualitativamente, com feedback direto dos usuários.

Documentação Técnica

Toda a construção, testes, erros e melhorias foram documentados.

Foram gerados:

- Diagramas,
- Prints de interface,
- Manual técnico,
- Manual do usuário.
- A Figura 1.1 apresenta a metodologia a ser utilizada no desenvolvimento do sistema.

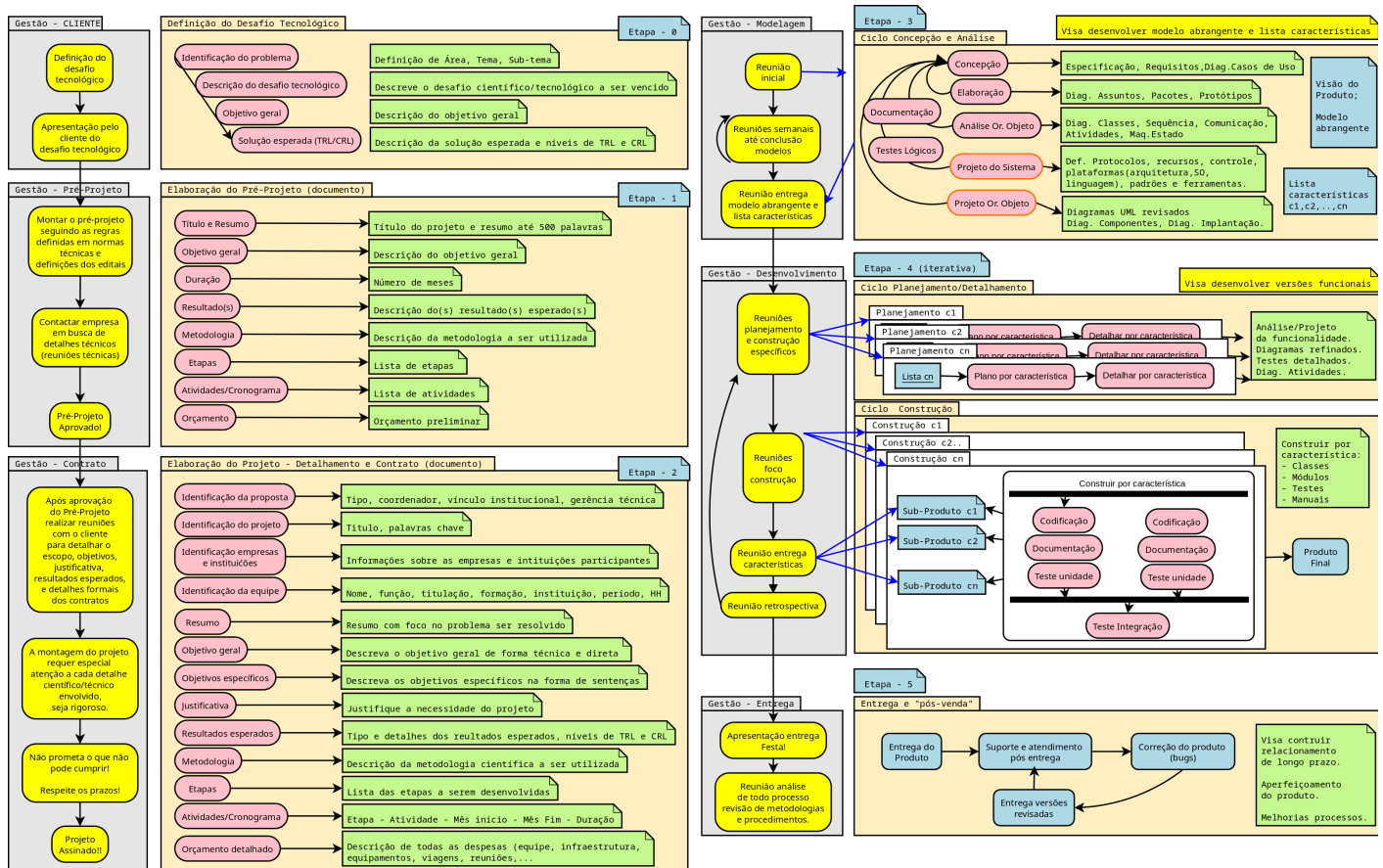


Figura 1.1: Metodologia utilizada no desenvolvimento do sistema

1.10 Mecanismos de Acompanhamento da Execução

A execução do projeto PetroPlanner foi estruturada com base em práticas da metodologia ágil SCRUM, adaptada ao contexto acadêmico e à carga horária dos integrantes da equipe. A seguir, detalham-se os mecanismos utilizados para monitorar o progresso, garantir entregas parciais e promover melhoria contínua ao longo do ciclo de desenvolvimento:

Ciclos de Desenvolvimento (Sprints)

- O desenvolvimento foi organizado em sprints curtos de 2 a 4 semanas, cada um com foco na implementação de um conjunto específico de funcionalidades (features).
- Cada sprint terminava com a liberação de um produto viável mínimo (MVP), testado e revisado pela equipe e por usuários convidados.

Gestão de Tarefas e Backlog

Foi utilizado um quadro de tarefas no GitHub Projects (ou Trello, quando necessário) para:

- Organizar as atividades em andamento, concluídas e pendentes;
- Visualizar as prioridades;
- Acompanhar bugs, testes e documentação.
- O backlog foi alimentado com sugestões dos próprios usuários e professores ao longo do projeto.

Reuniões e Feedback dos Usuários

Foram realizadas reuniões mensais com os "clientes" do sistema:

- Alunos reais da engenharia de petróleo (usuários finais);
- Professores e orientadores que acompanharam o projeto.
- Nessas sessões, eram coletados feedbacks parciais, validando o funcionamento das telas, fluxos e utilidade prática das funcionalidades implementadas.

Relatórios Semanais de Progresso

A equipe produziu relatórios semanais curtos, contendo:

- Etapas concluídas;
- Principais dificuldades enfrentadas;
- Estratégias e decisões tomadas para contornar problemas técnicos ou de planejamento;
- Tarefas previstas para a sprint seguinte.

Avaliação Final

- Ao final do semestre letivo, o sistema foi apresentado em uma sessão oral aberta, com demonstração em tempo real e espaço para questionamentos e sugestões.

A versão final do projeto incluiu:

- Código-fonte documentado;
- Interface funcional;
- Testes com usuários reais;
- Manual técnico e manual de uso.

1.11 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, 2005, de Moura Menezes, 2018, Pahl, 2005, Valeriano, 2015, Rosa, 2007]
- Produtos: [Abrantes, 2020].
- C++: [Bueno, 2003]

Capítulo 2

Etapas, Cronograma e Orçamento

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.

2.1 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

2.2 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

2.3 Orçamento

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

Etapas-Padrões/Programadas

Referências Bibliográficas

- [Abrantes, 2020] Abrantes, J. (2020). *Projeto e Engenharia de Produtos*. Ciencia Moderna. ISBN-13 : 978-8539910847. 9
- [autores, 2017] autores, V. (2017). *Projetos de engenharia - uma introdução*. LTC. ISBN-13 : 978-8521634454. 9
- [Blaha and Rumbaugh, 2006] Blaha, M. and Rumbaugh, J. (2006). *Modelagem e Projetos Baseados em Objetos com UML 2*. Campus, Rio de Janeiro. 9
- [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. 9
- [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). 9
- [de Moura Menezes, 2018] de Moura Menezes, L. C. (2018). *Gestão de Projetos*. Atlas. 9
- [Heldman, 2005] Heldman, K. (2005). *Gerência de projetos*. Elsevier. ISBN 13 : 978-8535216844, Rio de Janeiro. 9
- [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. 9
- [Pahl, 2005] Pahl, G. (2005). *Projeto na Engenharia: Fundamentos do Desenvolvimento Eficaz de Produtos - Métodos e Aplicações*. Blucher. ISBN-13: 978-8521203636. 9
- [Pires, 2012] Pires, A. M. S. (2012). *Projeto de Instalações Elétricas e Telecomunicações*. Instituto Superior de Engenharia de Coimbra. 9
- [Rosa, 2007] Rosa, M. O. (2007). *Gerenciamento de projetos de governo*. PMI-DF - PMInforma. 9
- [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. 9
- [Valeriano, 2015] Valeriano, D. (2015). *Moderno Gerenciamento de Projetos*. Pearson. 9
- [Woiler, 1996] Woiler, S. (1996). *Projetos: planejamento, elaboração, análise*. Atlas. 9

Índice Remissivo