

Introdução a Técnicas de Programação (2025.2)

Plano de Curso e Orientações para o Projeto

Prof. Fernando Figueira

DIMAp - UFRN

22 de Agosto de 2025

Agenda da Aula

1 Informações Gerais do Curso

Agenda da Aula

- 1 Informações Gerais do Curso
- 2 Metodologia e Avaliação

Agenda da Aula

- 1 Informações Gerais do Curso
- 2 Metodologia e Avaliação
- 3 Estrutura do Projeto

Agenda da Aula

- 1 Informações Gerais do Curso
- 2 Metodologia e Avaliação
- 3 Estrutura do Projeto
- 4 Critérios de Avaliação

Agenda da Aula

- 1 Informações Gerais do Curso
- 2 Metodologia e Avaliação
- 3 Estrutura do Projeto
- 4 Critérios de Avaliação
- 5 Sugestões de Projetos

Agenda da Aula

- 1 Informações Gerais do Curso
- 2 Metodologia e Avaliação
- 3 Estrutura do Projeto
- 4 Critérios de Avaliação
- 5 Sugestões de Projetos
- 6 Cronograma e Datas

Agenda da Aula

- 1 Informações Gerais do Curso
- 2 Metodologia e Avaliação
- 3 Estrutura do Projeto
- 4 Critérios de Avaliação
- 5 Sugestões de Projetos
- 6 Cronograma e Datas
- 7 Motivação e Expectativas

Agenda da Aula

- 1 Informações Gerais do Curso
- 2 Metodologia e Avaliação
- 3 Estrutura do Projeto
- 4 Critérios de Avaliação
- 5 Sugestões de Projetos
- 6 Cronograma e Datas
- 7 Motivação e Expectativas
- 8 Próximos Passos

Período Letivo

- **Duração:** 22/08/2025 a 15/12/2025
- **Encontros síncronos:** Segundas, Quartas e Sextas
- **Horário:** T56 (16:40 às 18:20)
- **Modalidade:** Remota
- **Acesso:** Google Meet via SIGAA

Importante

Aulas de conteúdo novo às quartas-feira (gravadas)
Segundas e sextas para revisão e dúvidas

Como Funciona a Avaliação?

Composição da Nota:

- **Listas de exercícios: 20%**
- **Projeto individual: 80%**

3 Unidades

U1: 22/08 - 30/09

U2: 01/10 - 27/10

U3: 29/10 - 28/11

Entrega Final: 05/12

O que Você Precisa Entregar?

Para cada unidade, você entrega:

- 1 **Link do repositório Git público** (GitHub/GitLab/BitBucket)
- 2 **Vídeo de 5-8 minutos** demonstrando o projeto
- 3 **Relatório técnico** em PDF (3-5 páginas)

Atenção!

Apenas o commit mais recente até **23:59** será corrigido.
Commits frequentes e descritivos são valorizados!

Evolução do Projeto por Unidade

Unidade 1 (22/8 até 30/09)

Variáveis, operadores, vetores, condicionais, repetição, funções básicas

Unidade 2 (1/10 até 27/10)

Tudo da U1 + strings, estruturas aninhadas, matrizes, ponteiros básicos

Unidade 3 (29/10 até 28/11)

Tudo anterior + ponteiros avançados, arquivos, registros, modularização

Projeto Final (até 05/12)

Versão completa com todos os conceitos integrados

Sugestão de organização

```
nome-do-aluno-itp-2025-2/  
|-- projeto/  
|   |-- src/           # Código-fonte  
|   |-- include/       # Cabecalhos  
|   |-- Makefile       # Compilacao  
|   '-- README.md      # Instrucoes  
|-- listas/  
|   |-- lista1/  
|   |-- lista2/  
|   '-- ...  
'-- README.md          # Descricao geral
```

Como Seu Projeto Será Avaliado?

Código (40%):

- Qualidade e organização (10%)
- Funcionalidade (15%)
- Aplicação dos conceitos (10%)
- Histórico de commits (5%)

Relatório (30%):

- Clareza e coerência (10%)
- Profundidade técnica (10%)
- Respostas orientadoras (10%)

Vídeo (30%):

- Demonstração (15%)
- Qualidade da explicação (10%)
- Tempo adequado (5%)

Estrutura do Relatório Técnico

- 1 **Introdução** - Contexto e objetivos
- 2 **Metodologia** - Ferramentas e abordagem
- 3 **Análise do Código** - Principais estruturas
- 4 **Dificuldades e Soluções** - Desafios enfrentados
- 5 **Conclusão** - Aprendizados e melhorias

Perguntas que você deve responder:

- Quais conceitos da unidade foram aplicados?
- Como a organização facilita a manutenção?
- Quais foram os principais desafios técnicos?

20 Ideias de Projetos

- 1 Sistema de Biblioteca
- 2 Jogo da Velha com IA
- 3 Calculadora Científica
- 4 Gerenciador de Tarefas
- 5 Simulador de Banco
- 6 Conversor de Unidades
- 7 Jogo de Campo Minado
- 8 Sistema de Cadastro
- 9 Agenda de Contatos
- 10 Shell Básico
- 11 Compressão de Texto
- 12 Quebra-Cabeça 8 Peças
- 13 Sistema de Vendas
- 14 Jogo da Vida
- 15 Calculadora de Matrizes
- 16 Reserva de Passagens
- 17 Jogo de Xadrez
- 18 Analisador Léxico
- 19 Sistema de Logs
- 20 Gerenciador Financeiro

Requisitos para Projetos Próprios

Se você quiser fazer um projeto diferente:

- Deve usar $\geq 70\%$ dos tópicos da unidade
- Ser desenvolvido em **linguagem C**
- Ter interface de **linha de comando (CLI)**
- Ser **original** (não copiado)
- Ter **complexidade média/alta**

Dica

Escolha algo que te motive! Você vai trabalhar nele por 4 meses.

Cronograma Resumido

Semanas 1-6: Unidade 1

Variáveis → Condicionais → Loops → Funções → Vetores

Semanas 7-10: Unidade 2

Strings → Loops Aninhados → Matrizes → Ponteiros Básicos

Semanas 11-15: Unidade 3

Ponteiros Avançados → Funções Complexas → Arquivos → Registros → Modularização

Semana 16: Projeto Final

Entrega final e encerramento

Entregas Importantes - Não Esqueça!

U1: 30/09 até 23:59

U2: 27/10 até 23:59

U3: 05/12 até 23:59

Feriado que afeta o curso

03/10/25 - Dia dos Mártires (sexta-feira) - Não haverá aula

Por Que Este Curso É Importante?

Você vai aprender:

- **Fundamentos sólidos** da programação
- **Pensamento lógico** e resolução de problemas
- **Boas práticas** de desenvolvimento
- **Trabalho com projetos reais**

Ao final do curso você será capaz de:

- Desenvolver programas completos em C
- Organizar código de forma profissional
- Documentar e apresentar seus projetos
- Trabalhar com controle de versão (Git)

Estudos:

- Pratique regularmente
- Não deixe acumular
- Participe das aulas
- Faça as listas

Projeto:

- Comece cedo
- Commits frequentes
- Documente bem
- Teste sempre

Lembre-se: A programação é uma habilidade prática!

Como tirar dúvidas?

- **Aulas síncronas** (segundas e sextas (quarta será conteúdo novo))
- **Discord** - mensagens e fóruns
- **E-mail:** fernando@dimap.ufrn.br

Materiais disponíveis:

- Gravações das aulas teóricas
- Listas de exercícios
- Material de apoio no SIGAA
- Exemplos de código no repositório do curso:
<https://github.com/fmarquesfilho/itp-2025-2>

O Que Fazer Agora?

1 Preparar o ambiente de desenvolvimento

- Instalar compilador C (GCC)
- Escolher um editor (VS Code, Dev-C++, etc.)
- Criar conta no GitHub/GitLab (se já não a tiver)

2 Começar a pensar no projeto

- Revisar as sugestões
- Escolher algo que te interesse
- Começar simples, evoluir gradualmente

3 Acompanhar o cronograma

- Próxima aula: Preparação do ambiente
- Primeira lista em breve

É normal...

- Sentir dificuldade no início
- Cometer erros de sintaxe
- Precisar de várias tentativas
- Ter bugs no código

O importante é...

- **Persistir** e praticar
- **Aprender** com os erros
- **Buscar ajuda** quando necessário
- **Celebrar** os pequenos sucessos

Alguma pergunta sobre:

- Metodologia do curso?
 - Avaliação?
 - Projetos?
- Cronograma?

Vamos esclarecer tudo agora!

Obrigado!

Nos vemos na próxima aula!

fernando@dimap.ufrn.br
DIMAp - UFRN