

Introdução a Técnicas de Programação (2025.2)

Planejamento Incremental para Projetos em C

Prof. Fernando Figueira

DIMAp - UFRN

29 de Agosto de 2025

Objetivo da Aula

- Entender desenvolvimento incremental para projetos em C
- Aprender a organizar backlog para aplicações CLI
- Planejar sprints adaptados ao curso de técnicas de programação
- Durante o mês de setembro vocês vão preparar o projeto para entrega da Unidade 1

- Dividir o projeto em funcionalidades menores
- Desenvolver por camadas: entrada \rightarrow processamento \rightarrow saída
- Testar cada função individualmente antes de integrar
- Exemplo: Calculadora com histórico
 - Sprint 1: Operações básicas (+, -, *, /)
 - Sprint 2: Histórico em vetor
 - Sprint 3: Persistência em arquivo

Exemplo: Calculadora com Histórico - Unidade 1

Função: soma(float a, float b)

Função: subtracao(float a, float b)

Vetor: historico[100] para armazenar resultados

Menu interativo com switch-case

Vantagens para Projetos em C

- Detecção precoce de erros de compilação
- Teste incremental de funções
- Facilita debugging de programas complexos
- Adaptação aos conceitos aprendidos em aula

Product Backlog para Projetos em C

- Funcionalidades core do programa
- Tratamento de erros e validações
- Melhorias de interface de usuário
- Otimizações de performance
- **Prioridade baseada nos conceitos da unidade**

Exemplo: Backlog para Sistema de Biblioteca (U1)

Alta: Cadastrar livro (struct Livro + vetor)

Alta: Listar livros (for + printf)

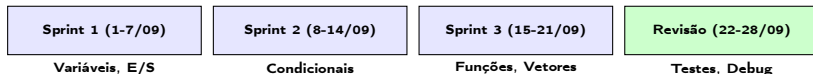
Média: Buscar livro por título (strcmp + repetição)

Baixa: Estatísticas

Sprints Adaptados para o Curso

- **Duração:** 1-2 semanas (alinhado com conteúdo das aulas)
- **Objetivo:** Entrega de funcionalidades que usem os tópicos recentes
- **Exemplo Sprint 1:** Variáveis, operadores, E/S básica
- **Exemplo Sprint 2:** Condicionais, repetições simples
- **Exemplo Sprint 3:** Funções, vetores

Planejamento: Projeto Unidade 1 (Setembro)



Entrega: 30/09 até 23:59

- Sprint 1: Entrada/saída básica, variáveis, operadores
- Sprint 2: if/else, switch, repetições simples
- Sprint 3: Modularização com funções, vetores
- Revisão: Integração, testes finais, documentação

Definição de Pronto para Projetos em C

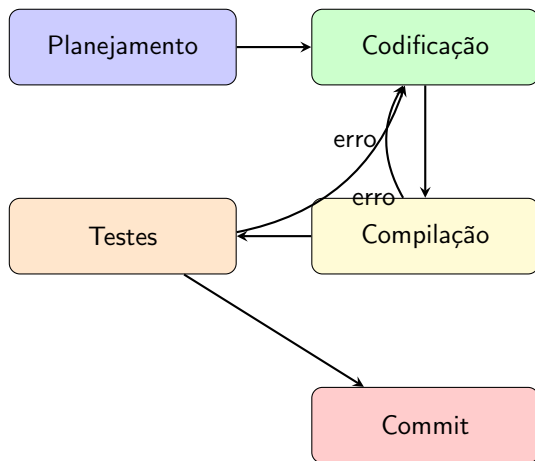
- Código compila sem warnings
- Funções testadas individualmente
- Documentação nos comentários
- **Commit no repositório Git**

Exemplo: Definição de Pronto para Função

Função: calcular_media(float notas[], int n)

- ✓ Compila sem warnings
- ✓ Retorna -1 se $n \leq 0$
- ✓ Calcula média corretamente
- ✓ Testada com valores limite
- ✓ Comentários explicativos
- ✓ Commit no Git com mensagem descritiva

Fluxo de Desenvolvimento para Projetos em C



Dicas para o Projeto - Unidade 1

- Comece com funcionalidades básicas
- Teste cada função individualmente
- Use nomes descritivos para variáveis e funções
- Faça commits frequentes no Git
- Documente com comentários
- **Foco:** Variáveis, operadores, condicionais, repetições, funções, vetores

Próximos Passos

- Escolher projeto até 05/09
- Configurar ambiente de desenvolvimento
- Criar repositório Git
- Backlog até 08/09
- Iniciar Sprint 1 na semana de 08/09
- Buscar ajuda nas aulas de dúvidas (segundas e sextas)