

Entrega Final do Projeto

Unidade 3 - MVP Final com Testes e Análises

Boas Práticas de Programação - 2025.2

UFRN

Entrega: 05/12/2025 às 23:59

U3: Evolução do Projeto

U1 focou em:

- ✓ Planejamento e MVP
- ✓ Código Limpo
- ✓ Code Smells
- ✓ Refatoração

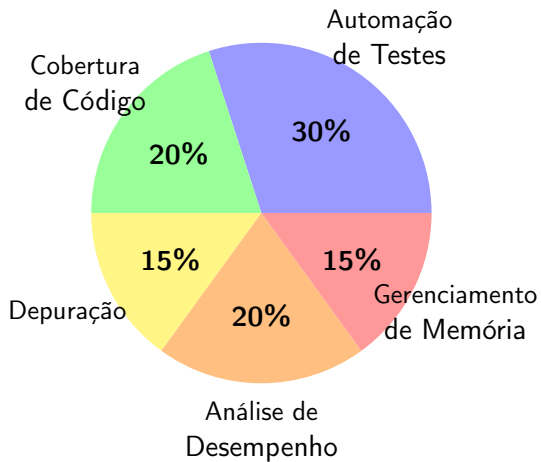
U3 adiciona:

- + Automação de Testes
- + Análise de Cobertura
- + Técnicas de Depuração
- + Análise de Desempenho
- + Gerenciamento de Memória

Importante

U3 é a continuação do mesmo projeto da U1! Você vai adicionar novas análises ao MVP existente.

Distribuição da Nota - U3



1. Automação de Testes (30%)

Requisitos Mínimos

- **15+** testes **unitários** cobrindo funcionalidades principais
- **5+** testes de **integração** testando interação entre componentes
- Seguir padrão **AAA** (Arrange-Act-Assert)
- Princípios **FIRST** (Fast, Independent, Repeatable, Self-validating, Timely)

Qualidade > Quantidade

Testes bem escritos e significativos valem mais que muitos testes triviais!

2. Análise de Cobertura (20%)

Métricas Obrigatórias

- **Cobertura de Linhas:** Mínimo 70%
- **Cobertura de Branches:** Mínimo 60%
- **Módulos Críticos:** Mínimo 85%

Entregáveis

- 1 Relatório HTML de cobertura na pasta `tests/coverage-results/`
- 2 Screenshot das métricas principais
- 3 Documento `coverage-report.md` com:
 - Evolução da cobertura (início vs final)
 - Justificativa para código não coberto (se $< 80\%$)

3. Técnicas de Depuração (15%)

Requisito: Documentar 3+ bugs

Para cada bug, documente em docs/debugging-log.md:

- 1 **Identificação:** Data, módulo, severidade
- 2 **Descrição:** Como reproduzir o problema
- 3 **Investigação:** Técnica de depuração usada
- 4 **Causa raiz:** O que estava errado
- 5 **Correção:** Código antes/depois
- 6 **Verificação:** Testes adicionados

Técnicas esperadas (escolha livre)

Debugger, logging, testes automatizados, binary search, stack trace, git bisect

4. Análise de Desempenho (20%)

Requisito: Identificar 2+ gargalos

Para cada gargalo, documente em `docs/performance-analysis.md`:

Análise:

- Módulo/função afetada
- Problema identificado
- Medição inicial (tempo)
- Complexidade (Big O)

Otimização:

- Técnica aplicada
- Código antes/depois
- Medição final
- Ganho de performance
- Trade-offs

Ferramentas

Python: cProfile, timeit | JS: Chrome DevTools | Java: JProfiler | C/C++: gprof, Valgrind

5. Gerenciamento de Memória (15%)

C/C++ (obrigatório):

- Usar Valgrind
- Detectar memory leaks
- Corrigir acessos inválidos
- Documentar correções

Outras linguagens:

- Uso eficiente de estruturas
- Evitar referências circulares
- Generators para grandes volumes
- Cache com limite (LRU)

Documento: docs/memory-analysis.md

- Análise realizada (ferramenta usada)
- Problemas encontrados
- Otimizações implementadas
- Medições antes/depois

Estrutura de Entrega

```
entrega-u3/
|-- README.md                # Visao geral + instrucoes
|-- docs/
|   |-- testing-report.md    # Relatorio completo
|   |-- coverage-report.md   # Analise de cobertura
|   |-- debugging-log.md     # Bugs encontrados
|   |-- performance-analysis.md # Gargalos e otimizacoes
|   '-- memory-analysis.md   # Analise de memoria
|-- tests/
|   |-- unit/                # Testes unitarios
|   |-- integration/         # Testes de integracao
|   '-- coverage-results/    # Relatorios HTML
|-- src/
|   '-- (codigo fonte refatorado)
'-- video-presentation.md    # Link para video
```

Vídeo de Apresentação (10-12 min)

1 Recapitulação (1-2 min)

- Visão do produto
- Evolução desde U1

2 Demonstração de Testes (3-4 min)

- Execução da suite
- Relatório de cobertura

3 Depuração (2 min)

- 1-2 bugs interessantes
- Como foram corrigidos

4 Performance (2 min)

- Gargalos encontrados
- Métricas before/after

5 Conclusão (2-3 min)

- Qualidade final
- Lições aprendidas

Documentação:

- ☐ README.md atualizado
- ☐ testing-report.md
- ☐ coverage-report.md
- ☐ debugging-log.md (3+ bugs)
- ☐ performance-analysis.md (2+ gargalos)
- ☐ memory-analysis.md

Código e Testes:

- ☐ 10+ testes unitários
- ☐ Todos passando
- ☐ Cobertura $\geq 70\%$
- ☐ Relatório HTML incluído

Apresentação:

- ☐ Vídeo 10-12 min
- ☐ Link funcionando

Informações Finais

Entrega

Data: 05/12/2025 às 23:59

Plataforma: SIGAA - Tarefa "Entrega U3"

Formato: ZIP único: u3-[projeto]-[integrantes].zip

Trabalho em Grupo

- Individual ou grupos de 2-4 pessoas
- Descrever contribuição de cada membro
- Todos devem dominar o código completo

Plantão de Dúvidas

05/12 no horário da aula (online) - Estarei disponível para tirar dúvidas sobre a entrega

Precisa de ajuda?

✉ **Atendimento:** Segundas 14h-16h (online)

🗨 **Discord:** <https://discord.gg/bbMFJBQRT8>

🐙 **GitHub:** <https://github.com/fmarquesfilho/bpp-2025-2>

Bom trabalho!