

Programação de Aplicativos

AULA 01 - APRESENTAÇÃO DA UNIDADE CURRICULAR

Lorrany B Marim



Conteúdo Programático

- Apresentação, objetivos e boas práticas.
- Preparação do ambiente: ferramentas, funções, repositórios/controle de versão, IDE; instalação, configurações e requisitos mínimos.
- Linguagem estruturada: tipos, controle de fluxo, modularização.
- POO: classes, objetos, encapsulamento, herança e polimorfismo.
- Conexão com banco de dados: drivers, consultas, transações e segurança básica.
- Técnicas de programação:
 - formatação e padronização;
 - documentação de código;
 - reutilização (funções/módulos);
 - otimização;
 - depuração;
 - rastreabilidade;
 - teste unitário.
- Ética profissional: sigilo, prudência, imparcialidade e honestidade.
- Trabalho e profissionalismo: planejamento da rotina, flexibilidade, acompanhamento de resultados.

Ambientes de programação

Componentes Essenciais

- IDE ou editor de código → onde o programador escreve o código.
- Compilador/interpretador → transforma o código em execução.
- Depurador → identifica e corrige erros durante a execução.
- Ferramentas de build → automatizam a compilação e organização do projeto.
- Biblioteca padrão → conjunto de funções/utilitários prontos.
- Controle de versão → mantém histórico e permite colaboração.

Ambientes de programação

Isolamento e Replicabilidade:

- Configurações consistentes entre máquinas.
- Uso de ambientes virtuais ou contêineres (Docker, venv).
- Evita o famoso problema: “funciona no meu computador, mas não no seu”.

Desempenho Aceitável:

- Ambiente deve ser leve e eficiente.
- Compilações rápidas.
- Depuração e testes sem lentidão que prejudiquem a produtividade.

Ambientes de Programação

Controle de Versão / Repositórios:

- Rastreia alterações no código.
- Permite reversão de erros.
- Facilita a colaboração em equipe.
- Exemplo: Git + GitHub.

Padronização e Consistência:

- Estilo de código unificado (indentação, convenções).
- Estrutura de pastas organizada.
- Documentação clara e atualizada.
- Facilita a leitura e manutenção por diferentes pessoas.



Ambientes de Programação

Testabilidade:

- Suporte a testes unitários e de integração.
- Ferramentas de depuração eficientes.
- Separação entre ambiente de desenvolvimento, teste e produção.

Modularidade e Configuração:

- Facilidade para adicionar, atualizar ou remover dependências.
- Parametrizações flexíveis sem comprometer o sistema.
- Possibilidade de trocar versões de bibliotecas de forma controlada.



Ambientes de Programação

Automação:

- Scripts de build para agilizar processos.
- Testes automatizados garantem qualidade contínua.
- Integração contínua (CI/CD) para builds e deploys frequentes.
- Reduz trabalho manual e erros humanos.

Segurança e Higiene:

- Controle de acesso a repositórios e servidores.
- Verificação de confiabilidade de bibliotecas externas.
- Ambientes isolados para reduzir riscos de contaminação.
- Boas práticas de permissões e atualização contínua.

Etapas da Preparação do Ambiente de Programação

- **Definição de requisitos do projeto:**
 - Quais linguagens serão usadas
 - Quais bibliotecas / dependências são esperadas
 - Tipo de aplicação (web, desktop, banco de dados, etc.)
- **Requisitos de hardware/software genéricos:**
 - Hardware mínimo para executar IDE/compilar/testar
 - Sistema de arquivos (estrutura de pastas, permissões)
 - Sistema operacional ou plataformas alvo (embora aqui o foco não seja específico)

Etapas da Preparação do Ambiente de Programação

- **Escolha de ferramentas:**
 - Editor ou IDE
 - Compilador / interpretador / runtime
 - Ferramentas de versionamento / repositório
 - Depurador, profiler, ferramentas de análise de código
 - Ferramentas de build / automação.
- **Configuração inicial:**
 - Instalação das ferramentas escolhidas
 - Ajustes de ambiente (variáveis de ambiente, PATH, config de IDE, configuração de linting / formatação)
 - Instalação das dependências do projeto

Etapas da Preparação do Ambiente de Programação

- **Padronização e boas práticas:**
 - Estabelecer convenções de código (formato, estilo, layout)
 - Documentação inicial (README, guias de instalação)
 - Uso de controle de versão desde o primeiro momento.
- **Testes e depuração:**
 - Configurar mecanismo de execução de testes (unitários ou outros tipos)
 - Depurador, logs, rastreabilidade de erros
 - Ambiente de teste isolado, de preferência replicável

Etapas da Preparação do Ambiente de Programação

- **Manutenção do ambiente:**
 - Atualizações de dependências, patches de segurança
 - Monitoramento de desempenho (se aplicável)
 - Backup ou versão de configurações.
- **Migrar ou replicar para produção / staging:**
 - Ter ambientes separados para teste / homologação / produção
 - Garantir que o ambiente de produção reflita o mais próximo possível o ambiente de desenvolvimento/teste

