

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DO RIO GRANDE DO NORTE
CAMPUS JOÃO CÂMARA

IMPLEMENTAÇÃO

Nickerson Fonseca Ferreira
nickerson.ferreira@ifrn.edu.br

Introdução

- É o processo de tradução do projeto para uma forma que seja possível de ser executada em máquina.
- Esta fase é responsável pela escrita do sistema em si.
- Inclui o uso em linguagens de programação, programação em banco de dados, geração de programas a partir de ferramentas CASE, etc.
- Foco em boas práticas de programação.

Introdução

3

- Alguns pontos importantes que serão vistos nesta fase do projeto:
 - Ambiente de desenvolvimento
 - Boas práticas de programação.
 - Documentação
 - Ferramentas de desenvolvimento

Ambiente de desenvolvimento

4

- Conjunto de ferramentas que dão apoio ao desenvolvimento de software.
- Seu principal objetivo é agilizar o processo de desenvolvimento.
- Possui como principais características e ferramentas:
 - Editor
 - Compilador
 - Depurador
 - Geração de códigos
 - Testes
 - Distribuição
 - Refatoração

Ambiente de desenvolvimento

- É tão importante que existe uma atividade no RUP que específica para o Ambiente.
- Exemplo de um ambiente de desenvolvimento ???
 - SO Linux, máquina virtual Java 8, Netbeans, SGBD MySQL 5, Internet, Intranet, etc.
- Pode ser instalado completamente na máquina dos membros da equipe de desenvolvimento.
- Como também pode estar distribuído, dependendo da arquitetura do sistema.

Ambiente de desenvolvimento

6



Boas práticas de programação

7

- O código deve ser:
 - Simples: código fácil de entender;
 - Direto: vai direto ao ponto, não dá “voltas” para atingir seu objetivo;
 - Eficiente: código que faz o que é proposto;
 - Sem duplicidade: não faz o que outra parte do código já faz;
 - Elegante: porque é diferente dos outros códigos;
 - Feito com cuidado: quem fez teve preocupação em produzir aquele código.

Boas práticas de programação

- A Sun criou uma convenção para padronizar a codificação com Java.
- O uso de uma convenção é importante por:
 - Aproximadamente 80% do custo de um sistema está em sua manutenção;
 - Dificilmente o autor original do código irá realizar a manutenção do código durante o tempo de vida do sistema;
 - A padronização deixa o código mais legível e fácil de ser entendido;

Boas práticas de programação

9

“ Sempre programe como se o cara que vai acabar dando manutenção no seu código fosse um psicopata violento que sabe onde você mora.”

Martin Golding



Documentação

10

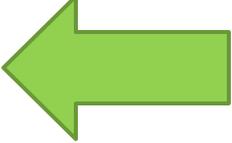
- Como toda obra bem planejada, o código fonte do sistema desenvolvido deve ser bem documentado.
- Os principais pontos para se documentar no código são:
 - Cabeçalho da classe (informações iniciais)
 - Principais variáveis utilizadas
 - Padronização do nome das variáveis e parâmetros
 - Pontos importantes no decorrer do código
- **Lembrem-se do psicopata!!**

```
package entity;

/**
 *
 * Objetivo: Esta classe tem o objetivo de imprimir uma
 * mensagem ao ser executada
 *
 * Autor: Nickerson Ferreira (nickerson.ferreira@ifrn.edu.br)
 *
 * Data de criação: 01/07/2016
 * #####
 * Última Alteração:
 *
 * Analista: João
 * Data: 10/07/2016
 * Alteração: Teste de
 *
 * #####
 *
 */
public class HelloWorld {
    /**
     * Objetivo: Executar a classe.
     * ...
     */
    public static void main(String[] args) {
        // msg: variável contendo a mensagem que será exibida
        String msg = "Olá Mundo!!";
        System.out.println(msg); // comando para exibir mensagem
    } // fim main()
}
```

Ferramentas de desenvolvimento

12

- São ferramentas criadas com o objetivo de auxiliar algumas tarefas do desenvolvimento de sistemas.
- Podem ser divididas em categorias, por exemplo:
 - Sistemas de controle de versão 
 - Ferramentas de build 
 - Geradores de interface
 - Editores de texto
 - Geração de documentação 

Sistemas de controle de versão

13

- Sistemas são desenvolvidos por equipes e seus membros têm que trabalhar paralelamente.
- **Como gerenciar as várias versões do sistema ???**
- Os sistemas de controle de versão são softwares utilizados no desenvolvimento de sistemas para controlar as diferentes versões – histórico e desenvolvimento – dos códigos-fontes e documentação.
- Também serve como backup.

Sistemas de controle de versão

14

- Os SCV mais utilizados atualmente são:
 - CVS
 - Mercurial
 - Git
 - SVN

Ferramentas de Build

15

- São ferramentas que auxiliam na compilação, gestão de dependências, empacotamento e distribuição dos sistemas desenvolvidos.
- Utilizados para a criação de procedimentos para automatizar as tarefas já citadas.
- As ferramentas de build mais utilizadas são:
 - Make
 - Apache Ant
 - Apache Maven

Geração de documentação

16

- Vimos que a documentação interna de código é uma atividade muito importante no desenvolvimento/manutenção de um sistema.
- **LEMBREM-SE DO PSICOPATA!!**
- Além da documentação interna é importante gerar uma documentação externa.
- Existem ferramentas que geram esse tipo de documentação:
 - Doxygen
 - Javadoc