Computação II

Java - Programação Orientada a Objetos

2ª Aula

Prof. Danilo S. Carvalho[1, 2]

1. Departamento de Ciência da Computação - UFRJ
2. Centro de Desenvolvimento Tecnológico em Saúde - FIOCRUZ

Índice

Erros de compilação e Erros de execução

Erros de compilação:

- São gerados pelo compilador quando encontrado código fonte inválido.
- O programa executável não pode ser obtido.
- O compilador aponta a linha que causou o problema.

Erros de Execução:

- São gerados pelo ambiente de execução (ou sistema operacional) quando um programa realiza uma operação inválida.
- O programa em execução é interrompido e todos os dados em processamento são perdidos.
- O ambiente de execução / SO produz um relatório de falha (*stack trace*) informando o motivo da interrupção do programa.

Tratamento de Exceção

- Consiste na antecipação, detecção e criação de um fluxo de código alternativo à ocorrência de um erro de execução.
- Tais erros são chamados de *exceções*.

1. Identifica-se um possível erro de execução.
2. Aplica-se um mecanismo de detecção, para caso o erro ocorra.
3. Cria-se um caminho alternativo no código, que será tomado quando o erro ocorrer.

Código ingênuo

Fonte

```
import java.util.Scanner;

public class IntroExcecoes {

    public static void main(String[] args) {
        int idade;
        Scanner scanner = new Scanner(System.in);

        System.out.println("Digite sua idade e aperte <ENTER>: ");
        String linha = scanner.nextLine();
        idade = Integer.parseInt(linha);
        System.out.println("Sua idade em binário é: " +
                           Integer.toBinaryString(idade));
    }
}
```

Fonte

```
int idade;
Scanner scanner = new Scanner(System.in);

System.out.println("Digite sua idade e aperte <ENTER>: ");
String linha = scanner.nextLine();

try {  // Bloco try: a exceção pode
    idade = Integer.parseInt(linha);

    System.out.println("Sua idade em binário é: " +
                       Integer.toBinaryString(idade));
}
catch (NumberFormatException e) {
    System.out.println("Idade inválida.");
}
```

Fonte

```
int idade = 0;
Scanner scanner = new Scanner(System.in);

while (idade == 0) {
    try {
        System.out.println("Digite sua idade e aperte <ENTER>: ");
        String linha = scanner.nextLine();
        idade = Integer.parseInt(linha);

        System.out.println("Sua idade em binário é: " +
                           Integer.toBinaryString(idade));
    }
    catch (NumberFormatException e) {
        System.out.println("Idade inválida.");
    }
}
```

Índice

Preciso mexer no código, e agora

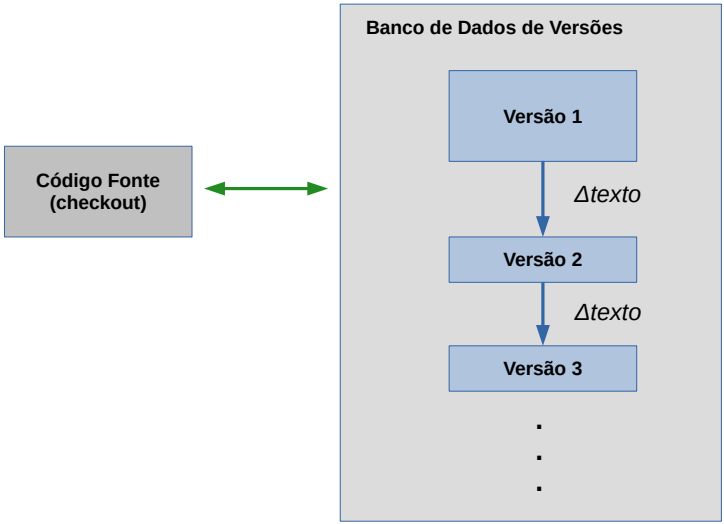
- Toda vez que precisamos fazer alguma alteração no código, estamos destruindo parte do código anterior e possivelmente criando problemas que precisarão ser analisados.
- Programadores podem cair na tentação de deixar o código anterior comentado como um registro histórico ou "*vai que eu preciso usar novamente...*"
- No pior dos casos, o programador cria vários arquivos com variações de nome, ex: ClasseImportante1(2,3).java.

Preciso mexer no código, e agora

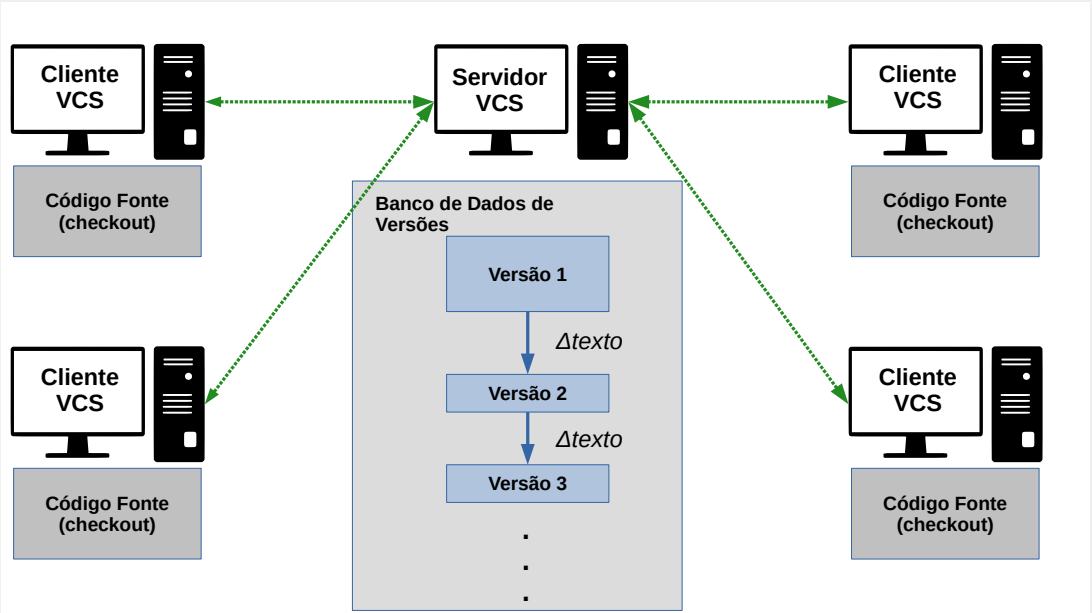
- Gerenciar vários arquivos da mesma base de código manualmente é ineficiente e problemático.

- Torna-se inviável para projetos grandes (ex: > 100 arquivos).

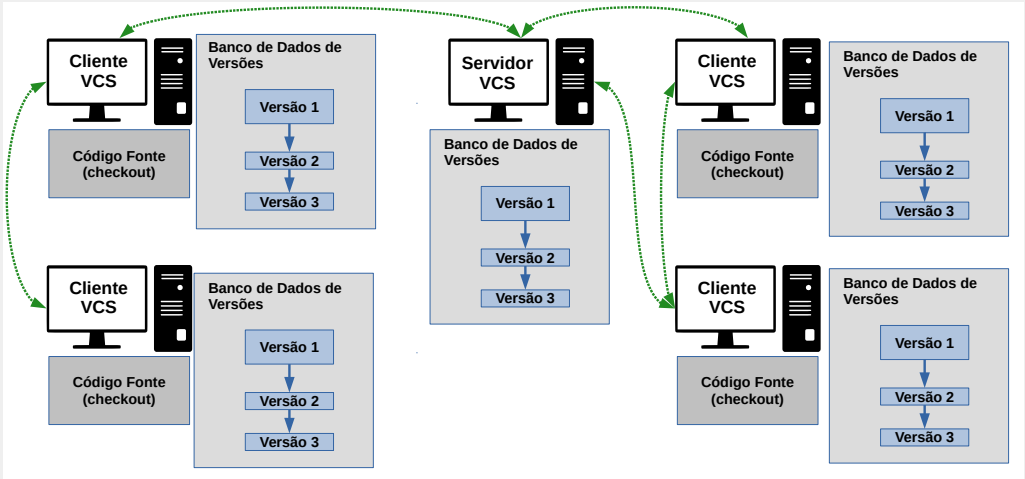
Sistema de Controle de Versão (VCS)



VCS Centralizado



VCS Distribuído



Índice

Git

- Sistema de controle de versão mais popular atualmente.
- Criado para gerenciar o código do kernel Linux.
- VCS distribuído.

GitHub

- Serviço de repositórios para o git.
- Um dos mais utilizados para distribuição de código aberto / livre.
- Disponibiliza outras ferramentas e serviços, como wikis.

Perguntas?