Programação Orientada à Objetos (POO)

CIÊNCIA DA COMPUTAÇÃO

Prof. Dr. João Paulo Aramuni

UNIVERSIDADE FUMEC

Sumário

* **Conceitos Básicos**
  * Introdução à orientação a objetos
  * Descrição da linguagem e ambiente de desenvolvimento Java

UNIVERSIDADE FUMEC
DE MINAS GERAIS PARA O MUNDO

Conceitos Básicos

* **Introdução à orientação a objetos**
  * O desenvolvimento de software é extremamente amplo.
  * Nesse mercado, existem diversas linguagens de programação que seguem diferentes ***paradigmas***.
  * Um desses paradigmas é a ***Orientação a Objetos***, que atualmente é o mais difundido entre todos.

Conceitos Básicos

* **Introdução à orientação a objetos**
  * Isso acontece porque se trata de um padrão que tem evoluído muito, principalmente em questões voltadas para ***segurança***, ***manutenção*** e ***reaproveitamento de código***, o que é muito importante no desenvolvimento de qualquer aplicação moderna.

Conceitos Básicos

* **Introdução à orientação a objetos**
  * A **Programação Orientada a Objetos** (POO) diz respeito a um padrão de desenvolvimento que é seguido por muitas linguagens, como C++, C# e Java.

  * Iremos entender as diferenças entre a POO e a **Programação Estruturada**, que era muito utilizada há alguns anos, principalmente com a linguagem C.

Conceitos Básicos

* **Introdução à orientação a objetos**
  * A **POO** se baseia em quatro pilares:
    * **Abstração**
    * **Encapsulamento**
    * **Herança**
    * **Polimorfismo**

  * Veremos detalhadamente cada um deles ao longo do curso.

* **Introdução à orientação a objetos**
  * Programação Estruturada x POO

Conceitos Básicos

* **Introdução à orientação a objetos**
  * Repare que, no paradigma estruturado, temos **procedimentos** (ou funções) que são aplicados globalmente em nossa aplicação.
  * No caso da orientação a objetos, temos **métodos** que são aplicados aos dados de cada **objeto**.
  * Essencialmente, os procedimentos e métodos são iguais, sendo diferenciados apenas pelo seu escopo.

Conceitos Básicos

* **Introdução à orientação a objetos**
  * A linguagem C é a principal representante da programação estruturada.
  * Se trata de uma linguagem considerada de baixo nível, que atualmente não é utilizada para projetos muito grandes.
  * A sua principal utilização, devido ao baixo nível, é em programação para sistemas embarcados ou outros em que o conhecimento do hardware se faz necessário para um bom programa.

Conceitos Básicos

* **Introdução à orientação a objetos**
  * Essa colocação nos traz um detalhe importante: a programação estruturada, quando bem feita, possui um desempenho superior ao que vemos na programação orientada a objetos.

  * Isso ocorre pelo fato de ser um paradigma sequencial, em que cada linha de código é executada após a outra, sem muitos desvios, como vemos na POO.

Conceitos Básicos

* **Introdução à orientação a objetos**
  * Além disso, o paradigma estruturado costuma permitir mais liberdade com o hardware, o que acaba auxiliando na questão do desempenho.

  * Entretanto, a programação orientada a objetos traz outros pontos que acabam sendo mais interessantes no contexto de aplicações modernas.

Conceitos Básicos

* **Introdução à orientação a objetos**
  * Como o desempenho das aplicações não é uma das grandes preocupações na maioria das aplicações (devido ao poder de processamento dos computadores atuais), a programação orientada a objetos se tornou muito difundida.

  * Essa difusão se dá muito pela questão da **reutilização de código** e pela capacidade de representação do sistema muito mais próximo do que veríamos no mundo real.

Conceitos Básicos

* **Introdução à orientação a objetos**
  * POO é apenas um modelo para projetar e implementar software.
  * As técnicas utilizadas não acrescentam nada ao produto final visto pelo usuário.
  * Entretanto, essas técnicas oferecem vantagens significativas para gerenciar problemas complexos, especialmente em grandes projetos.

Conceitos Básicos

* **Introdução à orientação a objetos**
  * O bom desenvolvedor precisa entender quais são as vantagens e desvantagens de cada um dos paradigmas de programação.
  * A escolha de qual linguagem de programação utilizar para desenvolver um programa, deve ser discutida em conjunto.
  * Deve-se unir as habilidades do *Arquiteto de Software*, do *Engenheiro de Sistemas*, dos *Desenvolvedores*, do *Coordenador de Sistemas* e até mesmo do *Gestor de Projetos*.

Conceitos Básicos

* **Introdução à orientação a objetos**
  * Hoje, como a demanda por software novo e mais poderoso está aumentando, construir softwares de maneira rápida, correta e econômica continua a ser um objetivo indefinido.

  * Objetos ou, mais precisamente, as classes de onde os objetos vêm são essencialmente componentes reutilizáveis de software. Há objetos data, objetos data/hora, objetos aúdio, objetos vídeo, objetos automóvel, objetos pessoas etc.

Conceitos Básicos

* **Introdução à orientação a objetos**
  * Quase qualquer substantivo pode ser razoavelmente representado como um objeto de software em termos dos ***atributos*** (por exemplo, nome, cor e tamanho) e ***comportamentos*** (por exemplo, calcular, mover e comunicar).

  * Grupos de desenvolvimento de software podem usar uma abordagem modular de projeto e implementação orientados a objetos para que sejam muito mais ***produtivos*** do que com as técnicas anteriormente populares como "programação estruturada" – programas orientados a objetos são muitas vezes mais fáceis de entender, corrigir e modificar.

Conceitos Básicos

* **Introdução à orientação a objetos**
  * Por ser uma das linguagens mais utilizadas no mundo, iremos implementar as **técnicas relacionadas ao paradigma da orientação a objetos** através da linguagem **JAVA.**
  * A POO têm maior importância para este curso do que a linguagem Java em si.

Conceitos Básicos

* **Descrição da linguagem e ambiente de desenvolvimento Java**
  * Java é uma linguagem simples, orientada por objetos, robusta, segura, independente de plataforma, interpretada, de alto desempenho.

  * Java é uma linguagem de propósito geral
    * Pode ser usada para a construção de pequenos programas (Applets) que rodam em browsers
    * Pode ser usada para a construção de complexas aplicações Desktop

Conceitos Básicos

* **Descrição da linguagem e ambiente de desenvolvimento Java**
  * Java possui sintaxe baseada em C.
  * Tipos de dados básicos similares a C.
  * Gerenciamento de memória automático.
  * Grande biblioteca que inclui recursos para Web, Interfaces Gráficas (GUI), Banco de Dados, Redes, etc.
  * Linguagem de uso gratuito.

Conceitos Básicos

* **Descrição da linguagem e ambiente de desenvolvimento Java**
  * O ***Java Standard Edition*** (JSE) contém os recursos necessários para desenvolver aplicativos de ***desktop e servidor***.
  * Antes do Java SE 8, a linguagem suportava três paradigmas de programação: ***programação procedural***, ***programação orientada a objetos*** e ***programação genérica***. O Java SE 8 acrescenta a ***programação funcional*** (utilizada para escrever programas de forma mais rápida, concisa, com menos bugs e que são mais fáceis de paralelizar, isto é, executar múltiplos cálculos ao mesmo tempo a fim de tirar proveito das atuais arquiteturas de hardware multiprocessadas com o intuito de melhorar o desempenho do aplicativo).

Conceitos Básicos

* **Descrição da linguagem e ambiente de desenvolvimento Java**
  * O Java é utilizado para um espectro de aplicações tão amplo que ele tem duas outras versões.
  * O ***Java Enterprise Edition*** (JEE) é adequado para desenvolver aplicativos em rede distribuída e em grande escala e também aplicativos baseados na ***web***.
  * O ***Java Micro Edition*** (JME), um subconjunto do Java SE, é voltado para o desenvolvimento de aplicativos para dispositivos embarcados com recursos limitados, como smartwatches, MP3 players, decodificadores de TV, medidores inteligentes, etc.

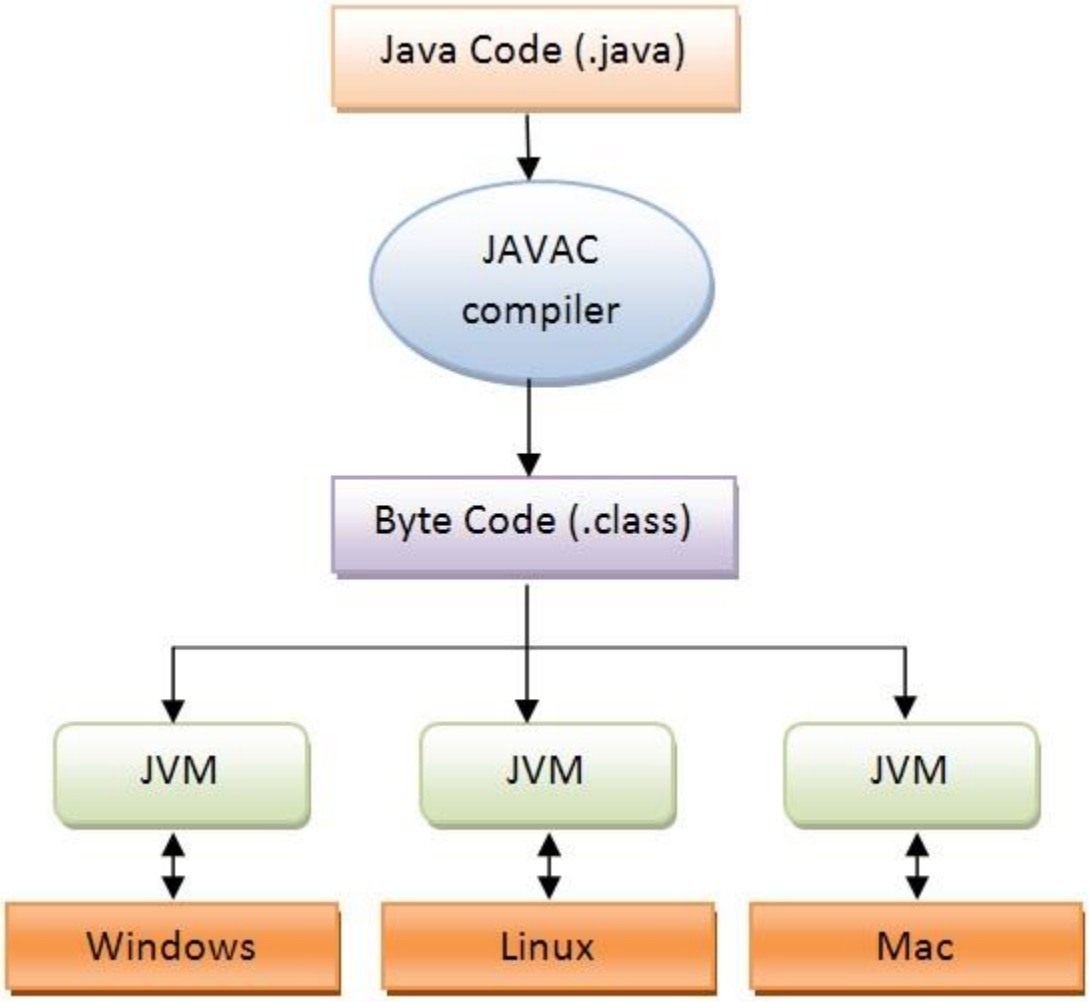
Conceitos Básicos

* **Descrição da linguagem e ambiente de desenvolvimento Java**
  * Programas Java são compostos por ***classes*** armazenadas em arquivos texto com extensão ***.java***.
  * Estes programas podem ser editados por um editor de texto convencional e são armazenados em disco como um arquivo convencional.
  * Através do processo de compilação, um código objeto é gerado a partir do código fonte. Este código objeto, denominado ***bytecode***, é armazenado em disco como um ou mais arquivos de extensão ***.class***.

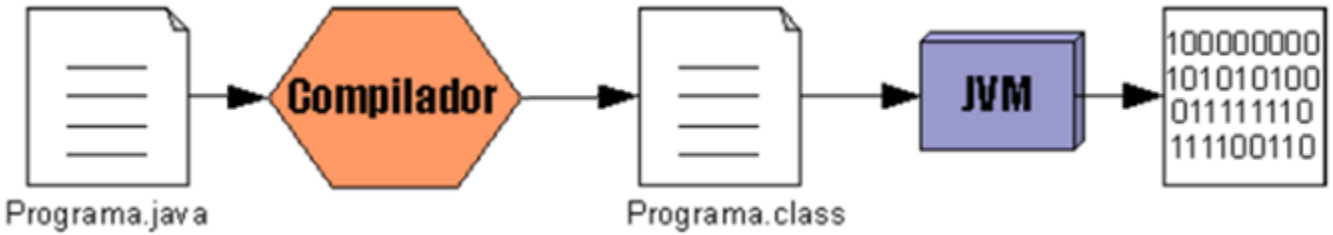
Conceitos Básicos

* **Descrição da linguagem e ambiente de desenvolvimento Java**
  * Uma vez gerado o código objeto Java (bytecodes), o mesmo é interpretado por uma máquina virtual (JVM), que traduz cada instrução do bytecode para uma instrução que o computador nativo possa entender.

Conceitos Básicos



* **Descrição da linguagem e ambiente de desenvolvimento Java**
  * As fases pelo qual passam um programa Java relacionam-se da seguinte forma:



  * 1) Criação do código fonte (Programa ***.java***).
  * 2) Compilação do código fonte e geração do bytecode (Programa ***.class***).
  * 3) Interpretação do bytecode pela máquina virtual JVM.
  * 4) Conversão do bytecode em linguagem de máquina.

Conceitos Básicos

* **Descrição da linguagem e ambiente de desenvolvimento Java**
  * O compilador (javac) atua no código fonte (.java) e gera código intermediário (.class - bytecodes independentes de plataforma).
  * A máquina virtual Java (JVM) carrega os bytecodes na memória, verifica os bytecodes e os interpreta diretamente para a arquitetura da máquina real.
  * A JVM é na verdade um emulador para uma máquina real.

Conceitos Básicos

* **Descrição da linguagem e ambiente de desenvolvimento Java**
  * A plataforma do Java é formada por três partes: JVM, Linguagem Java e Bibliotecas de Classes Java (API).
  * API (Application Programming Interface) Java
    * Complementa a linguagem Java com um conjunto de rotinas específicas para diversas tecnologias.
    * Possui milhares de métodos.

Conceitos Básicos

* **Descrição da linguagem e ambiente de desenvolvimento Java**
  * **JRE** e **JDK**
    * **JRE:** O Java Runtime Environment contém tudo aquilo que um usuário comum precisa para executar uma aplicação Java (JVM e bibliotecas), como o próprio nome diz é o "Ambiente de execução Java".
    * **JDK:** O Java Development Kit é composto pelo JRE, o compilador javac e as APIs Java.

Conceitos Básicos

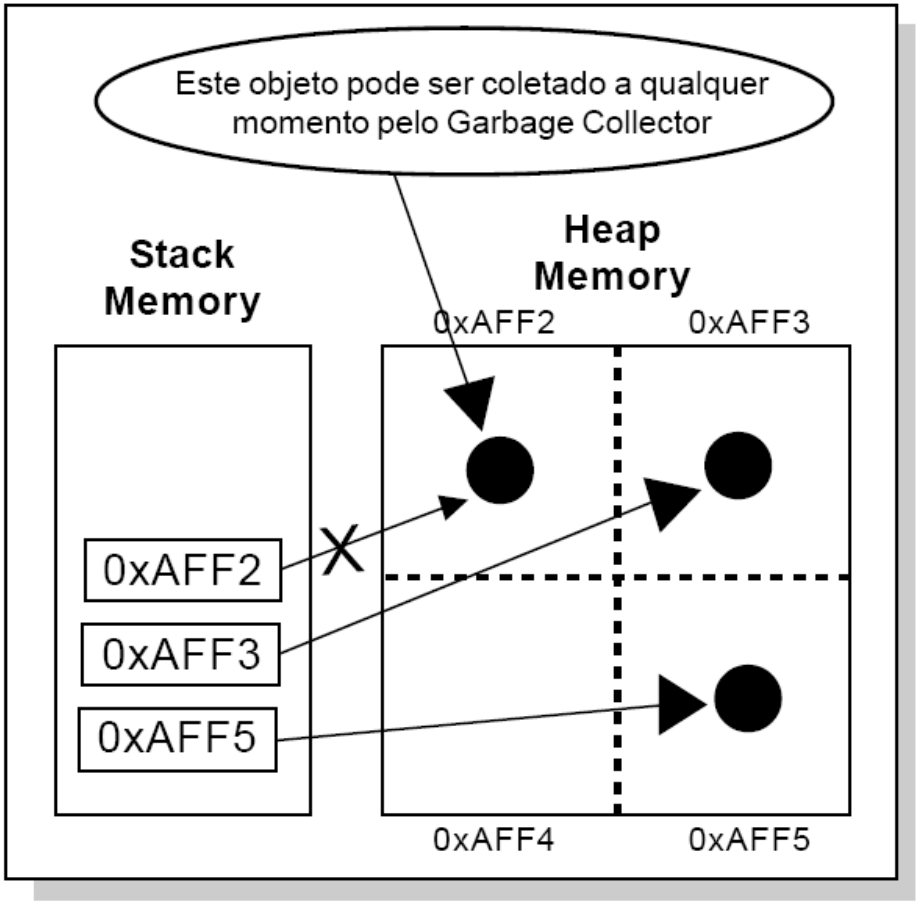
* **Descrição da linguagem e ambiente de desenvolvimento Java**
  * **Garbage Collector**
    * O Java possui um coletor de lixo. Um processo usado para a automação do gerenciamento de memória.
    * Com ele é possível recuperar uma área de memória inutilizada por um programa, o que pode evitar problemas de vazamento de memória, resultando no esgotamento da memória livre para alocação.

Conceitos Básicos

* **Descrição da linguagem e ambiente de desenvolvimento Java**
  * **Garbage Collector**
    * O Garbage Collector se resume em uma *Thread* em baixa prioridade.
    * Em Java a responsabilidade pela alocação de memória fica totalmente a critério da JVM.
    * Apenas os objetos de memória que não são mais necessários para sua aplicação são coletados.

Conceitos Básicos

* **Exemplo de funcionamento do Garbage Collector**



Conceitos Básicos

* **Descrição da linguagem e ambiente de desenvolvimento Java**
  * **Documentação da API Java 8**
    * https://docs.oracle.com/javase/8/docs/api/
  * **Site oficial**
    * https://www.oracle.com/br/java/index.html

Conceitos Básicos

- **Descrição da linguagem e ambiente de desenvolvimento Java**
  - **Ambientes de Desenvolvimento**
    - **Eclipse**
      - http://www.eclipse.org
    - NetBeans
    - Borland Jbuilder
    - Symantec Visual Café
    - Microsoft Visual J++
    - Asymetrix Super Cede
    - Visual Age (IBM)

Conceitos Básicos

* **Descrição da linguagem e ambiente de desenvolvimento Java**
  * **Path e Classpath**
    * A variável de ambiente PATH deve incluir o diretório contendo as ferramentas de desenvolvimento.
      * Em ambiente Windows, acrescente na variável de sistema PATH a pasta do JDK:
        * PATH=...;C:\Program Files\Java\jdk1.8.0_131\bin; ...
    * A variável de ambiente CLASSPATH deve incluir os diretórios contendo a estrutura de pacotes Java.
      * CLASSPATH=C:\Program Files\Java\jdk1.8.0_131\

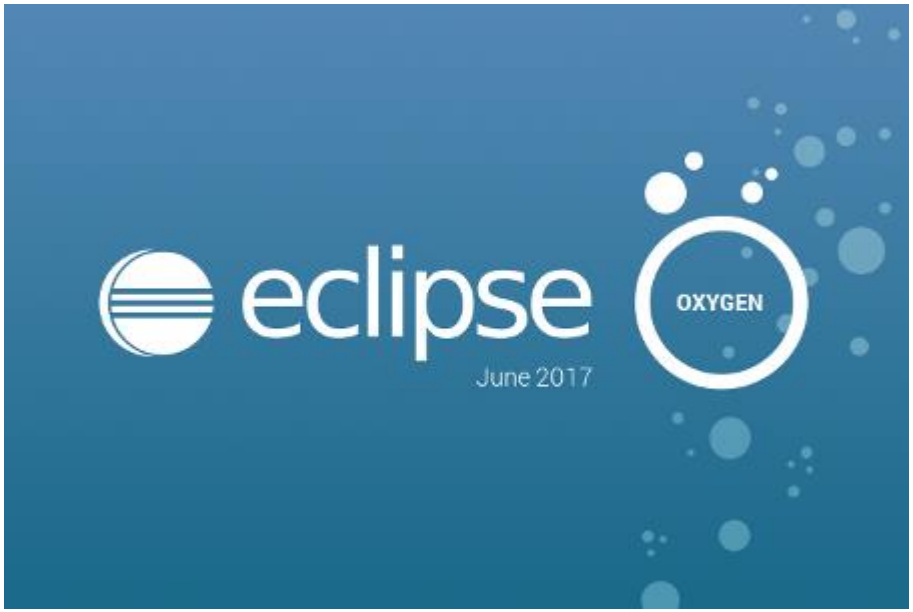
Conceitos Básicos

- **Descrição da linguagem e ambiente de desenvolvimento Java**
  - **Linha de Comando**
    - O programa fonte é um arquivo texto e pode ser digitado em qualquer editor de texto
      - NotePad (Bloco de Notas), TextPad ou outro qualquer.
    - Cada programa fonte é **compilado** usado o compilador **javac**.
      - Recebe como entrada um arquivo **.java**
      - Gera um ou mais arquivos **.class**
      - Ex: **javac** MeuPrograma.java
    - A interpretação e execução do programa são efetuadas usando o **interpretador java**
      - Recebe como entrada um arquivo **.class**
      - Executa o programa .class através da criação de uma máquina virtual. Ex: **java** MeuPrograma

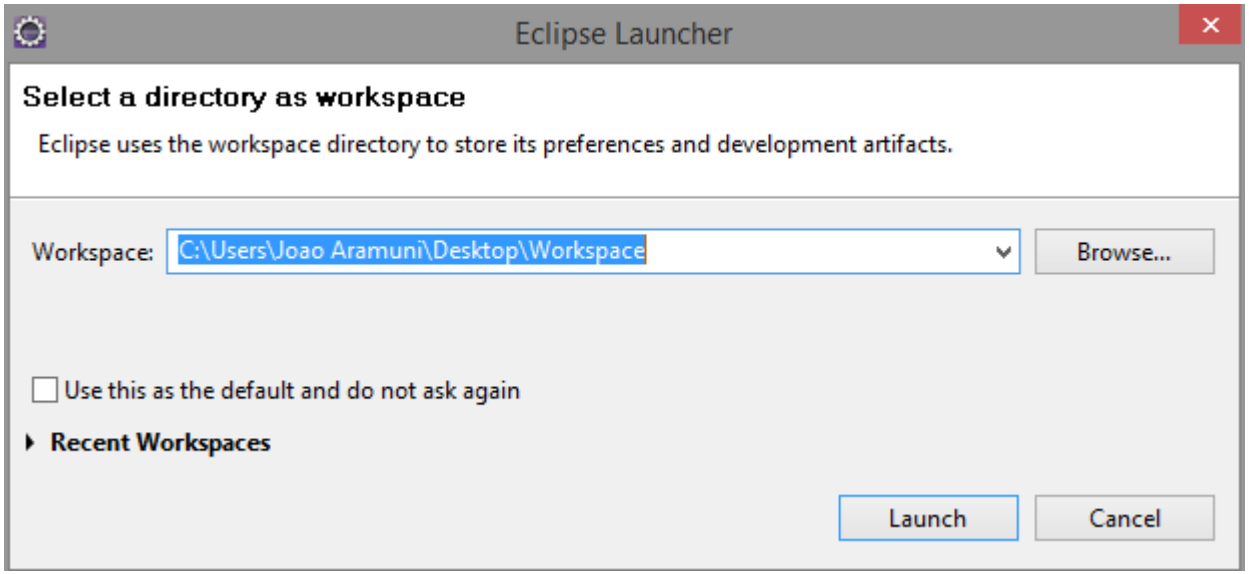
Conceitos Básicos

* **Descrição da linguagem e ambiente de desenvolvimento Java**
  * A seguir, escreveremos nossos primeiros programas em Java para começarmos a praticar as técnicas relacionadas ao paradigma da orientação a objetos.
  * Precisaremos criar um **Workspace** no diretório **U:** para guardar nossos futuros projetos Java.

* **Descrição da linguagem e ambiente de desenvolvimento Java**
  * **Abra o Eclipse:**



* **Descrição da linguagem e ambiente de desenvolvimento Java**
  * **Crie o Workspace:**



Obrigado.

joaopauloaramuni@gmail.com
joaopauloaramuni@fumec.br

UNIVERSIDADE FUMEC
50 Anos