

# **ÁREA DE LINGUAGENS DE PROGRAMAÇÃO**

**Mariza A.S. Bigonha**

**UFMG**

**25 de maio de 2019**

# Laboratório de Linguagens de Programação

---

## Um Pouco de História

**Criação: 4 de março de 1994 - Professor Roberto da Silva Bigonha**

**Coordenadores:**

**professor Roberto da Silva Bigonha**

**4 de março de 1994 até 2013**

**professora Mariza Andrade da Silva Bigonha 2013 . . .**

## Professores da Área

- **prof. Carlos Camarão (LLP)**
- **prof. Fernando Magno Quintão Pereira (LC)**
- **prof. Mariza A. S. Bigonha (LLP)**

# **DISCIPLINAS**

---

- **Disciplinas Graduação**

- Linguagens de Programação
- Compiladores I
- Compiladores II (optativa)
- Tópicos em Engenharia de Software: Modularidade (optativa)

- **Disciplinas Pós-Graduação**

- Tópicos em Linguagens: Linguagens Funcionais
- Tópicos em Linguagens: Sistemas de Tipos
- Tópicos em Compiladores: Otimização de Código
- Tópicos em Engenharia de Software: Modularidade

# Linguagens de Programação (DCC024)

---

- Oferecimento: alunos do quinto período
- Objetivos
  1. Identificar no tempo avanços metodológicos importantes:
    - Abstração de Endereços
    - Abstração de Estruturas de Controle
    - Abstração de Dados, Abstração de Tipos
    - Abstração de Componentes de Software (objetos)
  2. Ilustrar cada avanço com a apresentação de importantes LPs.
  3. Estabelecer espírito crítico em relação às LPs comparando seus avanços, acertos e erros.
  4. Conhecer as vantagens, desvantagens e limitações das LPs.
  5. Perceber o papel das LPs na qualidade de uma implementação.
  6. Entender a motivação do projeto de cada linguagem.

# **Compiladores I (DCC053)**

---

- **Oferecimento: alunos do sexto período**
- **Objetivos**  
**Ensinar técnicas básicas de construção de compiladores e conceitos relacionados com linguagens de programação.**

## **Compiladores II (DCC054/DCC888)**

---

- **Oferecimento: alunos do sétimo período**
- **Objetivos**

**Estudo de técnicas avançadas de construção de compiladores, especificamente para a geração e otimização de código em arquiteturas modernas. Concluindo o curso, os alunos deverão ser capazes de especificar um gerador de código otimizado.**

## **TES: Modularidade (DCC052/DCC890)**

---

- **Oferecimento: alunos do quinto período**
- **Objetivos**

**Mostrar um conjunto coerente de técnicas, princípios, e métodos que podem ser utilizados para desenvolver programas modulares de qualidade satisfatória, tendo certeza de que efetivamente possuem a qualidade desejada, independente do paradigma de linguagem de programação usada. Dentro desse objetivo, é apresentado**

- **Como desenvolver programas complexos de forma organizada e segura por construção e não por meio de inúmeras correções**
- **Como controlar a qualidade de cada módulo**
- **Como organizar o processo de integração dos módulos**

# Tópicos em Compiladores: Otimização de Código (DCC888)

---

- Oferecimento: pós-graduação

- Objetivos

**Estudo de técnicas avançadas de construção de compiladores, especificamente para a geração e otimização de código em arquiteturas modernas. Concluindo o curso, os alunos deverão ser capazes de especificar um gerador de código otimizado. Abordamos:**

- Visão geral da geração e otimização de código
- Linguagens de descrição de arquitetura
- Otimização Independente de Máquina
- Paralelismo de Instrução
- Alocação de Registradores (AR)
- Escalonamento de Instrução (EI)
- Interdependência entre EI e AR

# **Atividades de Pesquisa**

---

**Área do Conhecimento: Ciência da Computação**

**Sub-área do Conhecimento: Metodologia e Técnicas da Computação**

**Especialidade: Linguagens de Programação**

## **Linhas de Pesquisa**

- 1. Ferramentas para Definição, Implementação de LPs**
- 2. Metodologia e Técnicas de Computação**
  - 1. Ambientes para Definição e Implementação de Linguagens**

# Linhas de Pesquisa

---

## **Ambientes para Definição e Implementação de Linguagens**

*Esta linha investiga soluções para os problemas de projeto, definição e implementação de linguagens de programação. Em especial, no uso de computadores para automatizar essas atividades*

## **Ferramentas para Definição, Implementação de LPs**

*Esta linha compreende projetos de linguagens de programação e de desenvolvimento de ferramentas que auxiliem e, em especial, que acelerem todo o processo de desenvolvimento de compiladores e interpretadores.*

## **Metodologia e Técnicas de Computação**

*Esta linha de pesquisa compreende ambientes e ferramentas para dar suporte à programação em ponto grande, onde ressalta-se a importância dos fatores internos, modularidade e reuso, como os mais importantes para reduzir a complexidade de sistemas de grande porte. Outro fator considerado nas pesquisas realizadas é a manutenção, considerada extremamente relevante para a construção de software de qualidade.*

# **EQUIPE ATUAL**

---

## Alunos de Doutorado

- **Bruno Luan de Souza**
- **Mívia Marques Ferreira**

## Alunos de Mestrado

- **Talita Santana Orfanó**
- **Vanessa Cristiny Rodrigues Vasconcelos**
- **Rafael Prates Ferreira Trindade**
- **Cleiton Silva Tavares**
- **Charles Henrique Alvarenga**

## Professores Colaboradores

- **professora Kecia Aline Marques Ferreira (DECOM/CEFET-MG)**
- **professor Eduardo Figueiredo (DCC-UFMG)**
- **professor Elton M. Cardoso (UFOP/campus João Monlevade)**

## PROJETOS DE PESQUISA ATUAIS ...

---

**Temas centrais:** **Compiladores** e **Qualidade de *Software* no Paradigma de LP OO**

### Desafios para os Compiladores

- 1. Como possibilitar a exploração do poder crescente das DSLs, pequenas linguagens focadas em um domínio particular, se o projeto de compiladores não facilita a extensibilidade de linguagens, e se os custos de implementação e modificação de DSLs limitam o seu uso, especialmente quando deseja-se utilizar várias DSLs no mesmo sistema.*

### Objetivo Principal da Pesquisa

**Criar um modelo semântico uniforme no qual é possível expressar o comportamento de programas de forma independente da linguagem compilada.**

**Uma ferramenta com essa abordagem permitiria o compartilhamento de DSLs entre quaisquer compiladores que sejam baseados nessa ferramenta, sem o indevido esforço do programador.**

## ... PROJETOS DE PESQUISA ATUAIS ...

---

### ... Desafios para os Compiladores

2. *Como possibilitar a detecção e correção automática de erros lógicos em Haskell. Porque Haskell?*

#### Objetivo Principal da Pesquisa

Adaptar o sistema FixML, desenvolvido para OCaml, para a linguagem Haskell e automatizar a criação de testcases. FixML diagnostica e corrige erros lógicos em atribuições de programação funcional escritas em OCaml.

#### Proposta de Solução

Para localizar o erro lógico, usar um algoritmo que calcula uma pontuação estatística que representa o grau de probabilidade de o erro estar em determinada expressão. Conhecida sua localização, investigar por uma expressão correta para substituí-lo.

## ... PROJETOS DE PESQUISA ATUAIS ...

---

### Desafios com Relação a Qualidade de *Software* em Sistemas Desenvolvidos no Paradigma Orientado por Objeto

1. *Segurança e confiança de sistemas de software complexos*
2. *Medição da manutenibilidade de software*
3. *Dificuldade de validação de modelos propostos para medição da manutenibilidade de software devido à carência de dados experimentais de softwares reais*

### Objetivos Principais das Pesquisas em Qualidade de *Software* Orientados por Objeto

1. **Identificar as propriedades e a evolução das estruturas reais de software em Java**
2. **Avaliar os modelos existentes para medição de manutenibilidade de sistemas de software em Java em cenários reais de modificações de software.**

## ... PROJETOS DE PESQUISA ATUAIS ...

---

### Qualidade de *Software* em Sistemas Desenvolvidos no Paradigma OO: ...

1. *De que forma o conhecimento de um software OO é disseminado entre os diferentes contribuidores do projeto, e como a distribuição do conhecimento do software evolui?*

#### Objetivo Principal da Pesquisa

Este trabalho de pesquisa tem como principal objetivo compreender como o conhecimento dos autores em relação ao software é distribuído ao longo de seu ciclo de vida evolutivo. O escopo se delimita a softwares *open source* escritos na linguagem Java presentes nos repositórios GitHub1.

#### Proposta de Solução

1. Identificar como o conhecimento de cada autor está distribuído no projeto e como esse conhecimento evolui
2. Construir uma plataforma de software que permita a coleta e análise de dados de conhecimentos de software de projetos hospedados no GitHub

## ... PROJETOS DE PESQUISA ATUAIS ...

---

### ... Qualidade de *Software* em Sistemas Desenvolvidos no Paradigma OO

#### 2. Escassez de ferramentas para identificar *bad smells* em sistemas de software OO

##### Objetivo Principal da Pesquisa

Identificar e classificar *bad smells* em um programa de código aberto em Java, gerando um oráculo de *bad smells*.

##### Proposta de Solução

1. Realizar um Revisão Sistemática de Literatura sobre oráculos de *bad smells*
2. Desenvolver uma ferramenta para a detecção e validação colaborativa de *bad smells*
3. Identificar manualmente os *bad smells* em um conjunto de programas abertos desenvolvidos em JAVA
4. Validar o conjunto de *bad smells* com o auxílio de especialistas sobre o tema
5. Comparar o conjunto de *bad smells* com os resultados providos por ferramentas de identificação de *bad smells*

## ... PROJETOS DE PESQUISA ATUAIS ...

---

### ... Qualidade de *Software* em Sistemas Desenvolvidos no Paradigma OO

#### 3. *Identificar a natureza evolutiva dos bad smells em sistemas de software OO*

#### Objetivos Principais da Pesquisa

Realizar um estudo empírico aplicando valores de referência para métricas de software a fim de analisar a natureza evolutiva dos *bad smells* em softwares orientados por objetos, especificamente, investigar quando eles são introduzidos nos softwares.

#### Proposta de Solução

1. Definir os softwares que serão analisados no estudo empírico
2. Definir os *bad smells*, métricas, valores de referência para as métricas e as estratégias de detecção dos *bad smells*
3. Definir o modelo e estratégia de realização do estudo empírico
4. Analisar os resultados
5. Discutir os resultados e apresentar as diretrizes sobre a evolução dos *bad smells* no código

## ... PROJETOS DE PESQUISA ATUAIS ...

---

### ... Qualidade de *Software* em Sistemas Desenvolvidos no Paradigma OO

#### 4. *Análise Evolutiva de Bad Smells em Software com a Aplicação de Refatoração*

##### Objetivos Principais da Pesquisa

- Investigar se refatoração introduz *bad smells* em sistemas escritos em Java e
- Investigar se se existem ferramentas de refatoração que realmente não aumentam a quantidade de *bad smells* no código Java,
- caso exista, quais ferramentas e quais *bad smells* são suportados por elas

##### Proposta de Solução

Inicialmente será feita uma revisão sistemática da literatura centrada em refatoração, nos *bad smells* de Fowler, e na coocorrência entre esses *bad smells* e refatoração.

## ... PROJETOS DE PESQUISA ATUAIS ...

---

### ... Qualidade de *Software* em Sistemas Desenvolvidos no Paradigma OO

#### 5. *Análise da Evolução de Métricas OO em Sistemas de Software de Código Aberto*

##### Objetivos Principais da Pesquisa

Investigar como a qualidade interna de sistemas de software OO evolui, considerando métricas. Analisar como o comportamento de métricas pode ser descrito sobre a evolução do software, por exemplo: (1) se afeta a capacidade de reutilização e complexidade dos sistemas; (2) e a porcentagem de classes dos sistemas que impactam diretamente na evolução das métricas escolhidas.

##### Proposta de Solução

Inicialmente definimos uma metodologia que consiste em aplicar técnicas de regressão linear e testes de tendência em métrica de acoplamento para identificar a função que melhor explica o comportamento da evolução do acoplamento e os componentes dos sistemas que impactam no crescimento ou diminuição do acoplamento. Depois estenderemos a metodologia para outras métricas.

## ... PROJETOS DE PESQUISA ATUAIS ...

---

### ... Qualidade de *Software* em Sistemas Desenvolvidos no Paradigma OO

#### 6. *Investigação e Análise do Impacto de Mudanças na Prática de Manutenção de Software*

##### Objetivo Principal da Pesquisa

Este trabalho está concentrado na lacuna entre as técnicas de manutenção de software propostas pela academia e a prática de manutenção de software.

##### Proposta de Solução

Inicialmente investigamos como os profissionais entendem e aplicam os seguintes conceitos e técnicas principais relacionados à manutenção de software: *bad smells*, refatoração, métricas de software e análise de impacto de mudanças.

# Página do Laboratório de Linguagens de Programação

---

<http://www.llp.dcc.ufmg.br/>

**FIM**