

ES46A - Arquitetura de Software

Implementação DP

DACOM – Engenharia de Computação

Diego Addan

UTFPR - 2023

Apresentação

Implementação do PROJETO 1 -

Refatoração utilizando padrões de projeto

17 a 22/10/2023

Novembro e Avaliação 2 (Projeto e Seminário)

Diego Addan

https://www.inf.ufpr.br/dagoncalves/_disciplinas.html

PROJETO PEDAGÓGICO DO CURSO

Quadro 96 – Unidade Curricular Arquitetura de Software

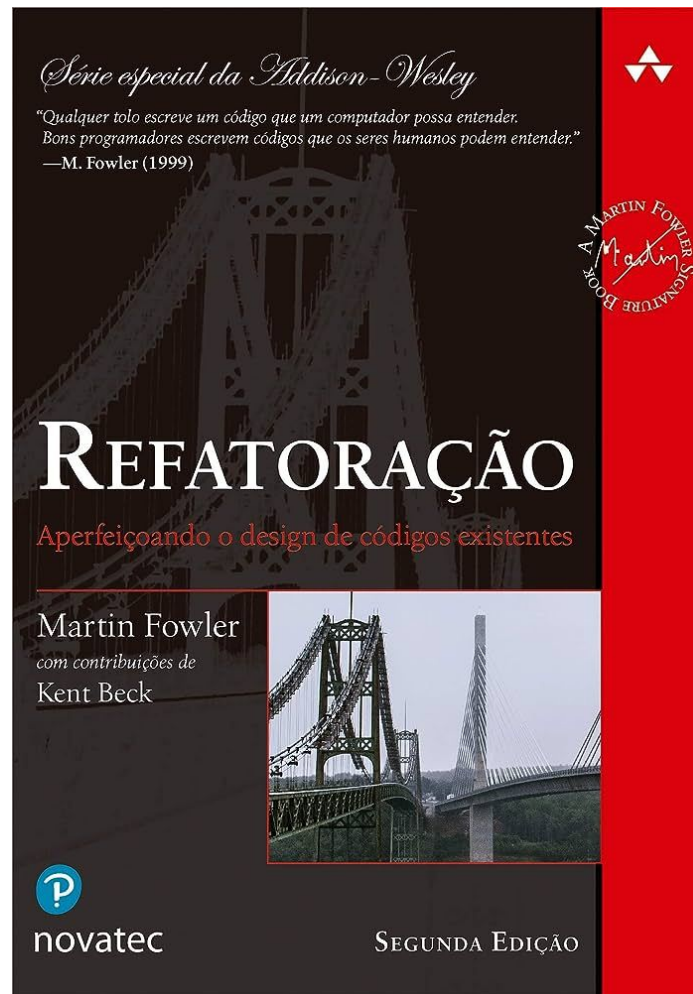
Unidade curricular	Arquitetura de Software
Núcleo	Específico
Carga-horária (Horas)	60
Temas de Estudo	
<ul style="list-style-type: none">▷ TE1: Orientação a objeto: bibliotecas de orientação a objeto, classes, métodos, acoplamento, coesão, manutenabilidade, métricas e aplicações (15h).▷ TE2: Bibliotecas e padrões: bibliotecas para organização de código, padrões de projeto e visualização da arquitetura (15h).▷ TE3: Arquitetura web: camadas, padrões web e REST (15h).▷ TE4: Estilos arquiteturais: nuvem (escalabilidade), eventos, filas de mensagens e microserviços (15h) .	
Resultados de Aprendizagem	
<ul style="list-style-type: none">▷ RA1: Identificar o nível de acoplamento, coesão e manutenabilidade de projetos de software orientado a objetos, considerando diferentes ferramentas, linguagens e tecnologias (TE1).▷ RA2: Entender os cenários nos quais técnicas de reuso de bibliotecas e padrões favorecem a arquitetura de projetos de software orientado a objetos, de forma adaptativa às novas tecnologias (TE2).▷ RA3: Refatorar a arquitetura de aplicações para uso de camadas, padrões web e REST, com qualidade e robustez para diferentes plataformas (TE3).▷ RA4: Entender os benefícios e as mudanças necessárias no uso de nuvem, eventos, filas de mensagens e microserviços, com objetividade e clareza (TE4).	

Refatoração de Código

Martin Fowler (+20 anos)

Necessidade do mercado

Alteração no código que não altera o
Comportamento observável do programa



Refatoração de Código

Alterações pontuais - Modulares

Integração ~ Novas tecnologias e adaptações

Refatoração é um processo presente na pesquisa e naturalmente criativo

Refatoração de Código

RESEARCH-ARTICLE



Understanding Machine Learning Practitioners' Data Documentation Perceptions, Needs, Challenges, and Desiderata

Authors: [Amy K. Heger](#), [Liz B. Marquis](#), [Mihaela Vorvoreanu](#), [Hanna Wallach](#), [Jennifer Wortman Vaughan](#)

[Authors Info & Claims](#)

Proceedings of the ACM on Human-Computer Interaction, Volume 6, Issue CSCW2 • Article No.: 340, pp 1–29

• <https://doi.org/10.1145/3555760>

RESEARCH-ARTICLE



Refactoring a full stack web application to remove barriers for student developers and to add customization for instructors

Authors: [Jack Cook](#), [Richard Weiss](#), [Jens Mache](#) [Authors Info & Claims](#)

Journal of Computing Sciences in Colleges, Volume 36, Issue 1 • pp 35–44

Published: 01 October 2020 [Publication History](#)

Refatoração de Código

- **Reorganizar sub-processos e módulos**
- **Eliminar redundâncias**
- **Facilitar adição de novos recursos**
- **Melhorar o projeto a nível código**
- **Adequar um sistema a arquitetura do projeto**

Refatoração de Código

Padrões de Projeto

Criação / Estrutura / Comportamento

Caminhos de refatoração (KERIEVSKY, 2008, p.57)

- **Aplicação de um padrão de projeto a um sistema**
- **Contrária a um padrão, baseada em otimização**

Refatoração de Código

Observação de interações e
Sub-processos

Ex: Federated Learning

Extract Class



```
class Person {
    get officeAreaCode() {return this._officeAreaCode;}
    get officeNumber() {return this._officeNumber;}
}
```



```
class Person {
    get officeAreaCode() {return this._telephoneNumber.areaCode;}
    get officeNumber() {return this._telephoneNumber.number;}
}
class TelephoneNumber {
    get areaCode() {return this._areaCode;}
    get number() {return this._number;}
}
```

Fonte: <https://refactoring.com/catalog/extractClass.html>

Refatoração de Código

Figura 2 – Candidato Wei *et al.* (2014)

```
//Create object of database logger
if(type == 'D'){
    //Connect database, code is omitted
    Logger logger = new database logger();
    //Initialize database logger, code is omitted
    return logger;
}
//Create object of file logger
else if(type == 'F'){
    //Create log file, code is omitted
    Logget logger = new Filelogger();
    //Initialize file logger, code is omitted
    return logger
}
else{
    return null;
}
```

Fonte – Adaptado de Wei *et ai.* (2014)

Figura 3– Refatoração Wei *et al.* (2014)

```
public abstract class LoggerFactory {
    public abstract Logger createLogger();
}
public class DatabaseLoggerFactory extends LoggerFactory {
    public Logger createLogger() {
        return new DatabaseLogger();
    }
}
public class FileLoggerFactory extends LoggerFactory {
    public Logger createLogger() {
        final Logger logger = new FileLogger();
        return logger;
    }
}
```

Fonte – Beluzzo (2018)

Refatoração de Código



Fonte – Beluzzo (2018)



Refatoração de Código



Fonte – Beluzzo (2018)

Refatoração de Código

Métodos de refatoração rumo a um Padrão (Design Pattern)

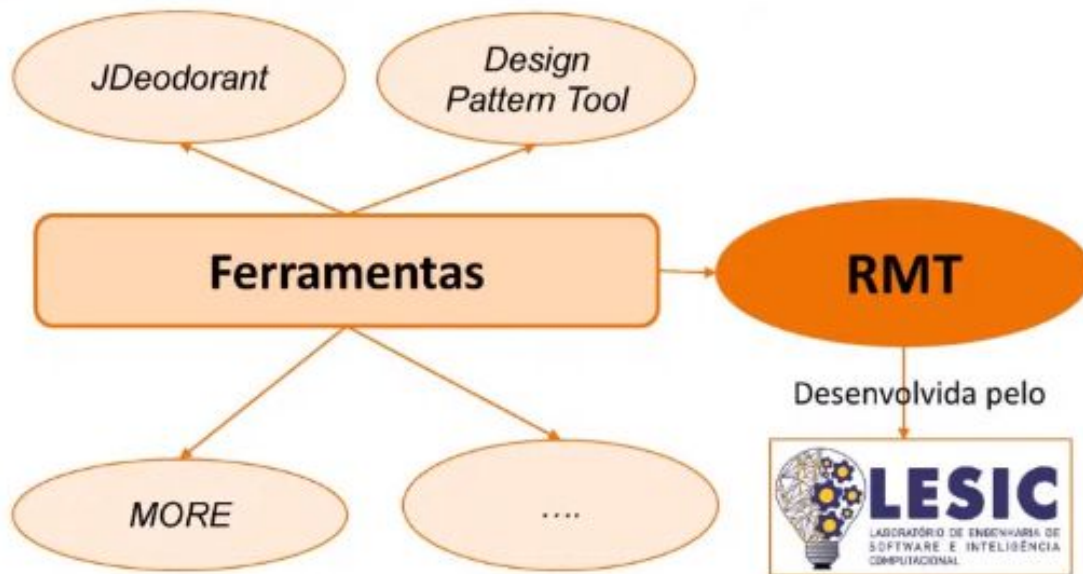
Gaitani et al. (2015)

Cinneide (2011)

Zafeiris et al. (2017)

Automatizações:

React Native



Refatoração de Código

Projeto 1

- **Seguir a modelagem: Mas aproveite que os padrões são adaptativos!**
- **Otimizar e simplificar**

Correções e notas!

Referências

T. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. Introdução a Algoritmos Complexos, Segunda Edição, The MIT Press, 2005