

Template de TCC-Pesquisa na Engenharia de Software: Formato da Sociedade Brasileira de Computação (SBC)

André Rodrigues de Freitas Faria¹

¹Instituto de Ciências Exatas e Informática - Pontifícia Universidade Católica de Minas Gerais (PUC MINAS) R. Dom José Gaspar, 500 - Coração Eucarístico, Belo Horizonte - MG, 30535-901

Abstract. *This meta-paper describes the style to be used in articles and short papers for SBC conferences. For papers in English, you should add just an abstract while for the papers in Portuguese, we also ask for an abstract in Portuguese (“resumo”). In both cases, abstracts should not have more than 10 lines and must be in the first page of the paper.*

Resumo. *Este meta-artigo descreve o estilo a ser usado na confecção de artigos e resumos de artigos para publicação nos anais das conferências organizadas pela SBC. É solicitada a escrita de resumo e abstract apenas para os artigos escritos em português. Artigos em inglês deverão apresentar apenas abstract. Nos dois casos, o autor deve tomar cuidado para que o resumo (e o abstract) não ultrapassem 10 linhas cada, sendo que ambos devem estar na primeira página do artigo.*

Bacharelado em Engenharia de Software - PUC Minas Trabalho de Conclusão de Curso (TCC)

Orientador de conteúdo (TCC I): Danilo Boechat - daniloboecat@pucminas.br
Orientador de conteúdo (TCC I): Joana Souza - joanasouza@pucminas.br
Orientador de conteúdo (TCC I): Leonardo Vilela - leonardocardoso@pucminas.br
Orientador acadêmico (TCC I): Cleiton Tavares - cleitontavares@pucminas.br
Orientador do TCC II: (A ser definido no próximo semestre)

Belo Horizonte, DIA de MÊS de ANO.

1. Introdução

A arquitetura de software é uma área da engenharia de software responsável por definir e projetar a estrutura do sistema, definindo seus componentes e a relação entre eles, sendo tal definição tanto numa camada macro quanto micro do software. Pode-se fazer uma analogia entre a arquitetura de software e a arquitetura propriamente dita onde a materialidade do software se vê a partir dos componentes e módulos do sistema. Segundo Martin Fowler: “Arquitetura de software é o conjunto de estruturas necessárias para raciocinar sobre um sistema de software”.

Por ser uma das partes mais cruciais de um software, uma arquitetura mal feita pode acarretar em vários problemas. Um dos modelos mais tradicionais de desenvolvimento é chamado de monolito, onde o sistema consiste em apenas um módulo responsável por resolver todos os problemas do software. O modelo monolítico pode levar o software

a ter vários problemas de escalabilidade, manutenção e desenvolvimento. Monolito distribuído foi um modelo arquitetural criado para tentar contornar os problemas do monolito tradicional, porém, tal prática levou apenas a mais problemas de dependência entre os módulos do monolito. Essa pesquisa tem como objetivo entender o real impacto de escalabilidade e desenvolvimento que um modelo de monolito distribuído possui.

Monolitos distribuídos podem causar grandes problemas a um sistema de software, entender seu real impacto é de grande importância. Muitos softwares grandes no mercado ainda utilizam tecnologias antigas e principalmente arquiteturas antigas e ultrapassadas com o monolito distribuído. Entender o real impacto do monolito distribuído pode ser de grande ajuda para ajudar desenvolvedores a migrarem seu sistema para uma arquitetura de microsserviços visto que, muitas vezes, um monolito distribuído muitas vezes é a tentativa fracassada de desenvolver microsserviços

link: <https://www.overleaf.com/read/wcbtkjzggwnq8b2fcd>

2. Fundamentação Teórica

2.1. Images

All images and illustrations should be in black-and-white, or gray tones, excepting for the papers that will be electronically available (on CD-ROMs, internet, etc.). The image resolution on paper should be about 600 dpi for black-and-white images, and 150-300 dpi for grayscale images. Do not include images with excessive resolution, as they may take hours to print, without any visible difference in the result.

2.2. Figures and Captions

Figure and table captions should be centered if less than one line (Figure 1), otherwise justified and indented by 0.8cm on both margins, as shown in Figure 2. The caption font must be Helvetica, 10 point, boldface, with 6 points of space before and after each caption.



Figura 1. A typical figure



Figura 2. This figure is an example of a figure caption taking more than one line and justified considering margins mentioned in Section 2.2.

3. Trabalhos Relacionados

3.1. Sections and Paragraphs

Section titles must be in boldface, 13pt, flush left. There should be an extra 12 pt of space before each title. Section numbering is optional. The first paragraph of each section should not be indented, while the first lines of subsequent paragraphs should be indented by 1.27 cm.

3.1.1. Subsections

The subsection titles must be in boldface, 12pt, flush left.

3.2. Tables

In tables, try to avoid the use of colored or shaded backgrounds, and avoid thick, doubled, or unnecessary framing lines. When reporting empirical data, do not use more decimal digits than warranted by their precision and reproducibility. Table caption must be placed before the table (see Table 1) and the font used must also be Helvetica, 10 point, boldface, with 6 points of space before and after each caption.

4. Materiais e Métodos

4.1. Exemplo de Listing

```

1 public class Phone {
2     private final String unformattedNumber;
3
4     public String getNumber() {
5         return unformattedNumber.substring(6,10);
6     }
7 }

```

Listing 1. Code Example

4.2. Exemplo de Algorithm

Tabela 1. Variables to be considered on the evaluation of interaction techniques

	Chessboard top view	Chessboard perspective view
Selection with side movements	6.02 ± 5.22	7.01±6.84
Selection with in- depth movements	6.29±4.99	12.22±11.33
Manipulation with side movements	4.66± 4.94	3.47±2.20
Manipulation with in- depth movements	5.71 ±4.55	5.37 ±3.28

Algorithm 1: Código fonte em Java

```

1 void printOwing (double amount){
2     print Banner();
3     //print details
4     System.out.println("name: " + _name);
5     System.out.println("amout: " + amount);
6 }

```

Tabela 2. Distribuição de tarefas por datas

Tarefas	2025						
	fevereiro		março		abril		maio
	1ªQ	2ªQ	1ªQ	2ªQ	1ªQ	2ªQ	1ªQ
Desenvolvimento do <i>script</i> para coleta dos repositórios	X						
Coleta de dados dos repositórios		X					
<i>Desenvolvimento da solução</i>		X					
Coleta de dados			X	X			
Aplicação do Método			X	X	X		
Geração dos painéis						X	
Discussão e avaliação dos resultados							X

4.3. Exemplo de Cronograma

Esta seção apresenta um exemplo de tabela para construção de cronograma.

4.4. References

Bibliographic references must be unambiguous and uniform. We recommend giving the author names references in brackets, e.g. [Knuth 1984], [Boulic and Renault 1991], and [Smith and Jones 1999].

Cormen et al. (2016) representa uma citação direta.

The references must be listed using 12 point font size, with 6 points of space before each reference. The first line of each reference should not be indented, while the

subsequent should be indented by 0.5 cm.

Referências

Boulic, R. and Renault, O. (1991). 3d hierarchies for animation. In Magnenat-Thalmann, N. and Thalmann, D., editors, *New Trends in Animation and Visualization*. John Wiley & Sons Ltd.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2002). Algoritmos: teoria e prática. *Editora Campus*, 2:296.

Knuth, D. E. (1984). *The T_EX Book*. Addison-Wesley, 15th edition.

Smith, A. and Jones, B. (1999). On the complexity of computing. In Smith-Jones, A. B., editor, *Advances in Computer Science*, pages 555–566. Publishing Press.

Apêndice

A. Exemplo de Seção em um Apêndice