



UMA AVALIAÇÃO DO ENSINO DE MÉTODOS FORMAIS NO BRASIL

Trabalho de Conclusão de Curso
Engenharia da Computação

Gemerson Gerardo Feitosa Gonçalves

Orientador: Prof. Gustavo Henrique Porto de Carvalho



Gemerson Gerardo Feitosa Gonçalves

UMA AVALIAÇÃO DO ENSINO DE MÉTODOS FORMAIS NO BRASIL

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia de Computação pela Escola Politécnica de Pernambuco - Universidade de Pernambuco

Universidade de Pernambuco

Escola Politécnica de Pernambuco

Graduação em Engenharia de Computação

Orientador: Prof. Gustavo Henrique Porto de Carvalho

Recife - PE, Brasil

29 de novembro de 2016

MONOGRAFIA DE FINAL DE CURSO

Avaliação Final (para o presidente da banca)*

No dia 30 de 12 de 2016, às 10:00 horas, reuniu-se para deliberar a defesa da monografia de conclusão de curso do discente **GEMERSON GERARDO FEITOSA GONCALVES**, orientado pelo professor **Gustavo Henrique Porto de Carvalho**, sob título **UMA AVALIAÇÃO DO ENSINO DE MÉTODOS FORMAIS NO BRASIL**, a banca composta pelos professores:

Joabe Bezerra de Jesus Júnior

Gustavo Henrique Porto de Carvalho

Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

Aprovada Aprovada com Restrições* Reprovada

e foi-lhe atribuída nota: 9,5 (nove e meia)

*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O discente terá 5 dias para entrega da versão final da monografia a contar da data deste documento.

JOABE BEZERRA DE JESUS JÚNIOR

GUSTAVO HENRIQUE PORTO DE CARVALHO

“Quando os problemas parecem absurdos, os desafios se tornam apaixonantes.”
(Dom Hélder Câmara)

Agradecimentos

Inicialmente, gostaria de agradecer a Deus, por ter iluminado a minha mente e ter me guiado pelo caminho até aqui, caminho este que me fez conhecer pessoas excepcionais e que tanto contribuíram para o meu crescimento pessoal e profissional.

Aos meus pais: Maria Sineyde Gonçalves Feitosa e Gerson Gerardo Feitosa e minha irmã: Edyenis Gonçalves Feitosa. Por sempre me encorajarem e ampararem no decorrer da minha vida, e sem os quais eu jamais conseguiria alcançar o êxito de concluir mais um curso superior.

Em especial a minha namorada, amiga e companheira dos últimos 11 anos: Monique Cardoso Santos. Por entender meus problemas, me ajudar nas soluções, abraçar quando preciso, reclamar se necessário. Juntamente com meus pais, você é uma grande fonte de inspiração para batalhar na vida e sempre observar a felicidade e alegria que nela existe. Principalmente, perante o pequeno problema de saúde nos últimos meses.

Ao meu orientador Gustavo Henrique Porto de Carvalho, por ser um dos melhores professores que encontrei em toda minha vida acadêmica, e o mais disciplinado orientador dos quais já tive a oportunidade de trabalhar.

Por fim, e não menos importante, aos amigos que fiz no curso de Engenharia da Computação da UPE: Breno Lapada, Caio Mínimo, Eduardo Retirante, Eva Solange, Felipe Jorjão, Fernando GibaHarry, Hartur L Barreto e Joninhas Eterno, pelas incontáveis horas de estudos, trabalhos e piadas. Aos recentes amigos que fiz no C.E.S.A.R e o total apoio que me deram nos últimos meses.

Resumo

Métodos Formais são técnicas que utilizam modelos matemáticos para especificar, analisar e verificar propriedades de um sistema, sendo importante o seu conhecimento pelos futuros cientistas e engenheiros da computação. Apesar da sua importância, nem sempre este é um conteúdo ensinado em cursos de computação no Brasil. Portanto, este trabalho tem como objetivo realizar um levantamento sobre o ensino de Método Formais nos cursos de ensino superior de computação do Brasil e analisar suas características de forma qualitativa e quantitativa. Foram analisados os sites das 30 primeiras universidades do ranking RUF, considerando somente cursos de computação, em busca de disciplinas de Métodos Formais, catalogando os dados presentes nas descrições dos mesmos. No levantamento, verificou-se que vários cursos não apresentam disciplinas de Métodos Formais em sua grade curricular, o que confirma a expectativa inicial desta pesquisa, além de perceber uma dificuldade em encontrar informações sobre as características das disciplinas pela pesquisa por sites. Desta maneira pode-se observar como ainda não é dada a devida importância para o ensino de Métodos Formais no cursos superiores de computação

Palavras-chave: métodos formais, ensino, matriz curricular.

Abstract

Formal Methods are techniques that use mathematical models in order to specify, analyse and verify the properties of a system; the knowledge of which is important to forthcoming scientists and computer engineers. Despite its importance, this subject is seldom taught in computing courses in Brazil. This survey has, therefore, the objective of gauging the teaching of Formal Methods in computing undergraduate courses in Brazil and analysing its characteristics in quantitative and qualitative ways. The first thirty universities in the RUF ranking were analysed, considering only computing courses, in search of Formal Methods disciplines, cataloging data found in their descriptions. In this data collection, we concluded that many of them do not appear to have formal methods disciplines on their programs, which confirms the initial suspicions of this survey and also finds that there is some difficulty in finding information about the characteristics of the disciplines through the website's search tool. This way, we can observe how the appropriate importance is not yet given to the teaching of Formal Methods in computing undergraduate courses.

Keywords: formal methods, teaching, school program matrix.

Lista de ilustrações

Figura 1 – Elementos da Rede de Petri	17
Figura 2 – Exemplo de implementação	18
Figura 3 – Cursos que oferecem disciplinas de MF	30
Figura 4 – Quantidade de cursos por área	30
Figura 5 – Obrigatoriedade das disciplinas	32
Figura 6 – Quantidade de cursos que informam o período da disciplina	33
Figura 7 – Disciplinas do curso de EC da UNICAMP	35
Figura 8 – Disciplinas do curso de CC da UNICAMP	36
Figura 9 – Erro ao acessar o portal do departamento de CC	36
Figura 10 – Erro no Siga de CC da UFRJ	37
Figura 11 – Erro no Siga de EC da UFRJ	37
Figura 12 – Erro no certificado da página	41

Lista de abreviaturas e siglas

CC	Ciência da Computação
EC	Engenharia de Computação
ES	Engenharia de Software
MF	Métodos Formais
RUF	Ranking Universitário Folha
THE	Times Higher Education
UML	Unified Modeling Language

Sumário

1	INTRODUÇÃO	10
1.1	Objetivos	11
1.1.1	Objetivos Específicos	11
1.2	Estrutura da monografia	12
2	REFERENCIAL TEÓRICO	13
2.1	Métodos Formais	13
2.1.1	Áreas de aplicação de métodos formais	14
2.1.2	Tipos de métodos formais	15
2.2	Exemplo de alguns métodos formais	16
2.2.1	Redes de Petri	16
2.2.2	Circus	17
2.3	Adoção, crescimento e impacto na indústria	18
2.4	O ensino de métodos formais	20
3	MÉTODO DE PESQUISA	24
3.1	Revisão bibliográfica	24
3.2	Seleção do ranking	25
3.3	Definição da metodologia de coleta de dados	25
3.4	Definição da metodologia de análise de dados	27
3.5	Levantamento dos dados	27
3.6	Limitações da pesquisa	28
4	RESULTADOS	29
4.1	Visão consolidada de todos os cursos pesquisados	29
4.2	Detalhamento das 5 primeiras universidades do RUF	35
4.2.1	Universidade Estadual de Campinas (UNICAMP)	35
4.2.2	Universidade Federal do Rio de Janeiro (UFRJ)	35
4.2.3	Universidade de São Paulo (USP)	38
4.2.4	Universidade Federal do Rio Grande do Sul (UFRGS)	40
4.2.5	Universidade Federal de Minas Gerais (UFMG)	42
5	CONCLUSÃO	43
5.1	Trabalhos Futuros	44
	REFERÊNCIAS	45

1 Introdução

A importância da tecnologia para o desenvolvimento da sociedade atual é incontestável. O desenvolvimento da interação homem-programa-máquina é a base de grandes avanços em diversas áreas de conhecimento, em um mecanismo complexo de manipulação de dados e informações que prescinde de rapidez e eficiência (JANOWSKI; BEDNARCZYK, 2016). Para tal processo de interação são utilizados sistemas computacionais (ALMEIDA et al., 2011a).

Entretanto, é notória a dificuldade em se desenvolver um sistema totalmente isento de falhas, principalmente quando este é complexo e envolve riscos elevados ao ser humano e ao meio ambiente (BABIN; AIT-AMEUR; PANTEL, 2016). Como exemplo, tem-se os sistemas críticos da indústria aeronáutica, aeroespacial, hospitalar e financeira, onde uma falha pode gerar consequências graves, envolvendo a potencial perda da vida de indivíduos e elevados prejuízos financeiros. Portanto, é importante que estes sistemas sejam desenvolvidos com extrema cautela e precisão, em particular, apoiado por um processo que minimize a quantidade de falhas (KUSAKABE et al., 2014).

Usar linguagem natural, como a língua portuguesa, na especificação de sistemas complexos pode gerar ambiguidade na expressão das ideias. Linguagens como UML, apesar de apresentarem uma sintaxe bem definida, não são matematicamente precisas (GLINZ, 2000). Diante deste cenário, são necessárias técnicas que priorizem a especificação precisa: métodos formais. Métodos Formais são procedimentos desenvolvidos a partir de princípios matemáticos com o objetivo de auxiliar no projeto, implementação, teste e concepção de software e hardware, aplicando técnicas de especificação formal, refinamento e verificação (KUSAKABE et al., 2014; WOODCOCK et al., 2009). Para apoiar o emprego de tais técnicas, pode-se fazer uso de ferramentas auxiliares como Skevoulis e Falidas (2002), explica detalhadamente. Tornam-se, deste modo, viáveis o teste e a elaboração de sistemas com uma reduzida taxa de erro.

Métodos Formais (MF) constituem, portanto, em uma abordagem matemática alternativa para garantia de confiabilidade, a qual a indústria vem progressivamente dando mais atenção (ALMEIDA et al., 2011a). MF está gradativamente saindo da situação em que era utilizado exclusivamente no meio acadêmico para ser efetivamente utilizado particularmente onde sistemas críticos têm relevância (ALMEIDA et al., 2011a; CRISTIÁ, 2006). No Brasil, onde o sistema financeiro, por exemplo, necessita de muita tecnologia para acompanhar o crescimento das movimentações financeiras continentais e reduzir fraudes, os métodos formais são de extrema importância (BEEK; GNESI; KNAPP, 2016). Nesse contexto de sistemas críticos, de precisão ou de risco, observa-se a importância da

aprendizagem de MF, em particular, em cursos superiores de computação.

A melhoria da qualidade de software pode ser obtida equipando os desenvolvedores com modos disciplinados de pensar via educação em MF (LIU et al., 2009), sendo o aprendizado e domínio de tais conceitos e ferramentas o ponto crucial. Para o desenvolvimento da indústria de software, necessário se faz formar recursos humanos os mais qualificados possíveis. O ensino de MF agrega, portanto, neste sentido. Na indústria algumas barreiras para a adoção de MF são a falta de profissionais preparados para aplicá-los, além dos custos de treinamento (CRISTIÁ, 2006). Isto ocorre, pois nem sempre o ensino destes conceitos está presente nos cursos superiores de engenharia e ciência de computação, o que gera dificuldades em encontrar profissionais com tal conhecimento no mercado de trabalho.

Ensinar MF é barato, uma vez que é possível lecionar boa parte dos métodos formais utilizando ferramentas gratuitas, ou licenças acadêmicas. Portanto, estudantes universitários podem cursar disciplinas de MF sem incorrer em custos adicionais, o que faz das faculdades uma espécie de centro de treinamento para a indústria (CRISTIÁ, 2006).

Para possibilitar que os estudantes se interessem e aprendam MF, é necessário demonstrar os claros benefícios destes conceitos no planejamento, desenvolvimento e manutenção de sistemas computacionais. Igualmente importante é o uso de estilos e técnicas de ensino adequadas, além de um sensato arranjo curricular (LIU et al., 2009). Portanto, o problema de pesquisa deste trabalho consiste em responder a seguinte pergunta: como é o ensino de métodos formais em cursos de graduação em computação no Brasil?

1.1 Objetivos

Conforme exposto, é importante entender como se dá o ensino de métodos formais na formação dos profissionais da área de computação, principalmente no que concerne ao cenário nacional, com o propósito de identificar estratégias de melhorias quanto ao modo de lecionar tal disciplina. O objetivo geral proposto neste trabalho, portanto, consiste em investigar como é ensinado métodos formais, em particular nos cursos de graduação em engenharia e ciência da computação no Brasil.

1.1.1 Objetivos Específicos

O objetivo geral desdobra-se nos seguintes objetivos específicos:

1. Elaborar um questionário de coleta de dados sobre o ensino de métodos formais;
2. Verificar as linguagens formais mais comumente ensinadas;
3. Elencar as ferramentas utilizadas no ensino de métodos formais;

4. Identificar os instrumentos de avaliação utilizados;
5. Identificar em que momento a disciplina é ofertada na grade curricular;
6. Criar um panorama do ensino de métodos formais a partir dos dados coletados;

1.2 Estrutura da monografia

Esta monografia está dividida em 5 partes. No Capítulo 2, a fundamentação teórica aborda conceitos-chave de Métodos Formais, exemplos de alguns métodos conhecidos, levantamentos realizados sobre o impacto na indústria e estudos similares ao desenvolvido nesta monografia.

No Capítulo 3 são descritas as etapas do processo utilizado para o levantamento das informações sobre o ensino de MF no Brasil; ou seja, apresenta a metodologia deste trabalho. O Capítulo 4 apresenta os resultados obtidos a partir da análise quantitativa e qualitativa da pesquisa.

Por fim, o Capítulo 5 encerra o trabalho com uma reflexão a partir dos resultados obtidos, bem como discute trabalhos futuros relevantes.

2 Referencial Teórico

As próximas seções apresentam a fundamentação teórica que contribui para o bom entendimento deste trabalho. A Seção 2.1 apresenta conceitos fundamentais de MF, enquanto que a Seção 2.2 ilustra estes conceitos considerando alguns métodos formais existentes. Em seguida, a Seção 2.3 discute sobre a adoção de MF na prática, na indústria. Por fim, a Seção 2.4 detalha estudos similares ao reportado nesta monografia.

2.1 Métodos Formais

Um desafio econômico e tecnológico é dominar a complexidade e os custos associados ao desenvolvimento de software. Cada vez mais há dependência de sistemas computacionais, e a complexidade dos mesmos cresce rapidamente. Manter a confiabilidade em softwares de uso intensivo e críticos é de extrema importância, pois as consequências de erros podem envolver prejuízos financeiros de grande porte, e principalmente risco a vidas humanas. Esse desafio deu origem à área de engenharia de software (ES), que abrange: linguagens, compiladores, ferramentas, métodos de desenvolvimento, técnicas de programação, métodos relacionados ao controle de qualidade, métodos de design, dentre outros tópicos que apoiam o planejamento, o desenvolvimento e a evolução de produtos de software.

Emergindo com os primeiros computadores e a própria área de ES, as ideias fundamentais sobre MF são estudadas desde os anos de 1960. Sua aplicação prática necessitou de desenvolvimento teórico e prático (apoio ferramental), cujos princípios e algoritmos foram sendo compreendidos ao longo das décadas seguintes. Mas por que o uso de Métodos Formais?

O uso de linguagem natural, para especificar sistemas complexos, pode gerar ambiguidade em função de possíveis interpretações diferentes. Até linguagens semi-formais, como UML, podem apresentar ambiguidade. No caso específico de UML, apesar de ser facilmente entendido por muitos, como a sua semântica é descrita em linguagem natural, casos particulares podem ter interpretações múltiplas, devido à informalidade e imprecisão da descrição (BORGES; MOTA, 2007; VALLES-BARAJAS, 2012).

As múltiplas interpretações podem levar a erros de grande magnitude, como presente no relatório da NASA elaborado por Denney (2005): este descreve projetos espaciais que tiveram falhas por detalhes que poderiam ter sido observados com o uso de Métodos Formais. Alguns exemplos mencionados pelo documento: falha do foguete Ariane 5 em 1996, onde um erro na conversão do software de 64bits para 16bits fez o foguete proceder em uma trajetória errada e se alto destruindo; Sonda espacial Spirit, que em 2004 ficou 10

dias desligada na superfície de Marte por uma sobrecarga na memória ROM, gerando um prejuízo de 4 milhões de dólares por dia.

Outros casos conhecidos foram os erros com os processadores Pentium (1993), que em algum momento do uso criavam erros em operações com ponto flutuante, fazendo com que a Intel gastasse 475 milhões de dólares em *recall* e processos movidos contra a mesma. Por fim, outro caso famoso foi a máquina de radioterapia Therac-25, onde de 1985 até 1987 foram confirmados 6 casos de pacientes que receberam grande dose de radiação, com 5 pessoas indo a óbito, por conta de um erro no software de controle da máquina.

Métodos de especificação formal permitem a descrição de sistemas complexos partindo de entidades abstratas, escrevendo especificações rigorosas, precisas e completas. Portanto, destacam-se os seguintes benefícios com a adoção de métodos formais:

- Semântica precisa, baseada em fundamentos matemáticos;
- Possibilidade de provar que um sistema satisfaz determinadas especificações;
- Gera uma boa base para o desenvolvimento de ferramentas de suporte;
- Menos tempo e esforços relacionados a testes, manutenção e codificação.

Em relação a este último tópico, [Hall \(1990\)](#) explica que com métodos formais é comum empregar mais tempo detalhando a especificação do que programando, o que é potencialmente interessante, uma vez que é menos custoso consertar problemas em etapas iniciais de desenvolvimento.

2.1.1 Áreas de aplicação de métodos formais

Métodos Formais podem ser aplicados em 4 principais áreas ([ALMEIDA et al., 2011a](#); [MONIN; HINCHEY, 2003](#)): especificação formal, prova formal, verificação de modelos, e processos de abstração. A seguir, estas áreas são explicadas.

- Especificação formal: especificação formal é o momento em que uma descrição não matemática (como texto em linguagem natural, diagramas e tabelas) é transformada em uma linguagem formal bem definida, tanto sintaticamente quanto semanticamente, descrevendo as propriedades de alto nível do sistema([ALMEIDA et al., 2011a](#)).
- Prova formal: utilizam-se argumentos para validar alguma propriedade do sistema descrito através de uma série de passos, onde cada um desses é determinado por pequenos conjuntos de regras, eliminando deste modo a ambiguidade das conclusões informais. Geralmente realizado com apoio de provadores de teoremas como Isabelle/HOL, mas podendo, também, ser realizado manualmente ([MONIN; HINCHEY, 2003](#)).

- Verificação de modelos: a base de verificação de modelos é explorar todos os caminhos possíveis de serem alcançados em um grafo, derivado do modelo de máquina de estados, determinando se a dada máquina satisfaz as propriedades expressas como fórmulas em determinada lógica.
- Processos de abstração: o processo de abstração em MF evita definições de projeto e escolhas de implementação prematuramente, generaliza as características e propriedades importantes do sistema, ignorando e simplificando, deste modo, detalhes não relevantes para a análise das propriedades em questão.

2.1.2 Tipos de métodos formais

Existem diferentes tipos de classificações para métodos formais, variando de autor para autor. O exemplo considerado neste trabalho baseia-se em (ALMEIDA et al., 2011b) e é constituído pelas seguintes categorias: orientados a propriedades (axiomáticos), orientados a modelos (denotacionais), orientados a comportamentos e híbridos.

Orientado a Propriedade (Axiomático)

Definem o comportamento a partir de propriedades, as quais o sistema deve satisfazer, normalmente na forma de axiomas. Esses axiomas são definidos em lógica de Hoare (HOARE, 1969), por exemplo, para cada tipo de instrução na linguagem. Nesta lógica, podem ser de pré-condição ou pós-condição.

- Pré-Condição: precede imediatamente uma instrução e descreve as variáveis dela neste ponto.
- Pós-Condição: segue imediatamente uma instrução e descreve as novas restrições a essas variáveis depois da execução da instrução.

Orientados a Modelos (Denotacional)

São adequados para representar e manipular estruturas de dados complexas a partir de estruturas matemáticas como conjuntos, sequências, tuplas, funções e relações. Por exemplo, em um sistema de votação, a partir da representação dos eleitores que já votaram como um conjunto (*listaVotos*), a operação de adicionar um novo eleitor a esta lista é representada como uma união de conjuntos ($listaVotos \cup \{eleitor\}$). Este tipo de representação é utilizado em linguagens como B (SCHNEIDER, 2001) e Z (WOODCOCK et al., 2009).

Orientados a Comportamento

Presente nas especificações de sistemas paralelos, distribuídos e concorrentes, tem como foco a modelagem do comportamento do sistema como uma sequência possível de estados. Métodos como CSP (ROSCOE, 1997) e Redes de Petri (REISIG, 2013) são exemplos de métodos desta categoria.

Híbridos

Utilizam características presentes em mais de uma das categorias antes citadas, geralmente, com características de uma categoria prevalecendo em relação às demais.

2.2 Exemplo de alguns métodos formais

Com o intuito de contextualizar a área de métodos formais, esta seção apresenta dois métodos formais (Redes de Petri e Circus), mostrando um breve histórico e uso dos mesmos.

2.2.1 Redes de Petri

É uma técnica que possui abrangência e aplicabilidade em diversas áreas, tais como: na ciência da computação, engenharia eletrônica, engenharia química e administração de empresas. Utilizada na especificação de sistemas de hardware ou software, avaliação de desempenho, especificação de protocolos de comunicação, diagnóstico de falhas e no projeto de software/hardware. Isto é possível uma vez que os elementos que possui são capazes de descrever partes de sistemas com características de concorrência, controle, conflito, sincronização e compartilhamento (REISIG, 2013).

Foi criada por Carl Adam Petri quando jovem e posteriormente utilizada pelo mesmo em sua tese de doutorado, intitulada *Kommunikation mit Automaten* (Comunicação com Autômatos), apresentada, em 1962, na faculdade de Matemática e Física de Darmstadt, na então Alemanha Ocidental. Este método constitui uma técnica de modelagem que permite a representação de sistemas, utilizando como alicerce uma forte base matemática. Como uma linguagem de modelagem, ela define graficamente a estrutura de um sistema distribuído como um grafo direcionado com anotações.

Elementos Básicos

As redes de Petri são formadas pelos seguintes componentes básicos:

- Estados/Lugares: estados são usados para modelar os componentes passivos dos sistemas, isto é, correspondem às suas variáveis de estado, representados por P1, P2, P3 e P4 na Figura 1.
- Ações/Ativos: ações são usadas para modelar os componentes ativos dos sistemas, ou seja, os eventos que levam o sistema de um estado a outro, representados pelas setas na Figura 1.
- Relação de fluxo/Transições: é usada para especificar como se dá a transformação de um estado em outro pela ocorrência das ações no sistema, representados por T1 e T2 na Figura 1.
- Tokens: denotam recursos e são representados graficamente por pontos pretos. Uma distribuição de tokens na rede é chamada de marcação e corresponde ao estado da rede de Petri, representados pelas marcações nos estados P1, P3 e P4 na Figura 1.

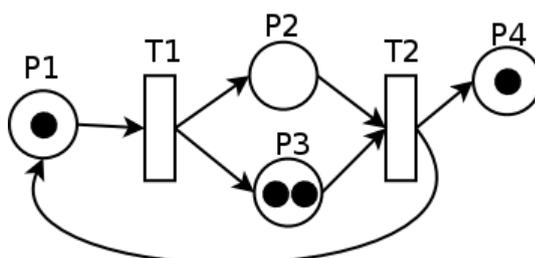


Figura 1 – Elementos da Rede de Petri

[Fonte: reproduzido de https://upload.wikimedia.org/wikipedia/commons/f/fe/Detailed_petri_net.png]

2.2.2 Circus

Foi criado e desenvolvido pelos professores Ana Cavalcanti e Jim Woodcock, do departamento de Ciência da Computação da Universidade de York (Reino Unido), e Marcel Oliveira, do departamento de Informática e Matemática aplicada da Universidade Federal do Rio Grande do Norte (Brasil).

Circus é uma linguagem de especificação formal que pode ser utilizada para representar o comportamento de sistemas concorrentes. Une os conceitos de Z e CSP, focando na parte estrutural e atuação das aplicações concorrentes. Uma especificação em Circus é formada por uma lista de parágrafos: parágrafos em Z, definição de canais ou definição de processos.

A partir da aplicação repetida e sistemática de leis de refinamento, uma especificação abstrata pode ser transformada em uma implementação concreta e correta por construção.

Portanto, Circus é usado para permitir a especificação de ambos os dados e os aspectos comportamentais de sistemas concorrentes e reativos, bem como para apoiar o desenvolvimento gradual através de refinamento. Na Figura 2, pode-se ver parte da especificação em Circus de um sistema de controle de uma bomba de combustível. O comportamento deste sistema é descrito pelo processo *Pump*, cujo estado é definido pela quantidade de combustível *fuelQ*; inicialmente, 5000. Ao retirar a mangueira (evento *liftNozzle*), a bomba se torna ativa. Se a mangueira for reposicionada (evento *putNozzle*), a bomba deixa de estar ativa. No entanto, se o sistema receber a informação do quanto será abastecido (*enterAmount?q*), após apertar o gatilho (*pressTrigger*), a variável *fuelQ* é diminuída de *q* (processo *Supply*). Em seguida, o gatilho é liberado e a bomba deixa de estar ativa. Outra ação possível é reabastecer a bomba (evento *reload?q*), que incrementa à variável *fuelQ* (processo *Reload*). Ainda é possível voltar a bomba para o estado inicial (evento *init* e processo *Init*).

```

channel init, liftNozzle, putNozzle, pressTrigger, releaseTrigger
channel enterAmount, reload :  $\mathbb{N}$ 

process Pump  $\hat{=}$ 
begin state PState  $\hat{=}$  [fuelQ :  $\mathbb{N}$ ]
  PInit  $\hat{=}$  [PState' | fuelQ' = 5000]
  Reload  $\hat{=}$  [ $\Delta$ PState; q? :  $\mathbb{N}$  | fuelQ' = fuelQ + q?]
  Supply  $\hat{=}$  [ $\Delta$ PState; q? :  $\mathbb{N}$  | fuelQ' = fuelQ  $\ominus$  q?]

  PumpIdle  $\hat{=}$  liftNozzle  $\rightarrow$  PumpActive
    □ reload?q  $\rightarrow$  Reload
    □ init  $\rightarrow$  PInit
  PumpActive  $\hat{=}$  putNozzle  $\rightarrow$  Skip
    □ enterAmount?q  $\rightarrow$  pressTrigger  $\rightarrow$  Supply; releaseTrigger  $\rightarrow$  Skip

  • init  $\rightarrow$  PInit;  $\mu$  X • PumpIdle; X
end

```

Figura 2 – Exemplo de implementação

[Fonte: Reproduzido parcialmente de (OLIVEIRA; CAVALCANTI; WOODCOCK, 2007)]

2.3 Adoção, crescimento e impacto na indústria

Cada vez mais missões importantes são delegadas a sistemas computacionais, os quais são, muitas vezes, constituídos pela integração de vários componentes e sistemas providos por diferentes empresas. Tendo isto em consideração, o trabalho de Foschi et al. (2003) relata a experiência de um projeto conjunto entre a Politecnico di Milano, a Ferroviária Estatal Italiana FS e o Departamento de Infra-estrutura Italiano onde foram definidas regras e procedimentos para a aquisição de softwares. Métodos formais foram adotados na fase de especificação suportados por ferramentas de verificação e validação adequadas.

No trabalho, destaca-se a importância de obedecer aos critérios de especificação, validação e verificação, os quais dão suporte a uma melhor interação entre o comprador e

fornecedor do componente, cuja finalidade é tornar o projeto e a implementação destes componentes uma atividade mais controlada e previsível. Como ponto de destaque, o artigo reporta diminuição de custos financeiros a partir da realização de simulações, antes mesmo dos testes em campo.

Dentre as conclusões obtidas com a execução do projeto, o artigo ressalta que métodos formais, principalmente nos quesitos especificação e verificação, vêm sendo mais valorizados e utilizados no ambiente industrial.

De uma forma mais geral, para uma melhor compreensão das tendências em métodos formais, o trabalho intitulado *Formal Methods: Practice and Experience* de [Woodcock et al. \(2009\)](#) realizou uma pesquisa para coletar informações de projetos industriais conhecidos por terem empregado técnicas formais. Foram avaliados os efeitos do uso de técnicas formais no tempo, custo e qualidade. Com relação ao tempo, foram relatados mais casos onde houve redução do que aumento do tempo levado para realização das atividades, especialmente na fase de especificação. Quanto ao custo, não houve diferença entre a quantidade de projetos que reportaram ter tido aumento de custo quando comparado aos que reportaram diminuição. Quanto ao critério, a percepção de quase todos os projetos é que a mesma foi melhorada impactando positivamente à detecção de erros, o projeto e o entendimento do sistema, entre outros aspectos. Tais resultados apontaram, portanto, para o benefício do uso de MF.

Por outro lado, a pesquisa pontuou que foram levantadas predominantemente aplicações de MF por especialistas usando ferramentas que, com poucas exceções, não são suficientemente robustas para aplicação em larga escala. Resultado este que indica a importância de produzir ferramentas que sejam suficientemente bem apoiadas para possibilitar uma maior aplicação comercial de MF. Foi identificado também que é preciso repensar com cuidado em como integrar práticas de MF com os processos de desenvolvimento de softwares e sistemas já em uso nas empresas.

De acordo com o artigo, a decisão de adotar uma tecnologia de desenvolvimento formal é baseada no risco e na evidência convincente de que as técnicas formais irão agregar valor na identificação de defeitos de forma a se sobrepôr ao custo de sua aplicação. Apesar dos casos de sucesso reportados, a tecnologia de verificação e os métodos formais não são amplamente adotados como parte da rotina prática de desenvolvimento de sistemas, exceto, no desenvolvimento de sistemas críticos em certos domínios.

As principais dificuldades encontradas para uma maior adoção de métodos formais na indústria foram: suporte a provas de propriedades de forma automática, formatos que sejam comuns para o intercâmbio de informações entre ferramentas, a falta de suporte para os usuários das ferramentas. O trabalho concluiu também que é possível observar um interesse da indústria pelo uso prático de MF, mas que uma maior aplicação depende da maturidade e avanços da teoria e de ferramentas de suporte.

2.4 O ensino de métodos formais

Trabalhos anteriores já investigaram aspectos relacionados ao ensino de métodos formais. A seguir, são apresentados e detalhados os estudos encontrados na literatura.

Ishikawa, Yoshioka e Tanabe (2015) destacam a existência de uma lacuna entre o que é ensinado na academia e o que é exigido na indústria em relação a métodos formais. No artigo, encontra-se o relato de experiência com um programa educacional chamado TopSE, desenvolvido para suprir essa lacuna, que por 10 anos têm proporcionado um lugar onde a academia e a indústria possam viabilizar o conhecimento técnico-prático de métodos formais.

O autor destaca a importância de considerar uma variedade de métodos e ferramentas, visando os engenheiros da indústria e os alunos de cursos de graduação, em uso tanto pela academia, como pela indústria. Na experiência reportada no trabalho, há destaque para o confronto com problemas práticos e uso de diferentes formas de aprendizado, por meio de palestras e exercícios em grupo, com a finalidade de se obter diferentes abordagens com relação ao uso de MF.

O artigo conclui que para um bom aproveitamento do ensino de MF é importante envolver no processo de ensino engenheiros que tenham a vivência prática do trabalho exercido na indústria. Além disso, o uso de MF oferece boas oportunidades para que os alunos discutam e definam abordagens de desenvolvimento rigorosas, reproduzíveis e objetivas, estabelecendo princípios claros, especialmente nas fases iniciais do desenvolvimento.

Já o trabalho de Spichkova e Zamansky (2016) analisa como unir o ensino de métodos formais ao de engenharia de software. Considerando este contexto, o trabalho discorre sobre os principais desafios no ensino de MF: elevada diversidade dos estudantes, desinteresse por conhecimentos de matemática e falta de motivação.

Quanto à alta diversidade dos estudantes, é citado que o número de alunos que saem do seu país de origem para estudar fora ou que se inscrevem em programas de ensino a distância no exterior tem crescido em todo o mundo. Esta diversidade tem que ser levada em consideração no ensino e avaliação dos alunos. Em um primeiro momento, princípios teóricos gerais e modelos matemáticos são introduzidos. Prossegue-se, então, com exemplos e aplicações desses princípios e modelos. Por fim, exercícios práticos são aplicados.

Já em relação à falta de interesse no aprendizado de matemática, a sugestão apresentada gira em torno de apresentar inicialmente especificações e expressões formais de menor complexidade e, gradativamente, apresentar exemplos de maior complexidade. Advoga-se, com o uso desta estratégia, tornar o aprendizado mais atrativo.

No que concerne à falta de motivação, a mesma é apontada como derivada da falta de conhecimento matemático. Os métodos formais exigem habilidades abstratas de

pensamento e conhecimento matemático prévio, sendo argumentado, portanto, que vários estudantes se intimidam por possuírem uma percepção negativa de notações matemáticas complexas. Uma solução apontada para este problema é a incorporação, no currículo de engenharia de software, de disciplinas como lógica e especificação formal.

Nesse contexto, algumas estratégias são apresentadas com o objetivo de integrar MF no currículo dos cursos de engenharia de software, como abranger conhecimentos básicos de lógica e não exigir um conhecimento profundo da matemática, onde apenas serão introduzidos os aspectos principais de MF. Por fim, mas não menos importante, considerar o uso de ferramentas visuais que possam deixar o material de estudo mais interessante e de mais fácil compreensão, tais como: MeLoISE platform (VOSINAKIS; KOUTSABASIS; ANASTASSAKIS, 2014), para ensino de lógica e programação, AutoFocus (SPICHKOVA; ZAMANSKY, 2016), para ensino de modelagem básica, KeY-Hoare (AHRENDT et al., 2016) e HAHA (SZNUK; SCHUBERT, 2014), para ensino de lógica. O uso dessas ferramentas é sugerido como forma de aumentar o interesse dos alunos pelo tema, diminuindo os desafios enfrentados pelos professores que se propõem a ensinar MF.

Já o trabalho de Liu et al. (2009) faz uma análise do ensino de métodos formais e lógica, demonstrando meios de ensinar, os principais problemas que podem aparecer e as possíveis soluções, através do levantamento realizado no arranjo da grade curricular do Departamento de Ciência da Computação na *Hosei University* no Japão

Para encorajar mais estudantes no estudo de métodos formais, os autores recomendam primeiro criar uma melhor motivação, demonstrando os claros benefícios dos métodos formais, melhorando assim a prática atual de engenharia de software.

O trabalho demonstra ainda que algumas técnicas específicas ajudam no ensino de métodos formais. Iniciar o ensino de MF com o uso de exemplos do dia-a-dia, que inspirem os alunos a solucionar um problema prático utilizando métodos formais. Introduzir gradualmente conceitos importantes. Utilizar de forma massiva exercícios, pois a escrita de especificações formais precisas exige que o desenvolvedor tenha uma boa compreensão dos recursos que podem ser aplicados nos diferentes tipos de dados e a forma mais eficiente de se aprender isso é através da prática de exercícios. Ensinar habilidades de abstração, pois o uso de métodos formais requer habilidade de abstração matemática. Adequar o conteúdo de MF ao tempo estipulado para o curso, que muitas vezes é curto. Por fim estimular a prática através de pequenos projetos onde o aluno possa ligar o uso de métodos formais à engenharia de software.

Desta forma, o artigo conclui que um importante fator para o sucesso do ensino de métodos formais é colocar o seu ensino no contexto da engenharia de software, sendo uma forma de transferir os uso de métodos formais de maneira efetiva para a indústria.

Mais alinhado aos objetivos desta monografia, tem-se o trabalho reportado em

(OLIVEIRA, 2004). Em 2001 o *Formal Methods Europe* (FME) criou o *FME Subgroup on Education* (FME-SoE) com o objetivo de desenvolver e recomendar um guia curricular para o ensino de métodos formais em cursos de graduação.

O artigo *A Survey of Formal Methods Courses in European Higher Education*, desenvolvido por Oliveira (2004), foi a base para este trabalho e apresenta um levantamento de cursos de métodos formais em ensino superior. A amostra de dados da pesquisa é composta por 117 cursos, distribuídos em 13 países europeus e envolveu 91 acadêmicos.

Foram analisadas as informações tanto de modo quantitativo, quanto qualitativo e como resultado propõe o seguinte corpo de conhecimento associado ao ensino de métodos formais:

- Fundamentos da disciplina
- Paradigmas de especificação formal
- Correção, verificação e cálculo formal
- Semântica formal
- Suporte para especificação executável
- Outros tópicos

Em um nível mais baixo de classificação, este corpo de conhecimento pode ser dividido em sub-áreas:

- Fundamentos topológicos de métodos formais
- Fundamentos lógicos de métodos formais
- Fundamentos teóricos de métodos formais
- Fundamentos algébricos de métodos formais
- Especificação orientada a propriedades
- Modelo orientado à especificação
- Especificação de múltiplos paradigmas
- Correção por construção
- Correção por verificação
- Correção por verificação de máquina de estados
- Técnicas de refinamento
- Semântica de linguagem de programação
- Formalização da distribuição, concorrência e mobilidade
- Programação declarativa

Na pesquisa realizada na web, através de 364 sites, pôde-se observar que muitos cursos não estão disponíveis on-line ou estão protegidos por senha. Muitas vezes também não se tem a informação se a disciplina é ofertada atualmente, ou se foi oferecida somente no passado. Eventualmente, a disciplina pode nem mais fazer parte da atual grade curricular do curso. Outro problema encontrado é a volatilidade de muitas URLs, o que torna difícil manter atualizada a lista dos cursos que ensinam métodos formais.

A amostra de cursos considerada pelo FME-SoE ainda não é totalmente representativa, pois não é geograficamente simétrica. Países como a Alemanha e a Espanha precisam ter seus cursos de métodos formais melhor analisados e documentados.

Segundo a publicação, o sucesso geral futuro do grupo FME-SoE dependerá de alcançar um consenso sobre a área de pesquisa e ensino de métodos formais, ajudando a responder perguntas chaves, tais como:

- O que caracteriza uma disciplina de métodos formais?
- Qual o limite entre as fundamentações matemáticas e os métodos formais?
- Qual o limite que separa os métodos formais e informais?

Por fim, o trabalho recomenda criar um repositório central provido pelo FME, que possa ser atualizado colaborativamente como uma Wiki. Desta maneira, ajudando os docentes a comparar seus métodos de ensino e adotar abordagens ou estilos particulares de ensino.

Apesar de ser uma publicação de 2004, nenhum artigo mais recente do grupo FME-SoE pode ser encontrado, o que sugere que atualmente este grupo não se encontra mais ativo.

3 Método de Pesquisa

A partir do que foi apresentado no Capítulo 2, pode-se observar que não foram encontrados estudos que analisam como se dá o ensino de métodos formais no Brasil. Com o intuito de levantar essa informação, esta pesquisa foi estruturada em uma sequência de etapas, descritas a seguir.

3.1 Revisão bibliográfica

Inicialmente, foi realizada uma revisão da literatura para verificar levantamentos similares sobre o ensino de métodos formais. Este levantamento foi feito através de consultas a materiais publicados nos portais ACM¹, SpringerLink² e IEEE Xplorer³, sendo estes trabalhos devidamente catalogados e resumidos. As strings de buscas utilizadas foram as seguintes:

- *Survey in Teaching Software Engineering*
- *Formal Methods Survey*
- *Teaching Formal Methods*
- *Formal Methods Curriculum*

Enquanto as 3 primeiras strings de busca estão diretamente relacionadas ao escopo deste trabalho, a última, apesar de tratar de ensino, não tem como foco métodos formais. Mesmo assim, esta *string* foi considerada para identificar *surveys* sobre o ensino de engenharia de software, que foram utilizados para estruturar o levantamento dos dados realizado neste trabalho.

Foram selecionados 22 artigos relacionados diretamente a pesquisas com métodos formais e no idioma Inglês, além de 3 livros referências no estudo de métodos formais (MONIN; HINCHEY, 2003; ALMEIDA et al., 2011a; BOULANGER, 2012).

Após a seleção dos artigos, foram lidos o resumo, a introdução e a conclusão de cada um, sendo criado um resumo das principais informações contidas nos trabalhos. Deste modo, dos 22 artigos inicialmente considerados, 14 foram selecionados por estarem mais alinhados ao escopo desta pesquisa.

¹ <http://dl.acm.org/>

² <http://link.springer.com/>

³ <http://ieeexplore.ieee.org/Xplore/home.jsp>

Após o resumo e escolha dos 14 artigos, foi realizada a leitura na íntegra dos mesmos. Neste momento, a partir da leitura completa dos trabalhos, 2 artigos foram desconsiderados, ficando 12 artigos como base para esta monografia.

3.2 Seleção do ranking

Considerando o tempo disponível, não seria possível levantar informações sobre todos os cursos de computação do Brasil; sendo, assim, necessário filtrar e considerar somente alguns cursos, idealmente, os mais bem conceituados. Desta forma, foram considerados inicialmente os seguintes rankings: o THE⁴ (*Times Higher Education World University Rankings*) e o RUF⁵ (*Ranking Universitário Folha*).

O THE é apoiado por diversas instituições ao redor do mundo e baseia seu levantamento em dados passados por instituições de ensino e pelo governo do país. Sua desvantagem em relação ao RUF é não possuir uma maneira fácil de agrupar as universidades por países, inviabilizando, deste modo, o levantamento de dados para o escopo brasileiro. Portanto, considerando o escopo deste trabalho, optou-se por não considerar o ranking THE.

O RUF, por sua vez, é restrito ao cenário brasileiro. É uma avaliação anual do ensino superior do Brasil feita pela Folha desde 2012. Possui ranking por universidade e por cursos (a qual foi utilizada). São utilizados 5 indicadores no ranking: pesquisa, internacionalização, inovação, ensino e mercado.

Os dados dos indicadores de avaliação do RUF são coletados por uma equipe da Folha em bases de patentes brasileiras, em bases de periódicos científicos, em bases do MEC e em pesquisas nacionais de opinião feitas pelo Datafolha.

Neste trabalho foi utilizado o *Ranking por Cursos de Computação de 2016*⁶ do RUF

3.3 Definição da metodologia de coleta de dados

Tomando como base os trabalhos de Woodcock et al. (2009), Pears et al. (2007) e principalmente Oliveira (2004), a metodologia de coleta de dados foi definida, considerando, em particular, o cenário de ensino de métodos formais no Brasil, bem como a necessidade de responder à pergunta de pesquisa deste trabalho: como é o ensino de métodos formais em cursos superiores de computação no Brasil?

⁴ <https://www.timeshighereducation.com/world-university-rankings>

⁵ <http://ruf.folha.uol.com.br/2016/>

⁶ <http://ruf.folha.uol.com.br/2016/ranking-de-cursos/computacao/>

Como principal instrumento de coleta de dados, definiu-se um questionário considerando estudos similares encontrados na revisão de literatura. O objetivo deste questionário é permitir coletar informações relevantes à construção de um panorama do ensino de métodos formais no Brasil. Um levantamento das informações disponíveis nos sites de cursos de Engenharia e Ciência da Computação das 5 primeiras instituições, de acordo com o Ranking Universitário Folha (RUF), foi realizada com o intuito de melhorar e consolidar o questionário de coleta de dados.

Como principal instrumento de coleta de dados, definiu-se um questionário considerando estudos similares encontrados na revisão de literatura. O objetivo deste questionário é permitir coletar informações relevantes à construção de um panorama do ensino de métodos formais no Brasil. Para validar o questionário desenvolvido inicialmente, realizou-se um piloto; ou seja, aplicou-se a versão inicial deste no levantamento de dados das 5 primeiras universidades de acordo com o ranking RUF. Após este piloto, foi possível identificar pontos de melhorias no questionário, que foram incorporadas em seguida.

Depois dos ajustes realizados no questionário a partir da realização do piloto as perguntas presentes no questionário de levantamento de dados ficaram apresentadas da seguinte forma:

1. Quais das 30 primeiras instituições de ensino superior, do Ranking Universitário Folha (RUF) e refinadas por "*Ranking de Cursos de Computação*", possuem cursos de Engenharia e/ou Ciência da Computação?
 - a) Alguma disciplina, presente no curso analisado, possui uma das seguintes palavras chaves no seu nome: formais, formal, especificação, verificação, análise e método(s)?
 - b) O resumo da ementa, apresentado no site, caracteriza uma disciplina de métodos formais com alguns dos seguintes detalhes:
 - i. Aborda lógica proposicional e de predicados?
 - ii. Aborda provadores de teorema?
 - iii. Na descrição da disciplina, apresenta métodos e linguagens de especificação formal, como: Redes de Petri, B-Method, Alloy, VDM, Z, CCS, CSP, OBJ, LOTOS, UNIT, entre outros?
 - c) Quais disciplinas, do curso analisado, foram consideradas como de MF, de acordo com os itens [1a](#) e [1b](#)?
 - i. Qual o nome da disciplina?
 - ii. O link do site com a descrição da cadeira.
 - iii. Nome do(s) docente(es) responsável(is).

- iv. Instrumento(s) de avaliação utilizado(s) na disciplina (provas, seminários, projetos, lista de exercícios).
- v. Qual a carga horária da(s) disciplina(s)?
- vi. A(s) disciplina(s) é(são) eletiva(s) ou obrigatória(s)?
- vii. Possui algum pré ou co-requisitos?
- viii. Pertence a qual período do referido curso?
- ix. Desde quando é ofertada?
- x. Quais são as linguagens adotadas na disciplina?
- xi. Quais ferramentas são utilizadas?
- xii. Quais são os livros-texto da disciplina?

3.4 Definição da metodologia de análise de dados

Para obter melhores conclusões deste trabalho, foram feitas análises qualitativas e quantitativas. A análise qualitativa permite um entendimento melhor e mais profundo, através da análise das perguntas apresentadas na seção anterior. A análise quantitativa, por sua vez, permite entender, de forma geral, indicadores de tendências e, com eles, criar um mapa analítico de como esses indicadores estão distribuídos em relação aos dados obtidos nas pesquisas.

Para isso, foram considerando estudos similares encontrados na revisão de literatura e elaborado um questionário de levantamento de dados para coletar informações relevantes à construção de um panorama do ensino de métodos formais no Brasil. Um levantamento das informações disponíveis nos sites de cursos de Engenharia e Ciência da Computação das 5 primeiras instituições, de acordo com o Ranking Universitário Folha (RUF), foi realizada com o intuito de melhorar e consolidar o questionário de coleta de dados. Desta forma, no próximo capítulo, são apresentadas análises quantitativas e qualitativas realizadas a partir dos dados levantados pelo questionário definido.

3.5 Levantamento dos dados

Após a realização do piloto, e considerando a versão final do questionário apresentada anteriormente, iniciou-se a coleta de dados propriamente dita: levantar dados sobre o ensino de métodos formais para os 30 melhores cursos de computação de acordo com o Ranking Universitário Folha (RUF). Em paralelo, tomando como base os dados coletados, foram realizadas análises quantitativas e qualitativas com o intuito de montar um panorama de como se dá o ensino de métodos formais no Brasil, atendendo ao objetivo desse trabalho.

3.6 Limitações da pesquisa

Assim como reportado no estudo ([OLIVEIRA, 2004](#)), esta pesquisa também teve dificuldade em levantar as informações necessárias exclusivamente a partir dos sites dos cursos. Antecipando esta dificuldade, inicialmente, foi realizado um levantamento das informações dos coordenadores de cada curso, com nome e endereço eletrônico. O intuito deste levantamento era entrar em contato com os coordenadores para sanar possíveis dúvidas ou adquirir informações extras. Porém, dentre dez coordenadores com os quais foi tentado contato, apenas quatro retornaram a mensagem eletrônica e em um intervalo grande de tempo (três semanas). Por este motivo, a abordagem via email foi descartada, ficando a pesquisa restrita às informações disponíveis online.

4 Resultados

Este capítulo descreve os achados do levantamento realizado em diversos sites de diferentes instituições de Ensino Superior do país. Na Seção 4.1 é apresentada uma visão geral consolidada do ensino de métodos formais no Brasil, de acordo com as 30 universidades melhor colocadas no ranking RUF. Em seguida, a Seção 4.2 traz mais informações sobre métodos formais no contexto das 5 primeiras universidades do mesmo ranking.

A principal fonte de informação para a pesquisa foram sites e portais de cada instituição de ensino superior.

Como dito anteriormente, um total de 30 universidades foram exploradas e registradas na pesquisa, de acordo com o Ranking Universitário Folha (RUF). Algumas dessas universidades possuem mais de um curso de computação em campi/unidades diferentes. Como exemplo a UNESP (Universidade Estadual Paulista), que possui três campi distintos com o curso de Engenharia de Computação. Desta maneira, considerando o levantamento de 30 universidades com cursos de Engenharia e Ciência da Computação, foram identificados 67 cursos distintos em campi/unidades diferentes.

4.1 Visão consolidada de todos os cursos pesquisados

Nesta seção serão apresentados de forma consolidada os dados obtidos a partir da análise dos 67 cursos em questão.

Disciplinas Encontradas

Dos 67 cursos analisados, somente 12 foram considerados com disciplinas de métodos formais. Dos 55 restantes, em 4 cursos não foi possível ter acesso à grade curricular e, assim, definir se métodos formais faz ou não parte do currículo (ver Figura 3).

No total, do levantamento, foram 40 cursos de Ciência da Computação (CC) e 27 de Engenharia de Computação (EC). Dos que apresentam disciplinas de Métodos Formais, 9 são cursos de CC e 3 são cursos de EC, como pode ser visto na Figura 4.

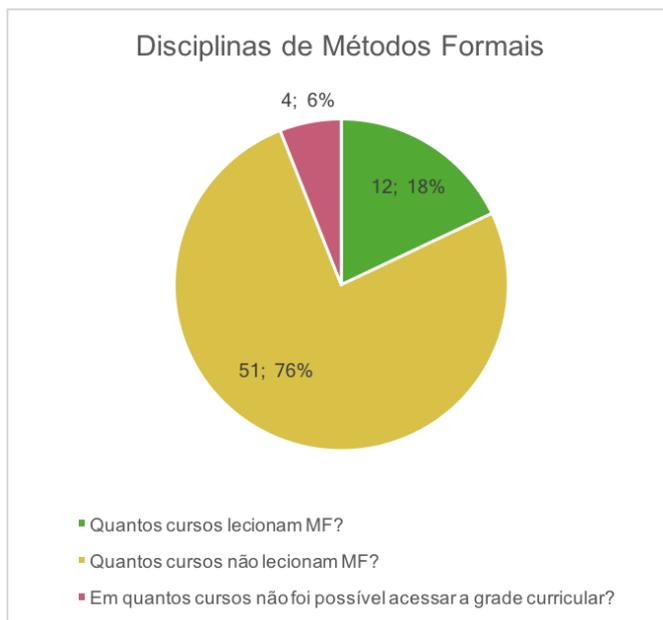


Figura 3 – Cursos que oferecem disciplinas de MF

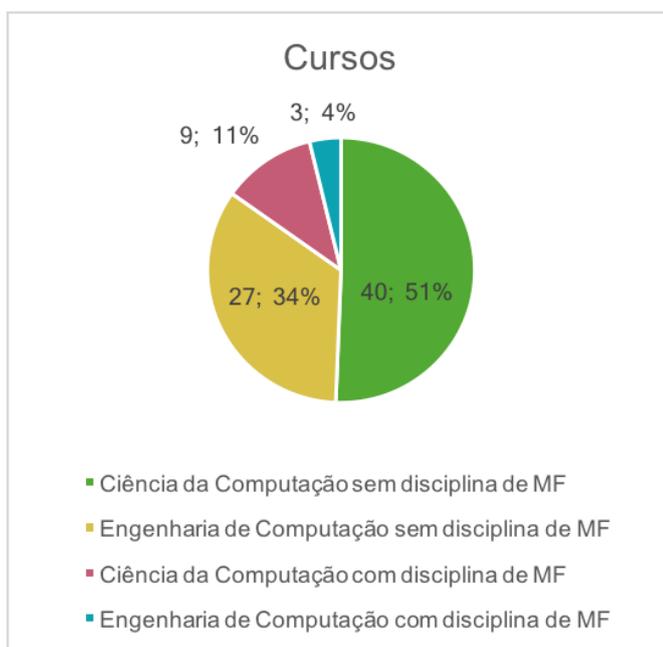


Figura 4 – Quantidade de cursos por área

Informações sobre docentes responsáveis e instrumentos de avaliação

As informações, sobre docentes responsáveis pelas disciplinas, foram uma das mais difíceis de serem encontradas no momento do levantamento. Dos 12 cursos com cadeiras de métodos formais, apenas 2 possuem esta informação. O IME (Instituto de Matemática e Estatística - São Paulo) possui 4 professores responsáveis pela matéria, sendo eles: Flavio

Soares C. da Silva, Leliane Nunes de Barros, Marcelo Finger e Renata Wassermann. A outra instituição que informa o professor é a Universidade Federal do Paraná (UFPR) com o docente Luis Allan Kunzle.

Na pesquisa realizada sobre os instrumentos de avaliação, apenas 3 cursos disponibilizam esta informação, sendo eles:

- Curso de Ciência da Computação do Instituto de Matemática e Estatística (IME): a descrição da disciplina de *Introdução à Lógica e Verificação de Programas* explica que a avaliação do discente é realizada através de provas e exercícios, porém não informa como é dado o cálculo para compor a nota final.
- Curso de Engenharia de Computação da Escola de Engenharia de São Carlos (EESC) e Curso de Engenharia de Computação do Instituto de Ciências Matemáticas e de Computação (ICMC): os cursos de ambas unidades compartilham a mesma grade curricular por serem interunidades. A descrição da disciplina de *Métodos e Técnicas para Análise e Projeto de Sistemas Reativos* descreve que os métodos de avaliação utilizados são provas, exercícios e trabalhos, sendo o último realizado tanto em classe como fora da mesma. A nota final é calculada pela média ponderada das notas obtidas pelo aluno nos trabalhos e provas.

Carga horária

Todos os 12 cursos com disciplinas de métodos formais possuem a carga horária descrita na ementa da disciplina. Os cursos com menor carga horária são os da Universidade Federal da Bahia (UFBA). Tanto o curso de Engenharia de Computação da Escola Politécnica, quanto o curso de Ciência da Computação do Departamento de Ciência da Computação possuem carga horária de 51 horas.

Os cursos com a maior carga horária são os da Universidade de São Paulo (USP), no curso de Ciência da Computação interunidades (EESC e ICMC) e o da Universidade Federal de Pernambuco (UFPE), no curso de Ciência da Computação do Centro de Informática (CIn). Todos com carga horária de 75 horas.

A média da carga horária de todas as disciplinas de MF encontradas é 62,25 horas.

Obrigatoriedade das disciplinas

Foram encontrados 10 cursos, do total de 12 com MF, que apresentam a descrição da obrigatoriedade ou não da disciplina (Ver Figura 5), sendo obrigatória em 3 (Ciência da Computação no Instituto de Matemática e Estatística - UPS; Ciência da Computação na PUCRio; Ciência da Computação na PUCRS) e eletiva em 7 (Ciência da Computação

na UFRGS; Ciência da Computação na UFPE; Ciência da Computação na UFPR; Ciência da Computação na UNISINOS; Ciência da Computação no Campus Campo Mourão - UTFPR; Ciência da Computação na UFBA; Engenharia de Computação na UFBA). Dois cursos não possuem a informação (Engenharia de Computação interunidades da USP).

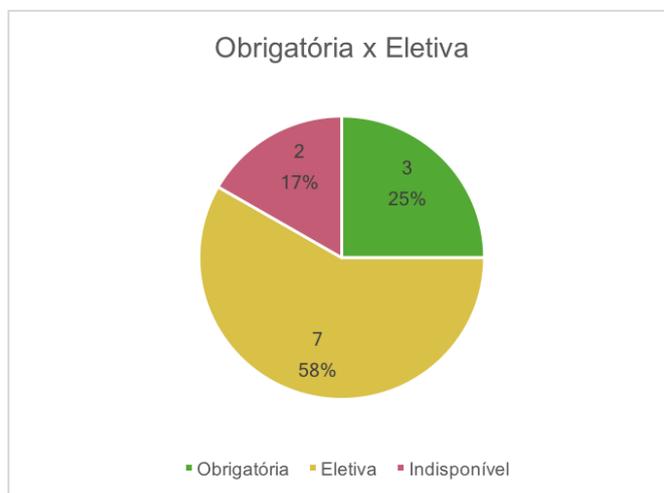


Figura 5 – Obrigatoriedade das disciplinas

Pré/Co-Requisito

Nove cursos informam sobre pré-requisitos e 3 não possuem esta informação (Ciência da Computação na UFPR; Ciência da Computação na PUCRS; Ciência da Computação no Campus Campo Mourão - UTFPR).

Período pertencente

Apenas 5 cursos informam a qual período pertence a disciplina de MF ofertada (Ver Figura 6). Desses, o curso de Ciência da Computação do Instituto de Matemática e Estatística (USP) é o que disponibiliza a cadeira mais cedo, no 4º período. O curso de Engenharia da Computação interunidades (EESC e ICMC), é o mais tardio dos cursos, sendo possível cursar apenas no 9º período. O período médio, de ofertas pelos cursos, é o 6º.

Linguagens utilizadas

Seis cursos informam qual método formal é ensinado durante a disciplina.

Três cursos, ICMC e EESC interunidades de Engenharia de Computação da USP e o curso de Ciência da Computação na UFPR descrevem o uso de Redes de Petri como base para o ensino de métodos formais.

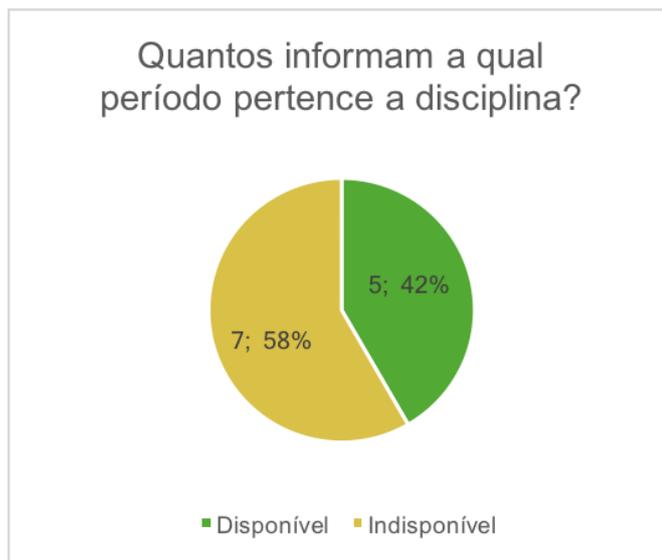


Figura 6 – Quantidade de cursos que informam o período da disciplina

O curso de Ciência da Computação da UFRGS descreve uma lista de linguagens que podem ser abordadas na disciplina, dependendo do docente. VDM, Z, CCS, CSP, OBJ, LOTOS, UNIT e outras.

No curso de Ciência da Computação da PUC-Rio, a disciplina tem o foco em CCS *Calculus of Communicating Systems*. CCS é um cálculo de processos desenvolvido por Robin Milner e é útil para avaliar a correção qualitativa das propriedades de um sistema concorrente como deadlock.

A descrição da disciplina no curso de Ciência da Computação da UFBA apresenta a Lógica de Hoare como base. Lógica de Hoare é um conjunto de regras lógicas para o raciocínio rigoroso sobre a correção de sistemas computacionais, proposta por Tony Hoare.

Data da primeira oferta e ferramentas adotadas

Somente os cursos de Ciência da Computação do Instituto de Matemática e Estatística (IME) e Ciência da Computação da UFPR apresentam o ano de início do ensino da disciplina de MF: sendo 2014 e 2011, respectivamente. Os cursos de Engenharia de Computação e Ciência da Computação da UFRN informam que 2011 foi o último ano que uma disciplina de MF foi ofertada.

Quanto a ferramentas adotadas no ensino de métodos formais, esta informação não está presente em nenhum dos sites analisados. Portanto, nada se pode afirmar sobre quais ferramentas são mais utilizadas com mais frequência.

Bibliografias

Abaixo, segue uma transcrição direta da bibliografia adotada nas disciplinas de métodos formais. Das 12 disciplinas, somente 5 disponibilizam esta informação no site.

- Ciência da Computação no Instituto de Matemática e Estatística (IME) da USP
 - F.S.C. da Silva, M. Finger, A.C.V. de Melo, *Lógica para Computação*, Cengage Learning, 2006.
 - D. Gries, *The Science of Programming*, Springer-Verlag, 1981.
 - M. Huth, M. Ryan, *Logic in Computer Science: Modelling and Reasoning about Systems*, 2nd ed., Cambridge University Press, 2004.
- Ciência da Computação no Instituto de Ciências Matemáticas e de Computação (ICMC) da USP
 - HAREL, D. *Algorithmics - The Spirit of Computing*. Addison-Wesley, 2. ed., 1992.
 - SIPSER, M. *Introduction to the Theory of Computation*. PWS, 2a ed, 1997.
 - HOPCROFT, M. & ULLMAN *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 2001.
- Engenharia de Computação no Instituto de Ciências Matemáticas e de Computação (ICMC) e Escola de Engenharia de São Carlos (EESC) da USP
 - PRESSMAN, R.S. *Software Engineering: - A Practitioner's Approach*. 4th Edition, McGraw-Hill, USA, 1997.
 - GOMAA, H. *A Software Design Method for Real-Time Systems*, CACM, vol. 27, no. 9, pp 938-949, 1984.
 - HAREL, D. et alli. *STATEMATE: A Working Environment for the Development of Complex Reactive Systems*, IEEE Transactions on Software Engineering, Vol. 16, No. 3, pp. 403-414, 1990.
 - HATLEY, D. and PIRBHAI, M. *Strategies for Real-Time Systems Specification*, Dorset-House, 1987.
 - Revistas e Periódicos da área.
- Ciência da Computação PUC-Rio
 - E. M. Clarke; O. Grumberg; D. Peled. *Model Checking*; Cambridge: MIT Press, 1999.

4.2 Detalhamento das 5 primeiras universidades do RUF

Nesta seção, apresenta-se uma visão mais detalhada sobre o ensino de métodos formais no contexto das 5 primeiras universidades do ranking RUF.

4.2.1 Universidade Estadual de Campinas (UNICAMP)

A primeira universidade de acordo com o RUF é a UNICAMP (Universidade Estadual de Campinas). A UNICAMP possui cursos de Engenharia e Ciência da Computação. O portal do curso de Engenharia de Computação¹ possui interface confusa, com informações pouco organizadas. É necessário passar por vários links e redirecionamentos até alcançar a área correspondente às informações sobre a grade curricular do curso. Em tal área depara-se com descrições das disciplinas de forma resumida conforme o apresentado na Figura 7. Não foi encontrada disciplina de MF em acordo com o questionário apresentado na seção 3.3.

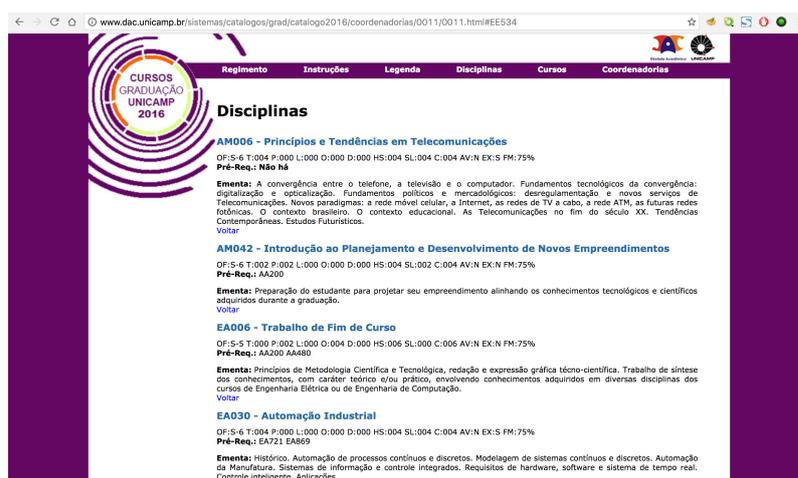


Figura 7 – Disciplinas do curso de EC da UNICAMP

Fonte: Reproduzido de

<http://www.dac.unicamp.br/sistemas/catalogos/grad/catalogo2016/coordenadorias/0011/0011.html#EE534>

O curso de Ciência da Computação² possui um portal (mostrado na Figura 8) intuitivo e fácil de utilizar. Nele é possível encontrar todas as informações de forma rápida e direta. No entanto, também foi encontrada disciplina de MF.

4.2.2 Universidade Federal do Rio de Janeiro (UFRJ)

A segunda posição no ranking RUF é ocupada pela UFRJ. Nessa universidade existem cursos de Ciência da Computação e Engenharia de Computação. O portal do

¹ <http://www.fee.unicamp.br/node/103>

² <http://www.ic.unicamp.br>



Figura 8 – Disciplinas do curso de CC da UNICAMP

Fonte: Reproduzido de <http://www.ic.unicamp.br/node/363>

Departamento de Ciência da Computação³ não estava acessível (Figura 9) nas seis tentativas realizadas, em 2 diferentes dias (10/10/2016 e 31/10/2016), bem como o perfil do curso de CC no Siga⁴ (Figura 10)

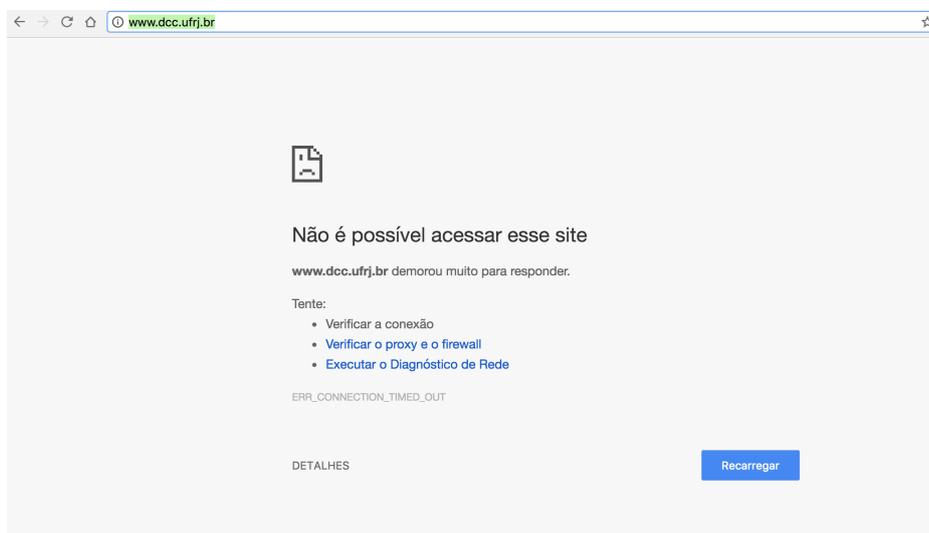


Figura 9 – Erro ao acessar o portal do departamento de CC

[Fonte: reproduzido de <http://www.dcc.ufrj.br/>]

O portal do curso de EC⁵ é intuitivo e fácil de encontrar as informações necessárias. Sobre a grade curricular, apresenta apenas um fluxograma das disciplinas e a qual período

³ <http://www.dcc.ufrj.br/>

⁴ <https://www.siga.ufrj.br/sira/temas/zire/frameConsultas.jsp?mainPage=/repositorio-curriculo/FA9F18A7-92A4-F79B-1A98-293E97D8939B.html>

⁵ <http://www.poli.ufrj.br/>

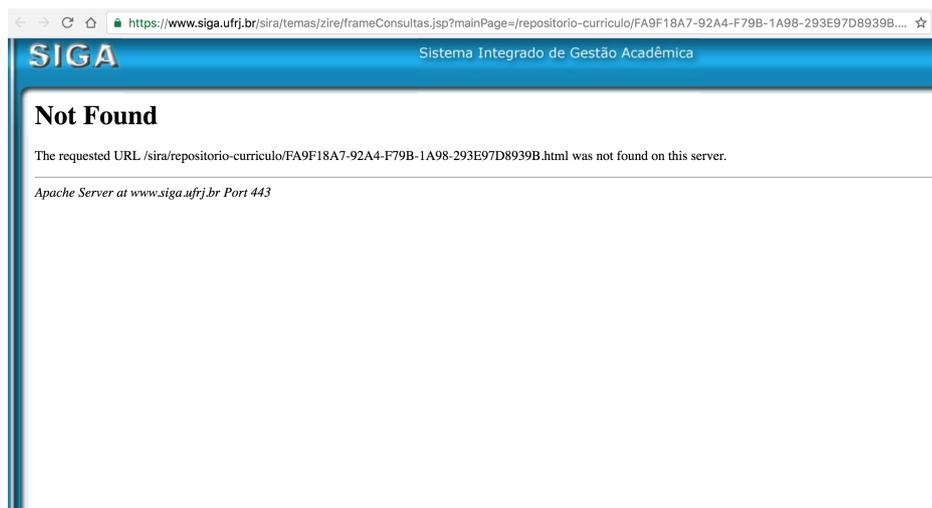


Figura 10 – Erro no Siga de CC da UFRJ

[Fonte: reproduzido de
<https://www.siga.ufrj.br/sira/temas/zire/frameConsultas.jsp?mainPage=/repositorio-curriculo/FA9F18A7-92A4-F79B-1A98-293E97D8939B.html>]

pertence, sem nenhuma informação extra. Na tentativa de obter mais informações, foi acessado o perfil do curso de EC no Siga⁶, porém, sem sucesso (Figura 11). Não foi encontrada disciplina de MF em acordo com o questionário definido anteriormente.

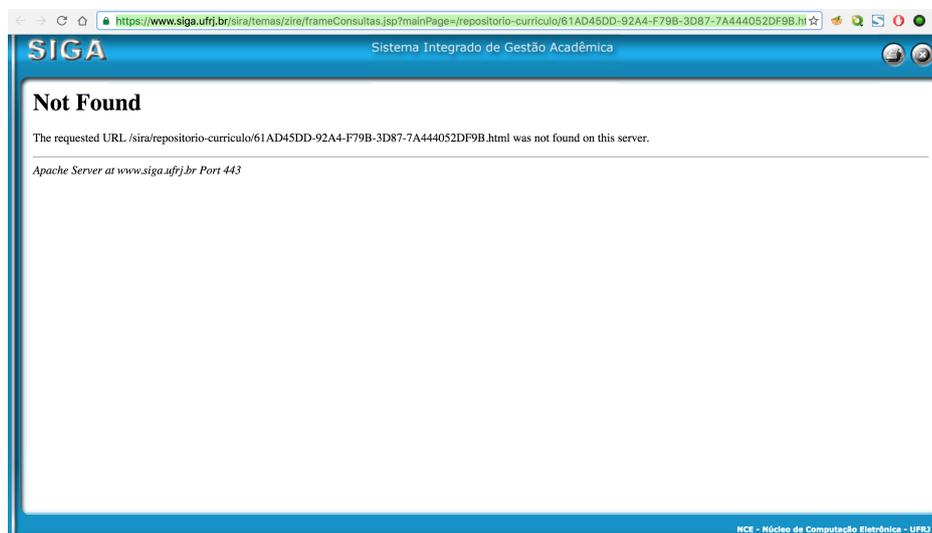


Figura 11 – Erro no Siga de EC da UFRJ

[Fonte: reproduzido de
<https://www.siga.ufrj.br/sira/temas/zire/frameConsultas.jsp?mainPage=/repositorio-curriculo/61AD45DD-92A4-F79B-3D87-7A444052DF9B.html>]

⁶ <https://www.siga.ufrj.br/sira/temas/zire/frameConsultas.jsp?mainPage=/repositorio-curriculo/61AD45DD-92A4-F79B-3D87-7A444052DF9B.html>

4.2.3 Universidade de São Paulo (USP)

A Universidade de São Paulo foi um das instituições com a maior quantidade de campi distintos e com cursos de Ciência e Engenharia da Computação. No total, foram cinco campi, dois com Ciência da Computação e três com Engenharia da Computação.

Instituto de Matemática e Estatística - São Paulo (IME)

No IME foi encontrado o curso de Ciência da Computação. O portal⁷ do instituto é bem elaborado, fácil de utilizar e com informações detalhadas sobre a unidade e os cursos.

Na página⁸ com a descrição da grade curricular de CC, foi encontrada uma disciplina de métodos formais de acordo com o questionário apresentado no capítulo anterior. A disciplina está registrada como MAC0239⁹ e começou a ser ofertada em 2002 com o nome de Métodos Formais em Programação. Seu nome foi alterado em 2014 para Introdução à Lógica e Verificação de Programas.

O programa da disciplina aborda: equivalência lógica e normas formais, métodos de prova e inferência lógica, noções sobre correção e completude, implicação lógica, equivalência lógica, extensão dos métodos de prova e inferência lógica para LPOO (linguagens de programação orientadas a objetos), dedução natural, verificação de programas utilizando lógica de Hoare, pré- e pós-condições, exemplos clássicos de provas de correção parcial e correção total de programas. Aplicando o questionário definido na Seção 3.3, temos as seguintes respostas:

1. Docentes responsáveis: Flavio Soares C. da Silva, Leliane Nunes de Barros, Marcelo Finger e Renata Wassermann;
2. As avaliações dos discentes são realizadas através de provas e exercícios em classe e extraclasse;
3. A carga horária é de 60h;
4. É uma disciplina obrigatória;
5. Tem como pré-requisito a disciplina de Introdução à Computação;
6. Pertence ao 4º Período do curso de CC;
7. É ofertada desde 2014;
8. Não informa sobre linguagens específicas adotadas na disciplina (fala de lógica de Hoare, mas não comenta se usa alguma linguagem que se baseia nesta lógica);

⁷ <https://www.ime.usp.br/>

⁸ <https://www.ime.usp.br/dcc/grad/grade>

⁹ <http://bcc.ime.usp.br/catalogo2015/disciplinas/MAC0239.html>

9. Não informa sobre ferramentas adotadas na disciplina;
10. Bibliografia básica:
 - a) D. Gries, *The Science of Programming*, Springer-Verlag, 1981.
 - b) F.S.C. da Silva, M. Finger, A.C.V. de Melo, *Lógica para Computação*, Cengage Learning, 2006.
 - c) M. Huth, M. Ryan, *Logic in Computer Science: Modelling and Reasoning about Systems*, 2nd ed., Cambridge University Press, 2004.

Instituto de Ciências Matemáticas e de Computação - São Carlos (ICMC)

No ICMC¹⁰ são ofertados os cursos de Ciência e Engenharia da Computação.

Considerando o curso de Ciência da Computação, não foi encontrada nenhuma disciplina de métodos formais. Já o curso de EC é interunidades e é ofertado pelo Instituto de Ciências Matemáticas e de Computação em São Carlos e pela Escola de Engenharia de São Carlos.

Todas as informações sobre as disciplinas estão disponíveis no JupiterWeb¹¹, sistema similar ao Siga, onde alunos e docentes podem acessar informações pertinentes aos seus respectivos perfis.

Foi encontrada uma disciplina de métodos formais: SSC0722 - Métodos e Técnicas para Análise e Projeto de Sistemas Reativos¹². O programa da disciplina apresenta características dos softwares reativos com classificações e aplicações, utilizando técnicas para especificação do comportamento de sistemas reativos, como máquinas de estado finito e redes de Petri, além de validar e simular modelos, abordando, assim, os principais métodos para análise e projeto de sistemas reativos.

Seguindo o questionário utilizado nesta pesquisa, têm-se as seguintes respostas:

1. Não tem informação sobre professor responsável pela disciplina;
2. As avaliações dos discentes são realizadas através de provas e exercícios em classe e extraclasse, assim como trabalhos;
3. Carga horária de 75h;
4. Não informa sobre a obrigatoriedade da disciplina;
5. Tem como pré-requisito as disciplinas de *SSC0620 - Engenharia de Software* e *SSC0640 - Sistemas Operacionais I*;

¹⁰ <http://www.icmc.usp.br/graduacao/cursos>

¹¹ <https://uspdigital.usp.br/jupiterweb/listarGradeCurricular?codcg=97&codcur=97001&codhab=0&tipo=N>

¹² <https://uspdigital.usp.br/jupiterweb/obterDisciplina?sgldis=SSC0722&codcur=97001&codhab=0>

6. Pertence ao 9º Período do curso de EC;
7. Não informa sobre a primeira vez que foi ofertada;
8. Adota redes de Petri como linguagem para o estudo de métodos formais;
9. Não informa sobre ferramentas adotadas na disciplina;
10. Bibliografia básica:
 - a) PRESSMAN, R.S. Software Engineering: - A Practitioner's Approach. 4th Edition, McGraw-Hill, USA, 1997.
 - b) GOMAA, H. A Software Design Method for Real-Time Systems, CACM, vol. 27, no. 9, pp 938-949, 1984.
 - c) HAREL, D. et ali. STATEMATE: A Working Environment for the Development of Complex Reactive Systems, IEEE Transactions on Software Engineering, Vol. 16, No. 3, pp. 403-414, 1990.
 - d) HATLEY, D. and PIRBHAI, M. Strategies for Real-Time Systems Specification, Dorset-House, 1987.

Escola de Engenharia de São Carlos (EESC)

A Escola de Engenharia de São Carlos possui o curso de Engenharia da Computação; curso esse, ofertado interunidades. Portanto, possui as mesmas características do curso de EC do ICMC, apresentado anteriormente.

Departamento de Engenharia de Computação e Sistemas Digitais (PCS)

No Departamento de Engenharia de Computação e Sistemas Digitais¹³ foi encontrado o curso de Engenharia da Computação.

Ao acessar a grade curricular, pelo portal, o mesmo redireciona para o JupiterWeb. Porém existe um erro no certificado, onde o nome do servidor é incompatível com o certificado instalado (Ver Figura 12). É possível ter acesso ao portal ignorando o aviso. Não foi encontrada disciplina de MF em acordo com o questionário deste estudo.

4.2.4 Universidade Federal do Rio Grande do Sul (UFRGS)

A Universidade Federal do Rio Grande do Sul possui apenas o curso de Ciência da Computação, curso esse, presente no Instituto de Informática¹⁴ da UFRGS. O site do departamento apresenta uma boa estrutura, com muitas informações fáceis de serem

¹³ <https://pcs.usp.br/>

¹⁴ <http://www.inf.ufrgs.br/site/ciencia-da-computacao/grade-curricular/>

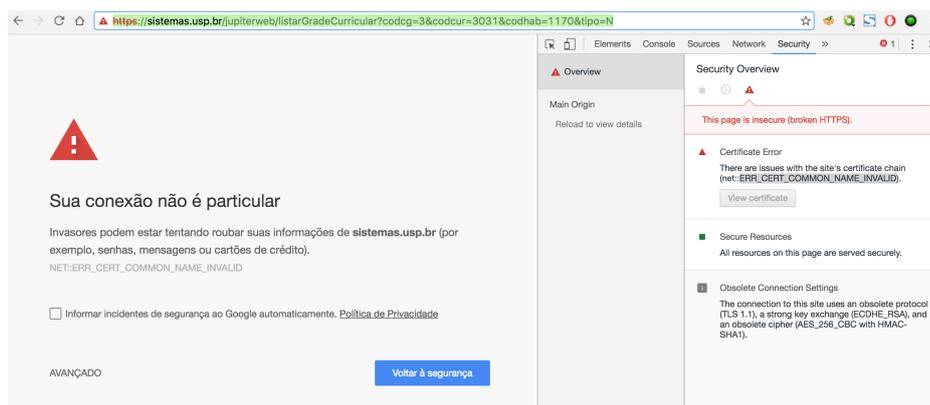


Figura 12 – Erro no certificado da página

encontradas. A grade curricular presente no portal detalha bem as disciplinas. Desse modo, foi encontrada uma matéria no contexto de ensino de métodos formais: INF01001 - Especificação Formal¹⁵.

Na ementa presente no portal, são relatados os seguintes tópicos da disciplina: revisão matemática, métodos e linguagens, classificação dos métodos formais, especificações executáveis (prototipação), além de apresentação e aplicação de métodos e linguagens de especificação formal como: VDM, Z, CCS, CSP, OBJ, LOTOS e UNIT. Outras informações relevantes obtidas são as seguintes:

1. Não tem informação sobre professor responsável pela disciplina;
2. Não tem informação sobre o método de avaliação do discente;
3. Carga horária de 60h;
4. É uma disciplina Eletiva;
5. Tem como pré-requisito a disciplina INF05501 - Teoria da Computação N;
6. Está disponível após ter cursado Teoria da Computação N (pode ser cursada a partir do 4º período);
7. Não informa desde quando é ofertada no curso;
8. As linguagens descritas pelo resumo da ementa são: VDM, Z, CCS, CSP, OBJ, LOTOS e UNIT;
9. Não informa sobre ferramentas utilizadas;
10. A bibliografia básica não é descrita nas informações da disciplina.

¹⁵ http://www.ufrgs.br/ufrgs/ensino/graduacao/cursos/exibeCurso?cod_curso=305

4.2.5 Universidade Federal de Minas Gerais (UFMG)

Na Universidade Federal de Minas Gerais foi encontrado o curso de Ciência da Computação, o qual é lecionado no Departamento de Ciência da Computação (DCC)¹⁶.

O site do departamento é bem estruturado e fácil de utilizar. Porém, a descrição das disciplinas presentes na grade curricular da unidade é escassa de informações. Não foi encontrado matérias de MF.

Portanto, como pode-se perceber, a grande dificuldade para a realização deste estudo é obter informações sobre o ensino de métodos formais a partir do site das instituições. No entanto, apesar desta dificuldade, a pesquisa atinge o seu objetivo de traçar um panorama inicial sobre o ensino de métodos formais em cursos de graduação em ciência e engenharia de computação no Brasil.

¹⁶ <http://www.dcc.ufmg.br/dcc/?q=pt-br>

5 Conclusão

Métodos formais (MF) constituem uma abordagem para garantir a confiabilidade, sendo procedimentos desenvolvidos a partir de princípios matemáticos com o objetivo de auxiliar no projeto, implementação, teste e concepção de software e hardware, aplicando técnicas de especificação formal, refinamento e verificação, que a indústria, por exemplo, vem progressivamente dando mais atenção.

Considerando esta motivação, percebe-se a importância de ter este conteúdo presente na formação de profissionais de computação. Para possibilitar que os estudantes se interessem e aprendam MF, é necessário demonstrar os claros benefícios destes conceitos na prática atual de desenvolvimento de sistemas computacionais. Sendo, assim, importante o uso de estilos e técnicas de ensino adequadas, além de um sensato arranjo curricular. Entretanto, nem sempre o ensino destes conceitos está presente nos cursos de graduação de Engenharia e Ciência de Computação, especialmente no cenário nacional. Portanto, este trabalho teve como objetivo traçar um panorama inicial de como se dá o ensino de MF no país.

O trabalho apresentou informações obtidas sobre MF nos cursos de Ciência e Engenharia da Computação das universidades melhor avaliadas de acordo com o ranking RUF. No levantamento e na análise realizadas no trabalho, verificou-se certa falta de importância dada ao ensino desse tema, uma vez que este conteúdo não faz parte da grade curricular de vários cursos. No que concerne à pesquisa na web, algumas dificuldades e limitações foram encontradas. Primeiramente, as informações sobre muitos cursos não estavam disponíveis online, como estrutura da disciplina, conteúdo abordado, métodos e materiais didáticos empregados, dificultando a caracterização dos conteúdos abordados na graduação. Em segundo lugar, alguns sites não são bem estruturados, contendo erros de certificado e links sem funcionar. Devido à dificuldade em se obter informações não é possível garantir que o cenário apresentado neste trabalho retrata fielmente como se dá o ensino de métodos formais no Brasil.

Em conformidade com pesquisas realizadas em outros países, o ensino de MF pode ser melhor estruturado, adotando boas práticas de didática, além de se fazer presente em uma quantidade maior de cursos de computação.

Apesar das limitações e dificuldades, o presente trabalho serve como um estímulo inicial para o compartilhamento das experiências de ensino e resultados obtidos a partir da inclusão de MF na grade curricular. As informações disponíveis podem ser utilizadas como modelo para a elaboração da ementa desta disciplina em cursos de CC e EC que não a possui. Desta forma, espera-se que, cada vez mais, surjam profissionais capacitados em

métodos formais, tanto no mercado de trabalho como no campo de pesquisa, para, assim, suprir as demandas existentes.

5.1 Trabalhos Futuros

Como trabalhos futuros, sugere-se, inicialmente, a ampliação do número de cursos de ensino superior analisados e maior detalhamento de informações no levantamento de seus perfis. Substituição da abordagem de linguagens utilizadas nas disciplinas por tipos de métodos formais ensinados nas disciplinas de MF. Com base em dados mais amplos, pode-se, então, utilizar cálculos estatísticos para realizar estudos comparativos, objetivando responder perguntas como: “Os cursos são mais focados em Hardware ou Software?” “Os cursos têm maior enfoque teórico ou prático?”, dentre outros questionamentos.

Além disso, deve-se buscar maneiras alternativas para o levantamento de dados; ou seja, com o intuito de superar a falta de informação detalhada online. Uma possibilidade é complementar o levantamento dos dados a partir do contato direto (email ou telefone) com os coordenadores dos cursos considerados.

Por fim, espera-se que um fiel panorama de como se dá o ensino de métodos formais no Brasil auxilie trabalhos futuros com o intuito de melhorar o ensino deste conteúdo no país, traçando estratégias que priorizem esta disciplina, assim como um melhor aproveitamento desta pelos discentes.

Referências

- AHRENDT, W. et al. (Ed.). *Deductive Software Verification – The KeY Book*. Springer International Publishing, 2016. Disponível em: <<http://dx.doi.org/10.1007/978-3-319-49812-6>>.
- ALMEIDA, J. B. et al. An overview of formal methods tools and techniques. In: *Rigorous Software Development*. Springer Nature, 2011. p. 15–44. Disponível em: <http://dx.doi.org/10.1007/978-0-85729-018-2_2>.
- ALMEIDA, J. B. et al. *Rigorous Software Development*. Springer London, 2011. Disponível em: <<http://dx.doi.org/10.1007/978-0-85729-018-2>>.
- BABIN, G.; AIT-AMEUR, Y.; PANTEL, M. Web services compensation at runtime: Formal modeling and verification using the event-b refinement and proof based formal method. *IEEE Transactions on Services Computing*, Institute of Electrical and Electronics Engineers (IEEE), p. 1–1, 2016. Disponível em: <<http://dx.doi.org/10.1109/TSC.2016.2594782>>.
- BEEK, M. H. ter; GNESI, S.; KNAPP, A. (Ed.). *Critical Systems: Formal Methods and Automated Verification*. Springer International Publishing, 2016. Disponível em: <<http://dx.doi.org/10.1007/978-3-319-45943-1>>.
- BORGES, R. M.; MOTA, A. C. Integrating UML and formal methods. *Electronic Notes in Theoretical Computer Science*, Elsevier BV, v. 184, p. 97–112, jul 2007. Disponível em: <<http://dx.doi.org/10.1016/j.entcs.2007.03.017>>.
- BOULANGER, J.-L. *Formal Methods: Industrial Use from Model to the Code*. Wiley, 2012. Disponível em: <<http://www.wiley.com/WileyCDA/WileyTitle/productCd-184821362X.html>>.
- CRISTIÁ, M. Teaching formal methods in a third world country: What, why and how. In: *Proceedings of the 2006 Conference on Teaching Formal Methods: Practice and Experience*. Swinton, UK, UK: British Computer Society, 2006. (TFM'06), p. 10–10. Disponível em: <<http://dl.acm.org/citation.cfm?id=2228206.2228216>>.
- DENNEY, E. *Formal Methods for the Certification of Auto-generated Flight Code*. 2005. Online. Disponível em: <[https://ti.arc.nasa.gov/m/pub-archive/1400h/1400%20\(Denney\).pdf](https://ti.arc.nasa.gov/m/pub-archive/1400h/1400%20(Denney).pdf)>. Acesso em: 14.11.2016.
- FOSCHI, U. et al. The role of formal methods in software procurement for the railway transportation industry. In: *Symposium on Formal Methods for Railway Operation and Control Systems (FORMS 2003), Budapest, Hungary*. [S.l.: s.n.], 2003. p. 15–16.
- GLINZ, M. Problems and deficiencies of uml as a requirements specification language. In: *Proceedings of the 10th International Workshop on Software Specification and Design*. Washington, DC, USA: IEEE Computer Society, 2000. (IWSSD '00), p. 11–. ISBN 0-7695-0884-7. Disponível em: <<http://dl.acm.org/citation.cfm?id=857171.857222>>.

3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0136744095>.

SCHNEIDER, S. *The B-method (Cornerstones of Computing)*. Palgrave Macmillan, 2001. ISBN 033379284X. Disponível em: <<https://www.amazon.com/B-method-Cornerstones-Computing-Steve-Schneider/dp/033379284X%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D033379284X>>.

SKEVOULIS, S.; FALIDAS, M. Integrating formal methods tools into undergraduate computer science curriculum. In: *Proceedings of the 7th Annual Conference on Innovation and Technology in Computer Science Education*. New York, NY, USA: ACM, 2002. (ITiCSE '02), p. 232–232. ISBN 1-58113-499-1. Disponível em: <<http://doi.acm.org/10.1145/544414.544500>>.

SPICHKOVA, M.; ZAMANSKY, A. Teaching of formal methods for software engineering. In: *Proceedings of the 11th International Conference on Evaluation of Novel Software Approaches to Software Engineering*. Scitepress, 2016. Disponível em: <<http://dx.doi.org/10.5220/0005928503700376>>.

SZNUK, T.; SCHUBERT, A. Tool support for teaching hoare logic. In: *Software Engineering and Formal Methods*. Springer Nature, 2014. p. 332–346. Disponível em: <http://dx.doi.org/10.1007/978-3-319-10431-7_27>.

VALLES-BARAJAS, F. Using lightweight formal methods to model class and object diagrams. *Computer Science and Information Systems*, National Library of Serbia, v. 9, n. 1, p. 411–429, 2012. Disponível em: <<http://dx.doi.org/10.2298/CSIS110210045V>>.

VOSINAKIS, S.; KOUTSABASIS, P.; ANASTASSAKIS, G. A platform for teaching logic programming using virtual worlds. In: *2014 IEEE 14th International Conference on Advanced Learning Technologies*. Institute of Electrical and Electronics Engineers (IEEE), 2014. Disponível em: <<http://dx.doi.org/10.1109/ICALT.2014.193>>.

WOODCOCK, J. et al. Formal methods: Practice and experience. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 41, n. 4, p. 19:1–19:36, out. 2009. ISSN 0360-0300. Disponível em: <<http://doi.acm.org/10.1145/1592434.1592436>>.