

## Um Panorama do Ensino de Programação Paralela e Distribuída em Universidades Brasileiras

Naylor Garcia Bachiega<sup>1</sup>, Paulo Sergio Lopes de Souza<sup>1</sup>,  
Sarita Mazzini Bruschi<sup>1</sup>, Simone do Rocio Senger de Souza<sup>1</sup>

<sup>1</sup>Instituto de Ciências Matemáticas e de Computação (ICMC)  
Avenida Trabalhador São-carlense – 400 – São Carlos – Brazil

naylor@usp.br, {pssouza, sarita, srocio}@icmc.usp.br

**Abstract.** *To master parallel and distributed programming is fundamental for today's computing professionals. Organizations such as ACM, IEEE, and SBC recommend teaching parallel and distributed programming and also provide details on how this topic should be studied in computer-based undergraduate courses. This article presents how the main institutions of higher education in Brazil approach teaching of parallel programming, with the purpose of evaluating its adherence to the syllabus proposed by ACM, IEEE, and SBC. Our results show that there are still differences regarding teaching and the syllabus considered, topics not covered and difficulty in finding documents related to the research.*

**Resumo.** *Dominar a programação paralela e distribuída é primordial para profissionais em Computação. Instituições como ACM, IEEE e SBC recomendam o ensino dessa programação e fornecem detalhamentos de como ela deve ser estudada em cursos de Computação. Este artigo apresenta um panorama do ensino de programação paralela e distribuída de algumas das principais instituições de ensino superior do país, com o propósito de avaliar sua aderência às diretrizes curriculares propostas por ACM, IEEE e SBC. Os resultados mostram que ainda há divergências em relação ao ensino praticado e às diretrizes curriculares consideradas, tópicos não abordados e dificuldade em encontrar documentos de consulta.*

### 1. Introdução

A Computação Paralela e Distribuída (CP&D) já é, há algum tempo, uma importante ferramenta para a sociedade. A CP&D permite usar o poder de processamento disponível nas arquiteturas atuais (*multicore* e com processadores heterogêneos) para a solução de problemas que exigem modelos de programação próprios, com um significativo maior grau de complexidade, para a execução de uma ou mais tarefas concomitantemente [Zarza et al. 2012].

O movimento conhecido como “pensamento paralelo”, proposto por [Blelloch 2009], enfatiza, já há quase uma década, que programas devem ser desenvolvidos com algum nível de paralelismo, na tentativa de se obter ganhos de desempenho, difíceis de serem obtidos apenas por melhorias pelo hardware, devido às limitações da Lei de Moore [Moore 2006].

Desse modo, a CP&D fornece soluções mais eficientes a problemas computacionalmente complexos, explorando as atuais arquiteturas paralelas e novos modelos de programação que permitem o uso de tais arquiteturas.

O ensino de CP&D, ao contrário da computação dita sequencial, aborda desafios especiais, como o uso de diferentes primitivas de comunicação e sincronização, abstrações de processos ou *threads* distintas que interagem, diferentes modelos de programação, problemas de balanceamento de carga e um maior impacto no desenvolvimento e desempenho das aplicações em função da plataforma computacional usada [Kumar et al. 1994].

Tais questões precisam ser compreendidas pelos alunos, que devem desenvolver habilidades e competências, conforme orienta os currículos de computação, para trabalhar com desenvoltura nesses ambientes computacionais, cada vez mais comuns no cotidiano de empresas e universidades [Shamsi et al. 2015]. Além disso, os alunos devem estudar uma gama considerável de conteúdos prévios de outras disciplinas, aumentando sua complexidade [Zarestky and Bangerth 2014], como por exemplo organização e arquitetura de computadores, sistemas operacionais, redes de computadores e programação de computadores.

Dada a importância do ensino de CP&D no contexto mundial, instituições internacionais, como a *Association for Computing Machinery* (ACM) e o *Institute of Electrical and Electronics Engineers* (IEEE), recomendam que a CP&D deve ser considerada uma das grandes áreas de conhecimento na Computação [ACM/IEEE-CS 2013]. No Brasil, esse cenário não é diferente, uma vez que a Sociedade Brasileira de Computação (SBC) recomenda, em seus referenciais de formação, que o ensino de CP&D seja uma das principais competências desenvolvidas em diferentes eixos de formação nos cursos de Computação.

Considerando esse contexto, os cursos de nível superior de Computação no Brasil necessitam incorporar em suas grades curriculares essas diretrizes sobre o ensino da CP&D, com o objetivo de preparar os alunos para um mundo cada vez mais paralelizado e distribuído.

O objetivo deste artigo é apresentar um levantamento dos projetos pedagógicos de Computação no Brasil, com foco no ensino de programação paralela, em algumas das mais importantes universidades brasileiras, dispersas nas diferentes regiões do país. As 45 universidades analisadas foram escolhidas de acordo com os sistemas de pontuação disponíveis em [Folha de São Paulo 2018] e a densidade demográfica de cada região [IBGE 2010]. O levantamento mostra quão aderente estão as universidades nacionais em relação ao ensino de CP&D sugerido pela ACM, IEEE e SBC, destacando convergências e divergências em relação a essas referências, tópicos não abordados e dificuldades em se encontrar documentos oficiais de consulta.

Este artigo é dividido em seis seções. A Seção 2 aborda como a CP&D deve ser estudada segundo a ACM, IEEE e a SBC. A metodologia seguida para obtenção dos dados apresentados é descrita na Seção 3. O panorama dos cursos de Computação no Brasil e suas características são discutidos na Seção 4. Ademais, a Seção 5 contém as limitações desta pesquisa. Por fim, a Seção 6 apresenta as conclusões, finalizando com a proposta para trabalhos futuros.

## 2. A COMPUTAÇÃO PARALELA E DISTRIBUÍDA NOS CURRÍCULOS DE COMPUTAÇÃO

O uso de computadores monoprocessados no passado levou a um ensino de programação baseado em algoritmos sequenciais, em que a CP&D, por muitas vezes, era oferecida usualmente como optativa [ACM/IEEE-CS 2013]. Com o crescente desenvolvimento tecnológico e a consequente adoção de processadores com vários núcleos, fez-se necessário reorganizar o currículo de Computação para essa nova necessidade de programação.

Para ajudar a uniformizar o ensino da CP&D, a ACM e IEEE-Computer Society uniram-se para estabelecer diretrizes curriculares internacionais para cursos de graduação em ciências da Computação e áreas correlatas [ACM/IEEE-CS 2013]. As diretrizes propostas pela ACM de 2013 apresentam dezoito áreas de conhecimento, as quais representam áreas de estudo relevantes em Computação. A CP&D é uma dessas áreas, em virtude da sua importância atual no desenvolvimento de soluções computacionais.

O ensino de CP&D, no currículo de Computação desenhado pela ACM/IEEE, é quantificado em horas (Tabela 1), unidade esta destinada ao tempo necessário para apresentar um material tradicional de leitura. Esse tempo não inclui trabalhos extras, aulas práticas de laboratórios, palestras, entre outros.

**Tabela 1. Ensino da CP&D nas diretrizes curriculares para os cursos de Computação, segundo a ACM/IEEE**

Tópicos	Horas		Inclui Eletivas
	Mínimas	Complementares	
Fundamentos do Paralelismo	2		Não
Decomposição Paralela	1	3	Não
Comunicação e Coordenação	1	3	Sim
Algoritmos Paralelos, Análise e Programação		3	Sim
Arquiteturas Paralelas	1	1	Sim
Desempenho Paralelo			Sim
Sistemas distribuídos			Sim
<i>Cloud Computing</i>			Sim
Modelos Formais e Semântica			Sim

Anteriormente à revisão do currículo de 2013, os conteúdos de CP&D, incluindo a programação paralela/distribuída, estavam espalhados dentro de outras áreas do conhecimento. Devido à sua atual importância, a CP&D foi, nessa versão, especialmente categorizada em uma nova área incluindo conteúdos específicos como: fundamentos do paralelismo, decomposição paralela, algoritmos paralelos, análise e programação e arquiteturas paralelas [ACM/IEEE-CS 2013].

A Sociedade Brasileira de Computação (SBC) é a referência acadêmica nacional que direciona o ensino de Computação no Brasil. Para isso, definiu o documento “Referenciais de Formação para os Cursos de Bacharelado em Ciência da Computação (RF-CC-17)”, para servir de base a Projetos Pedagógicos [Zorzo et al. 2017].

A Tabela 2 aponta os tópicos explicitados pela SBC e que são equivalentes aos tópicos referenciados pela ACM/IEEE em CP&D. O documento da SBC não traz os detalhes de cada disciplina, e também é possível observar que algumas áreas não são cobertas, como *Cloud Computing*, *Decomposição Paralela* e *Comunicação e Coordenação*. Entretanto, essas áreas podem estar cobertas em outros tópicos, como por exemplo, em *Programação Paralela e Distribuída*.

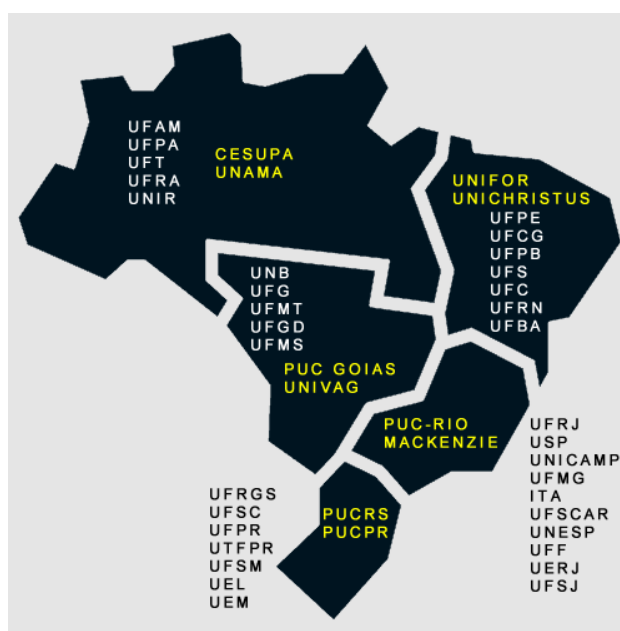
**Tabela 2. Relação dos tópicos de CP&D abordados pela SBC com a ACM/IEEE**

ACM/IEEE	SBC
<b>Fundamentos do Paralelismo</b>	Sistemas Concorrentes / Processamento Paralelo
<b>Decomposição Paralela</b>	
<b>Comunicação e Coordenação</b>	
<b>Algs. Par., Análise e Programação</b>	Programação Paralela e Distribuída
<b>Arquitetura Paralela</b>	Arquiteturas Paralelas de Computadores
<b>Desempenho Paralelo</b>	Avaliação de Desempenho
<b>Sistemas Distribuídos</b>	Sistemas Distribuídos
<b>Cloud Computing</b>	
<b>Modelos Formais e Semântica</b>	Métodos Formais

Considerando essas reestruturações do ensino da CP&D pela SBC, ACM e IEEE, e a importância da programação paralela/distribuída na formação dos futuros profissionais em Computação, faz-se importante determinar como tal programação vem sendo abordada pelas universidades brasileiras. As próximas seções visam a mostrar tal cenário.

### 3. METODOLOGIA PARA OBTENÇÃO DOS DADOS

Como metodologia para realizar o levantamento proposto neste artigo, foram escolhidas 45 universidades, referência no Ranking Universitário Folha (RUF) [Folha de São Paulo 2018], conforme mostra a Figura 1.



**Figura 1. Universidades pesquisadas em função das suas localizações.**

A escolha pelo RUF deve-se à sua abrangência, tanto em relação às instituições de ensino consideradas (todas as universidades, centros universitários e faculdades do país), quanto pelos uso de cinco indicadores diferentes: pesquisa, internacionalização, inovação, ensino e mercado.

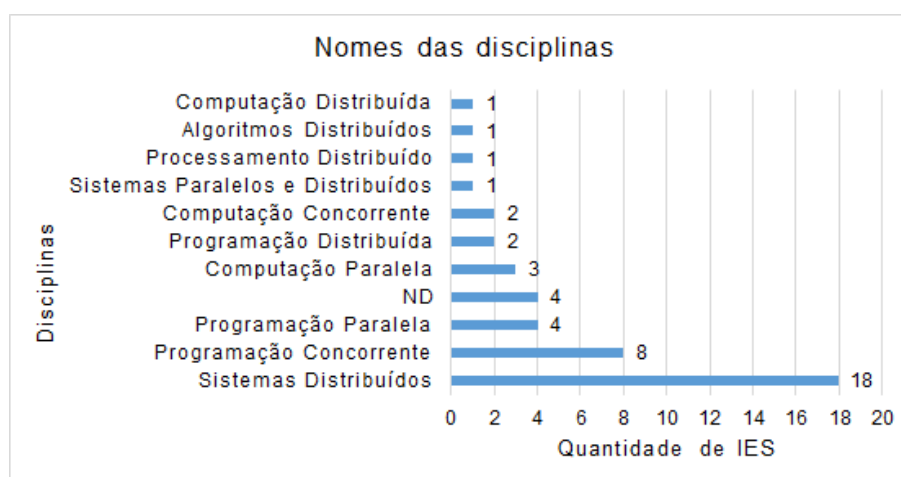
Em relação à pesquisa, por exemplo, são considerados itens como total de publicações científicas, citações, citações por docente, bolsistas produtividade, entre outros. Já em relação ao ensino, são considerados itens como percentual de docentes com mestrado e doutorado, dedicação integral e parcial e nota do Enade [Righetti 2015].

A quantidade de instituições selecionadas para o estudo neste artigo foi determinada em função da maior pontuação no ranking e da densidade demográfica de cada região do país. Desse modo, a quantidade de instituições selecionadas variou de sete (Norte e Centro-Oeste) a doze (Sudeste). Também foram selecionadas, intencionalmente, duas universidades privadas, de acordo com as melhores colocações no ranking, cujos nomes são apresentados na cor amarela na Figura 1.

Os materiais consultados para coleta dos dados estavam disponíveis nos sites de cada Instituição de Ensino Superior (IES), como matrizes curriculares, planos de ensino, planos de aula e os projetos pedagógicos. Os dados obtidos referem-se ao semestre que a disciplina é ofertada, à carga horária, obrigatória ou eletiva, referências bibliográficas, correquisitos, pré-requisitos e à ementa.

#### 4. O ENSINO DE PROGRAMAÇÃO PARALELA E DISTRIBUÍDA NO BRASIL

Para iniciar a discussão sobre o panorama do ensino de programação paralela no Brasil, foram selecionados quais os nomes mais comuns de disciplinas relacionadas com a CP&D, as quais remetem ao ensino da programação paralela e distribuída (Figura 2). A disciplina de Sistemas Distribuídos é a mais comum, seguida por Programação Concorrente.

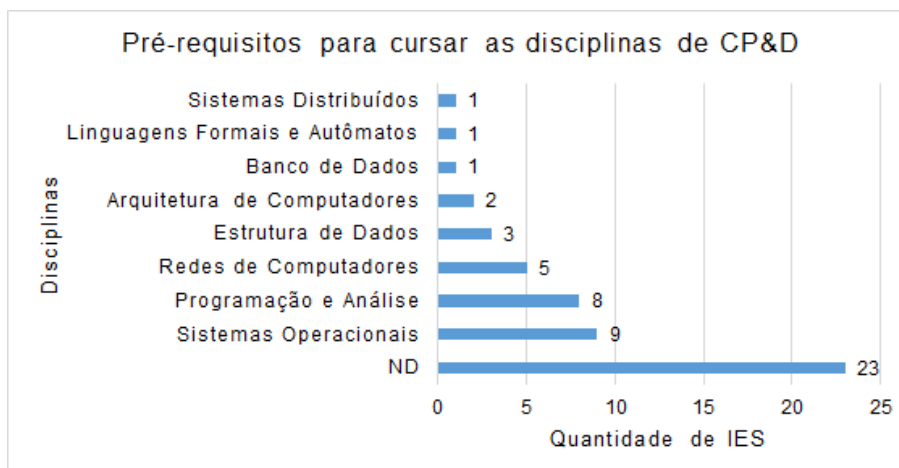


**Figura 2. Nome das disciplinas relacionadas com CP&D que remetem ao ensino de Programação Paralela e/ou Distribuída.**

Considerando as instituições pesquisadas, apenas a UFPB, UFSJ, PUC-RIO e UEL não possuem disciplinas relacionadas com paralelismo especificamente em suas ma-

trizes (indicadas como “ND”) ou não disponibilizaram a informação em seus *web-sites*. Ressalta-se que estes tópicos podem estar inseridos em outras disciplinas.

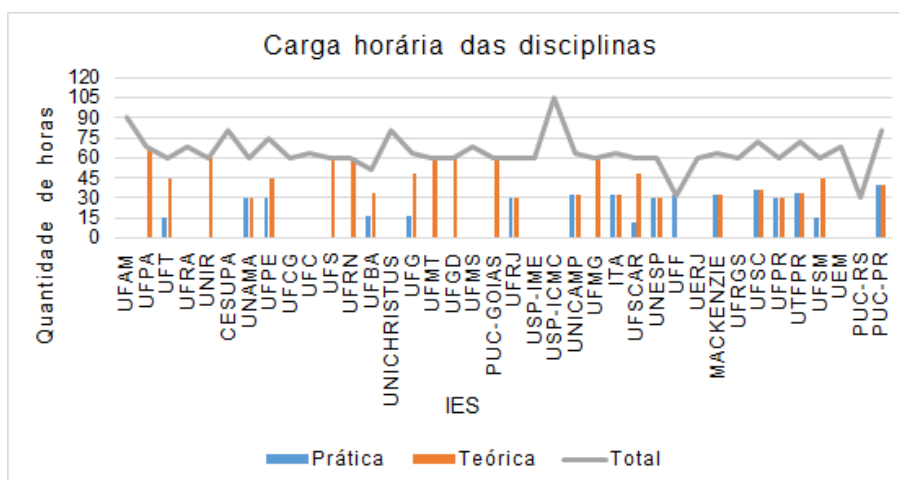
Na Figura 3 são apresentados os pré-requisitos (disciplinas que os alunos precisam cursar antes) mais utilizados pelas IES.



**Figura 3. Pré-requisitos das disciplinas relacionadas com CP&D.**

Nenhuma IES apresentou os correquisitos (tópicos que necessitam ser cursados concomitantemente à disciplina sobre computação paralela). As disciplinas de Sistemas Operacionais, Programação e Redes são as mais comuns. Desse total, vinte e três instituições não possuem pré-requisitos ou não foram encontrados.

A Figura 4 mostra a carga horária total, prática e teórica para as disciplinas. Desse total, três IES não disponibilizaram suas cargas horárias, sendo que quatro não possuíam essas disciplinas. Além disso, muitas universidades mostram a carga horária total, porém não especificam a quantidade de horas reservadas para o ensino prático e teórico.

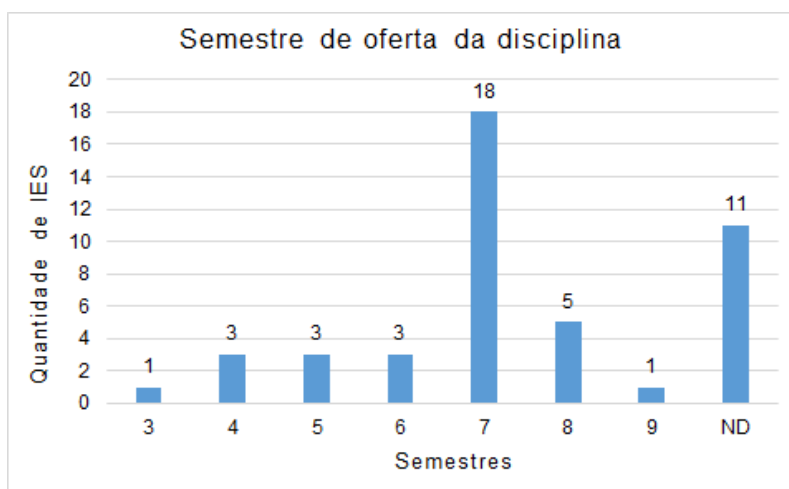


**Figura 4. Carga horária total das disciplinas relacionadas com CP&D.**

Na Figura 4, observa-se ainda que, para aquelas IES que disponibilizam suas informações por completo, oito instituições ministram somente aulas teóricas para a dis-

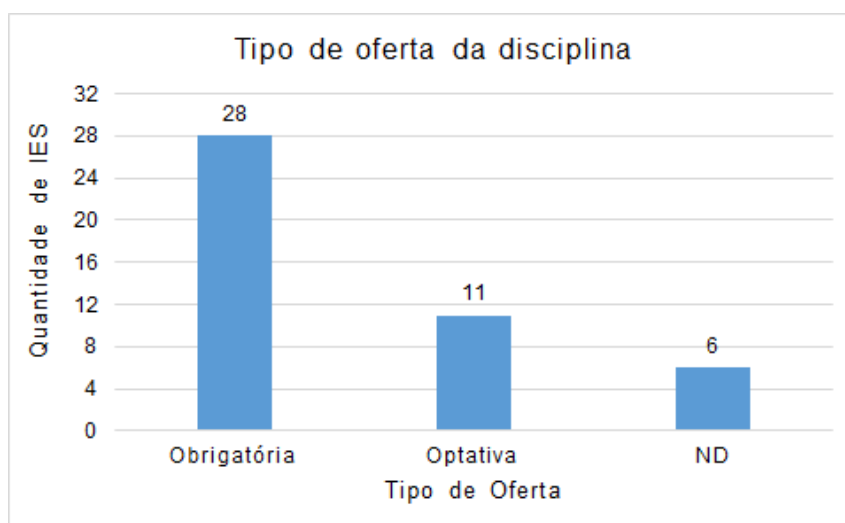
ciplina da grade de computação paralela, enquanto uma IES (UFF) ministra apenas aulas práticas. A média da carga horária total é de 60 horas, sendo a máxima 105 horas (ICMC-USP) e a mínima com 30 horas (PUC-RS). A média observada de aulas práticas é de 30 horas, e 45 horas para aulas teóricas.

A Figura 5 disponibiliza o semestre em que essas disciplinas são ofertadas, sendo que onze instituições não disponibilizaram os referidos dados. A maior parte das IES disponibiliza a disciplina no 7º semestre. É interessante observar também que nenhuma delas oferta no 1º e 2º semestres, o que provavelmente acontece em razão do tempo necessário para cumprir os pré-requisitos.



**Figura 5. Semestre de oferta das disciplinas relacionadas com CP&D.**

A Figura 6 mostra se a disciplina é ofertada como optativa ou obrigatória. Usualmente ela é obrigatória, sendo que seis instituições não disponibilizam a informação.

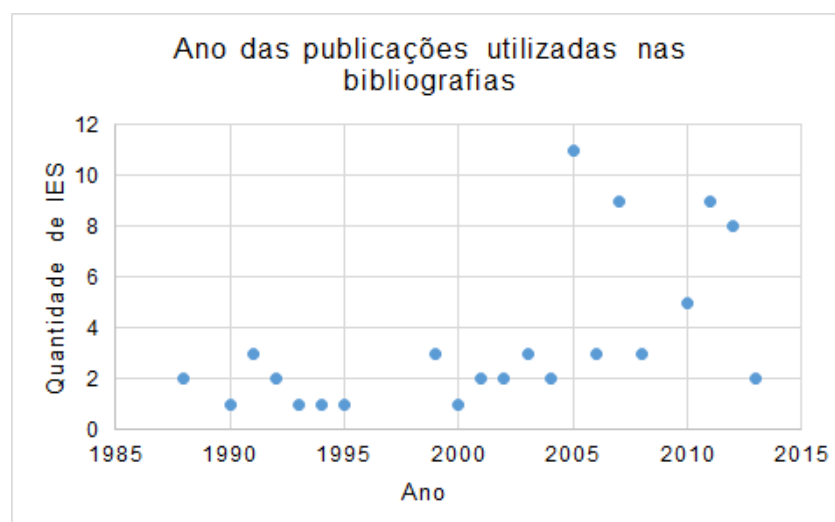


**Figura 6. Oferta das disciplinas relacionadas com CP&D nas universidades.**

A Figura 7 traz o agrupamento dos anos das bibliografias utilizadas nos planos de ensino daquelas IES que disponibilizaram *on-line*. A maior parte das instituições possui

bibliografias com ano de publicação entre 2005 e 2012. Todavia, algumas IES trabalham com bibliografias da década de 80 (1988), enquanto a mais atual é de 2013.

A Tabela 3 apresenta a aderência das disciplinas de CP&D em relação às diretrizes da ACM/IEEE e SBC. Ressalta-se que os referenciais de formação da SBC não trazem as especificações de cada disciplina; sendo assim, elas foram agrupadas de acordo com o nome e não seu conteúdo abordado. Pode-se observar que poucos cursos atingem os últimos tópicos da Tabela 3, como *Cloud Computing* e Métodos Formais. Além disso, 49% dos cursos investigados ofertam, explicitamente, em suas grades, a programação paralela, e apenas 20% abordam tópicos sobre desempenho paralelo.



**Figura 7. Ano das referências bibliográficas utilizadas nas disciplinas relacionadas com CP&D.**

## 5. LIMITAÇÕES

Nesta seção alguns pontos são destacados quanto a sua ameaça à validade. A consulta dos planos pedagógicos (PPC) foi realizada *on-line* e nos sites oficiais de cada instituição de ensino superior. É possível que a informação esteja desatualizada. Muitas IES não disponibilizaram o PPC completo, apenas partes que julgaram importantes, podendo afetar os resultados deste estudo.

Além disso, muitos tópicos, relativos à programação paralela e distribuída, podem estar inseridos dentro de outras disciplinas, como Programação de Computadores e Sistemas Operacionais.

## 6. CONCLUSÕES

A programação paralela e distribuída continua a ser mais difícil nos ambientes de programação contemporâneos [ACM/IEEE-CS 2013]. No entanto, compreender o processamento paralelo é vital para os cursos de Computação e, aprender esses modelos de programação o quanto antes, pode fortalecer o aprendizado dos alunos nesta área.

Todavia, é possível notar que a maioria dos cursos optaram por oferecer essa disciplina nos semestres mais tardios, devido ao seu alto grau de complexidade e para satisfazer seus pré-requisitos.



**Tabela 3. Relação da aderência das disciplinas de programação paralela e distribuída em relação aos currículos de referência da ACM/IEEE e SBC**

SBC	Sistemas Concorrentes / Processamento Paralelo			Programação Paralela e Distribuída	Arquiteturas Paralelas de Computadores	Avaliação de Desempenho	Sistemas Distribuídos		Métodos Formais
ACM/IEEE	Fundamentos do Parallelismo	Decomposição Paralela	Comunicação e Coordenação	Algoritmos Paralelos, Análise e Programação	Arquitetura Paralela	Desempenho Paralelo	Sistemas Distribuídos	Cloud Computing	Modelos Formais e Semântica
<b>IES</b>									
UFAM	X		X		X		X		
UFPA	X		X		X		X		
UFT	X		X		X		X		
UFRA	X		X		X		X		
UNIR	X	X	X	X	X	X			
CESUPA	X	X	X	X	X		X		
UFPE	X	X	X	X					
UFCG	X		X		X	X			
UFS	X		X		X		X		
UFC	X	X		X		X			
UFRN	X	X	X	X					
UFBA	X	X	X	X	X				
UNIFOR	X		X	X	X		X		
UFG	X	X	X	X	X	X			
UFMT	X		X				X		
UFGD	X		X	X			X		
UFMS	X		X	X	X	X	X		
PUC-GOIAS	X		X				X		
UFRJ	X	X	X	X			X		
USP-IME	X	X	X	X					
USP-ICMC	X	X	X	X	X	X			X
UNICAMP	X	X	X	X	X				
UFMG	X		X			X	X	X	
ITA	X	X	X	X			X		
UFSCAR	X		X	X	X		X	X	
UNESP	X						X	X	
UFF	X		X	X	X				
MACKENZIE	X			X	X				
UFRGS	X	X	X	X	X		X		
UFSC	X		X	X					
UFPR	X		X	X		X			
UTFPR	X		X		X		X		
UFMS	X		X				X		
UEM	X		X	X		X			
PUC-RS	X		X						X

Muitas IES utilizam bibliografias não contemporâneas, conseqüentemente, não abordando tópicos relativamente novos, como *Cloud Computing*. Destaca-se também que apenas 62% destas instituições ofertam a disciplina como obrigatória, sendo que as demais não ofertam como obrigatória ou nem possuem a oferta desta.

Outro fator importante para o sucesso de uma disciplina, em função do conhecimento adquirido pelos alunos, está em sua carga horária. Neste estudo, notou-se uma grande divergência em relação à quantidade de horas necessárias. A média apontada ficou em torno de 60 horas, enquanto a ACM/IEEE aponta 80 horas mínimas.

Em relação à aderência das instituições às diretrizes curriculares da ACM/IEEE e SBC, muitas IES vão além da proposta mínima, abrangendo tópicos atuais. Entretanto, muitas não abordam tópicos essenciais, como é o caso da Programação Paralela e Distribuída, tópico de extrema importância atualmente.

Para trabalhos futuros, espera-se levantar os pontos positivos e negativos de cada disciplina, analisar seus materiais de apoio, ferramentas utilizadas em sala de aula, bem como suas metodologias de ensino prático e teórico.

## Referências

- ACM/IEEE-CS (2013). Computer science curricula 2013. Technical report, ACM Press and IEEE Computer Society Press.
- Blelloch, G. E. (2009). Parallel thinking. *SIGPLAN Not.*, 44(4):1–2.
- Folha de São Paulo (2018). RUF: Ranking Universitário. <http://ruf.folha.uol.com.br/2017/ranking-de-cursos/computacao,18/04/2018>.
- IBGE (2010). Censo demográfico. <http://www.censo2010.ibge.gov.br>.
- Kumar, V., Grama, A., Gupta, A., and Karypis, G. (1994). *Introduction to Parallel Computing: Design and Analysis of Algorithms*. Benjamin-Cummings Publishing Co., Inc., Redwood City, CA, USA.
- Moore, G. E. (2006). Cramming more components onto integrated circuits, reprinted from electronics, vol 38, nr 8, 1965. *IEEE Solid-State Circuits Society Newsletter*, 11(3):33–35.
- Righetti, S. (2015). Fórum Nacional de Educação Superior.
- Shamsi, J. A., Durrani, N. M., and Kafi, N. (2015). Novelties in teaching high performance computing. In *IEEE Int Par and Dist Proc Sym Workshop*, pages 772–778.
- Zarestky, J. and Bangerth, W. (2014). Teaching high performance computing: Lessons from a flipped classroom, project-based course on finite element methods. In *2014 Workshop on Education for High Performance Computing*, pages 34–41.
- Zarza, G., Lugones, D., Franco, D., and Luque, E. (2012). An innovative teaching strategy to understand high-performance systems through performance evaluation. *Procedia Computer Science*, 9:1733 – 1742.
- Zorzo, A., Nunes, D., Matos, E., Steinmacher, I., Leite, J., Araujo, R., Correia, R., and Martins, S. (2017). Referenciais de formação para os cursos de graduação em computação. sociedade brasileira de computação (SBC). 153p. Technical report, ISBN 978-85-7669-424-3.