Revista Brasileira de Informática na Educação – RBIE
Brazilian Journal of Computers in Education
(ISSN online: 2317-6121; print: 1414-5685)
http://br-ie.org/pub/index.php/rbie

Submission: 01/Sep/2022; Camera ready: 02/Jul/2023; 1st round notif.: 10/Apr/2023; Edition review: 17/Aug/2023; Available online: 17/Aug/2023; Published: 17/Aug/2023;

A Syllabi Analysis of CS1 Courses from Brazilian Public Universities

Eryck Pedro da Silva
State University of Campinas (UNICAMP)
ORCID: 0000-0001-5141-6936
eryck.silva@ic.unicamp.br

Ricardo Caceffo
State University of Campinas (UNICAMP)
Virtual University of São Paulo (Univesp)
ORCID: 0000-0001-9152-0658
caceffo@ic.unicamp.br

Rodolfo Azevedo
State University of Campinas (UNICAMP)
ORCID: 0000-0002-8803-0401
rodolfo@ic.unicamp.br

Abstract
The design and administration of methodological interventions is a possible way to address dropout and failure rates in undergraduate introductory programming courses (CS1). However, to implement such strategies, it is required to identify how CS1 courses are organized and offered to students. In this work, we analyzed 225 syllabi from CS1 courses distributed in 95 Brazilian public universities. We collected these syllabi from Brazilian undergraduate programs with focus on Computer Science aspects. We report context information regarding the most covered topics, most common CS1 course names, when undergraduates take the course, total class hours, and the programming paradigms and languages taught. The results indicate that the Brazilian scenario has its own characteristics that differs from those presented in related work conducted in other countries or world regions. In Brazil, we identified 90% of the analyzed CS1 courses teaches the procedural paradigm, with C as the most common programming language (53% of the total). These results differ from those conducted in other countries where Java and object-oriented are the most common languages and paradigm taught in CS1 courses. We believe that our results can be used to: (1) provide an update to those interested in the Brazilian scenario of CS1 courses; (2) support future interventions in teaching and learning of CS1; and (3) support the Brazilian community in the development of future CS1 syllabi.
Keywords: *Syllabi Analysis; Undergraduate Studies; Introductory Programming; Course Contextualization; CS1.*

Cite as: Silva, E.P., Caceffo, R., & Azevedo, R. (2023). A Syllabi Analysis of CS1 Courses from Brazilian Public Universities. Brazilian Journal of Computers in Education (Revista Brasileira de Informática na Educação – RBIE), 31, 407-436. DOI: 10.5753/rbie.2023.2870.

1 Introduction

The utilization of computers, as a tool, continues to expand in most diverse areas. In 2005, it was estimated there would be over 90 million of end users in American workplaces, varying from spreadsheet users to actual programmers (Scaffidi, Shaw, & Myers, 2005). As consequence of this, more undergraduate programs are offering Computer Science (CS) classes to their students. These classes are often referred as *CS+X* in the literature (Sloan et al., 2020). Although CS+X classes are often in the Science, Technology, Engineering and Mathematics (STEM) field, such as Cognitive Psychology (Brodley et al., 2022) and Biology (Dodds, Libeskind-Hadas, & Bush, 2010), they can also be found in other areas, such as Law (Sloan, Taylor, & Warner, 2017).

A typical course within the teachings of CS regards the concepts of computer programming. In 1978, the Association for Computing Machinery (ACM) defined the terms CS1 and CS2 to describe the first two programming courses a student takes in an CS undergraduate program (Austing, Barnes, Bonnette, Engel, & Stokes, 1979). These terms have been used over 40 years, typically assigning CS1 to basic concepts of programming, and CS2 to data structures. However, there is no consensus of what exactly should be taught in these courses (Hertz, 2010). Literature reviews also beckon this statement by mentioning that publications regarding CS1 course contents are still stable over the last few decades (Becker & Quille, 2019; Luxton-Reilly et al., 2018). In Brazil, CS1 courses have many different names, even within a same higher education institution (Nascimento, 2018). Examples of the names include "Introduction to Programming", "Introduction to Algorithms", "Algorithms and Data Structures", "Algorithms and Computer Programming", among others.

Usually, undergraduate students start developing their logical and systematical thinking in CS1 courses. These courses also present a first programming language to the students. In other words, CS1 courses are an important background not only for the remainder of the CS undergraduate program, but also for the professional formation of the undergraduates (Campos, 2010; Sobral, 2019). However, along with this importance, CS1 teaching faces recurrent challenges regarding high rates of failure and dropout (Bosse & Gerosa, 2015; Kinnunen & Malmi, 2006; F. Pereira et al., 2020; Petersen, Craig, Campbell, & Tafliovich, 2016; Walker, 2017). To help solving these problems, several research with interventions in teaching and learning of CS1 have been made (L. Araujo, Bittencourt, & Chavez, 2021; Caceffo, Gama, & Azevedo, 2018; Caceffo, Wolfman, Booth, & Azevedo, 2016; Lima, Carvalho, Oliveira, Oliveira, & Pereira, 2021; Vahldick, Marcelino, & Mendes, 2021).

Given the importance and the challenges that CS1 courses have, how do we construct solutions that will work in the multiple scenarios in which teaching and learning of these courses happen? In terms of programming being hard to learn, Luxton-Reilly (2016) said that we make introductory courses difficult by establishing unrealistic expectations upon novices. The same author also stated that revisiting what is expected from students at the end of introductory programming courses might be the key to improve students' learning, address negative impacts of disciplinary measures, and create a more equitable environment in these classes. A possible first step to revisit what is expected from students would be to understand which concepts are being taught in CS1 courses. This step proves to be another challenge because some educators might say that all introductory programming courses teach the same thing: the basics of computer science and computer programming (Luxton-Reilly et al., 2018). However, if we are to follow this rationale, there would be no need for research about CS1 curricula - and that does not happen in practice.

This work was motivated by the potential that an assessment of the characteristics CS1 courses could have. Among these characteristics were the most covered topics. With this list of the frequency of the topics, researchers would be able to understand which topics are most or least covered so they could construct teaching and learning interventions. However, to ensure that this assessment would be holistic, the search ought to be done not only in terms of quantity, but also covering a broad geographic context. We decided to focus on Brazil because: (i) as far as we know, similar research in the literature do not state that Brazil was covered in their results (Avouris, 2018; Becker, 2019; Becker & Fitzpatrick, 2019; Hertz & Ford, 2013; Schulte & Bennedsen, 2006); and (ii) we did not find any other work that conducted a holistic search in Brazil to assess the characteristics of CS1 courses with focus on the most covered topics. Finally, we decided to limit our research to Brazilian public universities, which are composed by federal, state, and municipal institutions. Public universities were selected because they are present in all geographical regions of the country, and they are the ones that most appear in international rankings, such as the World University Rankings of the Times Higher Education[1].

An interesting approach to help with the research of the most covered topics would be to consult entities that are directly involved with the design of undergraduate programs. In Brazil, the organization and research of CS teaching have always been conducted by the Ministry of Education together with the Brazilian Computer Society (SBC). Among the contributions by both groups are discussions that elaborate and assess CS undergraduate programs. The Formation Guidelines for Computer Science Undergraduate Programs (Zorzo et al., 2017) are a result of several research that culminated in orientation to develop pedagogical projects. The document is organized by the types of CS undergraduate program in Brazil. In a global scenario, the Computing Curricula 2020 (CC2020 Task Force, 2020) provides recommendations in the same way. ACM and the Institute of Electrical and Electronic Engineers (IEEE-CS) elaborated this document, and it is also endorsed by SBC.

The Formation Guidelines and the Computing Curricula 2020 are well suited documents for the creation of pedagogical projects. However, they focus on competencies that each student must develop by the end of the undergraduate program, not necessarily being achieved in each of the courses. Considering this, they would not be the ideal source to obtain information specifically for CS1 courses. In this work, we chose to search and analyze documents that had already been created by the public universities: their pedagogical project and the syllabi for CS1 courses. By analyzing these documents, we would be able not only to identify the covered topics, but also other characteristics of the CS1 courses.

We created five research questions to contextualize the Brazilian CS1 courses from public universities:

RQ1: *What are the most common topics covered in CS1 courses from Brazilian public universities?*

RQ2: *What are the most common CS1 courses' names from Brazilian public universities?*

RQ3: *When do Brazilian public universities' curricula suggest students take the CS1 course?*

[1] https://www.timeshighereducation.com/world-university-rankings

RQ4: *What is the average of the total class hours of the CS1 courses from Brazilian public universities?*

RQ5: *What are the programming paradigms and programming languages taught in CS1 courses from Brazilian public universities?*

We manually searched the websites for 157 Brazilian public universities and curated 225 CS1 syllabi present in a total of 95 institutions. We used the pedagogical projects together with the CS1 syllabi to answer each of the aforementioned research questions. Our results indicate that the Brazilian scenario has characteristics that differs from those in other countries, such as the programming paradigm and languages that are taught.

We believe that our results can contribute to researchers interested in an update about the context of CS1 teaching in Brazil, especially in worldwide research where the spoken language can be a barrier (almost all Brazilian syllabi we found were only in Brazilian Portuguese, for example). The results can also be used to help in the construction of teaching and learning interventions of CS1 courses. Lastly, we believe that the list of most covered topics can also be used in the creation of new syllabi by Brazilian higher education institutes.

The remainder of this paper is organized as follows. In Section 2 we present the background and related work. In Section 3 we detail the methodology used, followed by the obtained results in Section 4. We discuss the obtained results in Section 5. In Section 6 we present the limitations and threats to validity of this research. Lastly, the conclusions are presented in Section 7.

2 Background and Related Work

The syllabus is a valuable tool in higher education because it is often the first formal way in which students receive information about a course. Syllabi analysis is important because the syllabus is an educational tool with functionalities that are commonly unknown to administration, faculty, and students (Eberly, Newton, & Wiggins, 2001). While designing a syllabus or course outline, the instructor needs to take careful consideration of topics covered, assignments' due, and learning objectives (McKeachie, 1978). McKeachie (1978) also says that, for undergraduates, the syllabus establishes expectations and directions for a particular course, thus providing a way of security. Even though the definitions of what a quality syllabus has not been clearly defined (Eberly et al., 2001), some suggested models go way beyond the aforementioned characteristics. A course calendar, grading information with the rubrics that are used, additional resources, and a guide to use the syllabus are examples of suggested items to be included in a syllabus (Grunert, 1997).

Becker and Fitzpatrick (2019) analyzed syllabi from CS1 courses of 916 institutions present in the QS World University Rankings from 2016-2017[2]. The authors were motivated to answer what exactly CS1 teachers expect from their students at the end of the introductory course. While they were searching for learning outcomes, a total of 15 topics were among the most covered: testing and debugging, writing programs, and selection statements were the top three. Becker and Fitzpatrick also reported information about the most used programming languages, in which Java, Python, and C++ were the most used. Finally, they also created an online tool for the community

[2] https://www.topuniversities.com/university-rankings/world-university-rankings/2016

in which it is possible to sort and analyze their gathered date about the syllabi. However, as mentioned in their work, they could only process English-created material. We believe this could possibly be the reason Brazil was not present in their analysis.

Porfirio, Pereira, and Maschio (2021) also did an analysis of syllabi from CS1 courses. They consulted 10 Brazilian federal universities (two for each geographic region) listed in the RUF 2018 Ranking[3] for Computing Programs. The authors were interested in discovering basic concepts that every student should master independently of the adopted approach in CS1 courses. In their work, they reported 10 most covered topics, with conditional structures, repetition structures, and data types being the top three. The authors' main goal was to create an automated assessment of computer programming skills by analyzing source code, called the *A-Learn EvId* method.

Syllabi analysis is also present in other CS related areas. Fréchet, Savoine and Dufresne (2020) analyzed syllabi of text-analysis courses from 45 graduate political science programs. The authors presented a systematic method for analyzing syllabi and retrieving information to help early-career professors and political science departments to build syllabi for text-analysis courses. The authors reported a method for evidencing most cited academic papers and books used in the syllabi, and about the choice of software between R or Python. Using the same systematic method, Abad, Ortiz-Holguin, and Boza (2021) analyzed syllabi of 51 Distributed Systems courses to answer if what is being taught in these courses matches important curricula initiatives. The authors reported the most covered topics, books, and papers listed in the analyzed syllabi of Distributed Systems courses.

The aforementioned studies used syllabi analysis approaches to find common topics in CS1 and CS related areas, sometimes extending it to other information also presented in these documents. However, analyzing syllabi has been deemed a challenging task to do in a large scale (Becker & Fitzpatrick, 2019; Tew & Guzdial, 2010). As a result of this, other ways of listing topics in CS1 were used in the literature, such as surveying academics (Hertz & Ford, 2013; Schulte & Bennedsen, 2006) and textbook analysis (Berges & Hubwieser, 2013; Tew & Guzdial, 2010). In general, these studies did not have the most covered topics as a main objective, but instead focused on aspects such as importance or difficulty perceived of said topics. Hertz and Ford (2013) culled a list with 17 topics from the literature and surveyed CS1 professors to investigate correlations between the importance of these topics and students' developed skills. Schulte and Bennedsen (2006) did a similar approach in surveying professors to find what they teach, what they believe that should be taught, and the CS1 topics students tend to have difficulty with. Their survey used a list of 28 topics to compose the analysis.

Berges and Hubwieser (2013) developed a semiautomated mechanism for textual analysis. The authors used this mechanism with five CS1 books, which addressed the object-oriented paradigm, to elaborate Concept Specification Maps. Their provided list of topics varied for each book, with their most populated listings ranging between 17 and 18 topics. Tew and Guzdial (2010) used a bottom-up approach with 12 textbooks to identify concepts taught by multiple CS1 courses. Their initial analysis culminated in a wide list with more than 400 topics. However, after further refinements that used other established curricula, and sorting concepts by programming paradigm, they ended with a list of 29 topics. In the end, the authors used their result to create a validated assessment of CS1 topics known as Foundational CS1 (FCS1) (Tew & Guzdial, 2011).

[3]https://ruf.folha.uol.com.br/2018/ranking-de-cursos/computacao/

We present a summary of the presented related work in Table 1. We briefly compare the methodology, total of respondents, syllabi or textbooks analyzed, and if each work covered Brazil in their analysis. We omitted research from Abad et al. (2021) and Fréchet et al. (2020) from the table because they did not evaluate CS1 courses.

Table 1: Summary of the related work presented in this section. Table is sorted alphabetically by the methodology.

Research	Methodology	N	Covered Brazil?
(Hertz & Ford, 2013)	Online survey with instructors	99	Not mentioned*
(Schulte & Bennedsen, 2006)	Online survey with instructors	349	Not mentioned*
(Becker & Fitzpatrick, 2019)	Syllabi analysis	234	No
(Porfirio et al., 2021)	Syllabi analysis	10	Yes
(Berges & Hubwieser, 2013)	Textbook analysis	5	N/A**
(Tew & Guzdial, 2010)	Textbook analysis	12	N/A**
This work	Syllabi analysis	225	Yes

*Authors did not mention Brazil among the respondents' location.
**Verification not applicable since the methodology used textbooks.

Our work differs from the studies done by Becker and Fitzpatrick (2019) and Porfirio et al. (2021) because we analyzed 225 CS1 syllabi from Brazilian public universities, covering all geographical regions. We also report more information such as most common names of CS1 courses, research question also presented in Abad et al. (2021) albeit their focus was on Distributed Systems courses. Our ranking of the most covered topics was done without filters of importance nor perceived difficulty of these topics, differing from other studies in the field that identify difficult topics and develop methods to mitigate these difficulties (A. Araujo et al., 2021; L. Araujo et al., 2021; Caceffo et al., 2016; R. Pereira, Peres, & Silva, 2021). These kind of research shows where they succeeded or failed, thus enriching the community (Valentine, 2004). However, the field could benefit more from empirical applications involving the results obtained for the already developed methods instead of creating new ones (Luxton-Reilly et al., 2018).

3 Methods

In this work, we gathered data directly from publicly available sources provided from Brazilian universities to answer each proposed research question. As previously stated, we decided to focus on public universities because they are the most present in rankings, and they cover all Brazilian geographical regions, thus providing a broad criterion for analysis.

In Brazil, public universities are higher education institutions financially maintained by the government. As consequence of this, one of their main characteristics is to serve the public and collective interest (FIA, 2019). Public universities are also known for their rigorous undergraduate admission process based on entrance exams, and the absence of tuition costs for the students. There are three types of public universities in Brazil: federal, state, and municipal. To guide our research in an ordered manner, we used three listings that described the universities from each desired group. According to the lists, there are 69 federal (Wikipedia, 2021), 47 state (Wikipedia, 2022a), and 41 municipal (Wikipedia, 2022b). Since many undergraduate programs can offer CS1 courses, we decided to limit our scope to Computer Science related undergraduate programs. As the Formation Guidelines (Zorzo et al., 2017) were organized by CS undergraduate programs, it seemed appropriate to use them in our work. We analyzed the following programs: Bachelor in Computer Science, Bachelor in Computer Science and Engineering, Bachelor in Software Engineering, and Bachelor in Information Systems. Another analyzed program can be described as a degree in Computer Science that prepares CS teachers to act in Brazilian's first and secondary educational levels. We did include undergraduate programs with minor difference in their names from the ones we sought, as well as programs which were in person or in distance learning format. It is important to state that even though undergraduate programs that offers a Technology degree are present in the Formation Guidelines, we did not include them in our research.

3.1 Data Collection

The entire process of data collection was done manually since data had to be collected directly from publicly available sources from the institutions. The following steps describe how this process occurred for all types of public universities (federal, state, and municipal):

1. Selection of a public university from the corresponding base listing (Wikipedia, 2021, 2022a, 2022b). This base listing was composed of two types of institutions: universities and colleges. However, as our focus was Brazilian public universities, we discarded the latter type of institution from our analysis.

2. Navigation to the university's official website.

3. Search for the desired undergraduate programs offered by the university. We decided to verify all possible campi that had each targeted program so that we could identify distinct versions that CS1 programs may have. This step repeated itself until all desired undergraduate programs were analyzed. If the university did not have any of the targeted programs, it was discarded from our analysis.

4. We used two criteria for determining if a course was considered as CS1: whether it had focus on teaching computer programming concepts (including or not the teaching of a programming language); and whether it was the first course with this former criterion listed in the university's suggested curriculum order. Our approach was similar to Guo's (2014) as we did not consider courses that only teach basic computer literacy. If more than one course had both criteria, we compared their syllabi and chose the one which had more topics covered. However, if a public university had two courses that taught different programming paradigms in the same curricular period, both courses were included. Also, if a CS1 course was divided into two, one taught in class and the other in the laboratory, we merged both.

5. Once a CS1 course had been identified, two documents were searched to answer the proposed research questions: the pedagogical project of the undergraduate courses, and the syllabus of the CS1 courses. We expected both documents to complement themselves regarding the information we needed, but our focus was the syllabi. However, these documents were not always sufficient to identify what we wanted. In some cases, the syllabi were already present in the pedagogical project, in others, though, they were not, leading us to search the institution's website for more documents. One example of other type of document was the program contents for a specific semester. If we could not find any other document within

their website, we Google searched "<institution name> + <CS1 course name> + <syllabi>" (in Brazilian Portuguese) to find instructors' personal websites or repositories that contained information about the syllabi. If we could not find or did not have access to any syllabi information using the aforementioned methods (sometimes the official websites were offline or they required login information to have access), the CS1 course was discarded from our analysis even if we had found other desired information about it.

6. Saving of the retrieved data by collecting and pasting directly from the sources.

At the end of the data collection process, we had one main document for each type of public university (federal, state, and municipal). The documents were organized by geographical region, institution, undergraduate program and CS1 course name. The data was collected in two cycles: the first happened during July and September of 2021, in which we focused on federal universities; and the second happened during April and July of 2022, in which we covered state and municipal universities.

3.2 Data Analysis

Since we used different approaches to answer the proposed research questions, we decided to describe the analysis methods for each one. We had identified two issues before our analysis began: the possibility of having multiple equivalent CS1 courses from the same institution in the assembled documents, and whether we would use or not the public universities' names in our results.

As we believe that having multiple equivalent CS1 courses would not add any impact in the results, we decided to maintain only one version of the equivalent courses, discarding the rest. This process only happened when the retrieved documents clearly stated that those courses were equivalent to each other. This also means that there was one syllabus per CS1 course. In the remainder of this article, we use these terms (*CS1 course* and *syllabi*) as synonyms when representing results. As for the other issue, we chose to present only the public university names in Appendix 1. The reason for this was because we did not have any intention of highlighting nor comparing these institutions in respect to how they structure their CS1 courses.

One final information we used in our general analysis was the specified year in which the data began to be *officially recognized* by the public university. We collected this information from each analyzed document e.g., the year in which the pedagogical project for the undergraduate program began. We decided to present this information to illustrate how old the collected data was. The emphasis in officially recognized means that if the university did have a more recent pedagogical project but it was still in the process of approval, we used the official one at that time. If we used more than one document to collect data about a CS1 course, we considered the year of the most recent one.

RQ1: *What are the most common topics covered in CS1 courses from Brazilian public universities?*

A CS education researcher analyzed the syllabi to identify the topics. For each found syllabus, he listed each of the covered topics in a worksheet. The list expanded with new topics as more syllabi were analyzed. During this analysis, the researcher used his experience to identify and group together topics with different names but meant the same concept. At the same time, the covered topics present in distinct syllabi were signaled and counted when the analysis ended.

Once the researcher had finished assembling the list with the topics and their frequencies, we sorted them decreasingly by the frequency. Since the goal was to identify the most common topics, we decided to report a subset of the initial list because the first assembled listing had many items. This subset was composed by applying two minimum thresholds regarding the frequencies. Finally, we decided to compare our listing with those assembled from related work. This comparison checked whether the topics were present or not in the other listings.

RQ2: *What are the most common CS1 courses' names from Brazilian public universities?*

We created a worksheet containing all the retrieved CS1 courses' names to answer this question. Even though courses that were divided into theoretical and practical classes were merged, we decided to keep only one name in the analysis (generally being the course regarding the theoretical class). The reason for that was because in most cases in which this situation occurred, the names were, for example, "Algorithms" and "Laboratory of Algorithms". If a course had numbered and not numbered names, we chose to consider both e.g., "Algorithms" and "Algorithms 1". This decision was made because this occurrence means that some universities have subsequent courses with the same name.

RQ3: *When do Brazilian public universities' curricula suggest students take the CS1 course?*

To answer this question, we analyzed how the public universities divide their curricula, as some institutions organize them by semester (semiannual), and others, by years (annual). Then we grouped together when their CS1 courses were suggested for the students to take e.g., first semester or first year. There were some cases in which even though the institution divided their curricula in years, the CS1 course only happens in one of its semesters. When this happened, the course was considered as semiannual.

RQ4: *What is the average of the total class hours of the CS1 courses from Brazilian public universities?*

Since not all CS1 courses explicitly stated how their class hours are divided (e.g., theory and practice), we chose to consider the absolute total. In other words, even if a specific CS1 course stated how its class hours are divided, the hours were summed. The same rule applied in cases when one CS1 course was divided into theoretical and practical classes. Before computing the average, we grouped courses with similar length. The reason for that was because it would not be reasonable to aggregate the total class hours for semiannual and annual courses. CS1 courses that happened in one semester of a scholarly year were considered as semiannual for this analysis. Finally, it is important to say that this data was not available for all found CS1 courses.

RQ5: *What are the programming paradigms and programming languages taught in CS1 courses from Brazilian public universities?*

The syllabi from the CS1 courses were used to answer this question. However, the retrieved syllabi did not always explicitly state the paradigm or the language. When this happened, the researcher responsible for the analysis decided to infer it from both the syllabus description and the suggested bibliography. While the paradigm could also be inferred by the programming language (e.g., Haskell would imply the functional paradigm), most object-oriented languages can be used to teach the procedural paradigm or the object-oriented paradigm (know together as the imperative paradigm) (Luxton-Reilly et al., 2018). Based on this, the researcher classified the paradigm based on his experience regarding the topics covered in the CS1 course. It was reported, inferred or not, a programming paradigm for every syllabus analyzed.

As for the programming language, the researcher consulted the suggested books in the bibliography (both basic and recommended) in the order they were listed. Then, he used the first one in which it was possible to infer a programming language. It is important to say that not all retrieved syllabi included a bibliography, and some books did not specify or used more than one programming language. In these cases, the language was classified as not possible to infer. We report the percentage of each inferred paradigm and language in our results in Section 4.

4 Results

In this section, we describe the results obtained with the execution of the data collection and analysis detailed in Section 3. We begin by presenting the general information about the public universities and the retrieved documents. Then we report the results for each research question, focusing on details about the application of the proposed methods. We present the discussion about the results and the answers for each research question in Section 5.

4.1 General Information

Table 2 presents the total amount of public universities and CS1 syllabi analyzed in this research. The associated type of public institution (federal, state, and municipal) is also described. For the public universities, we report the total that was present in the base listings (Wikipedia, 2021, 2022a, 2022b), the total universities that offered a targeted CS related undergraduate program for this research (detailed in Section 3), and the total universities that we used to compose our main results. For the CS1 syllabi, we report the total syllabi that we retrieved from the analyzed documents, and the distinct syllabi that we used to compose our main results. In total, we used 95 public universities and 225 CS1 syllabi in our main results.

Table 2: General distribution of the public universities and CS1 syllabi analyzed.

	Universities			**CS1 Syllabi**	
Type of institution	Present in Base Listings	Had Targeted CS Programs	Used	Retrieved	Used
Federal	69	63	61	195	150
State	47	35	32	88	72
Municipal	41	15	2	8	3
Total	157	113	**95**	291	**225**

Table 3 presents the geographic distribution of public universities and CS1 syllabi. We decided to report this distribution only for the universities and syllabi that composed our main results. In other words, Table 3 expands the numbers reported on both *Used* columns from Table 2. Our results managed to provide a broad analysis in terms of covering all Brazilian geographic regions in this research. This was only possible because of the federal universities since, after the discarding process mentioned in Section 3, state and municipal institutions were not present in all regions.

Table 3: Brazilian geographical distribution of the public universities and CS1 syllabi that composed the main results of this research.

	Universities			**CS1 Syllabi**		
Region	Federal	State	Municipal	Federal	State	Municipal
Center-west	8	3	0	17	10	0
North	9	0	0	22	0	0
Northeast	17	12	0	48	20	0
South	9	8	1	28	17	1
Southeast	18	9	1	35	25	2
Total	61	32	2	150	72	3

The distribution of the years of the documents in which we retrieved the syllabi information from is represented in Figure 1. As explained in Section 3, we only used the most recent documents that were officially recognized by the university. We assessed this factor even when we had to use other sources (such as class slides retrieved from instructors' websites) by checking if their publication year was within the official pedagogical project from the undergraduate program of the institution at that time. We present the distribution for each type of public university (federal, state, and municipal). The asterisk in 2022 indicates that the process of data collection and analysis happened between 2021 and 2022, meaning that the retrieved documents in 2022 could not be valid for the whole year. The numbers do not add up to 225 because we could not identify a year for 8 syllabi. While there were documents dating from as far as 2006, more than half (116) were from 2018 to 2022.

4.2 RQ1: CS1 Topics

As described in Section 3, we identified different CS1 topics by manually reading all the 225 syllabi from CS1 courses that were not considered equivalent to each other. We organized the initial listing by counting how many syllabi each topic covered, then we sorted the list decreasingly. In total, 72 different topics were identified. Although we decided to omit this initial list from our reports, we illustrate the frequency distribution of the 72 identified CS1 topics in Figure 2. The analysis of this distribution indicated that 51 different topics were present in less than 25 common syllabi. In other words, this means that approximately 71% of the identified topics were common to less than 11% of the total syllabi used in this research. On the other hand, there was no topic common to all 225 syllabi. We discuss the possible causes of both factors in Section 5.

Table 4 presents the ranking of the most covered topics. Since we were interested in the most common (RQ1), we decided to consider a subset of the initial listing with 72 topics. To do that, two thresholds were applied based upon the total of 225 syllabi used: the first was 10%, and the second was 33%. In total, 21 topics remained after applying the first threshold, and after the second, 12 topics remained. For each topic we report: the descriptive name, in which we tried to describe the different ways each syllabus referred to a same topic; the total number of the syllabi that each topic had in common; and, as a complement of the latter, the fraction of its frequency in terms of the 225 syllabi used. Table 4 is already represented with the initial threshold i.e., all the

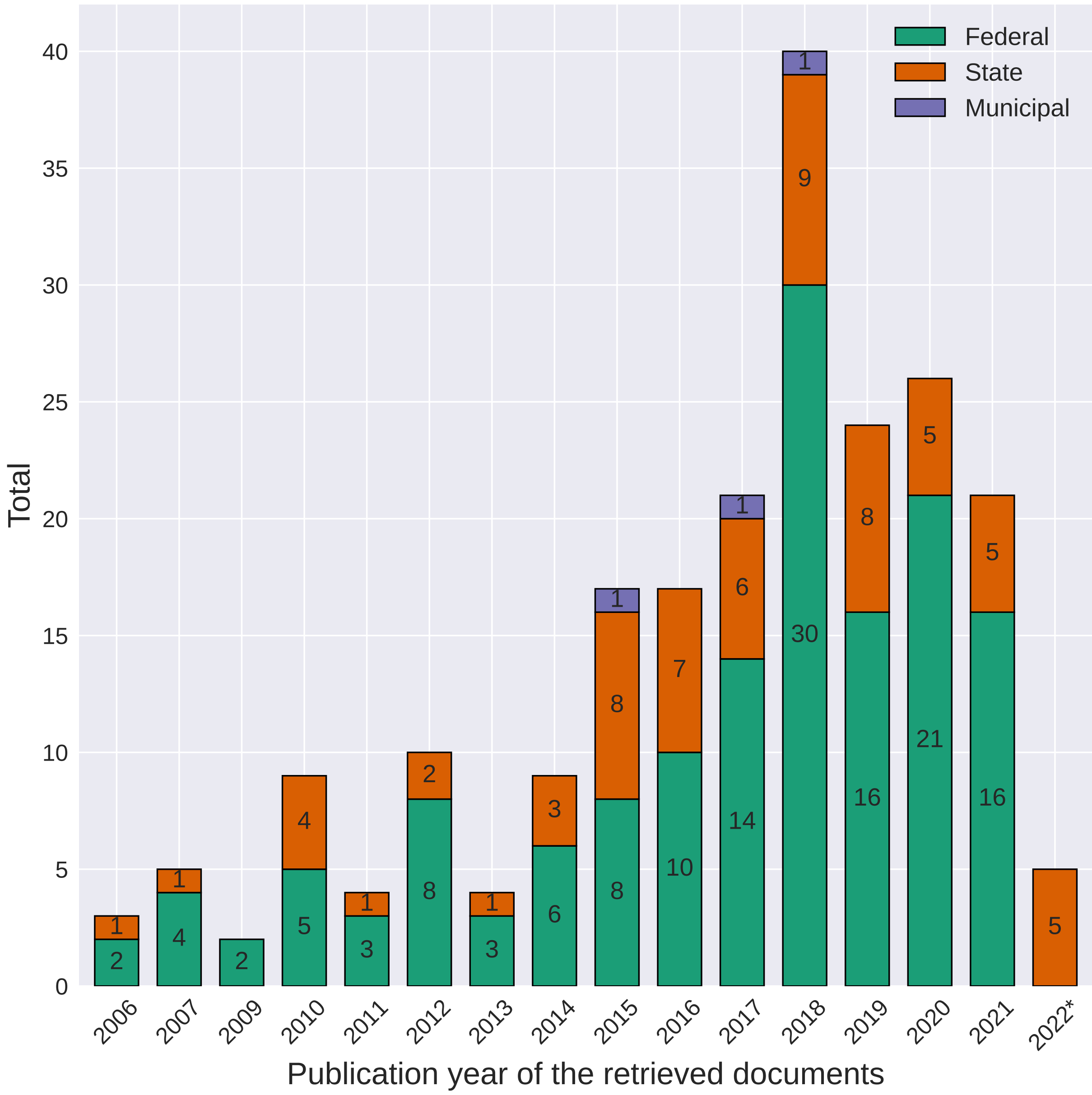

Figure 1: Distribution of the publication year of the retrieved documents that contained syllabi information. N = 217. It was not possible to retrieve the year in 8 syllabi. The asterisk in 2022 indicates that the corresponding retrieved syllabi might not be valid for the whole year since the data collection happened between 2021 and 2022.

21 topics appeared in at least 33% of total syllabi. A horizontal line defines the second threshold: the first 12 topics appeared in at least 10% of total syllabi.

The comparison of our listing with related work is also present in Table 4. We used the listings retrieved from Becker and Fitzpatrick (2019) and Porfirio et al. (2021) because they also analyzed CS1 syllabi. We also included the listings from Hertz and Ford (2013), and Schulte and Bennedsen (2006) to the comparison. It is important to notice that we only considered the 15 most covered topics from Becker and Fitzpatrick (2019) since the authors explicitly highlighted them. As explained in Section 3, we compared if a topic in our listing was present or not in the

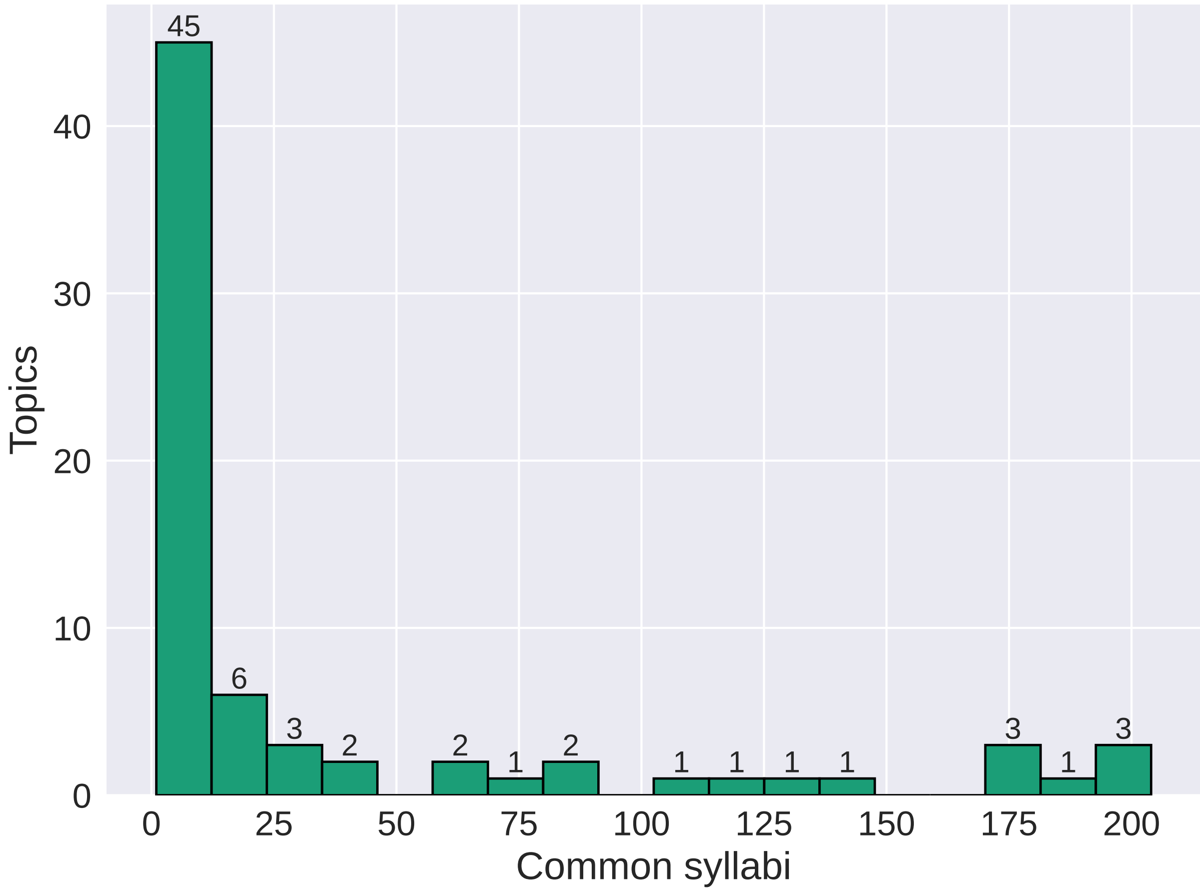

Figure 2: Frequency distribution of the presence of the initial 72 identified CS1 topics.

others, marking the presence with the ✓symbol. An example reading of Table 4 is that the topic *Conditional commands* appeared in 204 syllabi, and was common to 91% of all 225 syllabi. The same topic is also present in Becker and Fitzpatrick (2019), Porfirio et al. (2021), Hertz and Ford (2013), and Schulte and Bennedsen (2006).

4.3 RQ2: CS1 Courses' Names

We collected the CS1 course name directly from the used syllabi to calculate the results. The methodology was analogous to the one used in finding the most covered topics: we counted the frequency of each course name and ranked their total decreasingly. Table 5 presents the listing with the most common names. The total does not add up to 225 because we decided to omit names that appeared less than 5 times. It is important to notice that the translation of the course names was conducted by ourselves. This means that the names might not be exactly how each public university would translate them officially.

Table 4: Ranking of the most covered CS1 topics from Brazilian public universities. The horizontal line highlights the second threshold applied. Becker, Porfirio, Hertz, Schulte represent listings retrieved from (Becker & Fitzpatrick, 2019; Porfirio et al., 2021; Hertz & Ford, 2013; Schulte & Bennedsen, 2006), respectively.

Topic	Total	%	Becker	Porfirio	Hertz	Schulte
Conditional commands	204	91	✓	✓	✓	✓
Variables, constants, and assignments	201	89	✓	✓	✓	✓
Repetition commands	200	89	✓	✓	✓	✓
One-dimensional homogeneous composite variables	184	82	✓	✓	✓	✓
Arithmetical, logical, and relational expressions	181	80	✓	✓	✓	
Functions, modularization, subprograms	180	80	✓	✓	✓	✓
Multidimensional homogeneous composite variables	173	77	✓	✓		✓
Data input/output	141	63	✓	✓	✓	
Algorithm representation forms	127	56	✓			✓
Heterogeneous composite variables	118	52	✓			✓
Recursion	85	38	✓		✓	✓
Scope of variables and parameter usage	82	36			✓	✓
File handling	69	31	✓		✓	
Basic computer organization	64	28				
Pointers and dynamic memory allocation	61	27				✓
Debugging	40	18	✓		✓	✓
Documentation	37	16	✓			
Testing	33	15	✓		✓	
Sorting algorithms	30	13			✓	
Search algorithms	30	13				
Programming environments	22	10				✓

Table 5: Most common CS1 courses' names from Brazilian public universities.

Name (original)	Name (our translation)	Total
Programação 1	Programming 1	23
Introdução à Programação	Introduction to Programming	20
Algoritmos e Programação 1	Algorithms and Programming 1	16
Algoritmos e Estruturas de Dados 1	Algorithms and Data Structures 1	16
Algoritmos	Algorithms	15
Algoritmos e Programação	Algorithms and Programming	14
Algoritmos 1	Algorithms 1	11
Fundamentos de Programação	Programming Fundamentals	10
Introdução à Computação	Introduction to Computing	6
Técnicas de Programação 1	Programming Techniques 1	5
Programação de Computadores 1	Computer Programming 1	5
Algoritmos e Programação de Computadores	Algorithms and Computer Programming	5

4.4 RQ3: When Students Take the CS1 Course

We mostly used the pedagogical projects from each undergraduate program to identify when the public universities suggests that the students take the CS1 course. As explained in Section 3, each course was separated in terms of their duration and how each public university divided their curricula. Table 6 presents the results obtained in this analysis, sorted by the total number of CS1 courses for each suggested period. The *quarter* suggested period is composed of a four-month cycle, meaning that the institution divided its scholarly year in three quarters.

Table 6: Periods in which Brazilian public universities suggests that students should take the CS1 course.

Suggested Period	CS1 Courses
1st Semester	194
2nd Semester	17
1st Year	10
2nd Year	2
1st Quarter	1
4th Semester	1
Total	225

4.5 RQ4: Class Hours Duration

Since most of the CS1 courses happened in an annual or semiannual format, we decided to group them by this category independently of when the CS1 course is suggested in the curriculum. This means that we considered courses in the 1st or 2nd Year as *annual*, and courses in the 1st, 2nd, and 4th Semester as *semiannual* (these items are from Table 6).

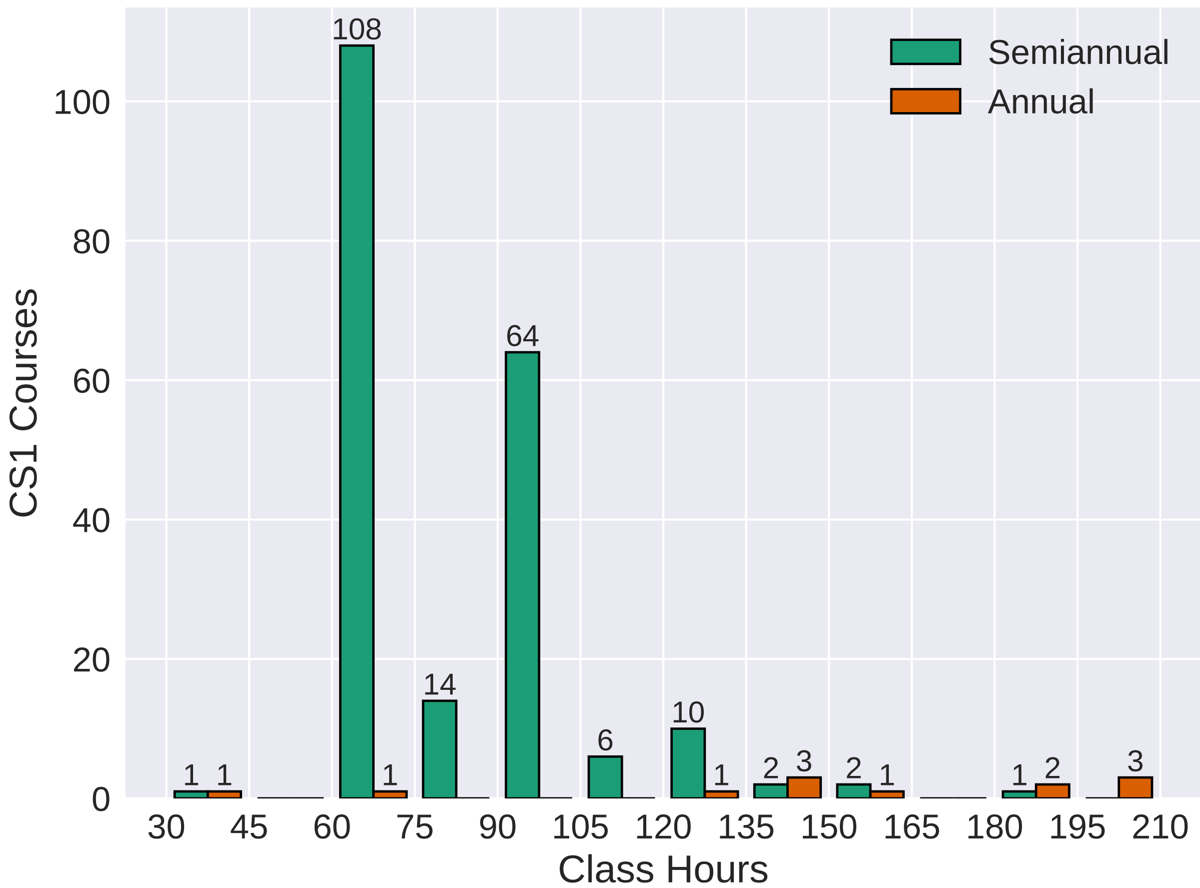

Figure 3: Total class hours distribution. N = 220. It was not possible to retrieve the class hours in 5 syllabi.

Figure 3 illustrates the distribution of total class hours (combining theory and practice) for both semiannual and annual CS1 courses. An example analysis of the figure reveals that 108 semiannual CS1 courses had a total class duration between 60 and 75 hours, while only 1 annual course fell within the same interval. On average, semiannual courses had a total class duration of 79 hours, with a standard deviation of 21. In contrast, annual courses had an average total class duration of 146 hours, with a standard deviation of 54. It should be noted that class hours for five courses, including the one classified in the 1st Quarter (Table 6), could not be found.

4.6 RQ5: Programming Paradigms and Languages

As mentioned in Section 3, the syllabi were used to identify the programming paradigms and languages taught in the CS1 courses. However, this information was not often explicitly stated. Because of that, we decided to infer it from the covered topics and the bibliography listed in the syllabi. Table 7 presents the programming paradigms taught in CS1 courses, sorted decreasingly by the total number of courses. Table 8 lists the programming languages, also sorted decreasingly by the total number of courses. For both tables, we also inform the approximate percentage of courses in which we had to infer the information. This means that, for example, in Table 7, from the total of 202 courses identified to teach the procedural paradigm, 92 (46%) were inferred from the syllabi (using the covered topics). Another example, in Table 8, from the total of 120 courses that taught the C programming language, 86 (72%) were inferred from the syllabi (using the first listed item in the bibliography in which it was possible to infer a programming language). On a final note, the CS1 courses from Table 7 do not add up to 225 because we omitted one CS1 course that taught computational thinking concepts. The CS1 courses from Table 8 also do not add up to 225 because we could not infer the programming language of 57 syllabi.

Table 7: Programming paradigms taught in CS1 courses from Brazilian public universities.

Paradigm	CS1 Courses	Total (%)*	Inferred (%)**
Procedural	202	90	46
Object-oriented	16	7	0
Functional	6	3	33

* % of all 225 CS1 courses.
** % of the corresponding number of CS1 Courses.

Table 8: Programming languages taught in CS1 courses from Brazilian public universities.

Language	CS1 Courses	Total (%)*	Inferred (%)**
C	120	53	72
Python	19	8	68
Java	10	4	80
Pascal	7	3	71
C++	6	3	83
Haskell	5	2	60
Scratch	1	<1	100

* % of all 225 CS1 courses.
** % of the corresponding number of CS1 Courses.

5 Discussion

We present the discussion of the obtained results in this section. We begin by discussing the assets used in this work (public universities and the syllabi), then we follow by the contextualization of the CS1 courses from public Brazilian universities (RQ2-RQ5), and finally we address implications regarding the most covered topics in these courses (RQ1).

5.1 Brazilian Public Universities

The distribution of public universities presented on Table 2 indicates that there are more federal institutions than state and municipal, although these last two have similar numbers. However, it becomes clear that not all institutions offer the CS undergraduate programs that we were interested in. Specifically in terms of the municipal institutions, approximately only one third of the total offers these CS undergraduate programs. This could be explained by the fact that there are municipals dedicated to specific areas such as Medicine, Humanities or Law: these institutions did not offer any CS undergraduate program. The same fact could also be applied to state and federal universities, albeit in a lower rate of occurrence. There is also the fact of the presence of programs that offers a Technology degree: they were sometimes present in institutions, but we did not analyze them. We also discarded CS1 courses that we were not able to retrieve the syllabi, thus, if this occurred for all CS1 courses from an institution, the whole institution was discarded. While the number of federal and state universities did not vary much between the ones that had a CS undergraduate program of interest and the ones used, these totals varied for municipal institutions. In fact, we had difficulties in finding the syllabi for the CS1 courses present in the municipal universities.

As mentioned in Section 4, the geographic distribution of the public universities used in this research (Table 3) covered all Brazilian geographic regions albeit it was only possible because of the federal institutions. Northeast, southeast, and south are the regions with most universities and syllabus for CS1 courses used in this research. We believe that the reason for this was the presence of different CS1 courses in different campi from the same institution that were not equivalent to each other. These regions are the most populated in Brazil. Porfirio et al. (2021) mentioned to have retrieved ten syllabi from ten federal public universities: two for each Brazilian geographic region. Since we managed to analyze 225 syllabi, our results can be seen as an update to theirs.

The year distribution of the retrieved documents (Figure 1) concentrates more than half in more recent years, with 2018 being most frequent year. This result could indicate that these documents might be close to what is being currently taught in the CS1 classes from Brazilian public universities. However, as explained before, we did not consider pedagogical projects still under approval at that time. This means that an updated version of the CS1 could be in effect by the time of the publication of this research if the new pedagogical project was approved in the meantime. The documents retrieved in the years 2020 and 2021 are also important to consider. As consequence of the *Sars-Cov-2* pandemic, many Brazilian institutions created Emergency School Periods to be able to teach. The adaptation necessary for the implementation of these periods could have impacts on the concepts taught in CS1 classes, especially for those universities in which the in person learning format was predominant. While Becker and Fitzpatrick (2019) and Porfirio et al. (2021) do not directly inform about the years of their retrieved syllabi, all authors used universities from rankings created between 2016 and 2018. Since our results go from 2006 to 2022, they can be seen as a complement of the research of these authors.

5.2 CS1 Syllabi

The syllabi used in this search were not homogeneous, containing distinct levels of detail. We found syllabi that were defined in few lines and others that detailed the topics taught per week. We also identified that syllabi used different ways to express the same topic e.g., loops were sometimes called “repetition commands”, “iterative commands” or even represented as “control structures” (combined with conditional commands). As explained in Sections 3 and 4, this was the reason a CS education researcher had to use his experience to identify equivalent topics represented with different names. We illustrate the heterogeneity of the syllabi by presenting examples from different public universities below, classified as S1 and S2. It is important to remember that these syllabi were originally elaborated in Brazilian Portuguese and were translated by the authors.

S1: Algorithms, fundamental programming concepts, expressions, control flow, functions and procedures, pointers, vectors and matrices, strings, dynamic allocation, structured types, files.

S2: Basics of programming logic: algorithms characteristics, algorithm representations, programs, instruction, sequences, successive refinement. Concepts of a procedural language: the C programming language, compiler, basic data types, constants and variables, comments, reserved words, logic and arithmetic expressions, assignment commands, data input and output. Control structures: conditional structures, iterative structures. Structured data types: vectors, matrices, dynamic memory allocation, pointers, user defined types. Modularization: functions, scope, parameters, recursion, file handling.

Porfirio et al. (2021) mentioned that syllabi may use different ways to express the same topic. In S1, we considered that *control flow* is related to both conditional and iterative commands, topic explicitly stated as *control structures* in S2. Becker and Fitzpatrick (2019) also reported this factor in a similar way since the authors presented a concept frequency in terms of being explicit or not explicit in the analyzed syllabi. In our analysis, we found topics with vague meanings e.g., *fundamental programming concepts* (present in S2) could have different meanings depending on the programming paradigm.

We did not find any CS1 syllabi that was described with the detail levels that Grunert (1997) specified. At best, we encountered the total class hours dedicated to each covered topic. In fact, most descriptions of the syllabi could be classified as course outlines. However, as McKeachie (1978) stated that the instructor needs to take careful considerations regarding the covered topics, assignments’ due, and learning objectives while constructing both syllabus and course outline, we did not discard any documents regarding its level of details. The data collection process used in this research was challenging, since we had to search the universities websites. If there was a national repository of syllabi, future research like our work would be simpler because researchers would be able to retrieve data by applying searching techniques like those used in systematic reviews, thus dedicating their focus on the assessment of topics. There are initiatives of repositories with these characteristics (Hislop et al., 2009; Tungare et al., 2007).

We do not believe that the absence of a topic in a syllabus implies that the related CS1 course does not cover it. S1, for example, does not explicitly state variables and constants, but they state vectors, matrices, and strings. This means that the presence of advanced topics that require basic ones implies that the latter topics are covered in the course but were omitted in the syllabus. This could explain why no topic appeared in all 225 syllabi (Figure 2 and Table 4). On the other hand, the presence of many different topics in few syllabi (Figure 2) could be explained by topics covered by other programming paradigms (such as the functional, which only appeared in 6 out of the 225 syllabi (Table 7)). The autonomy Brazilian public universities have could also have influenced the topic distribution, because institutions could be teaching specific topics to prepare professionals with a particular set of skills required for that geographic region, or to balance students with different background.

5.3 Contextualization of Brazilian CS1 Courses

Based on the results obtained from RQ3 (Table 6), almost 99% of the analyzed CS1 courses are offered within the first scholarly year of the undergraduate programs. This was expected since most Brazilian universities divide their curricula in semiannual periods. This result also means that students are exposed to CS1 concepts early in their academic life. The only exception for this is one course that is offered in the fourth semester. We identified that this CS1 course is from a Bachelor in Computer Science and Engineering. While we could not find any reasons for this occurrence, the CS1 course is present at the end of the basic cycle. The basic cycle is a biannual period that aggregates common courses between Brazilian Bachelor in Engineering undergraduate programs. RQ4 expresses that annual CS1 courses have an average total class hours doubled than semiannual ones. This factor indicates consistency among these averages even though annual CS1 courses are not common when compared to semiannual courses identified in this research.

As explained in Section 3, we considered numbered entries as individual names to answer RQ2 (Table 5). The presence of these numbered entries in our results could indicate that there are other programming courses in sequence. In fact, we identified courses like that while searching for the introductory programming courses: in these cases, the CS1 course was a prerequisite to the subsequent one. This indicates that some Brazilian public universities organize their curricula by providing CS1 and CS2 courses although we only focused on the first one. We exemplify this factor by the presence of 16 CS1 courses named *Algorithms and Data Structures 1*. While data structures can be seen as a CS2 topic, some universities might divide its topics between CS1 and CS2, thus corroborating that there is no consensus among these courses (Hertz, 2010). Examples of course names that were omitted in Table 5 (because they were present in less than five courses) were: variations from the listed courses (e.g., Programming and Data Structures 1, Programming Principles); names containing the taught programming paradigm (e.g., Functional Programming, Introduction to Structured Programming, Object-oriented Programming Language); and other specific names (e.g., Applied Informatics, Information Processing).

Table 7 shows the programming paradigm of all used syllabi. There is a major gap between procedural (202 syllabi) and the object-oriented paradigm (16 syllabi), and this gap is increased when compared to the functional paradigm (6 syllabi). Using our criteria to identify CS1 courses (Section 3), we identified one course that taught computational thinking (Wing, 2006) concepts. Since it might not be fit to classify it as a programming paradigm, we decided to omit it from the table, thus the total does not add up to 225. Our inferring methodology was the same for all syllabi, used only when there was no explicit word or topic containing the paradigm. It is important to notice that we did not need to infer any course that teaches the object-oriented paradigm because it was explicitly stated in all 16 corresponding syllabi.

The other result that complements the answer to RQ5 is the programming language distribution (Table 8). Although C, Python, and Java are the most common languages, there is a major gap between C and the rest. C is more than 6 times higher than Python (119 compared to 19, respectively). This result differs from others found in the literature, which has reports indicating that Java is the predominant language (Becker, 2019; Becker & Fitzpatrick, 2019; Siegfried, Herbert-Berger, Leune, & Siegfried, 2021). We identified some reasons for this. First, our results are limited to Brazilian public universities. We did not find any related work that explicitly said to cover Brazil, but there are research that focused on other specific countries. Avouris (2018) analyzed 121 CS1 courses from Greece and C was the most used language, appearing in 37 courses. Becker (2019) surveyed instructors from 39 CS1 courses and Java was the most used language, appearing in 49% of the courses. Second, the methodology used could also have impacted in the results. Research that surveyed instructors directly are more precise than those who used syllabi (like this research) that might be outdated. The third reason is a direct consequence of the second because we retrieved information from documents published from 2006 and 2022, even though most of them were from 2018 and further. Lastly, the publication year of the used documents could also have impacted in our inferring methodology, since the bibliography used to infer the programming language could also be outdated.

To compare if there was any difference when limiting to recent years, we filtered the results from Table 8 to include only CS1 courses in which we obtained information from 2018 to 2022. The results obtained from this filtering is presented in Table 9. There were 124 syllabi from 2018 to 2022, but it was not possible to infer a language from 28 of them. In general, the results from the filtering do not differ much from our initial findings. The gap between C and Python was not altered (45% of the total courses before and after). Pascal and C++ also appeared less times (both were 3% of the total courses before, and 2% and 1% after, respectively).

We conclude that these results indicate that the Brazilian scenario has its own contexts for teaching CS1, differing from those in other countries or world regions. In general, all the obtained information about the contextualization of CS1 courses is a consequence of the aforementioned autonomy that Brazilian public universities have. The majority of CS1 courses teach the procedural paradigm, perhaps due to a general consensus that it is the best approach for beginner programmers. Regarding the programming language, there may be several factors that contribute to C being the most taught. One factor could be instructors' traditionalist view about the language, preferring it over others, arguably. However, there may also be factors related to the costs associated with training faculty to teach other programming languages, such as Python. We make this statement because the ability to program in a language and the ability to effectively teach it are two distinct skills.

5.4 Covered Topics

As detailed in Section 4, we omitted the full list with 72 topics because we wanted to identify the most covered topics in RQ1 (Table 4). Our decision to create two thresholds (common topics present in 10% and 33% of the total syllabi, respectively) was established to highlight the topics that were most common. Topics that remained outside the thresholds (i.e., appeared in less

Table 9: Comparison of the programming languages identified from documents ranging from 2006-2022 to those obtained from 2018-2022. It was possible to infer a programming language from 168 syllabi in the 2006-2022 period, and from 96 syllabi in the 2018-2022 period.

	2006-2022			2018-2022		
Language	**CS1 Courses**	**Total (%)***	**Inferred (%)*****	**CS1 Courses**	**Total (%)****	**Inferred (%)**
C	120	53	72	70	56	67
Python	19	8	68	14	11	64
Java	10	4	80	8	6	75
Pascal	7	3	71	2	2	0
C++	6	3	83	1	1	100
Haskell	5	2	60	1	1	0
Scratch	1	<1	100	0	0	-

* % of 225 CS1 courses present in the 2006-2022 period.
** % of 124 CS1 courses present in the 2018-2022 period.
*** % of the corresponding number of CS1 Courses.

than 33% of total syllabi) include, but are not limited to: object-oriented and functional paradigm concepts (e.g., classes, objects, monads, function overloading); basic computer usage (word editors, spreadsheets, operating system, Internet browser); and advanced algorithms (geometric, non-deterministic). The absence of some of these topics in the thresholds corroborate with the procedural paradigm being the one most taught. As stated earlier, universities have the autonomy to select topics based on the desired professional outcome.

The 12 topics listed in the second threshold (Table 4) represent concepts present in the procedural paradigm. A special note is that the topic *Algorithm representation forms* was identified in syllabi that did not start with a specific programming language, teaching the other concepts via pseudocode. We noted CS1 courses that only begins teaching a programming language at the end of the course, while others did not mention any language. We conclude that this also contributed to the number of courses in which we could not infer the programming language. We also decided to list a group of concepts in a specific topic because of the different ways syllabi referred to them. Some examples of topics were: *One-dimensional homogeneous composite variables*, which included topics as vectors, lists, arrays, and strings; *Multidimensional homogeneous composite variables*, which included matrices; and *Heterogeneous composite variables*, which contained structures, unions, and dictionaries.

We also noted discrepancies regarding the frequency of topics that are closely related to each other. For example, in Table 4, *Functions, modularizations, and subprograms* appear in 80% of all syllabi whereas *Scope of variables and parameter usage* and *Recursion* appear in 36% and 38%, respectively. There could be a couple reasons for this, and they depend on the programming paradigm that each course teaches. *Scope of variables and parameter usage* could just have been omitted in the syllabus since it already had listed *Functions, modularizations, and subprograms*. However, since there were syllabi reporting both, we decided to treat them as separate topics. *Recursion*, on the other hand, could be related to the elaboration of the CS1 course itself. Since recursion is often considered a difficult topic to teach and learn (Caceffo et al., 2016), some curricula might delay its teaching to subsequent programming courses. Another relatable example regards the gaps between composite variables: there is one gap between one and multidimensional homogeneous, and an even greater gap with the heterogeneous types.

5.4.1 Comparison with Topics Listed in Related Work

The 12 topics present in the second threshold (at least 33% of total syllabi) appears in at least 2 out of the 4 analyzed related work (Table 4). This is an important factor because it establishes a certain degree of similarity with listings obtained from related work even though they were conducted in a different context (except for Porfirio et al. (2021)).

We identified that some topics which were classified as distinct in Table 4 were sometimes grouped together in related work. For instance, in our work, we reported *Conditional commands* and *Repetition commands* as distinct topics. Becker and Fitzpatrick (2019) listed *Selective statements (if/else/etc.)* separately from *Repetition & loops*, Porfirio et al. (2021) listed *Conditional structures* separately from *Repetition structures*, Hertz and Ford (2013) grouped these concepts as *Control constructs*, and Schulte and Bennedsen (2006) grouped them as *Selection and Iteration*. Another example is *Variables, constants, and assignments* and *Arithmetical, logical, and relational expressions* were grouped as *Variables, types, expressions* (Hertz & Ford, 2013) and as *Variables, assignment, arithmetic operators, declarations, data types* (Becker & Fitzpatrick, 2019). Since all reported listings were built upon the authors' experience, including our work, we consider this difference in grouping as just a matter of opinion in reporting the topics.

Our listing contemplated 9 of the 10 topics listed in Porfirio et al. (2021). Table 4 shows only 8 because they listed *Data types* as a separate topic, and we considered it within *Variables, constants, and assignments*. Another topic presented in this related work was *Introduction to programming*. This was an abstract concept in which we did not consider as a CS1 topic per se, thus we did not include it in our listing. Since Porfirio et al. (2021) assessed a subset of 10 Brazilian federal universities, we consider our listing as an expansion of their initial work regarding the Brazilian scenario.

Debugging and *Testing* are topics that were not present in our second threshold but appeared in 75% and 50% of the related work, respectively. *Debugging* appeared in 40 out of 225 syllabi (18%). This concept is related to the teaching of specific tools used to debug code and some public universities might not teach them in the introductory programming course. Similarly, *Testing* appeared in 33 out of 225 syllabi (15%). Specific teachings of this topic could be covered in future courses that focuses more on software engineering concepts. The same analogy could also be applied to *Documentation*.

Lastly, we also searched for topics that were reported in the related work but not in our listing. Examples of topics in this classification were: abstract concepts (e.g., problem solving, writing programs, mental models); object-oriented concepts (e.g., classes and objects, inheritance and polymorphism, encapsulation); advanced data structures (e.g., graphs, trees, linked-lists); and algorithm efficiency (e.g., big-O notation).

The comparison of our listing with those obtained from related work makes the identification of common topics possible. However, except for Porfirio et al. (2021), all research were conducted regarding different contexts, especially nationwide. This is corroborated by the results obtained from RQ5 (Tables 7 and 8). Since the programming paradigm taught in CS1 courses from Brazilian public universities (procedural was the most common) is different from the one taught in other countries (object-oriented), topic listings would certainly be different. All of this contributes to Hertz's (2010) view that there is no consensus of what is taught in CS1 and CS2.

5.4.2 Grouping of the Second Threshold

As consequence of the discussion this section provided, the second threshold applied in Table 4 answers RQ1, our main research question. We decided to group the 12 highlighted topics by joining the closely related, using data from the listings we had found in related work. Table 10 presents the final grouping, indicating the group name and the grouped topics from Table 4. A description of each group is stated below.

Table 10: Final grouping of the most covered topics in Brazilian public universities. Table sorted alphabetically by the group name.

Group	**Topics** (Table 4)
Algorithm representations	Algorithm representation forms
Basic concepts of algorithm construction	Variables, constants, and assignments Arithmetical, logical, and relational expressions Data input/output
Composite variables	One-dimensional homogeneous composite variables Multidimensional homogeneous composite variables Heterogeneous composite variables
Control structures	Conditional commands Repetition commands
Functions, scope, and parameter usage	Functions, modularization, subprograms Scope of variables and parameter usage
Recursion	Recursion

- **Algorithm representations**: This group consists of topics related to other ways of constructing and representing algorithms, not necessarily using a programming language. Examples include pseudocode and flowcharts.
- **Basic concepts of algorithm construction**: This group consists of primordial topics necessary for the construction of simple algorithms. Examples include variables, constants, basic data types, and expressions. Assignment and input/output (from keyboard) commands are also present in this group.
- **Composite variables**: This group consists of composite types of data such as vectors, strings, matrices, and structures. Other types that depend on the programming language (e.g., lists, tuples, dictionaries) are also present in this group.
- **Control structures**: This group consists of commands for selection and iteration of code. These structures can vary based on the programming language or algorithm representation.
- **Function, scope, and parameter usage**: This group consists of concepts related to functions, subprograms, and modularization. Since scope of variables and parameter usage is closely related, we decided to also include them in this group.
- **Recursion**: This group consists of the specific topics regarding recursion. Again, we decided to create a separate group for this because the analyzed CS1 syllabi listed them separately. Recursion is also presented as a specific topic in related work.

6 Limitations and Threats to Validity

The main limitations regarding this work are consequences of the restrictions related to data collection explained in Section 3. Since our data collection and analysis had to be done manually, we had to limit our assessment. Choosing Brazilian public universities provides a broad context of assessment, since it covered all national geographical regions. Brazilian public universities are also the most present in international rankings. However, there are other types of higher education institutions in the country, especially private universities and colleges. It is possible that our results could be different if we assessed data from these other institutions. Another limitation factor was the availability of the documents from the universities' websites. Data from public universities that we were not able to retrieve could also have changed our results.

We identify three main threats to the validity of this research. First, we used our empirical knowledge to identify and list the distinct covered topics present in each syllabus. Second, the dominance of the procedural paradigm taught in CS1 courses influenced in our discussion about the presence of topics in the syllabi. Third, our inferring methodology about the programming languages taught was vastly dependent on each syllabus. Since the analyzed syllabi were not homogeneous (sometimes not even listing the bibliography) and included documents ranging from as far as 2006, this could also have impacted on our results. We tried to mitigate these threats by comparing our topics with others retrieved from related work, and by applying a filter to analyze the programming languages from syllabi from 2018 to 2022.

7 Conclusions

In this work, we presented an assessment of characteristics of CS1 courses from Brazilian public universities. Our focus was the most covered topics, but we also reported the most common course names, when undergraduates take the course, time that is dedicated to teaching the course, and the programming paradigms and languages taught. To answer each research question, we gathered data directly from the public universities' websites, searching for undergraduate Computer Science programs listed in the Formation Guidelines for Computer Science Undergraduate Programs (Zorzo et al., 2017). The main document analyzed was the syllabi of CS1 courses.

In total, our results derived from 225 syllabi within 95 Brazilian public universities. The most covered topics were from concepts related to the procedural programming paradigm. We identified 12 topics among the most frequent and grouped them in 6 categories: Algorithm representations; Basic concepts of algorithm construction; Composite variables; Control structures; Functions, scope, and parameter usage; and Recursion. We also concluded that the CS1 course is within the first scholarly year in 99% of the Brazilian public universities. The most common names of CS1 courses vary, and some of them have numbered entries that implies subsequent courses with the same name in the curriculum. Regarding the total class hours, we identified that it is dependent on the course being semiannual or annual, but in average, the latter is double the former. Lastly, we concluded that the most common taught programming paradigm is the procedural one, and the most common programming language is C.

Acknowledgments

We would like to thank the Brazilian National Council for Scientific and Technological Development (CNPq) for the financial support (process numbers 142476/2020-0 and 312088/2020-5), as well as the São Paulo Research Foundation (FAPESP) for grant number 2013/08293-7.

Extended Awarded Article

This publication is an extended version of an awarded paper at the II Brazilian Symposium on Computing Education (EduComp 2022), entitled "Análise dos Tópicos Mais Abordados em Disciplinas de Introdução à Programação em Universidades Federais Brasileiras" (Silva, Caceffo, & Azevedo, 2022). DOI: 10.5753/educomp.2022.19196.

References

Abad, C. L., Ortiz-Holguin, E., & Boza, E. F. (2021). Have We Reached Consensus? An Analysis of Distributed Systems Syllabi. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (p. 1082–1088). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3408877.3432409 [GS Search]

Araujo, A., Filho, D., Oliveira, E., Carvalho, L., Pereira, F., & Oliveira, D. (2021). Mapeamento e análise empírica de misconceptions comuns em avaliações de introdução à programação. In *Anais do Simpósio Brasileiro de Educação em Computação* (pp. 123–131). Porto Alegre, RS, Brasil: SBC. doi: 10.5753/educomp.2021.14478 [GS Search]

Araujo, L., Bittencourt, R., & Chavez, C. (2021). Python Enhanced Error Feedback: Uma IDE Online de Apoio ao Processo de Ensino-Aprendizagem em Programação. In *Anais do Simpósio Brasileiro de Educação em Computação* (pp. 326–333). Porto Alegre, RS, Brasil: SBC. doi: 10.5753/educomp.2021.14500 [GS Search]

Austing, R. H., Barnes, B. H., Bonnette, D. T., Engel, G. L., & Stokes, G. (1979, mar). Curriculum '78: Recommendations for the Undergraduate Program in Computer Science— a Report of the ACM Curriculum Committee on Computer Science. *Commun. ACM*, *22*(3), 147–166. doi: 10.1145/359080.359083 [GS Search]

Avouris, N. (2018). Introduction to Computing: A Survey of Courses in Greek Higher Education Institutions. In *Proceedings of the 22nd Pan-Hellenic Conference on Informatics* (p. 64–69). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3291533.3291549 [GS Search]

Becker, B. A. (2019). A Survey of Introductory Programming Courses in Ireland. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education* (p. 58–64). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3304221.3319752 [GS Search]

Becker, B. A., & Fitzpatrick, T. (2019). What Do CS1 Syllabi Reveal About Our Expectations of Introductory Programming Students? In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (p. 1011–1017). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3287324.3287485 [GS Search]

Becker, B. A., & Quille, K. (2019). 50 Years of CS1 at SIGCSE: A Review of the Evolution of Introductory Programming Education Research. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (p. 338–344). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3287324.3287432 [GS Search]

Berges, M., & Hubwieser, P. (2013). Concept Specification Maps: Displaying Content Structures. In *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education* (p. 291–296). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2462476.2462503 [GS Search]

Bosse, Y., & Gerosa, M. (2015). Reprovações e Trancamentos nas Disciplinas de Introdução à Programação da Universidade de São Paulo: Um Estudo Preliminar. In *Anais do XXIII Workshop sobre Educação em Computação* (pp. 426–435). Porto Alegre, RS, Brasil: SBC. doi: 10.5753/wei.2015.10259 [GS Search]

Brodley, C. E., Hescott, B. J., Biron, J., Ressing, A., Peiken, M., Maravetz, S., & Mislove, A. (2022). Broadening Participation in Computing via Ubiquitous Combined Majors (CS+X). In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1* (p. 544–550). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3478431.3499352 [GS Search]

Caceffo, R., Gama, G., & Azevedo, R. (2018). Exploring Active Learning Approaches to Computer Science Classes. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (p. 922–927). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3159450.3159585 [GS Search]

Caceffo, R., Wolfman, S., Booth, K. S., & Azevedo, R. (2016). Developing a Computer Science Concept Inventory for Introductory Programming. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (p. 364–369). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2839509.2844559 [GS Search]

Campos, R. (2010). Metodologia ERM2C: Para melhoria do processo de ensino-aprendizagem de lógica de programação. In *XVIII Workshop sobre Educação em Computação* (pp. 961–970). XXX Congresso da Sociedade Brasileira de Computação. Retrieved 08/29/2022, from https://docplayer.com.br/52972769-Metodologia-erm2c-para-melhoria-do-processo-de-ensino-aprendizagem-de-logica-de-programacao.html [GS Search]

CC2020 Task Force (2020). *Computing Curricula 2020: Paradigms for Global Computing Education.* New York, NY, USA: Association for Computing Machinery. doi: https://doi.org/10.1145/3467967 [ISBN: 9781450390590. GS Search]

Dodds, Z., Libeskind-Hadas, R., & Bush, E. (2010). When CS 1 is Biology 1: Crossdisciplinary Collaboration as CS Context. In *Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education* (p. 219–223). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/1822090.1822152 [GS Search]

Eberly, M. B., Newton, S. E., & Wiggins, R. A. (2001). THE SYLLABUS AS A TOOL FOR STUDENT-CENTERED LEARNING. *The Journal of General Education*, *50*(1), 56–74. Retrieved 2022-08-02, from http://www.jstor.org/stable/27797862 [GS Search]

FIA (2019). *Universidades públicas: O que são, importância e lista de instituições.* Retrieved 08/30/2022, from https://fia.com.br/blog/universidades-publicas/

Fréchet, N., Savoie, J., & Dufresne, Y. (2020). Analysis of Text-Analysis Syllabi: Building a Text-Analysis Syllabus Using Scaling. *PS: Political Science & Politics*, *53*(2), 338–343.

doi: 10.1017/S1049096519001732 [GS Search]

Grunert, J. (1997). *The Course Syllabus: A Learning-Centered Approach* (1st ed.). Bolton, MA, USA: Anker Publishing Co. Inc. [ISBN: 9781882982189. GS Search]

Guo, P. (2014). *Python Is Now the Most Popular Introductory Teaching Language at Top U.S. Universities.* Communications of the ACM Blog (BLOG@CACM). Retrieved 08/29/2022, from https://cacm.acm.org/blogs/blog-cacm/176450-python-is-now-the-most-popular-introductory-teaching-language-at-top-us-universities/fulltext

Hertz, M. (2010). What Do "CS1" and "CS2" Mean? Investigating Differences in the Early Courses. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (p. 199–203). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/1734263.1734335 [GS Search]

Hertz, M., & Ford, S. M. (2013). Investigating Factors of Student Learning in Introductory Courses. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education* (p. 195–200). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2445196.2445254 [GS Search]

Hislop, G. W., Cassel, L., Delcambre, L., Fox, E., Furuta, R., & Brusilovsky, P. (2009). Ensemble: Creating a National Digital Library for Computing Education. In *Proceedings of the 10th ACM Conference on SIG-Information Technology Education* (p. 200). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/1631728.1631783 [GS Search]

Kinnunen, P., & Malmi, L. (2006). Why Students Drop out CS1 Course? In *Proceedings of the Second International Workshop on Computing Education Research* (p. 97–108). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/1151588.1151604 [GS Search]

Lima, M. A. P., Carvalho, L. S. G., Oliveira, E. H. T. d., Oliveira, D. B. F. d., & Pereira, F. D. (2021, set.). Uso de atributos de código para classificar a dificuldade de questões de programação em juízes online. *Revista Brasileira de Informática na Educação*, *29*, 1137–1157. doi: 10.5753/rbie.2021.29.0.1137 [GS Search]

Luxton-Reilly, A. (2016). Learning to Program is Easy. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education* (p. 284–289). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2899415.2899432 [GS Search]

Luxton-Reilly, A., Simon, Albluwi, I., Becker, B. A., Giannakos, M., Kumar, A. N., . . . Szabo, C. (2018). Introductory Programming: A Systematic Literature Review. In (p. 55–106). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3293881.3295779 [GS Search]

McKeachie, W. J. (1978). *Teaching tips: A guidebook for the beginning college teacher* (7th ed.). Lexington, MA, USA.: D.C. Heath and Company. [ISBN: 978066901151. GS Search]

Nascimento, P. (2018). *Recomendação de ação pedagógica no ensino de introdução à programação por meio de raciocínio baseado em casos*. Unpublished master's thesis, Programa de Pós-graduação em Informática. Retrieved from https://tede.ufam.edu.br/handle/tede/6837 [GS Search]

Pereira, F., Fonseca, S., Oliveira, E., Oliveira, D., Cristea, A., & Carvalho, L. (2020). Deep learning for early performance prediction of introductory programming students: a comparative and explanatory study. *Revista Brasileira de Informática na Educação*, *28*(0), 723–748. doi: 10.5753/rbie.2020.28.0.723 [GS Search]

Pereira, R., Peres, L., & Silva, F. (2021). Hello World: 17 habilidades para exercitar desde o início da graduação em computação. In *Anais do Simpósio Brasileiro de Educação em Computação* (pp. 193–203). Porto Alegre, RS, Brasil: SBC. doi: 10.5753/educomp.2021.14485 [GS Search]

Petersen, A., Craig, M., Campbell, J., & Tafliovich, A. (2016). Revisiting Why Students Drop CS1. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research* (p. 71–80). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2999541.2999552 [GS Search]

Porfirio, A. J., Pereira, R., & Maschio, E. (2021, jul.). A-Learn EvId: A Method for Identifying Evidence of Computer Programming Skills Through Automatic Source Code Assessment. *Revista Brasileira de Informática na Educação*, *29*, 692–717. doi: 10.5753/rbie.2021.29.0.692 [GS Search]

Scaffidi, C., Shaw, M., & Myers, B. (2005). Estimating the Numbers of End Users and End User Programmers. In *Proceedings of the 2005 IEEE Symposium on Visual Languages and Human-Centric Computing* (p. 207–214). USA: IEEE Computer Society. doi: 10.1109/VLHCC.2005.34 [GS Search]

Schulte, C., & Bennedsen, J. (2006). What Do Teachers Teach in Introductory Programming? In *Proceedings of the Second International Workshop on Computing Education Research* (p. 17–28). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/1151588.1151593 [GS Search]

Siegfried, R. M., Herbert-Berger, K. G., Leune, K., & Siegfried, J. P. (2021). Trends Of Commonly Used Programming Languages in CS1 And CS2 Learning. In *2021 16th International Conference on Computer Science & Education (ICCSE)* (p. 407-412). doi: 10.1109/ICCSE51940.2021.9569444 [GS Search]

Silva, E., Caceffo, R., & Azevedo, R. (2022). Análise dos Tópicos Mais Abordados em Disciplinas de Introdução à Programação em Universidades Federais Brasileiras. In *Anais do II Simpósio Brasileiro de Educação em Computação* (pp. 29–39). Porto Alegre, RS, Brasil: SBC. doi: 10.5753/educomp.2022.19196 [GS Search]

Sloan, R. H., Barr, V., Bort, H., Guzdial, M., Libeskind-Hadas, R., & Warner, R. (2020). CS + X Meets CS 1: Strongly Themed Intro Courses. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (p. 960–961). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3328778.3366975 [GS Search]

Sloan, R. H., Taylor, C., & Warner, R. (2017). Initial Experiences with a CS + Law Introduction to Computer Science (CS 1). In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education* (p. 40–45). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3059009.3059029 [GS Search]

Sobral, S. (2019). 30 YEARS OF CS1: PROGRAMMING LANGUAGES EVOLUTION. In *ICERI2019 Proceedings* (p. 9197-9205). IATED. doi: 10.21125/iceri.2019.2214 [GS Search]

Tew, A. E., & Guzdial, M. (2010). Developing a Validated Assessment of Fundamental CS1 Concepts. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (p. 97–101). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/1734263.1734297 [GS Search]

Tew, A. E., & Guzdial, M. (2011). The FCS1: A Language Independent Assessment of CS1 Knowledge. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science*

Education (p. 111–116). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/1953163.1953200 [GS Search]

Tungare, M., Yu, X., Cameron, W., Teng, G., Pérez-Quiñones, M. A., Cassel, L., … Fox, E. A. (2007, mar). Towards a Syllabus Repository for Computer Science Courses. *SIGCSE Bull.*, *39*(1), 55–59. doi: 10.1145/1227504.1227331 [GS Search]

Vahldick, A., Marcelino, M. J., & Mendes, A. J. (2021, dez.). Analyzing novices' fun and programming behaviours while playing a serious blocks-based game. *Revista Brasileira de Informática na Educação*, *29*, 1337–1355. doi: 10.5753/rbie.2021.2069 [GS Search]

Valentine, D. W. (2004). CS Educational Research: A Meta-Analysis of SIGCSE Technical Symposium Proceedings. In *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education* (p. 255–259). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/971300.971391 [GS Search]

Walker, H. M. (2017, oct). ACM RETENTION COMMITTEE Retention of Students in Introductory Computing Courses: Curricular Issues and Approaches. *ACM Inroads*, *8*(4), 14–16. doi: 10.1145/3151936 [GS Search]

Wikipedia (2021). *Lista de universidades federais do brasil — wikipédia, a enciclopédia livre.* Retrieved 07/06/2022, from https://pt.wikipedia.org/w/index.php?title=Lista_de_universidades_federais_do_Brasil&oldid=61551015

Wikipedia (2022a). *Lista de universidades estaduais do brasil — wikipédia, a enciclopédia livre.* Retrieved 03/02/2022, from https://pt.wikipedia.org/w/index.php?title=Lista_de_universidades_estaduais_do_Brasil&oldid=63117788

Wikipedia (2022b). *Lista de universidades municipais do brasil — wikipédia, a enciclopédia livre.* Retrieved 04/17/2022, from https://pt.wikipedia.org/w/index.php?title=Lista_de_universidades_municipais_do_Brasil&oldid=63408268

Wing, J. M. (2006, mar). Computational Thinking. *Commun. ACM*, *49*(3), 33–35. doi: 10.1145/1118178.1118215 [GS Search]

Zorzo, A. F., Nunes, D., Matos, E., Steinmacher, I., de Araujo, R. M., Correia, R., & Martins, S. (2017). *Referenciais de Formação para os Cursos de Graduação em Computação*. Sociedade Brasileira de Computação (SBC). [ISBN: 978-85-7669-424-3. GS Search]

Appendix 1

Table 11 presents all the Brazilian public universities used in this work, composed by federal (ID 1 to 61), state (ID 62 to 93) and municipal (ID 94 and 95) institutions. We decided to list only their original names in Brazilian Portuguese (Wikipedia, 2021, 2022a, 2022b) because it is the best way to identify them via on-line search.

Table 11: Brazilian public universities used in this work.

ID	Name (Brazilian Portuguese)	ID	Name (Brazilian Portuguese)
1	Universidade de Brasília	49	Universidade Federal do Rio de Janeiro
2	Universidade Federal da Grande Dourados	50	Universidade Federal dos Vales do Jequitinhonha e Mucuri
3	Universidade Federal de Goiás	51	Universidade Federal Fluminense
4	Universidade Federal de Mato Grosso	52	Universidade Federal Rural do Rio de Janeiro
5	Universidade Federal de Mato Grosso do Sul	53	Universidade Tecnológica Federal do Paraná
6	Universidade Federal de Catalão	54	Universidade Federal da Fronteira Sul
7	Universidade Federal de Jataí	55	Universidade Federal de Pelotas
8	Universidade Federal de Rondonópolis	56	Universidade Federal de Santa Catarina
9	Universidade Federal da Bahia	57	Universidade Federal de Santa Maria
10	Universidade Federal do Sul da Bahia	58	Universidade Federal do Pampa
11	Universidade Federal do Recôncavo da Bahia	59	Universidade Federal do Paraná
12	Universidade Federal da Lusofonia Afro-Brasileira	60	Universidade Federal do Rio Grande
13	Universidade Federal da Paraíba	61	Universidade Federal do Rio Grande do Sul
14	Universidade Federal do Cariri	62	Universidade Estadual de Goiás
15	Universidade Federal de Alagoas	63	Universidade do Estado do Mato Grosso
16	Universidade Federal de Campina Grande	64	Universidade Estadual do Mato Grosso do Sul
17	Universidade Federal de Pernambuco	65	Universidade do Estado da Bahia
18	Universidade Federal de Sergipe	66	Universidade Estadual de Feira de Santana
19	Universidade Federal do Ceará	67	Universidade Estadual de Santa Cruz
20	Universidade Federal do Maranhão	68	Universidade Estadual do Sudoeste da Bahia
21	Universidade Federal do Piauí	69	Universidade Estadual do Ceará
22	Universidade Federal do Rio Grande do Norte	70	Universidade Estadual da Paraíba
23	Universidade Federal do Vale do São Francisco	71	Universidade de Pernambuco
24	Universidade Federal Rural de Pernambuco	72	Universidade do Estado do Rio Grande do Norte
25	Universidade Federal Rural do Semi-Árido	73	Universidade do Estado do Amazonas
26	Universidade Federal de Rondônia	74	Universidade do Estado do Pará
27	Universidade Federal do Acre	75	Universidade Estadual de Roraima
28	Universidade Federal do Amapá	76	Universidade do Tocantins
29	Universidade Federal do Amazonas	77	Universidade do Estado de Minas Gerais
30	Universidade Federal do Oeste do Pará	78	Universidade Estadual de Montes Claros
31	Universidade Federal do Pará	79	Universidade do Estado do Rio de Janeiro
32	Universidade Federal do Tocantins	80	Universidade Estadual da Zona Oeste
33	Universidade Federal Rural da Amazônia	81	Universidade Estadual do Norte Fluminense Darcy Ribeiro
34	Universidade Federal do Sul e Sudeste do Pará	82	Universidade de São Paulo
35	Universidade Federal de Alfenas	83	Universidade Estadual de Campinas
36	Universidade Federal do Itajubá	84	Universidade Estadual Paulista "Júlio de Mesquita Filho"
37	Universidade Federal de Juiz de Fora	85	Universidade Virtual do Estado de São Paulo
38	Universidade Federal de Lavras	86	Universidade Estadual de Maringá
39	Universidade Federal de Minas Gerais	87	Universidade Estadual de Ponta Grossa
40	Universidade Federal de Ouro Preto	88	Universidade Estadual de Londrina
41	Universidade Federal de São Carlos	89	Universidade Estadual do Oeste do Paraná
42	Universidade Federal de São João del-Rei	90	Universidade Estadual do Centro-Oeste
43	Universidade Federal de São Paulo	91	Universidade Estadual do Paraná
44	Universidade Federal de Uberlândia	92	Universidade Estadual do Norte do Paraná
45	Universidade Federal de Viçosa	93	Universidade do Estado de Santa Catarina
46	Universidade Federal do ABC	94	Universidade de Taubaté
47	Universidade Federal do Espírito Santo	95	Universidade Regional de Blumenau
48	Universidade Federal do Estado do Rio de Janeiro		