State University of Feira de Santana

Computer Science Graduate Program

Designing and Evaluating a Computational Thinking and Computing Attitudes Instrument for Educational Purposes

Eduardo Carneiro de Oliveira

Feira de Santana

2021

State University of Feira de Santana

Computer Science Graduate Program

Eduardo Carneiro de Oliveira

Designing and Evaluating a Computational Thinking and Computing Attitudes Instrument for Educational Purposes

Master's Thesis presented to State University of Feira de Santana as a partial requirement for obtaining the title of Master in Computer Science.

Advisor: Roberto Almeida Bittencourt

Feira de Santana

2021

Ficha catalográfica - Biblioteca Central Julieta Carteado - UEFS

Oliveira, Eduardo Carneiro de
O46d Designing and evaluating a computational thinking and computing attitudes instrument for educational purposes / Eduardo Carneiro de Oliveira. - 2021.
38f.

Orientador: Roberto Almeida Bittencourt

Dissertação (mestrado) - Universidade Estadual de Feira de Santana. Programa de Pós-Graduação em Ciência da Computação, 2021.

1. Pensamento computacional. 2. Questionário. 3. Educação em Computação. 4. Atitudes em relação à computação. I. Bittencourt, Roberto Almeida, orient. II. Universidade Estadual de Feira de Santana. III. Título.

CDU: 004.4

Rejane Maria Rosa Ribeiro – Bibliotecária CRB-5/695

Eduardo Carneiro de Oliveira

Designing and Evaluating a Computational Thinking and Computing Attitudes Instrument for Educational Purposes

Dissertação apresentada à Universidade Estadual de Feira de Santana como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Feira de Santana, 3 de dezembro de 2021

BANCA EXAMINADORA

Roberto Almeida Bittencourt (Orientador(a))
Universidade Estadual de Feira de Santana

Dalton Dario Serey Guerrero
Universidade Federal de Campina Grande

Rodrigo Silva Duran
Instituto Federal do Mato Grosso do Sul

Abstract

Computational Thinking (CT) is a set of thinking processes used by computer scientists to formulate problems and describe solutions. Given that the skills typically associated with CT, such as abstraction and algorithmic thinking, are largely demanded for the 21st century, there have been several claims on the need to introduce CT to the educational community. One crucial aspect in this regard is the perceptions that students hold on the subject as well as on Computing in general. Various studies have been carried out to measure students' attitudes towards Computing. However, assessing the understanding of CT and Computing attitudes in undergraduate education, and particularly in teacher education, has not been fully explored yet. Most current instruments are either not aimed at educators or not designed to cover elements related to both CT and Computing. In this work, our goal was to design and evaluate a survey instrument on perceptions of CT and attitudes towards Computing for undergraduate students. Our work aimed to evaluate whether the instrument has potential to grasp the aspects involved in five different intended constructs: 1) understanding of CT, 2) application of CT in classroom, 3) confidence to learn Computing, 4) interest for Computing and 5) perception of Computing usefulness. After applying the instrument in two different phases, we computed its reliability and used confirmatory factor analysis results to calculate its internal validity. Results showed evidence that the instrument is both reliable and valid. Thus, researchers and educators may reuse our survey instrument to measure students' impressions of CT and Computing attitudes in educational contexts.

Keywords: Survey instrument; Computational thinking; Attitudes towards computing; Computing education.

Resumo

Pensamento Computacional (PC) é um conjunto de processos de pensamentos usados por cientistas da computação para elaborar problemas e soluções. Considerando que as habilidades tipicamente associadas ao PC, como abstração e pensamento algorítmico, são consideradas muito necessárias para a sociedade do século XXI, temos visto várias discussões sobre a necessidade de apresentar o PC para a comunidade educacional. Um aspecto crucial nesse sentido é das percepções que os estudantes possuem sobre PC e sobre Computação de forma geral. Diversos estudos foram realizados para avaliar as atitudes dos estudantes em relação à Computação. Entretanto, a avaliação da compreensão de PC e das percepções sobre Computação no ensino superior, e particularmente nos cursos de licenciatura, ainda não foi totalmente explorada. A maioria dos instrumentos atuais não é focada para educadores ou não é projetada para cobrir tanto PC quanto Computação. Neste trabalho, nosso objetivo foi projetar e avaliar um questionário sobre as percepções de Pensamento Computacional e Computação para estudantes do ensino superior. Nosso trabalho focou em avaliar se o instrumento tem potencial para capturar os aspectos envolvidos em cinco categorias: 1) compreensão de PC, 2) aplicação de PC na sala de aula, 3) confiança para aprender Computação, 4) interesse por Computação, 5) percepções da utilidade da Computação. Depois de aplicar o instrumento em duas fases, nós calculamos sua confiabilidade e usamos a análise fatorial confirmatória para medir a sua validade. Os resultados sugerem que o instrumento é confiável e válido. Portanto, pesquisadores e educadores podem aplicar o nosso questionário para avaliar as impressões de Pensamento Computacional e atitudes em relação à Computação dos estudantes em contextos educacionais.

Palavras-chave: Questionário; Pensamento computacional; Atitudes em relação à computação; Educação em computação.

Preface

This Master's Thesis was submitted to State University of Feira de Santana (UEFS) as a partial requirement for obtaining the title of Master in Computer Science.

The Master's Thesis was developed in the Computer Science Graduate Program (PGCC), supervised by Dr. **Roberto Almeida Bittencourt**.

Acknowledgements

Portuguese Version Eu gostaria de agradecer primeiramente à minha família, por todo o amor e apoio prestados. Meus pais, Graça e Erenildo, que me ensinaram que o caminho do estudo e trabalho é o correto a ser seguido. E ao meu irmão e melhor amigo, Evandro, que é o meu exemplo de que nunca devo parar de estudar e buscar mais. Obrigado, família. Cada dia com vocês me torna uma pessoa melhor!

Agradeço também ao meu orientador, Roberto, por todos os anos de parceria e ensinamentos. Em um período tão difícil e cheio de incertezas que tem sido a pandemia do Covid-19, foi quem me sugeriu e incentivou a cursar o mestrado do PGCC-UEFS. Desde a época da graduação, nas disciplinas e orientações, tenho aprendido muito a cada encontro. Obrigado, professor!

Agradeço ao SENAI-BA e Colégio Helyos, instituições nas quais tive o prazer de ser professor nos últimos anos. Tenho gratidão pela oportunidade de trabalhar e crescer profissionalmente em ambas as instituições. Pelo incentivo ao meu desenvolvimento acadêmico e pessoal, bem como pela compreensão e flexibilidade em várias situações. Esta conquista também é de vocês!

Por fim, sou grato ao corpo docente e colegas do PGCC e ao grupo de pesquisa do LESS. Primeiro, pela chance de conhecer várias pessoas incríveis. E também por todo o conhecimento adquirido nesse período. Tenho muito orgulho de ter feito o mestrado do PGCC-UEFS e ter sido parte do LESS!

English Version First, I would like to thank my family, for all love and support given. My parents, Graça and Erenildo, who taught me that studying and working hard is the right path to be taken in life. My brother and best friend, Evandro, who is my example to never stop studying and keep going. Thank you, my family. I become a better person each day with you all!

Also, I would like to thank my supervisor, Professor Roberto, for his guidance and friendship. In such a tough period of the Covid-19 pandemic, you recommended and supported me to take my Master's degree at the PGCC-UEFS. I have been learning a lot with you every day. Thank you, Professor Roberto!

I would like to thank SENAI-BA and Helyos International School, educational institutions in which I had the pleasure to work as a teacher in the latest years. I am grateful for the opportunity to work and develop professionally in both institutions, as well as for the help and empathy offered in several situations. This achievement is also yours!

Finally, I am grateful to the PGCC faculty and the LESS research team, for sharing your knowledge. I feel privileged for receiving my Master's degree from PGCC-UEFS and for have been part of the LESS team!

Summary

Alignment with the Research Line

Research Line: Software and Computing Systems

This Master's Thesis contributes an instrument to measure students' perceptions of Computational Thinking and attitudes towards Computing. It also contributes the evaluation of that instrument. Computing Education is needed to provide background to Software and Computing Systems professionals. In this sense, this thesis is aligned with the Software and Computing Systems research line by providing a potentially relevant instrument to the Computing Education community that may affect the success of professional development programs.

Papers and Awards

Oliveira, E. C., Correia, R. C., & Bittencourt, R. A. (2021, April). Evaluating a Computational Thinking and Computing Attitudes Instrument for Educational Purposes. In 2021 IEEE Global Engineering Education Conference (EDUCON) (pp. 748-754). IEEE.

List of Tables

List of Acronyms

Acronym	**Description**
CFA	Confirmatory Factor Analysis
CS	Computer Science
CT	Computational Thinking
PD	Professional Development

Chapter 1

Introduction

The use of digital technology has been playing an increasing key role in our lives in the latest decades. As this sort of technology tends to be even more present in daily activities, researchers have been arguing on the relevance of preparing people to become technology makers, instead of keeping them as mere consumers (Hsu et al., 2017). According to Crick and Sentance (2011), a logical way to include Computing in the K-12 education is by allowing students to be introduced to Computing principles at a young age. Thus, they could be able to better comprehend the technology surrounding them (Crick and Sentance, 2011).

Gal-Ezer and Stephenson (2014) suggested that a crucial condition for comprehending technology is teacher development. Their study showed that there have been several efforts to qualify K-12 educators in Computing, and some countries have already included Computer Science education requirements in their K-12 curricula. In Brazil, there has been a growth of undergraduate programs focused on teacher education in Computing (Linhares and Santos, 2021). In the United States, the Computer Science for All (CS4All) initiative was created to promote teacher development in Computing. From the CS4All[1], projects like Home4CS[2] and CSVisions were developed to prepare K-12 educators to teach computer science. In England, the National Centre for Computing Education was established to improve the supply of the field in K-12 education, developing programs such as TeachComputing.org[3] and Isaac Computer Science[4]. These efforts are typically concerned with educators' perceptions about the Computing field, since these perceptions may have an impact on how the subject is taught. Another common goal is to ensure that teachers are provided with technical, content and pedagogical knowledge to teach Computing-related topics, such as Computational Thinking (Gal-Ezer and Stephenson, 2014).

[1]https://www.csforall.org/
[2]https://www.csfored.org/
[3]https://teachcomputing.org/
[4]https://isaaccomputerscience.org/

Computational Thinking (CT) is recognized by several researchers from the Computing Education community as one of the essential set of skills demanded for education in the 21st century (Lorenceau et al., 2019). Educators can be a crucial stakeholder in this regard. However, before teaching CT, they need to be qualified. Also, due to the connection between CT and Computer Science (CS) principles (Wing, 2006), learning CT may change teachers' attitudes towards Computing, and improve their own perceptions of CT (Barr and Stephenson, 2011).

The definition of Computational Thinking has been first presented by Wing (2006) in 2006 (Wing, 2006). In her work, Wing (2006) stated that CT comprises the thought processes involved to formulate problems and their solutions, in such a way that humans and computers are able to understand. Since then, the interpretation of CT has been frequently discussed and updated by both researchers, government bodies and teachers' associations.

Currently, there is not a consensus on the definition of Computational Thinking. Recent publications have also focused on presenting strategies to teach CT and develop its skills. For instance, Selby (2014) described CT considering five skills: Abstraction, Algorithmic Design, Decomposition, Evaluation and Generalization (Selby, 2014). In the educational context, Santos et al. (2018) reviewed, in a systematic mapping study, six approaches to promote CT skills in class, such as the use of unplugged computing, robotics concepts and practices and the development of games and animations (Santos et al., 2018).

Furthermore, researchers have also been discussing educators' attitudes towards Computing. Eagly and Chaiken (1998) explained that these attitudes involve perceptions about a specific group of people or some of their characteristics (Eagly and Chaiken, 1998). As educators are the main stakeholders who can incorporate CT in classroom, they may need to have positive perceptions about Computing in general. Thus, comprehending the definition of CT and its use in classroom as well as obtaining positive attitudes towards the field are complementary needs for educators.

According to Fennema and Sherman (1976)'s renowned instrument, attitudes can be, for example, student's confidence to learn the subject or their beliefs about the usefulness of studying this subject (Fennema and Sherman, 1976). More specifically, the confidence construct measures one's ability to learn and perform well on tasks related to the field. In addition, the usefulness category grasps learner's perceptions about the utility of the subject when associated to their career goals.

Various efforts have been made to qualify educators in CT and Computing, such as in Yadav et al. (2014) and Oliveira et al. (2019). Yadav et al.'s work is particularly relevant for qualifying and evaluating students in both aspects of the understanding of CT: its application in school and their perceptions of Computer Science. However, the increasing number of these efforts, combined with a lack of consensus on the definition of CT, have also provoked a need to evaluate these courses (Cutumisu et al., 2019). To assess such efforts, numerous instruments that measure one's understanding of CT and attitudes towards CS have been used (Hoegh and Moskal, 2009; Magerko et al., 2016; Dorn and Elliott Tew, 2015). Hoegh and Moskal's instrument, for instance, was developed in 2009 and it is still largely used, due to the quality of its questions to cover elements of several categories as well as to its flexibility to be applied both before and after an intervention.

Nonetheless, there is still a demand for validated instruments for teachers, especially to measure their perceptions of CT and Computing. Current instruments are not designed for educators or are exclusively focused either on CT or on attitudes towards Computing. Thus, designing validated questionnaires that holistically investigates CT skills and attitudes might contribute not only to analyze teachers' perceptions, but also to comprehend their understanding on CT and how this subject can be incorporated in classroom.

1.1 Goals and Research Questions

In this work, our goal is to design and evaluate an instrument to measure students' (from both Educational and Computing tracks) impressions of CT as well as their attitudes towards Computing. We aim to evaluate whether the survey has potential for grasping the elements related to five designed constructs: 1) comprehension of CT, 2) application of CT in classroom, 3) confidence to learn Computing, 4) interest for Computing and 5) awareness of Computing utility.

The research questions that conduct this study are:

- RQ1 – Which survey questions would better evaluate students' understanding on a definition of Computational Thinking from the literature?
- RQ2 – Which survey questions would better grasp students' perceptions on the use of typical approaches to teach Computational Thinking?
- RQ3 – Do the survey questions from Hoegh and Moskal's work remain valid and reliable to measure students' attitudes towards Computing?

1.2 Contributions

This work has two major contributions. The first one is the development of a survey instrument to capture one's comprehension of Computational Thinking and its application in an educational context. In addition, the instrument is designed to grasp one's impressions about Computing in terms of their confidence in studying the field, their interest in continue studying and using Computing concepts, as well as their perceptions about the usefulness of Computing for professional goals.

The second major contribution of this work is the evaluation of the survey instrument. The instrument was applied with two large and potentially qualified samples, from several backgrounds. From the analyses, we show preliminary evidence towards statistical reliability and internal validity of the instrument.

Chapter 2

Background

In this chapter, we present the background needed to provide better comprehension of this work: Computational Thinking, Assessment of CT, Attitudes towards Computing and Professional Development in Computing. Later, we discuss relevant works related to these topics.

2.1 Computational Thinking

In 2006, Jeannette Wing popularized the concept of Computational Thinking (CT). Initially, Wing described CT as the processes of thinking used by computer scientists to formulate problems and provide solutions. For her, these problems and solutions need to be expressed at a level of abstraction so that both humans and machines could effectively comprehend (Wing, 2006).

However, Wing (2006)'s definition of CT is not universally accepted and has been often reviewed in the latest years, culminating in a range of views on Computational Thinking. León et al. (2019)'s work indicate 16 publications regarding a definition of CT (León et al., 2019). As an example, the Computer Science Teachers Association (CSTA) and the International Society for Technology in Education (ISTE) described some core characteristics of CT, such as solving problems with the focus on the most efficient and effective blend of steps and resources.

The latest studies on CT have also aimed at how to teach it and develop its skills. For instance, Selby (2014)'s work presents Computational Thinking by describing it in five core skills (Selby, 2014). In her Ph.D. thesis, Selby identified the following CT skills: Abstraction, Algorithmic Design, Decomposition, Evaluation and Generalization. According to the author, although these skills were already defined in the literature, a taxonomy was demanded. "Abstraction" regards the ability of rationally consider the essential parts of a problem. "Algorithmic Design" can be basically interpreted as the ordering of the steps required by a solution, whilst "Decomposition" refers to the skill of breaking a problem down into smaller, more easily solved, sections. Furthermore, "Evaluation" means the addition of a criticism of weaknesses in solutions, and why some are better or worse than alternative approaches, while "Generalization" is described as the capacity of expressing a solution in generic terms, so that it can be applied to similar situations (Selby, 2014).

The previous definitions of Computational Thinking are considered of a broader applicability, since humans and computers are seen as computational agents and that CT should be accessible to everyone. However, some authors advocate for a narrower, ”traditional view” of CT and its use. According to Denning (2017), CT should rely on computational models and algorithms. He also stated that one cannot claim that improving coding skills help develop general problem-solving skills, as there is no current evidence to prove this point (Denning, 2017). Pears et al. (2021) extended his work, listing eight troublesome ideas in CT, including that learning CT does not assure a comprehension of how computers and virtual worlds work (Pears et al., 2021).

Although there is little consensus in some core aspects of what Computational Thinking should be, such as its breadth of applicability and whether humans can be considered computational agents or not, the scientific community found general agreement on other topics related to CT. Curzon et al. (2019) point out that most researchers recognize that abstraction and algorithmic thinking are essential skills applied by computational agents to provide solutions to general problems, and the focus on algorithmic solutions is what differentiate CT from other problem-solving methods (Curzon et al., 2019). From our perspective, it is more relevant to expose and discuss about the cognitive skills related to CT than obtaining a consensus on its definition. In this sense, we have embraced the work from Selby (2014).

Another frequent discussion is whether Computational Thinking needs to be used by the educational community. Considering the well-known skills demanded for the 21st century, various researchers stated that CT should be accessible to everyone held in K-12 education (Wing, 2006; Curzon et al., 2019). This can be done through the incorporation of CT in classroom. Barr and Stephenson (2011) suggested short-term strategies to promote CT in teacher education, such as the advance of a positive relationship between computer scientists and educational associations to advocate for CT, as well as the professional development of CT for educators (Barr and Stephenson, 2011).

In the methodological context, Santos et al. (2018) reviewed six approaches that can be used to teach CT and Programming (Santos et al., 2018). These approaches include: 1) the traditional way of lecturing, aiming at algorithms to solve mathematical problems; 2) the employment of robotics concepts and practices; 3) the use and development of games; 4) the application of code to handle hardware components, like engines and sensors; 5) the adoption of playful activities without using a computer and 6) the development of animations and stories.

2.2 Assessment of CT

In the latest years, there have been various interventions on Computational Thinking for K-12 educators. Nonetheless, there is still a lack of widely-accepted instruments to measure CT skills and other related abilities (Zhong et al., 2016). According to Cutumisu et al. (2019), a reason for this can be the lack of consensus on a CT definition, which makes it harder to design a valid instrument to measure students' CT skills (Cutumisu et al., 2019). In Santana et al.'s work, the authors listed some works focused on the assessment of CT, including Bebras (Dagiene and Stupuriene, 2016), Computational Thinking Test (CTT) (Román-González et al., 2017) and Dr. Scratch (Moreno-León et al., 2015).

Bebras is an international contest intended to promote Informatics and Computational Thinking in K-12 education (Dagiene and Stupuriene, 2016). More specifically, it consists of a challenge with 18 to 24 questions to be solved by students, according to their grade level. Students are often supervised by teachers, who incorporate Bebras in their teaching activities.

Román-González et al. (2017) developed the Computational Thinking Test (CTt) to assess both CT and coding skills of middle school students. This multiple choice test contains 28 items, which are related to programming concepts, like conditionals, loops and functions (Román-González et al., 2017).

Another instrument used to assess programming and CT skills is Dr. Scratch. According to Moreno-León et al. (2015), this tool allows educators and students to automatically analyze Scratch projects. From the analysis, users receive feedback to improve their code as well as develop CT skills, such as abstraction, data representation and parallelism (Moreno-León et al., 2015).

2.3 Attitudes towards Computing

Currently, a common topic of interest in the Computer Science Education community is the students' attitudes towards the field. According to Eagly and Chaiken (1998), attitude is a psychological trend manifested by the evaluation of an entity in a level of favor or disfavor (Eagly and Chaiken, 1998). Luxton-Reilly et al. (2018) add that attitudes involve self-perceptions, and, in the context of computer programming, these perceptions may vary from abstract aspects, such as a discipline, to a more concrete one, such as a programming tool (Luxton-Reilly et al., 2018). Various studies evaluated students' attitudes towards Computing, such as in Yadav et al. (2014), Hoegh and Moskal (2009), Magerko et al. (2016), Dorn and Elliott Tew (2015), Ko (2009), Bockmon et al. (2020) and Gomes et al. (2012).

A typically studied attitude is the interest in the subject. Dewey (1913) defined student interest as the engagement or engrossment in an activity that is considered worthy (Dewey, 1913). Specifically in Computing, Hildebrandt and Diethelm (2012) addressed essential factors that impact students' interest in studying the subject:

quality of the material and the instructional model, social integration and support of autonomy (Hildebrandt and Diethelm, 2012).

Several validated instruments designed for Computing attitudes relied on Fennema and Sherman (1976)'s work. In the 1970's, they designed an instrument to measure students' attitudes towards the learning of mathematics. Their instrument included categories such as the confidence in learning the subject. This scale is designed to grasp confidence in student's ability to learn and perform well on tasks related to the field. Another category is the usefulness, which is intended to measure one's beliefs about the current utility of the subject, associating it to their career goals and other activities (Fennema and Sherman, 1976). These categories are commonly used in Computing instruments, such as in Wiebe et al. (2003) and Hoegh and Moskal (2009).

To better comprehend how students form their first perceptions on computers and programming, Ko (2009) collected autobiographies of students from a Computing course, which described their initial contacts with the subjects (Ko, 2009). Findings suggested that students' attitudes towards programming did not change because of a single particular experience with computers and programming. On the contrary, digital technology has slowly started playing a more prominent role in their lives, as a result of various meetings, both positive and negative, until it became relevant to them. Thus, the persistent and positive exposure of students to computers and programming may help foster future generations of information technology professionals.

In another work, Gomes et al. (2012) studied the various causes of difficulty in CS1 courses, from study methods to students' attitudes towards learning programming (Gomes et al., 2012). They used the Inventory of Usual Study Attitudes and Behaviors (IACHE) to measure, among other variables, students' study methods and their perceptions about their own skills (Tavares et al., 2004). Results showed a strong correlation between students' grades and their own perceptions of competence and motivation. Thus, these works illustrate how perceptions of competence are relevant factors to the success in learning programming.

2.4 Professional Development in Computing

The importance of including Computer Science in K-12 education has been broadly discussed by the scientific community. According to Grover and Pea (2013), the study of CS in schools has potential to increase students' career possibilities, as well as to develop skills such as critical thinking, problem analysis and design of solutions (Grover and Pea, 2013). A typical way to promote Computing in K-12 education is by offering professional development (PD) for pre- and in-service teachers, and several countries have been making efforts in this direction (Ni et al., 2021).

For instance, in the United States, although CS courses are not compulsory at the K-12 level, various initiatives have been promoted in recent years to qualify educators. These efforts are either funded by the National Science Foundation, higher education institutions or private companies (Menekse, 2015). Also, Australia has its own national curriculum regarding digital technologies in K-12 education (Falkner et al., 2018), whereas the United Kingdom developed a curriculum on Information and Communication Technology, focusing on the instruction of Computer Science (Brown et al., 2014).

In Brazil, the first undergraduate program in Computing with focus on teacher education was released in 1997, at the University of Brasília (UnB). In 2002, the Brazilian Computer Society (SBC) developed national guidelines and curricula for teacher education undergraduate programs in Computing (Zorzo et al., 2017). Since then, there has been a sharp increase in the number of degrees in Computing dedicated to teacher education. Currently, there are more than 150 degrees programs in Computing established in Brazil (Linhares and Santos, 2021).

These efforts have contributed to a growing interest in the incorporation of Computer Science in K-12 education. Nevertheless, along with the development of PD programs, a demand emerges to evaluate these programs. Most interventions on Computer Science for educators are still to be analyzed and validated.

2.5 Related Work

Hoegh and Moskal (2009) developed a survey to assess undergraduate students' attitudes towards Computer Science (Hoegh and Moskal, 2009). Their survey consisted of five constructs to be measured, such as students' interest and their beliefs in the usefulness of studying CS. Findings suggested that the survey is statistically reliable. However, Hoegh and Moskal's work was not validated in multiple languages, including Portuguese. Doing this could increase the external validity of the instrument.

Yadav et al. (2014) designed a Computational Thinking course to pre-service teachers (Yadav et al., 2014). They created a survey to assess participants' comprehension on CT and their attitudes towards Computing, relying on a previous study of Yadav et al. (2011) and including some categories from Hoegh and Moskal's work. The outcomes from Yadav et al.'s study indicated that incorporate CT modules into teacher education can contribute to the development of CT at the K-12 level. Nonetheless, Yadav et al.'s work does not capture the typical approaches to teach CT, such as the use of unplugged computing, games and animations and educational robotics. Also, from our point of view, the phrasing in some questions regarding the definition of CT can be improved.

Several studies evaluated Hoegh and Moskal's and Yadav et al.'s survey instruments. For instance, Heersink and Moskal (2010) evaluated the instrument from Hoegh and Moskal (2009) in a Computing course for high school students. Results from their study implied that the survey could measure all intended constructs for a high school population (Heersink and Moskal, 2010). Oliveira et al. (2019)'s study had similar findings. In their work, the authors designed a CT course for in-service teachers and evaluated them with an instrument based on both Yadav et al.'s and Hoegh and Moskal's studies. Findings from this work showed evidence of a positive correlation between attitude categories, such as "Interest" and "Usefulness" (Oliveira et al., 2019).

Leonard et al. (2018) evaluated Yadav et al.'s instrument in a course for in-service teachers. Their results suggested that participants exposed to blended game/robotics design obtained higher scores in understanding CT, and how it could be incorporated into their classroom practices, compared to the ones who were exposed to a single pedagogical strategy (Leonard et al., 2018). Furthermore, Mouza et al. (2017) evaluated Yadav et al.'s instrument with an intervention for pre-service teachers, in a pre- and post-test analysis. Their outcomes indicated a significant gain in the comprehension of Computational Thinking after the course, but did not find strong evidence on how participants could explore pedagogical strategies for infusing CT knowledge (Mouza et al., 2017). Results from these studies may imply the need for an update in Yadav et al.'s survey questions to assess multiple strategies to teach CT.

Also concerning teacher development in Computational Thinking, Jocius et al. (2021) designed the Virtual Pivot. This work relied on Infusing Computing, a long-term project in the United States created to stimulate in-service teachers to incorporate CT into their classes (Jocius et al., 2021). Their findings suggest that three categories of support (digital tools, formats, and supports for teacher engagement and collaboration) contributed to participants' increase in self-efficacy in teaching CT and to design CT-infused content-area lessons.

Additionally, Magerko et al. (2016) created the Attitudes Towards Computing Scale (Magerko et al., 2016). The survey has 19 items split into five categories, including Computing enjoyment and motivation to succeed in Computing. In 2019, Wanzer et al. (2019) evaluated this instrument (Wanzer et al., 2019), showing that the survey is moderately valid to measure attitudes towards Computing.

Another example of instrument has been developed by Dorn and Elliott Tew (2015): the Computing Attitudes Survey (CAS) (Dorn and Elliott Tew, 2015). Their instrument has 26 items distributed into five constructs, like strategies for problem solving and real-world connections. CAS has been rigorously designed and validated for supporting researchers to evaluate learners' attitudes and beliefs related to Computing.

Also, Bockmon et al. (2020) has extended Dorn and Elliott Tew (2015)'s work by adding nine questions on gender issues and five questions to capture students' attitudes towards the utility of Computing (Bockmon et al., 2020). Their outcomes support the results found in Dorn and Elliott Tew (2015) as well as indicated that participants who have a strong bias towards males in Computing are more likely to have a fixed mindset when attempting to solve problems.

Our work aims to measure both perceptions about Computational Thinking and attitudes towards Computing. However, our study differs from the others by including a large sample of first-year university students, where half of them are enrolled in a teacher education program. It extends Yadav et al.'s work by designing and validating an instrument to deal with two dimensions of Computational Thinking related to K-12 education. In addition, it reuses three categories of Hoegh and Moskal's survey on Computing attitudes. Finally, our work is similar to Magerko et al., Dorn and Elliott Tew and Bockmon et al. for dealing with attitudes towards Computer Science, but it differs in focus, since we are interested in the context of professional development of educators that will teach CT in K-12 education.

Chapter 3

Phase 1: Instrument Design

The design and evaluation of our instrument was split into two phases. In this chapter, we present the methods and results of Phase 1.

3.1 Methodology

First, we present the scenario, participants, description of the CT course, development and evaluation of the instrument and the processes of data collection and analysis in Phase 1.

3.1.1 Scenario and Participants

In the first semester of 2020, we established a partnership with researchers from the Virtual State University of São Paulo (UNIVESP) to evaluate a Computational Thinking course using a survey instrument. The course was conducted in an online format, and split into eight modules, being each module conducted in a week.

The course was offered to nearly 16,000 first-year university students at UNIVESP. After completing the full course, they were surveyed. We obtained the answers from 2,290 students. They read an informed consent form and, if agreed to participate in the research, provided demographic data, such as age, gender and the college major that they were most likely to enroll in.

The course members' average age was 36.6 ± 9.8. The gender figures were 51.4% of female, 48.3% of male and 0.2% of others. In addition, the participants' major choice ranged from six different options: Education (27%), Language and Arts (10%) and Mathematics (7%) from the Educational track; Computer Engineering (24%), Data Science (11%) and Information Technology (8%) from the Computing track. Approximately 13% of participants had not choose their major course yet.

3.1.2 Course Description

The course was split into eight modules. The author and facilitators of this course rely their views about CT on Wing (2006) and Selby (2014). Units 1 and 2 regarded the introduction to CT and programming, presenting their impact, skills and tools. Units 3 and 4 exposed two coding platforms: Scratch and App Inventor. Students were taught how to design simple projects in these platforms with programming basics. Unit 5 concentrated on digital literacy, as the participants studied how computers and internet work. Unit 6 discussed the application of Computational Thinking in K-12 education, focusing on ongoing standards and CT curricula. Unit 7 concerned problem-solving fundamentals and techniques, such as abstraction and decomposition. In Unit 8, the course content was reviewed. The course planning can be seen in Table 3.1.

Table 3.1: Course Planning

Unit	Content
1	Introduction to CT: Motivation; Pillars, Skills, Contributions and Unplugged Computing.
2	Algorithms: concepts of problem, representation, logical reasoning and programming languages.
3	Scratch learning environment.
4	App Inventor learning environment.
5	Digital literacy: Digital technologies for learning processes, social life of individuals.
6	Brazilian standards for computing curriculum; Guidelines for computing in K-12 Education.
7	Problem solving: Pólya and Pozo methods; Problem solving techniques. Examples of applications.
8	Review and questions.

3.1.3 Instrument Development and Evaluation

The survey included questions regarding participants' comprehension on Computational Thinking and attitudes towards Computing. This survey was based on the studies of Yadav et al. (2014) and Hoegh and Moskal (2009), considering these specific categories: from the former work, we kept Definition and Classroom constructs; from the latter, we selected the questions in Confidence, Interest and Usefulness groups.

As stated in Yadav et al. (2014), the "Definition" construct relies on four questions concerning participants' understanding on the definition of Computational Thinking. For example, it includes items on the association of logical thinking and CT, as well as the application of general solutions to a range of situations.

The "Classroom" category consists of two statements regarding learners' perceptions about CT in their classrooms. The first question covers the incorporation of CT in the classroom by including the use of computers in the lesson plan, whereas the second question correlates CT in the classroom with activities that encourage problem solving.

According to the study in Hoegh and Moskal (2009), the "Confidence" construct has eight items bearing on respondents' self-confidence to obtain and apply Computer

Science skills. The “Interest” category groups ten statements about participants' attraction to learn CS and use it to solve problems. Finally, “Usefulness” measures their thoughts on the implications of studying CS for their career goals, with a set of six questions.

Later, we analyzed the Definition of CT section in order to correlate it to the work of Selby (2014) on CT skills. Three statements were included in the survey to better represent Abstraction, Decomposition and Evaluation skills. We also examined Santos et al. (2018)'s study and added four statements in CT in Classroom, concerning distinct approaches to incorporate CT into classroom: App Development, Game Development, Robotics and Unplugged Computing. From this draft survey, we asked an expert panel, composed by three computing faculty, to review the instrument. The panelists offered qualitative feedback about question concepts and phrasing. Based on the results, question phrasing was improved. The complete set of statements used in Phase 1, translated to English, is shown in Table 3.2. Additionally, the Portuguese version of the survey is shown in Appendix A.

Table 3.2: Computational Thinking and Computing Attitudes Survey - Phase 1

Category	Statement	Source[a]
Definition	D1. Computational Thinking is understanding how computers work.	1
Definition	D2. Computational Thinking involves thinking logically to solve problems.	1
Definition	D3. Computational Thinking involves using computers to solve problems.	1
Definition	D4. Computational Thinking involves abstracting general principles and applying them to other situations.	1
Definition	D5. Computational Thinking involves considering only the most relevant aspects of problems to design their solutions.	2*
Definition	D6. Computational Thinking involves breaking problems into smaller parts to facilitate their solution.	2*
Definition	D7. Computational Thinking involves evaluating possible approaches analytically before making a decision.	2*
Classroom	E1. Computational Thinking can be incorporated in the classroom by using computers in the lesson plan.	1
Classroom	E2. Computational Thinking can be incorporated in the classroom by allowing students to problem solve.	1
Classroom	E3. Computational Thinking can be incorporated in the classroom by developing apps.	3*
Classroom	E4. Computational Thinking can be incorporated in the classroom by using robots.	3*
Classroom	E5. Computational Thinking can be incorporated in the classroom by creating games and animations.	3*
Classroom	E6. Computational Thinking can be incorporated in the classroom by promoting activities to understand Computing without using a computer.	3*
Confidence	C1. I am comfortable with learning Computing concepts.	4
Confidence	C2. I have little self-confidence when it comes to Computing courses.	4
Confidence	C3. I do not think that I can learn to understand Computing concepts.	4
Confidence	C4. I can learn to understand Computing concepts.	4
Confidence	C5. I can achieve good grades (C or better) in Computing courses.	4
Confidence	C6. I am confident that I can solve problems by using computer applications.	4
Confidence	C7. I am not comfortable with learning Computing concepts.	4
Confidence	C8. I doubt that I can solve problems by using computer applications.	4
Interest	I1. I would not take additional Computer Science courses if I were given the opportunity.	4
Interest	I2. I think Computer Science is boring.	4
Interest	I3. I hope that my future career will require the use of Computer Science concepts.	4
Interest	I4. The challenge of solving problems using Computer Science does not appeal to me.	4
Interest	I5. I like to use Computer Science to solve problems.	4
Interest	I6. I do not like using Computer Science to solve problems.	4
Interest	I7. The challenge of solving problems using Computer Science appeals to me.	4
Interest	I8. I hope that I can find a career that does not require the use of Computer Science concepts.	4
Interest	I9. I think Computer Science is interesting.	4
Interest	I10. I would voluntarily take additional Computer Science courses if I were given the opportunity.	4
Usefulness	U1. Developing Computing skills will not play a role in helping me achieve my career goals.	4
Usefulness	U2. Knowledge of Computing will allow me to secure a good job.	4
Usefulness	U3. My career goals do not require that I learn Computing skills.	4
Usefulness	U4. Developing Computing skills will be important to my career goals.	4
Usefulness	U5. Knowledge of Computing skills will not help me secure a good job.	4
Usefulness	U6. I expect that learning to use Computing skills will help me achieve my career goals.	4

1 – Yadav et al. (2014).
2 – Selby (2014).
3 – Santos et al. (2018).
4 – Hoegh and Moskal (2009).
[a] An asterisk (*) indicates a statement based on the authors' work. Its absence indicates a replicated statement from the authors' original instrument.

3.1.4 Data Collection and Analysis

In the online survey, applied right after the course, we scrambled questions from different constructs, and participants responded the survey by rating each statement in a 5-point Likert scale, ranging from 1 (Not True) to 5 (Very True). Negatively phrased statements, which were already part of the work from Hoegh and Moskal (2009), were kept with their original wording and reversely scored for data analysis, in order to represent positive answers with high scores.

We conducted data analysis with R 4.0.1 software, using *lavaan* and *psych* packages. First, we measured Cronbach's Alpha to verify the internal consistency of the five constructs as well as their statements. O'Rourke and Hatcher (2013) state that Cronbach's Alpha is an index of reliability, ranging from 0 to 1, related to the variation of the true score of the designed statements in their categories (O'Rourke and Hatcher, 2013). An index of 0.7 is generally considered as acceptable.

To validate our questionnaire, we considered that applying Exploratory Factor Analysis (EFA) was not needed, as we already knew the actual dimensions of interest to our analysis, which were based on previous work in the field of computing education. Our demand was to analyze how properly the items represented these dimensions. Thus, we used Confirmatory Factor Analysis (CFA).

According to Thompson (2004), CFA can be used for testing the fit of a factor model, considering its number of factors, which questions reflect these factors and the correlations between them (Thompson, 2004). Statistics considered for our initial CFA analysis included Comparative Fit Index (CFI) and Tucker-Lewis Index (TLI), as well as Root Mean Square Error of Approximation (RMSEA) and Standardized Root Mean Square Residual (SRMR). Bentler (1990) puts that these indices are used to measure the goodness-of-fit of a structural model (Bentler, 1990). Typically, $CFI \geq 0.8$, $TLI \geq 0.8$, $RMSEA < 0.08$ and $SRMR < 0.07$ suggest a good model fit.

After checking the preconditions, we performed the factor analysis by computing loading factors and standard errors for each statement. We considered the threshold of loading factors less than 0.30 to remove questions that did not represent the designed construct.

3.2 Results

In this section, our results are presented in terms of the instrument's reliability and internal validity in Phase 1.

3.2.1 Reliability

We used Cronbach's Alpha to calculate the reliability for the set of questions of each construct. In our first analysis, we found Cronbach's Alpha values above 0.70 for

Classroom, Confidence, Interest and Usefulness categories. However, we found an initial α of 0.61 for Definition. Thus, we analyzed which questions were decreasing the reliability (D1, D3 and D5) and removed them from the set, achieving an acceptable α for Definition. Table 3.3 shows the Cronbach's Alpha calculated for each construct.

Table 3.3: Phase 1 - Cronbach's Alpha: Constructs

Construct	Original α	Questions Removed	Adjusted α
Definition	0.61	D1, D3, D5	0.73
Classroom	0.71	-	0.71
Confidence	0.85	-	0.85
Interest	0.91	-	0.91
Usefulness	0.79	-	0.79

3.2.2 Internal Validity

In order to perform CFA, our first step was to validate our model by computing fit statistics. Initially, we computed CFI and TLI indices. Next, we calculated RMSEA and SRMR indices. In this analysis, we found $CFI = 0.80$, $TLI = 0.79$, $RMSEA = 0.079$ and $SRMR = 0.065$, suggesting that our model was adequate.

After validating the model, we performed the factor analysis for the complete instrument. For Confidence, Interest and Usefulness categories, all statements had loading factors greater than 0.30. In Classroom category, question E1 (0.29) had a loading factor just below the cut-off. Nevertheless, we kept this statement in our survey, in order to maintain Classroom's construct reliability at an acceptable level. In Definition construct, we found loading factors less than the threshold for D1 (0.06), D3 (0.07) and D5 (0.21), the same ones that led to poor reliability. Thus, we removed these questions from the final set of questions. Results computed in the initial version of the survey (Phase 1) are displayed in Table 4.3.

Chapter 4

Phase 2: Instrument Review

This chapter presents the methods and results of Phase 2. The Methodology section includes the description of scenario and participants of our second intervention, the instrument review and the processes of data collection and analysis. Results are described in the same terms of Phase 1.

4.1 Methodology

After applying the first version of the instrument, we still had questions to be better answered, particularly for some statements on the Definition and CT in Classroom categories: were these statements genuinely unreliable and invalid? Was it possible to improve them? To answer these questions and support the potential of our instrument, we applied its revised version with another large sample.

4.1.1 Scenario and Participants

We maintained the partnership with researchers from UNIVESP for another evaluation of the CT course, in the second semester of 2021. The course followed the same structure from the 2020 intervention: an eight-week online course designed for first-year undergraduate students.

The 2021 course was offered to nearly 8,000 first-year university students, and we obtained 1,939 answers. Their average age was 33.4 ± 9.5. The gender figures were 57.4% of male, 42.1% of female and 0.3% of others. Additionally, the students' major choice were: Computer Engineering (23%), Data Science (20%) and Information Technology (12%) from Computing Track; Education (20%), Language and Arts (10%) and Mathematics (9%) from Educational Track. Nearly 5% of students had not decided on their major by that time.

4.1.2 Instrument Review

The survey applied in Phase 2 was based on the survey from Phase 1. After the intervention in Phase 1, we tested our set of questions for reliability and validity, and observed three statements (D1, D3 and D5) that were decreasing the results of Definition in both analyses. In the Classroom construct, the statement E1 also had poor values. Finally, even though we found adequate values for E4, we were still not satisfied with its wording.

As we concluded that these statements might have not been properly phrased and have not expressed what they were meant to, they were redesigned in Phase 2. The questions in Confidence, Interest and Usefulness categories remained the same. The final version of the instrument can be seen in Table 4.2, and the redesigned questions are highlighted in bold. In addition, the Portuguese version of the survey is shown in Appendix B.

4.1.3 Data Collection and Analysis

The online survey was applied in the first week of the course. Similarly to the settings in Phase 1, we shuffled the question order, and the answer for each item ranged from 1 (Not True) to 5 (Very True). Also, we used Cronbach's Alpha to evaluate the reliability of the constructs and applied a Confirmatory Factor Analysis to examine the validity of the questions.

4.2 Results

Our results in Phase 2 are shown in terms of the instrument's reliability and internal validity.

4.2.1 Reliability

In Phase 2, we applied Cronbach's Alpha to estimate the reliability of each category in the reviewed version of the survey. In this phase, the Definition category (0.71) already had an α greater than 0.70 in the first analysis. Additionally, we also found acceptable α values in the set of questions in Classroom (0.77), Confidence (0.80), Interest (0.88) and Usefulness (0.75). Table 4.1 shows the Cronbach's Alpha calculated for each construct.

4.2.2 Internal Validity

Similarly to Phase 1, we calculated fit statistics to validate our model, using CFI, TLI, RMSEA and SRMR indices. In this analysis, we found $CFI = 0.80$, $TLI = 0.78$, $RMSEA = 0.069$ and $SRMR = 0.059$, which indicate that our model was adequate.

Table 4.1: Phase 2 - Cronbach's Alpha: Constructs

Construct	Cronbach's α
Definition	0.71
Classroom	0.77
Confidence	0.80
Interest	0.88
Usefulness	0.75

The next step was to perform the factor analysis in each question of the instrument. In Definition category, all questions had loading factors above the cut-off (0.30). Apart from D1 (0.33), all questions had values greater than 0.4. In the Classroom construct, each statement had a loading factor higher than 0.6, except for E6 (0.46). Furthermore, we found decent loading factors for the questions in Confidence, Interest and Usefulness. Table 4.3 displays the loading factor of each question in the final version (Phase 2) of the survey, compared to the values in Phase 1. The values in bold represent the results for the redesigned questions.

Table 4.2: Computational Thinking and Computing Attitudes Survey - Phase 2

Category	Statement
Definition	**D1. Computational Thinking is understanding how computer scientists work.**
Definition	D2. Computational Thinking involves thinking logically to solve problems.
Definition	**D3. Computational Thinking involves solving problems similarly to how computer scientists do.**
Definition	D4. Computational Thinking involves abstracting general principles and applying them to other situations.
Definition	**D5. Computational Thinking involves considering primarily the most relevant aspects of problems to design their solutions.**
Definition	D6. Computational Thinking involves breaking problems into smaller parts to facilitate their solution.
Definition	D7. Computational Thinking involves evaluating possible approaches analytically before making a decision.
Classroom	**E1. Computational Thinking can be incorporated in the classroom by using computer programming in the lesson plan.**
Classroom	E2. Computational Thinking can be incorporated in the classroom by allowing students to problem solve.
Classroom	E3. Computational Thinking can be incorporated in the classroom by developing apps.
Classroom	**E4. Computational Thinking can be incorporated in the classroom by using educational robotics concepts and practices.**
Classroom	E5. Computational Thinking can be incorporated in the classroom by creating games and animations.
Classroom	E6. Computational Thinking can be incorporated in the classroom by promoting activities to understand Computing without using a computer.
Confidence	C1. I am comfortable with learning Computing concepts.
Confidence	C2. I have little self-confidence when it comes to Computing courses.
Confidence	C3. I do not think that I can learn to understand Computing concepts.
Confidence	C4. I can learn to understand Computing concepts.
Confidence	C5. I can achieve good grades (C or better) in Computing courses.
Confidence	C6. I am confident that I can solve problems by using computer applications.
Confidence	C7. I am not comfortable with learning Computing concepts.
Confidence	C8. I doubt that I can solve problems by using computer applications.
Interest	I1. I would not take additional Computer Science courses if I were given the opportunity.
Interest	I2. I think Computer Science is boring.
Interest	I3. I hope that my future career will require the use of Computer Science concepts.
Interest	I4. The challenge of solving problems using Computer Science does not appeal to me.
Interest	I5. I like to use Computer Science to solve problems.
Interest	I6. I do not like using Computer Science to solve problems.
Interest	I7. The challenge of solving problems using Computer Science appeals to me.
Interest	I8. I hope that I can find a career that does not require the use of Computer Science concepts.
Interest	I9. I think Computer Science is interesting.
Interest	I10. I would voluntarily take additional Computer Science courses if I were given the opportunity.
Usefulness	U1. Developing Computing skills will not play a role in helping me achieve my career goals.
Usefulness	U2. Knowledge of Computing will allow me to secure a good job.
Usefulness	U3. My career goals do not require that I learn Computing skills.
Usefulness	U4. Developing Computing skills will be important to my career goals.
Usefulness	U5. Knowledge of Computing skills will not help me secure a good job.
Usefulness	U6. I expect that learning to use Computing skills will help me achieve my career goals.

Table 4.3: Phases 1 and 2 - Factor Analysis: Statements

Construct	Question	Loading Factor Phase 1	Loading Factor Phase 2
Definition	D1	0.068	**0.334**
Definition	D2	0.676	0.662
Definition	D3	0.079	**0.444**
Definition	D4	0.702	0.624
Definition	D5	0.211	**0.613**
Definition	D6	0.585	0.613
Definition	D7	0.569	0.596
Classroom	E1	0.293	**0.607**
Classroom	E2	0.781	0.63
Classroom	E3	0.475	0.642
Classroom	E4	0.343	**0.692**
Classroom	E5	0.425	0.629
Classroom	E6	0.475	0.466
Confidence	C1	0.812	0.764
Confidence	C2	0.552	0.514
Confidence	C3	0.47	0.451
Confidence	C4	0.765	0.712
Confidence	C5	0.698	0.653
Confidence	C6	0.755	0.711
Confidence	C7	0.565	0.515
Confidence	C8	0.432	0.413
Interest	I1	0.629	0.536
Interest	I2	0.74	0.655
Interest	I3	0.821	0.762
Interest	I4	0.687	0.611
Interest	I5	0.818	0.775
Interest	I6	0.665	0.624
Interest	I7	0.857	0.803
Interest	I8	0.63	0.586
Interest	I9	0.67	0.628
Interest	I10	0.705	0.604
Usefulness	U1	0.623	0.549
Usefulness	U2	0.518	0.536
Usefulness	U3	0.557	0.55
Usefulness	U4	0.793	0.791
Usefulness	U5	0.508	0.45
Usefulness	U6	0.67	0.635

Chapter 5

Discussion

In this chapter, we discuss our results in terms of our research questions. Then, we examine the threats to the validity of this work. In our analysis, obtaining a high rating (4 or 5) in the Definition of CT category means that the participant could effectively comprehend what is involved in Computational Thinking and its skills, whereas a good grade in CT in Classroom represents participant's potential ability to incorporate CT in classes in multiple ways. A high score in Confidence could indicate that the student feels comfortable when taking Computing courses, whilst in Interest it suggests that they are more likely to continue studying Computing as well as apply its concepts. Finally, a high rating in Usefulness could express the student's plan on following a professional career in the Computing field.

5.1 RQ1 – Questions on the Definition of CT

Answering RQ1, our statements covered the five CT skills identified by Selby (2014): Abstraction (D5), Algorithmic Design (D2), Decomposition (D6), Evaluation (D7) and Generalization (D4). From our analysis, we considered that the concept of Abstraction was not accurately represented in D5 in Phase 1, as it reduces CT to consider only the most relevant aspects of problems. In Phase 2, we rephrased this question to replace the word 'only' by 'primarily'. Thus, the loading factor in D5 in Phase 1 was 0.21, whereas this value was 0.61 in the final phase.

Moreover, we analyzed that D1 (0.06) and D3 (0.07) may have been understood by participants as a restrictive point of view in Phase 1, since they have been taught, in their CT course, that CT is more related to using the thought processes that a computer scientist would use (Wing, 2006), instead of understanding computers and using them to solve problems. In this sense, we revised both questions in Phase 2, associating CT to the practices and skills of computer scientists. In the second phase, D1 and D3 had loading factors of 0.33 and 0.44, respectively.

There have been various researchers discussing the definition of CT, with multiple and occasionally antagonistic points of view, such as Wing (2006) and Denning

(2017). As this work is aimed for educational purposes, we focused on designing questions which could cover the skills involved in CT, rather than emerging with a new definition of the topic. Therefore, we based our Definition questions on the work of Selby (2014). Results in Phase 2 indicate that our group of seven questions can capture these skills related to the comprehension of Computational Thinking.

5.2 RQ2 – Questions on the use of CT in Classroom

To answer RQ2, we designed questions to illustrate common approaches for teaching CT, according to Santos et al. (2018): teaching CT in the traditional way (E2, E3), applying robotics concepts and practices (E4), creating games and animations (E5), and developing activities in the context of unplugged computing (E6).

In the CFA performed in Phase 1, we found poor results in E1 (0.29) and E4 (0.34). Question E1 concerned the use of computers in classroom to promote CT. The students were taught in the CT course that using a computer in the classroom does not mean that CT is being promoted. Then, this question was rephrased in Phase 2, concerning the incorporation of computer programming in classroom as a way to promote CT. With this new phrasing, its loading factor was 0.60.

Question E4 regarded the use of robotics to teach CT. However, the students were not exposed to activities involving robots throughout the course, and may have misunderstood this statement. The question was also rephrased in Phase 2, regarding the use of educational robotics as a tool to incorporate CT in classroom. The loading factor in E4 was 0.69 in Phase 2. Furthermore, we found suitable loading factors for E2, E3, E5 and E6 statements in both phases.

Researchers have been working on strategies to promote Computing and CT in education with different contexts, such as the use of the unplugged computing (Bell et al., 2009), games (Barnes et al., 2007), medias (Guzdial, 2003) and block-based programming (Resnick et al., 2009). We consider that the revised questions in the Classroom category are suitable to capture some of these typical ways to teach CT in K-12 education.

5.3 RQ3 – Questions on the attitudes towards Computing

We answered RQ3 by verifying that the statements on Confidence, Interest and Usefulness are valid and reliable to measure participants' Computing attitudes. We computed positive results for all categories: In Phase 1, the α values were above 0.79 for reliability and the loading factors were greater than 0.51 in each statement in the CFA. In Phase 2, the α figures were greater than 0.75 and the loading factors were higher than 0.41. These values are consonant with the ones found in Hoegh and

Moskal (2009)'s work. Thus, this set of questions remains as a potentially suitable tool to measure students' attitudes towards Computing.

The scientific community has been interested in analyzing students perceptions of Computing, such as in Wanzer et al. (2019), Schulte and Knobelsdorf (2007) and Funke et al. (2016). Using Hoegh and Moskal (2009)'s survey on Computing attitudes and finding positive results may support their work and disseminate the relevance of this topic for Computing Education.

Even though the teaching of Computational Thinking and the analysis of students' attitudes towards Computing are correlated, these topics have not usually been investigated together. Designing and evaluating an instrument on both CT and Computing attitudes turned out to be pertinent, as we could capture students' understanding of the subject as well as their perceptions on the field.

5.4 Threats to Validity

We consider that the audience is a relevant factor in our study. Offering this course to an audience which included not only pre-service teachers could be a threat to the validity of this work. However, we applied the instrument with 2,290 undergraduate students in Phase 1 and 1,939 students in Phase 2. Hence, the survey was used by a large number of potentially qualified participants, from both the Educational and Computing tracks. We also consider that offering the course to an audience with different backgrounds and major preferences was positive. The participants are likely to have distinct interests and perceptions about Computing and CT, allowing us to obtain a wider range of beliefs on both fields.

Concerning external validity, the evaluation of the instrument was performed at a large university that offers distance education. Our audience consisted of first-year students, who are typically not biased towards their fields of study. Results from this study may neither be generalized to every undergraduate audience nor to in-service teachers. Nevertheless, we tried to mitigate this validity threat by using a large sample of undergraduate students from different majors and different locations, which have all been exposed to concepts of CT and Computing during one academic term.

Concerning internal validity threats, our study aimed to verify both the reliability and the validity of the designed survey, applying acknowledged statistics and psychometric methods for these. In addition, the survey design was based on other previously validated studies. Thus, our results indicate that the instrument is both statistically reliable and valid.

Moreover, the fact that the participation in the survey was voluntary could contribute to a self-selection of the audience. Nonetheless, we consider that our sample was large enough to capture not only the students with positive attitudes towards

CT, but also those who did not feel comfortable with the course and the field of Computing in general.

The instrument was applied in different stages of the course in each phase: in Phase 1, it was applied right after the course; in Phase 2, it was applied before the intervention. When we established the partnership to evaluate the CT course in 2020, it had already started. Thus, we could only apply our instrument after the intervention. However, since some students in Phase 2 had previous experience with computing, while other did not, there was a potential variability in their pre-intervention attitudes towards computing and their implicit perceptions of CT. For such, we applied the instrument before the course in Phase 2.

Currently, there is not an universal definition of CT. Thus, the wording of some questions might have not grasped exactly what they were meant to measure. However, we designed our set of questions based on instruments previously validated, namely Yadav et al. (2014) and Hoegh and Moskal (2009). In addition, our statements about Definition of CT relied on acclaimed definitions from the literature, such as Selby (2014). For CT in Classroom construct, we considered the most common approaches used to teach CT and Programming, including those reviewed by Santos et al. (2018). Also, we asked an expert panel to analyze and suggest improvements to the survey before its application in Phase 1. Finally, we revised its wording after the first application, and used the improved version of the survey in Phase 2. Our results indicate that the final instrument provides both reliability and validity for the five intended constructs.

Chapter 6

Conclusions

In this work, we designed and evaluated a survey instrument on Computational Thinking and attitudes towards the field of Computing. We relied the instrument design on the studies of Yadav et al. (2014), Hoegh and Moskal (2009), Selby (2014) and Santos et al. (2018). To evaluate the instrument, we established a partnership with researchers from a large public university in Brazil, which offers a CT course to undergraduate students from six different majors.

Our results were significantly positive. After applying Cronbach's Alpha to calculate the reliability of the constructs, we found α values greater than 0.70 for all constructs. Moreover, we performed a Confirmatory Factor Analysis to examine the internal validity of each question, finding loading factors above 0.30 for all questions. Thus, our outcomes indicate that the final version of this work shows preliminary evidence towards statistical reliability and internal validity of the instrument.

We consider that the main contributions of this work are: 1) the design of a survey instrument with 37 questions to capture five dimensions associated with Computational Thinking and Computing attitudes; 2) the evaluation of the instrument with two large and potentially qualified samples.

Researchers interested in evaluating interventions on Computational Thinking for educational purposes may reuse this instrument, as it enables to capture both the dimensions of comprehension of CT, the application of the subject in school, and students' perceptions on Computing. This instrument may be considered valuable for the scientific community, as it was statistically validated in two phases, covering the latest definitions and applications of CT in education, and allowing its use in different scenarios.

As future work, we suggest the application of the instrument in educational contexts. This could be done in different settings, such as the assessment of groups receiving different types of instruction, or an evaluation in a pre- and post-intervention format. Additionally, we consider that other instruments may be designed based on our work, improving the question wording or including other CT categories besides “Definition” and “Classroom”. Finally, to increase its external validity, our instrument may be reevaluated with other audiences, such as in-service teachers or high school students.

References

Barnes, T., Richter, H., Powell, E., Chaffin, A., and Godwin, A. (2007). Game2learn: building cs1 learning games for retention. In *Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education*, pages 121–125.

Barr, V. and Stephenson, C. (2011). Bringing computational thinking to k-12: what is involved and what is the role of the computer science education community? *Acm Inroads*, 2(1):48–54.

Bell, T., Alexander, J., Freeman, I., and Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, 13(1):20–29.

Bentler, P. M. (1990). Comparative fit indexes in structural models. *Psychological bulletin*, 107(2):238.

Bockmon, R., Cooper, S., Gratch, J., and Dorodchi, M. (2020). Validating a cs attitudes instrument. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, pages 899–904.

Brown, N. C., Sentance, S., Crick, T., and Humphreys, S. (2014). Restart: The resurgence of computer science in uk schools. *ACM Transactions on Computing Education (TOCE)*, 14(2):1–22.

Crick, T. and Sentance, S. (2011). Computing at school: stimulating computing education in the uk. In *Proceedings of the 11th Koli Calling International Conference on Computing Education Research*, pages 122–123.

Curzon, P., Bell, T., Waite, J., and Dorling, M. (2019). *Computational Thinking*, pages 513–546. Cambridge Handbooks in Psychology. Cambridge University Press.

Cutumisu, M., Adams, C., and Lu, C. (2019). A scoping review of empirical research on recent computational thinking assessments. *Journal of Science Education and Technology*, 28(6):651–676.

Dagiene, V. and Stupuriene, G. (2016). Bebras–a sustainable community building model for the concept based learning of informatics and computational thinking. *Informatics in education*, 15(1):25–44.

Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, 60(6):33–39.

Dewey, J. (1913). *Interest and effort in education*. Houghton Mifflin.

Dorn, B. and Elliott Tew, A. (2015). Empirical validation and application of the computing attitudes survey. *Computer Science Education*, 25(1):1–36.

Eagly, A. H. and Chaiken, S. (1998). Attitude structure and function.

Falkner, K., Vivian, R., and Williams, S.-A. (2018). An ecosystem approach to teacher professional development within computer science. *Computer Science Education*, 28(4):303–344.

Fennema, E. and Sherman, J. A. (1976). Fennema-sherman mathematics attitudes scales: Instruments designed to measure attitudes toward the learning of mathematics by females and males. *Journal for research in Mathematics Education*, 7(5):324–326.

Funke, A., Berges, M., and Hubwieser, P. (2016). Different perceptions of computer science. In *2016 International Conference on Learning and Teaching in Computing and Engineering (LaTICE)*, pages 14–18. IEEE.

Gal-Ezer, J. and Stephenson, C. (2014). A tale of two countries: Successes and challenges in k-12 computer science education in israel and the united states. *ACM Transactions on Computing Education (TOCE)*, 14(2):1–18.

Gomes, A. J., Santos, A. N., and Mendes, A. J. (2012). A study on students' behaviours and attitudes towards learning to program. In *Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education*, pages 132–137.

Grover, S. and Pea, R. (2013). Computational thinking in k–12: A review of the state of the field. *Educational researcher*, 42(1):38–43.

Guzdial, M. (2003). A media computation course for non-majors. In *Proceedings of the 8th annual conference on Innovation and technology in computer science education*, pages 104–108.

Heersink, D. and Moskal, B. M. (2010). Measuring high school students' attitudes toward computing. In *Proceedings of the 41st ACM technical symposium on Computer science education*, pages 446–450.

Hildebrandt, C. and Diethelm, I. (2012). The school experiment intech: how to influence interest, self-concept of ability in informatics and vocational orientation. In *Proceedings of the 7th Workshop in Primary and Secondary Computing Education*, pages 30–39.

Hoegh, A. and Moskal, B. M. (2009). Examining science and engineering students' attitudes toward computer science. In *2009 39th IEEE Frontiers in Education Conference*, pages 1–6. IEEE.

Hsu, Y.-C., Baldwin, S., and Ching, Y.-H. (2017). Learning through making and maker education. *TechTrends*, 61(6):589–594.

Jocius, R., Joshi, D., Albert, J., Barnes, T., Robinson, R., Cateté, V., Dong, Y., Blanton, M., O'Byrne, I., and Andrews, A. (2021). The virtual pivot: Transitioning computational thinking pd for middle and high school content area teachers. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, pages 1198–1204.

Ko, A. J. (2009). Attitudes and self-efficacy in young adults' computing autobiographies. In *2009 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 67–74. IEEE.

León, J. M., Robles, G., González, M. R., and Rodríguez, J. D. (2019). Not the same: a text network analysis on computational thinking definitions to study its relationship with computer programming. *Revista Interuniversitaria de Investigación en Tecnología Educativa*, (7):26–35.

Leonard, J., Mitchell, M., Barnes-Johnson, J., Unertl, A., Outka-Hill, J., Robinson, R., and Hester-Croff, C. (2018). Preparing teachers to engage rural students in computational thinking through robotics, game design, and culturally responsive teaching. *Journal of Teacher Education*, 69(4):386–407.

Linhares, A. C. O. and Santos, K. S. (2021). A licenciatura em computação no brasil: histórica e contexto atual. *Revista Brasileira de Informática na Educação*, 29:188–208.

Lorenceau, A., Marec, C., and Mostafa, T. (2019). Upgrading the ICT questionnaire items in PISA 2021. (202).

Luxton-Reilly, A., Albluwi, I., Becker, B. A., Giannakos, M., Kumar, A. N., Ott, L., Paterson, J., Scott, M. J., Sheard, J., and Szabo, C. (2018). Introductory programming: a systematic literature review. In *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, pages 55–106.

Magerko, B., Freeman, J., Mcklin, T., Reilly, M., Livingston, E., Mccoid, S., and Crews-Brown, A. (2016). Earsketch: A steam-based approach for underrepresented populations in high school computer science education. *ACM Transactions on Computing Education (TOCE)*, 16(4):1–25.

Menekse, M. (2015). Computer science teacher professional development in the united states: a review of studies published between 2004 and 2014. *Computer Science Education*, 25(4):325–350.

Moreno-León, J., Robles, G., and Román-González, M. (2015). Dr. scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *RED. Revista de Educación a Distancia*, (46):1–23.

Mouza, C., Yang, H., Pan, Y.-C., Ozden, S. Y., and Pollock, L. (2017). Resetting educational technology coursework for pre-service teachers: A computational thinking approach to the development of technological pedagogical content knowledge (tpack). *Australasian Journal of Educational Technology*, 33(3).

Ni, L., Bausch, G., and Benjamin, R. (2021). Computer science teacher professional development and professional learning communities: a review of the research literature. *Computer Science Education*, pages 1–32.

Oliveira, E. C., Bittencourt, R. A., and Trindade, R. P. (2019). Introduction to computational thinking for k-12 educators through distance learning. In *2019 IEEE Frontiers in Education Conference (FIE)*, pages 1–9. IEEE.

O'Rourke, N. and Hatcher, L. (2013). *A step-by-step approach to using SAS for factor analysis and structural equation modeling.* Sas Institute.

Pears, A., Tedre, M., Valtonen, T., and Vartiainen, H. (2021). What makes computational thinking so troublesome?

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., et al. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11):60–67.

Román-González, M., Pérez-González, J.-C., and Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? criterion validity of the computational thinking test. *Computers in human behavior*, 72:678–691.

Santana, B. L., Chavez, C. v. F. G., and Bittencourt, R. A. (2020). Uma proposta de avaliação de conceitos, práticas e perspectivas de pensamento computacional. In *Anais do V Congresso sobre Tecnologias na Educação*, pages 158–167. SBC.

Santos, P. S., Araujo, L. G. J., and Bittencourt, R. A. (2018). A Mapping Study of Computational Thinking and Programming in Brazilian K-12 Education. In *2018 IEEE Frontiers in Education Conference (FIE)*, pages 1–8. IEEE.

Schulte, C. and Knobelsdorf, M. (2007). Attitudes towards computer science-computing experiences as a starting point and barrier to computer science. In *Proceedings of the third international workshop on Computing education research*, pages 27–38.

Selby, C. C. (2014). *How can the teaching of programming be used to enhance computational thinking skills?* PhD thesis, University of Southampton.

Tavares, J., Almeida, L., Vasconcelos, R., and Bessa, J. (2004). Inventário de atitudes e comportamentos habituais de estudo–iache. *Universidade de Aveiro/Universidade do Minho, Centro de Investigação em Educação, Portugal.*

Thompson, B. (2004). *Exploratory and confirmatory factor analysis.* American Psychological Association.

Wanzer, D., McKlin, T., Edwards, D., Freeman, J., and Magerko, B. (2019). Assessing the attitudes towards computing scale: A survey validation study. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, pages 859–865.

Wiebe, E., Williams, L. A., Yang, K., and Miller, C. S. (2003). Computer science attitude survey. Technical report, North Carolina State University. Dept. of Computer Science.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., and Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education (TOCE)*, 14(1):1–16.

Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S., and Korb, J. T. (2011). Introducing computational thinking in education courses. In *Proceedings of the 42nd ACM technical symposium on Computer science education*, pages 465–470.

Zhong, B., Wang, Q., Chen, J., and Li, Y. (2016). An exploration of three-dimensional integrated assessment for computational thinking. *Journal of Educational Computing Research*, 53(4):562–590.

Zorzo, A. F., Nunes, D., Matos, E., Steinmacher, I., de Araujo, R. M., Correia, R., and Martins, S. (2017). Referenciais de formação para os cursos de graduação em computação.

Appendix A

Phase 1 Instrument - Portuguese

Table A.1: Instrumento - Versão Preliminar em Português

Categoria	Frase
Definição	D1. Pensamento Computacional é compreender como os computadores funcionam.
Definição	D2. Pensamento Computacional envolve pensar logicamente para resolver problemas.
Definição	D3. Pensamento Computacional envolve usar computadores para resolver problemas.
Definição	D4. Pensamento Computacional envolve extrair princípios gerais sobre soluções de problemas e aplicá-los a outras situações.
Definição	D5. Pensamento Computacional envolve considerar apenas os elementos mais relevantes dos problemas para projetar soluções.
Definição	D6. Pensamento Computacional envolve decompor problemas em partes menores para facilitar suas resoluções.
Definição	D7. Pensamento Computacional envolve analisar criticamente possíveis abordagens antes de tomar uma decisão.
Escola	E1. Pensamento Computacional pode ser incorporado nas aulas através do uso de computadores.
Escola	E2. Pensamento Computacional pode ser incorporado nas aulas através do incentivo à resolução de problemas.
Escola	E3. Pensamento Computacional pode ser incorporado nas aulas através da criação de aplicativos (apps).
Escola	E4. Pensamento Computacional pode ser incorporado nas aulas através do uso de robôs.
Escola	E5. Pensamento Computacional pode ser incorporado nas aulas através da criação de jogos ou animações digitais.
Escola	E6. Pensamento Computacional pode ser incorporado nas aulas através de atividades de compreensão da computação sem o uso de computadores.
Confiança	C1. Eu me sinto confortável em aprender conceitos de computação.
Confiança	C2. Eu tenho pouca autoconfiança quando se trata de disciplinas de computação.
Confiança	C3. Eu não acho que eu possa aprender conceitos de computação.
Confiança	C4. Eu posso aprender conceitos de computação.
Confiança	C5. Eu posso conseguir boas notas (5,0 ou mais) em disciplinas de computação.
Confiança	C6. Estou confiante de que posso resolver problemas criando programas de computador.
Confiança	C7. Não me sinto confortável em aprender conceitos de computação.
Confiança	C8. Eu duvido que eu possa resolver problemas criando programas de computador.
Interesse	I1. Eu não faria disciplinas adicionais de computação se me dessem a oportunidade.
Interesse	I2. Eu acho que a computação é chata.
Interesse	I3. Espero que minha carreira futura exija o uso de conceitos de computação.
Interesse	I4. O desafio de resolver problemas usando computação não me atrai.
Interesse	I5. Eu gosto de usar a computação para resolver problemas.
Interesse	I6. Eu não gosto de usar computação para resolver problemas.
Interesse	I7. O desafio de resolver problemas usando a computação me atrai.
Interesse	I8. Espero encontrar uma carreira que não exija o uso de conceitos de computação.
Interesse	I9. Eu acho que a computação é interessante.
Interesse	I10. Eu participaria voluntariamente de disciplinas adicionais de computação se me fosse dada a oportunidade.
Utilidade	U1. Desenvolver habilidades de computação não terá um papel em me ajudar a alcançar meus objetivos de carreira.
Utilidade	U2. O conhecimento de computação me permitirá garantir um bom trabalho.
Utilidade	U3. Meus objetivos de carreira não exigem que eu aprenda habilidades de computação.
Utilidade	U4. Desenvolver habilidades de computação será importante para meus objetivos de carreira.
Utilidade	U5. O domínio das habilidades de computação não me ajudará a garantir um bom emprego.
Utilidade	U6. Eu espero que aprender a usar habilidades de computação me ajude a alcançar meus objetivos de carreira.

Appendix B

Phase 2 Instrument - Portuguese

Table B.1: Instrumento - Versão Final em Português

Categoria	Frase
Definição	D1. Pensamento Computacional envolve compreender como os profissionais de Computação trabalham.
Definição	D2. Pensamento Computacional envolve pensar logicamente para resolver problemas.
Definição	D3. Pensamento Computacional envolve resolver problemas de forma similar à que profissionais de Computação fazem.
Definição	D4. Pensamento Computacional envolve extrair princípios gerais sobre soluções de problemas e aplicá-los a outras situações.
Definição	D5. Pensamento Computacional envolve considerar sobretudo os elementos mais relevantes dos problemas para projetar soluções.
Definição	D6. Pensamento Computacional envolve decompor problemas em partes menores para facilitar suas resoluções.
Definição	D7. Pensamento Computacional envolve analisar criticamente possíveis abordagens antes de tomar uma decisão.
Escola	E1. Pensamento Computacional pode ser incorporado nas aulas através da programação de computadores.
Escola	E2. Pensamento Computacional pode ser incorporado nas aulas através do incentivo à resolução de problemas.
Escola	E3. Pensamento Computacional pode ser incorporado nas aulas através da criação de aplicativos (apps).
Escola	E4. Pensamento Computacional pode ser incorporado nas aulas através do uso da robótica educacional.
Escola	E5. Pensamento Computacional pode ser incorporado nas aulas através da criação de jogos ou animações digitais.
Escola	E6. Pensamento Computacional pode ser incorporado nas aulas através de atividades de compreensão da computação sem o uso de computadores.
Confiança	C1. Eu me sinto confortável em aprender conceitos de computação.
Confiança	C2. Eu tenho pouca autoconfiança quando se trata de disciplinas de computação.
Confiança	C3. Eu não acho que eu possa aprender conceitos de computação.
Confiança	C4. Eu posso aprender conceitos de computação.
Confiança	C5. Eu posso conseguir boas notas (5,0 ou mais) em disciplinas de computação.
Confiança	C6. Estou confiante de que posso resolver problemas criando programas de computador.
Confiança	C7. Não me sinto confortável em aprender conceitos de computação.
Confiança	C8. Eu duvido que eu possa resolver problemas criando programas de computador.
Interesse	I1. Eu não faria disciplinas adicionais de computação se me dessem a oportunidade.
Interesse	I2. Eu acho que a computação é chata.
Interesse	I3. Espero que minha carreira futura exija o uso de conceitos de computação.
Interesse	I4. O desafio de resolver problemas usando computação não me atrai.
Interesse	I5. Eu gosto de usar a computação para resolver problemas.
Interesse	I6. Eu não gosto de usar computação para resolver problemas.
Interesse	I7. O desafio de resolver problemas usando a computação me atrai.
Interesse	I8. Espero encontrar uma carreira que não exija o uso de conceitos de computação.
Interesse	I9. Eu acho que a computação é interessante.
Interesse	I10. Eu participaria voluntariamente de disciplinas adicionais de computação se me fosse dada a oportunidade.
Utilidade	U1. Desenvolver habilidades de computação não terá um papel em me ajudar a alcançar meus objetivos de carreira.
Utilidade	U2. O conhecimento de computação me permitirá garantir um bom trabalho.
Utilidade	U3. Meus objetivos de carreira não exigem que eu aprenda habilidades de computação.
Utilidade	U4. Desenvolver habilidades de computação será importante para meus objetivos de carreira.
Utilidade	U5. O domínio das habilidades de computação não me ajudará a garantir um bom emprego.
Utilidade	U6. Eu espero que aprender a usar habilidades de computação me ajude a alcançar meus objetivos de carreira.

Appendix C

List of Relevant Terms Translated to Portuguese

English	**Portuguese**
K-12 Education	Educação Básica
In-service Teacher	Docente em Atividade
Pre-service Teacher	Estudante de Licenciatura
Professional Development	Qualificação Docente