

Universidade Federal do Estado do Rio de Janeiro  
Centro de Ciências Exatas e Tecnologia  
Escola de Informática Aplicada

PROGRAME-SE: O PENSAMENTO COMPUTACIONAL NA EDUCAÇÃO BÁSICA

Livia Costa Pereira

**Orientador**

Sean Wolfgang Matsui Siqueira

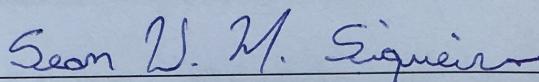
Rio de Janeiro, RJ – Brasil  
Dezembro de 2016

# PROGRAME-SE: O PENSAMENTO COMPUTACIONAL NA EDUCAÇÃO BÁSICA

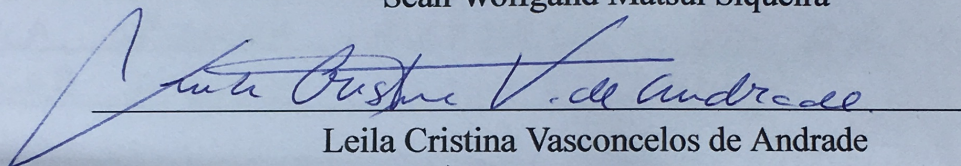
Lívia Costa Pereira

Projeto de Graduação apresentado à Escola de Informática Aplicada da Universidade Federal do Estado do Rio de Janeiro (UNIRIO) para obtenção do título de Bacharel em Sistemas de Informação.

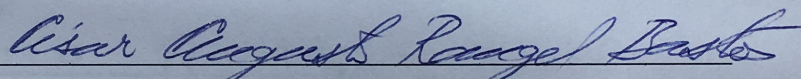
Aprovada por:



Sean Wolfgang Matsui Siqueira



Leila Cristina Vasconcelos de Andrade



César Augusto Rangel Bastos

Rio de Janeiro, RJ – Brasil.

Dezembro de 2016

*“Ninguém caminha sem aprender a caminhar, sem aprender a fazer o caminho caminhando, refazendo e retocando o sonho pelo qual se pôs a caminhar”.*

(Paulo Freire)

Dedico este projeto ao meu grande amor e companheiro de viagem, Felipe.

## Agradecimentos

A Deus, pelas oportunidades que a vida me trouxe e traz, pela saúde, intuições e aprendizado que foram fundamentais ao longo de todos os longos anos de estudo que me permitiram chegar até aqui.

Ao meu grande companheiro Felipe, pelo amor, paciência, e incentivo ao longo de todo o processo.

Aos meus pais Alice e Rubem, pela vida, pelo apoio, preocupação e incentivo ao estudo.

Ao avô Salomão (*In memoriam*) pelo carinho eternizado e pela referência que foi e é na minha dedicação aos estudos.

Aos amigos Kim, Larissa e Bruno, pela amizade, o apoio e carinho de sempre.

Aos amigos Natália e Gabriel, pela amizade, carinho e finais de semana de recarga de energia.

À amiga Marcia Arissawa, pelo carinho e sabedoria compartilhada, essenciais durante a jornada de graduação.

Ao professor Sean, por aceitar me orientar nesta etapa, pela atenção, pelas dicas valiosas e dedicação durante todo o processo.

Aos professores Ismênia, Julio e Victor da Escola Modelar Cambaúba, por toda a disponibilidade e atenção em me receber em suas aulas, e pela oportunidade que tive em vivenciar na prática o trabalho do Pensamento Computacional na educação básica. Ao professor Charles Pimentel da Escola SESC, pela indicação da visita à Escola Modelar Cambaúba, por toda a atenção e disponibilidade.

Ao professor César Bastos, pela oportunidade de ver na prática uma experiência que rema contra a maré da crise educacional brasileira em que estudantes voluntariamente ficam após o horário para poder aprender um pouco de robótica pedagógica. Por toda a disponibilidade e atenção dispendida na visita de sua aula na FAETEC Ferreira Viana.

Às professoras Leila e Geiza, pelo incentivo contínuo a “continuar a nadar”.

Aos colegas Maria das Graças, Daiane, Felipe, Leonardo, William, Gabriel, Cláudio, César, Caroline e Bianca. Pelos trabalhos, aflições, risos e alívios compartilhados.

Aos professores Márcio Barros e Leonardo Azevedo, cujos desafios enriqueceram-me de coragem e disposição para prosseguir no desenvolvimento de *software*. Ao professor Gladson Antunes, por tornar Cálculo 2 simples e prazeroso, e mostrar tantas aplicações práticas. Ao professor Celso Campos, por tamanha visão, sabedoria e análise (visionária) política, econômica e social transmitida. Ao Douglas Brito, por tornar as burocracias do dia a dia acadêmico sempre tranquilas com sua calma e atenção, pelos bolinhos e lanchinhos que foram fundamentais na hora do aperto.

À todos os professores que se esforçam em manter a educação pública de qualidade, apesar de todas as dificuldades.

## RESUMO

A necessidade de reformulação do ensino básico tradicional é fato aceito por grande parte da comunidade acadêmica internacional. O modelo de ensino largamente utilizado, onde alunos perfazem papéis de agentes passivos e receptores de “conhecimento” em única direção (do professor para o aluno), sem interferir no processo e apenas aceitando conteúdos e informações, traz resultados a curto, médio e longo prazo que prejudicam a formação integral dos estudantes ao longo do processo. Pensando na necessidade de reformulação da educação básica e em como ela é concebida, muitos defendem a idéia de inserir e explorar o Pensamento Computacional na educação básica, como uma das diversas mudanças que poderiam auxiliar no maior desenvolvimento dos estudantes enquanto seres pensantes, agentes questionadores e transformadores do meio social. Considerando estas questões, o presente trabalho visa poder contribuir na divulgação da idéia de inclusão do Pensamento Computacional na educação básica brasileira, trazendo uma discussão detalhada do assunto “Pensamento Computacional na Educação Básica” tratando do que é e o que não é Pensamento Computacional, das iniciativas já realizadas, abordagens, benefícios e desafios, entre outros. O trabalho também propõe uma plataforma disponível na web que objetiva a divulgação e consulta de objetos de aprendizagem que trabalhem o desenvolvimento do pensamento computacional na educação básica.

**Palavras-chave:** Pensamento Computacional, Informática na Educação, objeto de aprendizagem

## ABSTRACT

*The need for reformulation of traditional basic education (k-12) is a fact accepted by many in the international academic community. The widely used teaching model, where students play roles of passive agents and recipients of “knowledge” in a single direction (from teacher to student), without interfering in the process and only accepting content and information, brings short, medium and long-term results that undermine the integral education of students throughout the process. Thinking about the need to reformulate basic education and how it is conceived, many advocate the idea of adding and exploring Computational Thinking in basic education as one of the many changes that could aid in the further development of students as thinking beings, questioning agents, and changers of the social environment. Considering these questions, we aim to contribute to the dissemination of the idea of inclusion of Computational Thinking in Brazilian basic education, bringing a detailed discussion of the subject "Computational Thinking in Basic Education" dealing with what is and what is not Computational Thinking, initiatives already undertaken, approaches, benefits and challenges, etcetera. We also propose a platform that aims at the dissemination and search of learning objects that work the development of computational thinking in basic education.*

**Keywords:** Computational Thinking, Education Informatics, Learning Objects.

## Índice

<b>1.</b>	<b>Introdução</b>	
1.1.	Motivação.....	8
1.2.	Objetivos .....	10
1.3.	Organização do texto .....	10
<b>2.</b>	<b>O Pensamento Computacional</b>	
2.1.	O que é Pensamento Computacional ?.....	11
2.2.	Por quê estimular o Pensamento Computacional na ..... educação básica ?	14
2.3.	Experiências concretizadas.....	16
2.4.	Benefícios do desenvolvimento do Pensamento ..... Computacional	19
2.5.	O “Pensar Computacional” como meio de ..... transformação social	23
2.6.	Como abordar ?.....	23
2.7.	Desafios da Implantação do Pensamento Computacional na Educação Básica...29	
<b>3.</b>	<b>Proposta de Trabalho</b>	
3.1.	Objeto de aprendizagem e SCORM.....	31
3.2.	Requisitos, Modelos e Ferramentas Utilizadas.....	33
<b>4.</b>	<b>Um <i>overview</i> sobre a plataforma.....</b>	<b>42</b>
<b>5.</b>	<b>Conclusão</b>	
5.1.	Considerações finais.....	53
5.2.	Limitações do projeto.....	54
5.3.	Possibilidades de trabalhos futuros.....	54
<b>6.</b>	<b>Referências</b>	



## Índice de imagens

Figura 1: Diagrama de Casos de Uso da plataforma “Programe-se” .....	33
Figura 2 - Modelo de classes da plataforma Programe-se.....	37
Figura 3: Modelo Lógico da plataforma Programe-se .....	39
Figura 4: Classes da plataforma Programe-se criadas no padrão MVC.....	40
Figura 5: Tela de Login e Cadastro de novo Usuário da plataforma Programe-se .....	41
Figura 6: Tela de Cadastro da plataforma Programe-se .....	42
Figura 7: Cadastro de Usuário na plataforma Programe-se .....	43
Figura 8: Usuário Cadastrado e com dados inseridos para se logar.....	44
Figura 9: Tela principal da plataforma Programe-se .....	45
Figura 10: Tela de Cadastro da plataforma Programe-se com dados do objeto preenchido...	46
Figura 11: Tela de Cadastro com dados do objeto preenchido.....	47
Figura 12: Tela de confirmação do cadastro do objeto de aprendizagem na plataforma Programe-se.....	47
Figura 13: Tela principal da com novo objeto cadastrado.....	48
Figura 14: Detalhes do objeto cadastrado.....	49
Figura 15: Detalhes do objeto cadastrado.....	49
Figura 16: objeto de aprendizagem salvo no formato .txt.....	50
Figura 17: Filtro de objeto de aprendizagem.....	51
Figura 18: Filtro de objeto de aprendizagem .....	51



# 1. Introdução

Este capítulo apresenta o tema Pensamento Computacional e as questões que motivam e justificam a sua inclusão na educação básica, tratando os objetivos gerais e específicos, metodologia utilizada e a estruturação do texto ao longo do documento.

## 1.1. Motivação

A necessidade de reformulação do ensino básico tradicional é fato aceito por grande parte da comunidade acadêmica internacional (Brasil, 2000). O modelo de ensino largamente utilizado, onde alunos perfazem papéis de agentes passivos e receptores de “conhecimento” em única direção (do professor para o aluno), sem interferir no processo e apenas aceitando conteúdos e informações, traz como resultados a curto prazo pouco interesse e engajamento por parte dos estudantes no processo de aprendizagem, falta de perspectiva de como e aonde aplicar o conhecimento, aversão a determinadas áreas de conhecimento (geralmente as que o aluno enfrenta maiores dificuldades no aprendizado), entre outros. A médio prazo, é possível observar deficiências em lógica e pouco desenvolvimento da criatividade, devido a esse modelo de educação privilegiar a memorização (Carvalho et al., 2015). Já a longo prazo, estudantes prejudicados por esse modelo tradicional de ensino-aprendizagem na educação básica podem vir a ter analfabetismo funcional, dificuldades de organização lógica de ideias e, por consequência, de comunicação clara e no ingresso e/ou acompanhamento de um curso de nível superior, na argumentação e sustentação de ideias e pontos de vista, seja no âmbito pessoal ou profissional, entre outros. Pensando na necessidade de reformulação da educação básica e em como ela é concebida, pesquisadores defendem a ideia de inserir e explorar o Pensamento Computacional na educação básica, como uma das diversas mudanças que poderiam auxiliar no maior desenvolvimento dos estudantes enquanto seres pensantes, agentes questionadores e transformadores do meio social.(Araújo et al., 2015 ;Cardoso & Antonello, 2015; França & Tesdeco, 2015; Pimentel et al., 2015 ; Raabe et al., 2015)

Quando o computador foi inserido nas escolas na década de 90, a proposta era a de torná-lo familiar aos estudantes e objeto de uso para comunicação, informação e entretenimento. Porém, tal finalidade já não contempla mais as necessidades atuais da sociedade (Ramos, 2014). Atualmente, observa-se certo domínio das sociedades no uso das tecnologias, especialmente pela gerações que desde a infância já se utilizam de smartphones, tablets, internet, televisão e videogame. O uso excessivo dessas tecnologias por crianças e adolescentes pode estar relacionado com atrasos cognitivos, dificuldades de aprendizagem, impulsividade e problemas em lidar com sentimentos de raiva, além de obesidade, privação do sono e risco de dependência por tecnologia (Linn, 2004). Em contrapartida, o contato e o entendimento do *backstage* dessas tecnologias, ou seja, a compreensão do funcionamento a nível de *software* das tecnologias e habilidades envolvidas na construção destas, pode favorecer o desenvolvimento de habilidades que vão muito além do âmbito acadêmico, como o desenvolvimento do raciocínio lógico, de confiança na resolução de problemas e em lidar com a complexidade, de persistência no trabalho de problemas difíceis, de tolerância a ambiguidades e incertezas, da capacidade de comunicação e trabalho em equipe, de incentivo ao trabalho colaborativo, entre outros (Ramos, 2014).

Há relatos de iniciativas de trabalho do Pensamento Computacional na educação básica em escolas no Brasil e em diversos outros países (Aguilhar, 2013 ; Pimentel et al., 2015). Estas diferem-se entre si em vários aspectos como a forma de abordar o Pensamento Computacional, a tecnologia utilizada na abordagem, os recursos utilizados, a faixa etária dos estudantes participantes da abordagem, entre outros. Tais iniciativas ainda estão pouco presente nas escolas brasileiras, especialmente nas públicas onde recursos são mais escassos, seja por desconhecimento da iniciativa por parte dos educadores ou de como seria possível implementá-las nas escolas.

## **1.2. Objetivos**

O Objetivo principal deste trabalho é contribuir na divulgação da idéia de inclusão do Pensamento Computacional na educação básica brasileira, bem como promover a colaboração de iniciativas através da plataforma “Programe-se”. Desta forma, o presente trabalho faz uma discussão detalhada do assunto “Pensamento Computacional na Educação Básica”, tratando de definir o que é e o que não é Pensamento Computacional, das iniciativas já realizadas, formas de abordagem, benefícios e desafios. O trabalho também propõe uma plataforma que objetiva a divulgação e consulta de objetos de aprendizagem que trabalhem o desenvolvimento do pensamento computacional na educação básica. Espera-se que ao final deste trabalho o leitor compreenda o que é e o que não é pensamento computacional, os benefícios e desafios de sua inclusão na educação básica, formas de abordar, entre outros. Espera-se também que o trabalho incentive iniciativas de abordagem do pensamento computacional na educação básica e promova a colaboração das iniciativas através da plataforma “Programe-se”, desenvolvida neste trabalho.

## **1.3. Organização do Texto**

O presente trabalho está estruturado em capítulos da seguinte forma:

- O capítulo 2 contém uma revisão bibliográfica em torno de: definir o que é e o que não é pensamento computacional, experiências concretas de trabalho do pensamento computacional em salas de aula, benefícios da abordagem, o potencial de transformação social e formas de abordar.
- O capítulo 3 contém a descrição dos requisitos e o modelo conceitual do sistema, utilizando-se de uma combinação de diagramas de casos de uso, descrições de casos de uso, modelos de dados e modelos de classes. Além disso, o capítulo descreve as tecnologias utilizadas no desenvolvimento.
- O capítulo 4 contém a descrição das funcionalidades da plataforma, apresentando suas telas e uma breve descrição de cada funcionalidade. Aborda também a escolha do padrão seguido para o desenvolvimento dos objetos de aprendizagem.
- O capítulo 5 contém as considerações finais, pontua as contribuições do trabalho e sugere possíveis continuações posteriores.

## 2. O Pensamento Computacional

Este capítulo abordará os conceitos relacionados ao tema, experiência concretas de trabalho do pensamento computacional em salas de aula, benefícios da abordagem, o potencial de transformação social e formas de abordar.

### 2.1. O que é Pensamento Computacional?

Jeannette Wing (2006) foi pioneira em tratar do assunto Pensamento Computacional, e o define como uma habilidade fundamental a ser desenvolvida não apenas por cientistas da computação, mas para qualquer ser humano, e que juntamente com a leitura, a escrita e a aritmética, o Pensamento Computacional deveria ser trabalhado na habilidade analítica de cada criança. Apesar da concepção do termo datar mais de uma década, não existe um consenso na definição do mesmo. Em um documento elaborado pelo National Research Council nos Estados Unidos (NRC, 2011), observam-se diferentes visões acerca do termo. Não se estabeleceu se é adequado chamar de “pensamento”, se a idéia é nova ou uma combinação de pensamentos existentes (pensamento matemático e de engenharia) (Raabe et al., 2015).

Zanetti e Oliveira (2015) afirmam que o Pensamento Computacional está mais próximo de um pensamento analítico, que compartilha características com o pensamento matemático (onde encontram-se as habilidades iniciais para resolução de problemas), pensamento sistêmico (projetar e avaliar sistemas complexos onde operam restrições específicas) e pensamento científico (conhecimento teórico sobre o assunto, análise de resultados e senso crítico). Portanto, trata-se de uma maneira específica de se pensar e analisar uma situação ou artefato, sendo influenciado por todas as áreas de conhecimento e diferentes tecnologias.

O Pensamento Computacional também é definido na literatura como um arranjo de pensamentos que aplicam conceitos e metodologias da computação a fim de solucionar problemas em um amplo espectro de assuntos e oferecendo um arcabouço de habilidades essenciais para qualquer ciência contemporânea (Quin, 2009). Desta forma, o Pensamento

Computacional poderia ser comparado a um canivete suíço cognitivo, que pertenceria a um kit de sobrevivência intelectual, capaz de oferecer um conjunto de habilidades que potencializa a resolução de problemas complexos e desconhecidos, aplicando o raciocínio computacional (Quin, 2009).

Seguindo raciocínio semelhante, Ramos (2014) também compara o Pensamento Computacional a um canivete suíço. Para Ramos, o Pensamento Computacional consiste num conjunto de capacidades intelectuais comparadas à leitura, à escrita e às operações aritméticas. Segundo o pesquisador, divulgar o Pensamento Computacional é divulgar um conjunto de habilidades que auxiliam na resolução de problemas ao longo de toda a vida, em um espectro biopsicossocial, sendo um casamento entre a dimensão social e a dimensão cognitiva, focado na comunicação e na colaboração entre as pessoas e não pode ser encarado como uma proposta de tornar as pessoas um computador. Ramos afirma que computar é uma ação profundamente e essencialmente humana, e consiste em uma operação cognitiva de automatização de um conjunto de operações, onde a máquina apenas estende a nossa capacidade de resolver problemas, trazendo recursos e velocidade para se obter um alcance mais longo, comparado ao exercício humano de computar.

Para Ramos, o Pensamento Computacional consiste essencialmente em: formular problemas em uma maneira que permita a utilização do computador e outras ferramentas para resolver, organizar e analisar dados de forma lógica, representar dados através de abstrações, como por exemplo através de modelos e simulações, automatizar soluções através do pensamento algorítmico, identificar, analisar e implementar soluções possíveis, generalizar e transferir estes processos para resolver uma grande variedade de situações (similar a idéia de um canivete suíço). Adicionalmente, o pesquisador sugere aspectos complementares ao Pensamento Computacional, são eles: entender o que se resolve e o que não se resolve computacionalmente, avaliar e escolher as ferramentas e técnicas adequadas para resolução de um problema, entender limitações das ferramentas e técnicas computacionais, aprender estratégias de resolução de problemas, como por exemplo a técnica de divisão e conquista.

Muitas das práticas de Pensamento Computacional já são realizadas em escolas sem este nome. Observa-se, por exemplo, na filosofia o desenvolvimento de habilidades de abstração, lógica e comunicação; na física há a modulação na resolução de problemas (Ramos, 2014).

Alguns aspectos são frisados na literatura como não sendo Pensamento Computacional. Por exemplo, o Pensamento Computacional não é sinônimo de programação, mas sim uma maneira de compreender os sistemas computacionais, como eles funcionam, como são projetados e programados. A programação não é um fim, mas um meio para que se atinja um objetivo (Zanetti e Oliveira, 2015). O Pensamento computacional também não propõe aos estudantes a pensarem como máquina, não propõe a utilização do computador na perspectiva de um usuário em atividades como ler email, fazer uma pesquisa na internet, utilizar uma rede social, entre outras. Por fim, por ser uma atividade essencialmente humana, o Pensamento Computacional não requer um computador para ser trabalhado e desenvolvido entre estudantes e professores.

A definição que parece ser a mais presente na literatura, construída pela International Society for Technology in Education- ISTE em conjunto com a Computer Science Teachers Association - CSTA(CSTA, 2011) , é a de que Pensamento Computacional é um processo de resolução de problemas que possui as seguintes características: “formulação de problemas de forma que computadores e ferramentas possam ajudar a resolvê-los, organização lógica e análise de dados, representação dos dados através de abstrações como modelos e simulações, automatização de soluções através do pensamento algorítmico, identificação, análise e implementação de soluções visando a combinação mais eficiente e eficaz de etapas e recursos e generalização e transferência de soluções para uma ampla gama de problemas”(Raabe et al., 2015).

O cerne do escopo da resolução de problemas encontra-se nos processos mentais envolvidos na forma como um profissional da computação age quando coloca em ação técnicas e práticas da ciência da computação para solucionar problemas. Embora essas definições de pensamento computacional e as habilidades associadas a este variem entre si, há um consenso na literatura de três habilidades associadas ao Pensamento Computacional, são elas: abstração, decomposição e construção de algoritmos. Abstração é a habilidade em entender a essência, o cerne do que se está observando e buscando entender, podendo-se separar a abstração em

diferentes níveis de detalhes e complexidade. Decomposição é uma estratégia para lidar com problemas complexos, e consiste na divisão do problema em problemas menores e mais fáceis de gerenciar e solucionar. Construção de algoritmos é o processo de planejar e organizar uma sequência de passos lógicos para resolver um problema ou alcançar um objetivo (Araújo et al., 2015).

## **2.2. Por quê estimular o Pensamento Computacional na educação básica?**

A Meta 6 do Plano Nacional da Educação (Brasil, 2014) é: “oferecer educação em tempo integral em, no mínimo, 50% (cinquenta por cento) das escolas públicas, de forma a atender, pelo menos, 25% (vinte e cinco por cento) dos (as) alunos (as) da educação básica”.

Desta forma, com uma carga horária maior, cria-se a oportunidade de propiciar aos estudantes atividades inovadoras de cunho interdisciplinar, estabelecendo um cenário que propicie atividades que viabilizem o desenvolvimento do pensamento computacional na educação básica, como na utilização de jogos de programar, na construção de aplicações em ambientes de programação, na robótica pedagógica, na mecatrônica educacional, na computação desplugada, entre outros (Raabe et al., 2015).

Estudiosos na década de 1980, já afirmavam que era necessário promover o desenvolvimento do Pensamento Computacional junto aos alunos, argumentando sobre o pensamento analítico (chamado em seu trabalho de pensamento procedimental), utilizando a linguagem de programação Logo, que precedeu várias ferramentas pedagógicas de ensino de programação (Zanetti e Oliveira, 2015).

Outro aspecto da adoção do Pensamento Computacional na educação básica é a potencial contribuição na resolução de problemas de suas respectivas futuras carreiras e áreas de atuação. Por focar em desenvolver a capacidade de resolução de problemas, o Pensamento Computacional cria desafios e oportunidades para estudantes, pesquisadores, educadores e toda a comunidade escolar, possuindo um cunho multidisciplinar e integrador, com a capacidade de proporcionar uma interdisciplinaridade efetiva, colaborar para a melhoria constante do aprendizado escolar e



proporcionar um uso mais eficaz da tecnologia em benefício da sociedade (França e Tesdeco, 2015).

Alunos pertencentes à educação básica desde o início do século XXI até os dias atuais pertencem à geração Z, que tem como características um comportamento interativo, muito familiarizado com tecnologias e a busca por um retorno imediato a suas ações expectativas. As aulas expositivas tradicionais nas escolas se mostram ineficazes em prender a atenção dos alunos dessa geração, que costuma acompanhar somente os primeiros 10 a 15 minutos de aula. O ensino do Pensamento Computacional possui um grande potencial de estimular esses alunos, seja pela proposta interativa e desafiadora, seja pela visão de interdisciplinaridade e a possibilidade de trabalhar com o lúdico, com os possíveis jogos que podem ser realizados no trabalho em sala (Ferreira et. al, 2015).

Por fim e não menos importante, uma consequência do ensino do Pensamento Computacional na educação básica é a de desmistificação da computação. Segundo levantamento do SEMESP (Sindicato de Estabelecimentos de Ensino Superior no Estado de São Paulo), os cursos de nível superior em computação são os que possuem o maior índice de evasão de acadêmicos (Ferreira et. al, 2015). Segundo Piva Jr. e Freitas (2011), o conhecimento de base dos alunos ingressantes em cursos superiores de computação é deficiente, especialmente em Português (na interpretação de textos) e Matemática (resolução de equações matemáticas). O aprendizado em programação na graduação pode ser complexo e até mesmo desmotivador, visto que o tempo despendido no ensino é curto e a complexidade da abordagem é alta. Esses fatores contribuem com o alto índice de evasão, troca de curso, baixo rendimento em disciplinas como Linguagem de Programação, Estrutura de Dados, Análise de Algoritmos, entre outros. Como consequência à longo prazo, esses fatores levam à uma escassez de mão de obra em Computação, e um distanciamento de outras áreas com a computação, tornando remotas as possibilidades interessantes para a evolução dessas áreas (Cardoso e Antonello, 2015). Como exemplo de possibilidades interessantes pode-se imaginar a aplicação do Pensamento Computacional na rotina de um profissional de Química, na automatização de um procedimento laboratorial, ou um profissional de Biologia, na utilização da Bioinformática e desenvolvimento de programas mais focados na sua pesquisa, ou de um médico na forma de instruir um paciente,

ou um pai ou professor de matemática, ensinando o filho/aluno a verificar a existência e calcular raízes de uma equação de segundo grau quando este aprende a fórmula de Bháskara na escola.

### **2.3. Experiências concretizadas**

Iniciativas de introduzir o Pensamento computacional têm sido realizadas e seus resultados têm mostrado engajamento de estudantes e um grande potencial interdisciplinar envolvido. Nos Estados Unidos e na Europa, iniciativas de promover e adotar o Pensamento Computacional nas escolas vêm sendo crescente (Aguilhar, 2013). Na Inglaterra, o ensino já faz parte do currículo escolar obrigatório (Aguilhar, 2013). A partir dos cinco anos, os estudantes ingleses são estimulados na criação de programas de computador e teste de seus programas, além de aulas sobre segurança na internet, prevenção de situações de perigo na web e como manter privacidade na rede. Já no ensino secundário, esses estudantes têm aulas sobre sistemas de computação, linguagens de programação variadas, utilização de impressoras 3D e robótica (Aguilhar, 2013).

No Brasil há poucas iniciativas de desenvolver o Pensamento Computacional na educação básica. A abordagem mais comum é mais frequentemente realizada no ensino técnico, na graduação de cursos de ciências exatas e em pós-graduação (França e Tesdeco, 2015). Na Paraíba, há o relato na literatura da realização de olimpíadas para apresentar estruturas de linguagens de programação e prática de técnicas usadas na construção de algoritmos (abstração, depuração de erros, testes, melhoria de algoritmos simples) (França e Tesdeco, 2015). Em Pernambuco, há iniciativas em prol da disseminação do Pensamento Computacional e divulgação dos cursos de licenciatura em computação e desmistificação do papel desses profissionais na sociedade, através de cursos de lógica de programação a estudantes do ensino médio (pela plataforma App Inventor) (França e Tesdeco, 2015). Na Bahia, há iniciativas para incentivar e descobrir jovens talentos para a área de computação em escolas do ensino fundamental e médio, através do trabalho do raciocínio lógico e matemático, do incentivo à participação feminina na Olimpíada Brasileira de Informática (OBI) e da introdução ao raciocínio algorítmico, de forma

que estudantes pudessem compreender a utilização dos modelos de raciocínio na solução de problemas (França e Tesdeco, 2015). No Rio Grande do Sul há iniciativa de proposição de atividades direcionadas ao ensino fundamental que envolvem Pensamento Computacional (França e Tesdeco, 2015). No Amazonas há relatos de realização de peças teatrais expondo fundamentos de computação de forma clara e lúdica. No Rio de Janeiro, há iniciativas de introdução ao Pensamento Computacional através da mecatrônica e da Robótica (França e Tesdeco, 2015). Em São Paulo há investigação de como as competências relacionadas à matemática são absorvidas por estudantes do ensino médio através da construção de jogos digitais (França e Tesdeco, 2015). Há também o relato de um projeto que objetivou proporcionar estudantes de ensino médio a oportunidade de adquirir competências relacionadas ao desenvolvimento de raciocínio lógico, na resolução de problemas por meio de algoritmos e estrutura de dados, e como resultado obteve um aumento de interesse dos estudantes nos cursos de computação, havendo a participação deles em atividades da OBI e o ingresso em cursos superiores da área de computação (França e Tesdeco, 2015). Em Minas Gerais, há um projeto para inserção do pensamento computacional e conceitos de Tecnologia da Informação (TI) para ensino médio e focado à empregabilidade, empregado em 11 escolas e há uma meta de inserção de todas as escolas estaduais no projeto (França e Tesdeco, 2015).

Brito e Madeira (2015) relatam a experiência de uma implementação de um framework para ensino de introdução a programação baseado em gamificação, como proposta para tratar o problema das altas taxas de reprovação, retenção e evasão em cursos de computação. Embora tratado especificamente em cursos de computação, a proposta se estende para qualquer iniciativa introdutória ao Pensamento Computacional. Os autores se utilizam da técnica de *Design Thinking*, que consiste em um processo focado no entendimento profundo de um problema, para só depois propor soluções que a resolvam efetivamente. Busca-se as verdadeiras raízes da questão investigada para só então projetar-se uma solução que possa resolvê-la com eficácia. A técnica é composta de 3 etapas, são elas: imersão, ideação e prototipação. A imersão consiste no estudo aprofundado do problema junto ao público alvo, de forma que este termine a etapa entendendo completamente o problema a ser resolvido. A ideação é a etapa onde várias idéias

são geradas para compor possíveis soluções para a problemática. Já na prototipação, são realizados testes e validação das soluções propostas junto ao público-alvo.

Com base nessa ideologia, os autores aplicaram um framework para gamificação proposto por Werbach (Werbach, 2012). O modelo consiste em seis etapas que representam passos para serem tomados em uma construção de solução, centrado no entendimento do problema por parte da equipe proponente. Estas etapas são: definição de objetivos do negócio, que consiste no entendimento do problema e o ambiente em que ele ocorre; definição de metas, onde os objetivos desejados são definidos; descrição dos jogadores, onde são conhecidas as pessoas envolvidas no processo e que interagem com o problema no dia a dia; definição de ciclos de atividades, onde, após entendido o problema, deve-se pensar em ações que poderão ser efetivadas; diversão dos jogadores, onde espera-se o estímulo e a participação dos envolvidos no processo; disponibilização de ferramentas apropriadas, onde, após definida uma proposta de solução, deve-se escolher técnicas, modelos ou ferramentas adequadas para colocá-las em prática.

O Modelo foi aplicado através de pontuações associadas a exercícios realizados, relacionados a cada conceito da matéria introdutória de programação. A turma na qual a atividade foi realizada foi dividida em grupos de acordo com os perfis em relação ao tempo despendido no estudo em casa e se trabalha concomitantemente com o estudo. O projeto revelou um maior engajamento dos estudantes com o processo de resolução de exercícios e o método foi avaliado como justo e descontraído. Justo pois os exercícios eram distribuídos igualmente para cada conceito, de forma que se um aluno tivesse dificuldades com exercícios de um conceito, poderia recuperar em exercícios de outros. Descontraído, pois eles recebiam prêmios de acordo com os objetivos alcançados e acompanhavam em um quadro suas pontuações preliminares (Brito e Madeira, 2015).

Pimentel et. al (2015) relata a aplicação da mecatrônica através de uma experiência prática de desenvolvimento de um protótipo de estacionamento inteligente, no ensino de matemática e física, para estudantes do ensino secundário. Os autores ressaltam a preocupação durante o projeto de não tornar o que é pra ser uma prática pedagógica educativa em um momento puramente de diversão. Sendo um projeto de mecatrônica, isto é, uma união da

mecânica, da eletrônica e de recurso de Tecnologia da Informação, este contempla desde o trabalho com sucata, marcenaria, soldagem e geometria, até o uso de eletrônicos como a plataforma arduino para programação dos componentes do estacionamento. A proposta dos autores foi na construção do estacionamento inteligente utilizando a metodologia denominada *Problem Based Learning* (PBL), que, entre outras questões, parte do princípio que o aluno seja o coautor de sua aprendizagem. O projeto consistiu na primeira etapa em planejamento e desenho do projeto através de encontros semanais de 3h durante um período de 4 meses. Após, foi realizada a confecção da planta baixa e a construção do circuito e dos sensores de presença conectados a uma placa de arduino. Através da prática, foi possível a abordagem de conceitos como circuitos elétricos, geometria espacial e conceitos de programação como variáveis, *loops* e condicionais. Ao final do projeto, a experiência foi avaliada pelos alunos, que em sua maioria apontou o projeto como uma experiência motivadora de aprendizado. Adicionalmente, foi avaliado o quanto os alunos aprenderam em relação aos conceitos de matemática, física e programação abordados, e verificou-se um aprendizado expressivo dos conceitos após a prática.

## **2.4 Benefícios do desenvolvimento do Pensamento Computacional**

Diversos estudos demonstram os benefícios de se incorporar o Pensamento Computacional de forma ampla, dinâmica e interdisciplinar, e em como o Pensamento Computacional pode influenciar diversas áreas do conhecimento. Ramos (2014) defende que o Pensamento Computacional contribui positivamente em aspectos de dimensão pessoal como por exemplo na confiança em lidar com a complexidade, na persistência em se trabalhar em problemas difíceis, na tolerância à ambiguidade e incertezas, na capacidade de lidar com problemas abertos (infinitos), na capacidade de comunicar e trabalhar com outras pessoas, entre outras possibilidades.

Rodrigues et. al (2015) realizaram estudo com 103 estudantes do estado da Paraíba e avaliaram se o Pensamento Computacional possui alguma relação com o desempenho dos estudantes no ensino básico e, em caso afirmativo, se o Pensamento Computacional poderia melhorar o desempenho desses estudantes. Através de uma análise estatística com resultados do

ENEM (Exame Nacional do Ensino Médio) de dois grupos de estudantes (um grupo que possuía experiência em programação e outro que não possuía), verificou-se que *“existe uma correlação moderada entre o desempenho dos estudantes no ENEM e a experiência destes em programação. Além disso, alunos com experiência em programação obtiveram um desempenho melhor em todos os eixos cognitivos abordados no exame em relação aos que não estudaram programação. Esses resultados de certa forma eram esperados, pois a matriz de referência do ENEM tem como fundamento a habilidade de resolução de problemas que também é fortemente explorada por parte do Pensamento Computacional”* (Rodrigues et al., 2015).

Bastos et al. (2016) realizaram uma pesquisa em neurociência que, através da coleta de sinais cerebrais de estudantes secundários antes e depois de um treinamento em lógica computacional, investigou possíveis contribuições do treinamento de lógica computacional para o desenvolvimento de raciocínio lógico dos estudantes. De acordo com os autores *“pensar, aprender e memorizar são processos biológicos realizados no cérebro, porém distintos. O arquivamento seletivo, a evocação de informações, envolve um conjunto de processos neurobiológicos e neuropsicológicos que caracterizam a memória. A aprendizagem é decisiva para o pensamento, o qual deriva da capacidade de lidar com as informações das áreas de associação motora, sensorial e mnemônica. Já o pensamento é o referencial para a orientação do comportamento, pois seu processamento envolve o recebimento, a percepção, a compreensão, o armazenamento, a manipulação, o monitoramento e o controle essenciais para lidar com o fluxo constante de dados objetivando planejar ações .”*A aprendizagem de lógica depende das habilidades de linguística e raciocínio. Sendo o raciocínio a maneira mais complexa de pensar, a lógica estuda a “correção do raciocínio”, dando um formato de ordem para o pensamento. Desta forma, a lógica se mostra fundamental para a vida de qualquer ser humano, pois ao buscar pensar, falar, agir e escrever corretamente, é necessário o uso da lógica para se colocar uma “ordem” no pensar. O raciocínio lógico desenvolvido é fundamental para resolução de problemas cotidianos, que muitas vezes são problemas mal estruturados, onde não apresentam recursos claros e disponíveis para a solução, e quem se propõe a resolvê-los apresenta dificuldades para elaborar representações mentais de modelagem do problema e da solução. Já um problema, quando é bem estruturado, possui representação e recursos claros para a solução,

bem como uma estratégia de solução clara, como uma fórmula matemática. As habilidades cognitivas são influenciadas pelo meio social. Diante da plasticidade neuronal, é possível a reconfiguração das redes neurais e reconstrução de conhecimento em todas as fases, especialmente na fase da adolescência, onde há um intenso desenvolvimento cerebral (Bastos et al., 2016).

Os autores explicam que o cérebro humano é dividido em dois hemisférios especializados. O hemisfério esquerdo é responsável pelo processamento da linguagem, na realização mental de cálculos matemáticos, no comando da escrita e na compreensão dela pela leitura. Já o hemisfério direito é melhor na percepção de sons musicais e reconhecimento de faces. Grandes regiões do encéfalo humano chamadas lobos apresentam funções determinadas como: o lobo occipital está relacionado ao sentido da visão e dividido em diversas áreas visuais diferentes, o lobo parietal é em parte dedicado ao sentido do tato, agrupando funções de sensibilidade corporal e reconhecimento espacial. O lobo temporal processa dados auditivos, aspectos de memória, visão e lingüística. O lobo frontal é fundamental para o planejamento de ações cognitivas, memória e movimento. Para a leitura de sinais cerebrais, foi utilizada a Tecnologia BCI (*Brain Computer Interface*). A Tecnologia BCI é composta por ferramentas que possibilitam uma comunicação a partir da atividade neural gerada pelo cérebro e funcionam como uma interface entre o usuário e dispositivos externos, como interruptores, próteses e computadores. Movimentos musculares são criados pelos comandos cerebrais apenas pelo fato de pensar no movimento em si, sem exigir qualquer outro estímulo, de forma que sinais elétricos são detectados por sensores no escalpo da superfície cortical ou em áreas subcorticais e convertidos em comandos no dispositivo elétrico. Para o experimento, os autores utilizaram uma touca com recepção para 32 eletrodos posicionados nas 4 regiões ( lobo frontal, temporal, parietal e occipital) inseridos previamente no couro cabeludo. O sinal elétricos captados pelos eletrodos foram amplificados, digitalizados e transferidos para um computador por entrada USB, para armazenamento e exibição dos dados.

Foi realizada a observação e caracterização de mudanças pré e pós treinamento de lógica computacional, para investigar possíveis contribuições do treinamento de lógica computacional



como auxiliador no desenvolvimento do raciocínio lógico nos estudantes de ensino médio, utilizando sinais cerebrais coletados em dois momentos antes e após o treinamento em lógica computacional, através de eletroencefalograma.

A fim de obter esta observação e caracterização, foi aplicado um questionário de perfil e teste de lógica para selecionar os estudantes. Após, foi escolhido o grupo de alunos dos dois últimos anos do ensino médio e realizada a coleta de sinais cerebrais antes do treinamento em lógica computacional. Logo após, foi realizado o treinamento em lógica computacional em formato de oficina, utilizando a linguagem Scratch e com duração de 24 horas total de aula, ministrada em um período de 2 meses. Por fim, foi realizada a coleta dos sinais cerebrais dos estudantes após o término da oficina e os dados foram analisados.

Foi relatado por completo a atividade pré e pós treinamento de um dos alunos observados no experimento. Os autores levantaram 2 hipóteses, uma para um comportamento esperado na coleta de sinais cerebrais mais intensos pré-treinamento e outra para o pós-treinamento.

No pré treinamento, os autores formularam a hipótese de que haveria um maior comportamento cerebral em diversas regiões. Considerando-se que os problemas de lógica são problemas mal estruturados, ou seja, problemas onde a maior dificuldade costuma ser na elaboração de um plano para que se siga sequencialmente uma série de passos que avance em direção a uma solução, o que favorece elaboração de percursos que não solucionem o problema e um maior tempo despendido, esperou-se uma maior atividade cerebral em maior número de regiões não específicas para problemas de lógica. Esta hipótese foi confirmada pelos resultados do eletroencefalograma pré treinamento.

Já no pós treinamento, os autores formularam a hipótese, que se mostrou válida pelos resultados do eletroencefalograma referente a esta etapa, de que, após os alunos passarem por treinamento em lógica computacional, a atividade cerebral mais intensa estaria mais concentrada em um número menor de regiões, comparando à quantidade de regiões ativadas no experimento pré treinamento, e mais intensa nas regiões esperadas, que são as regiões relacionadas ao raciocínio lógico, de forma que o aluno mais experiente dedicaria mais tempo na fase inicial decidindo como solucionar um problema, e menos tempo solucionando-o efetivamente, pois ao

dedicarem mais tempo na decisão antecipada do que fazer, o aluno possui menor chance de ter um início de raciocínio e percurso errados (Bastos et al., 2016).

## **2.5. O “Pensar Computacional” como meio de transformação social**

O tráfico de drogas no Brasil é um empreendimento forte, que apresenta plano de carreiras e abraça pessoas com baixa qualidade de vida em termos de saúde, educação e entretenimento e que não possuem informação do potencial destruidor das drogas. A ilusão de enriquecimento à curto prazo e a baixa perspectiva de vida recruta cada vez mais jovens das comunidades mais pobres de grandes cidades brasileiras, como Rio de Janeiro e São Paulo. Com vista nessas questões, iniciativas foram travadas a fim de buscar atuar positivamente nesse cenário. O CDI ( Comitê de Democratização da Informática) é uma organização social voltada ao empoderamento digital, que busca formar jovens autônomos, conscientes e conectados, aptos a reprogramar o sistema em que estão inseridos, através do uso da tecnologia. Atuam em parceria com centros comunitários, bibliotecas e escolas públicas para formar multiplicadores de sua própria metodologia, que replicam o programa para o público final. Formam uma rede de educadores, professores e bibliotecários para promover uma nova consciência e gerar oportunidades aos jovens brasileiros. Na cidade do Rio de Janeiro atuam no DEGASE (Departamento Geral de Ações Sócio-educativas) onde, entre outras atividades, realizam o ensino de programação. Como exemplo de ações bem sucedidas, há a experiência de Wanderson Skrock, que atuava como gerente do tráfico no Complexo do Alemão e em sua segunda passagem pelo DEGASE agentes socioeducativos estimularam o jovem a deixar a vida de tráfico (Heringer, 2014).

Experiências como essas levantam questões do potencial transformador que o Pensamento Computacional pode ter diante dessa realidade degradante. Além disso, abre-se a oportunidade de aproximação do meio acadêmico de Tecnologia da Informação e sociedade, de forma que universidades podem atuar na divulgação de Pensamento Computacional na sociedade civil e nos grupos marginalizados.

## 2.6. Como abordar?

O ISTE/CSTA publicou um documento em 2011(CSTA, 2011), que tem como proposta servir como modelo no incentivo do desenvolvimento de pensadores criativos, que se beneficiem da computação para a resolução de problemas. Objetiva o auxílio a docentes e gestores educacionais na compreensão e implantação do pensamento computacional na educação básica. O documento faz a descrição da importância do pensamento computacional para estudantes nos tempos atuais, e detalha recursos para sua introdução no ambiente escolar, são eles: definição operacional de pensamento computacional, a qual permite melhor compreensão e uma discussão mais direcionada do pensamento computacional, vocabulário dos termos chaves envolvidos, que facilita o diálogo entre os atores envolvidos, utilização de um modelo para uma mudança sistemática. Esse modelo se baseia em 4 passos, são eles: **guiar**, que refere-se à necessidade dos envolvidos no processo em ter uma postura de liderança e de conduta para tomada de decisão para poder então construir (próximo passo); **construir**, que sugere a construção de consciência do pensamento computacional entre demais professores e líderes; **conectar**, que diz respeito ao potencial interdisciplinar de conectar o pensamento computacional com iniciativas educacionais inovadoras que construirão valores e conhecimento, incluindo preparação de materiais e recursos e adequação para cada faixa etária; **praticar**, que diz respeito a prática em si de atividades que introduzam o pensamento computacional em sala de aula. O documento também sugere práticas adequadas por faixa etária com suas estratégias de implementação, objetivos e pontos que deverão ser abordados e desenvolvidos (Raabe et. al, 2015).

Quanto maior o envolvimento do aluno no processo de aprendizado, maior é o potencial de aquisição de conhecimento por parte dele (Ferreira et al., 2015). Morin (2001) evidenciou, em suas investigações, sete classificações de saberes para serem fomentadores de metodologias educacionais para que estas sejam transdisciplinares e, assim, atender ao futuro da educação em uma perspectiva integral. Morin elenca entre outras a competência de enfrentar as incertezas e descreve o futuro como algo ilimitado e imprevisível. Isto aponta que há uma quantidade considerável de projetos complexos que nenhum ser humano ainda pensou em esboçar (Farias et al., 2015).

Uma forma de abordar o Pensamento Computacional em salas de aula é através do método *Problem Based Learning* (PBL). PBL é uma metodologia que parte do princípio de que o aluno deve ser o coautor de sua aprendizagem, que não deve ser imposta pelo docente, de forma que o professor age como facilitador, na proposição de problemas e no auxílio dos alunos no desenvolvimento de tarefas. Os problemas escolhidos através desta metodologia englobam os diferentes conceitos do currículo escolar. Alguns dos benefícios desta metodologia são: desenvolvimento de diferentes habilidades do aluno, visualização da processo de construção de um projeto ou de um problema apresentado para que se alcance a visualização não só do resultado, mas também das metas parciais e da importância de cada etapa do processo (Pimentel et. al, 2015).

A impressão inicial de estudantes com conceitos de programação pode ser determinante na forma como eles perceberão os desafios e enfrentarão as dificuldades próprias da aprendizagem da lógica de programação. Fatores psicológicos como motivação pessoal e autoconfiança exercem grande influência sobre alunos de programação pois estes estão diante da necessidade de dominar uma forma completamente nova de pensar. A abordagem através de Jogos no ensino do Pensamento Computacional demonstra ser motivadora e efetiva ao que se propõe. O jogo, um dos meios de se trabalhar o desenvolvimento do Pensamento Computacional na educação básica, promove um ambiente lúdico e os estudantes exercitam habilidades necessárias ao desenvolvimento integral, como a autodisciplina, a sociabilidade, a afetividade, valores morais, espírito de equipe e bom senso (Cardoso e Antonello, 2015).

Há três formas de se trabalhar os jogos na educação básica, são elas: competição, cooperação e individuação. A competição consiste necessariamente em um jogador buscar superar um ou mais oponentes, e possui o intuito de motivar cada jogador na superação do oponente. Como efeito colateral da competição, há o fato de que nem todos os jogadores se empolgam, e muitas vezes preferem evitar confrontos e se sentem mais confortáveis com situações cooperativas ou individuais, onde não ocorrem perdedores. Para os mais afeiçoados ao assunto do jogo e/ou a um comportamento competitivo, é uma tendência buscar um maior preparo para competir e de elevar o nível de disputa. Isso pode causar retração cada vez maior de participantes menos afeiçoados à disputa e/ou ao assunto e um possível afastamento destes. Já os

jogos cooperativos consistem em todo o grupo trabalhar em conjunto a fim de se alcançar um objetivo coletivo. Possuem como vantagem a preparação dos jogadores para o trabalho em equipe, essencial para boas relações profissionais. Há de se ter cuidado, porém, de se manter certo equilíbrio de esforço dentre os jogadores, de modo que todos dediquem esforços igualmente. Por fim, a experiência individual consiste em um jogo de apenas um jogador, onde este interage com o jogo de forma a alcançar solitariamente os objetivos. A experiência individual não corre os riscos de efeitos colaterais da experiência competitiva, nem o risco de desequilíbrio de esforço entre os jogadores, porém seu sucesso depende de quanto o jogo se tornou atrativo pelo enredo, mecânica e elementos de game design (Galdino et al., 2016).

Quanto a forma de abordar o jogo no campo educacional, esta pode ser realizada de 3 formas: gamificação (*gamification*), jogos sérios (*serious game*) ou interação jogável. O jogo do tipo *gamification* possui uma finalidade técnica com elementos de jogos. Por exemplo, o aplicativo Waze é um exemplo de jogo tipo *gamification*. O *serious game* possui a finalidade de ser um jogo interativo, porém é voltado mais para educar ou treinar do que entreter. O jogo Insuonline é um exemplo de *serious game*, que possui o objetivo de treinar médicos para o tratamento da diabetes e o uso de insulina. Já a interação jogável possui a finalidade de mudança de comportamento com produtos e serviços interativos (Galdino et al., 2015).

Outra forma de trabalhar o Pensamento Computacional é através da Robótica Pedagógica. Jovens geralmente não estão interessados em abordagens tradicionais, tornando-se motivados quando as atividades de Robótica Pedagógica são introduzidas como uma maneira de conectar outras disciplinas ou áreas de interesse. A robótica pedagógica propicia a construção de um ambiente de trabalho colaborativo e a realização de atividades interpessoais, gerando motivação e aquisição de conhecimento em informática, eletrônica, mecânica e design, através da resolução de um determinado problema ou desafio (Zanetti e Oliveira, 2015). Os estudantes podem, através da robótica pedagógica, construir seus próprios robôs e programar seus comportamentos. As ferramentas mais comumente utilizadas para trabalhar a robótica pedagógica são o Lego Mindstorm, o WeDo também da Lego e a plataforma Arduino (Zanetti e Oliveira, 2015).

Uma outra forma de abordar o Pensamento Computacional é através da Computação Desplugada. A computação desplugada consiste no ensino de fundamentos de ciência da computação sem a necessidade de um computador, sendo totalmente independente de hardware e software, e pode ser trabalhada de forma lúdica e divertida (Arantes e Ferreira, 2015). A computação desplugada é descrita em um documento em formato de livro, escrito por Tim Bell (University of Canterbury), Ian Witten (University of Waikato) e Michael Follows (Charles Darwin University), com contribuições de dezenas de outros pesquisadores, e propõe uma abordagem lúdica dos fundamentos de computação, sem o uso de computadores e sem detalhes técnicos. É possível sua utilização a partir do ensino fundamental até o ensino superior. Por não depender de recursos tecnológicos, a proposta da computação desplugada possibilita uma maior popularização do pensamento computacional e adequa-se bem a ambientes com poucos recursos em tecnologia da informação. O livro trata de conceitos de computação como números binários, representação de imagens, compreensão de texto, detecção e correção de erros, algoritmos de ordenação, criptografia, entre outros, e é essencialmente voltado ao professor. Possui as atividades descritas em um formato de plano de aula, junto com materiais que podem ser impressos para auxiliar a aplicação das atividades pelos professores (CSUnplugged, 2015).

Similar à computação desplugada, a Computação Criativa é um guia desenvolvido pela *Harvard University* que propõe atividades de introdução à computação com o uso do Scratch de forma a estimular a criatividade, imaginação e interesses pessoais dos alunos. O Scratch é uma linguagem de programação desenvolvida pelo *MIT Media Lab* bastante simples, voltada para o público infanto-juvenil e possui a estrutura de blocos. A computação criativa sugere a adoção do Scratch para o desenvolvimento da Pensamento Computacional por professores que buscam abordar a computação de forma introdutória e interdisciplinar, englobando conceitos de criatividade e arte. As atividades do guia contemplam: conceitos de pensamento computacional como sequência, ciclos, execução em paralelo, eventos, condições, operadores, dados; práticas de trabalho iterativo e incremental, testes, correção e depuração, reutilização e refatoramento, abstração e modularização; perspectivas como expressão, conexão e questionamentos.

Por fim, há um movimento global denominado *Hour of Code* impulsionado pela organização sem fins lucrativos [Code.org](https://code.org), que tem como proposta trabalhar o Pensamento

Computacional de forma a desmistificar a programação e mostrar que qualquer pessoa pode aprender fundamentos básicos de computação e desenvolver habilidades de resolução de problemas, lógica e criatividade. Possui o foco de ensinar a programação e é voltado diretamente para os estudantes, trazendo como proposta uma sequência de atividades de introdução a programação com duração estimada de uma hora, utilizando a notação de programação em blocos das ferramentas scratch e blockly. As atividades possuem personagens de desenho animado e são esteticamente bem construídas. São promovidas especialmente na primeira semana de dezembro, quando ocorre o evento Hora do Código em si, durante a Semana de Educação em Computação (*Computer Science Education Week*) nos Estados Unidos. Pessoas de destaque da computação como Mark Zuckerberg e Bill Gates foram alguns dos propagadores da idéia. Na segunda edição em 2014, mais de 60 milhões de estudantes acessaram a atividade, uma marca sem precedentes (Raabe et al., 2015).

Enquanto que as iniciativas Computação Desplugada e Computação Criativa estimulam a disseminação do conhecimento sobre computação e aumento da fluência no uso de tecnologias, os recursos da Hora do Código estão mais voltados para o ensino de habilidades em programação e técnicas computacionais de resolução de problemas. O material elaborado pelo ISTE incorpora ambas as perspectivas, tendendo mais para a primeira. Segundo Raabe et al. (2015), décadas já foram desperdiçadas nesta abordagem mais pragmática, que possui também o intuito de preencher vagas no mercado. Para eles, o pensamento computacional terá maior amplitude e promoverá mudanças mais profundas no desenvolvimento de estudantes e, conseqüentemente, maior impacto na sociedade, se estiver alinhado à abordagem construcionista. A abordagem construcionista foi o termo cunhado por Seymour Papert (1980) para mostrar o nível de construção de conhecimento quando um aluno constrói um objeto de seu interesse, como uma obra de arte, um relato de experiência ou um programa de computador. A perspectiva sobre construcionismo de Papert afirma que o aprendiz constrói seu conhecimento por meio do fazer e o fato deste estar construindo algo de seu interesse e que há motivação envolvida, onde o envolvimento afetivo torna a experiência do aprendizado mais significativa.

Desta forma, os autores afirmam a necessidade de se estimular a abordagem construcionista do pensamento computacional em salas de aula, abordagem esta que propicia um



trabalho mais profundo no sentido de envolver os alunos e trabalhar de forma mais ampla aspectos computacionais sem o foco em mercado ou tecnologia específica, mas sim na construção de uma visão mais holística pelo aluno. Adicionalmente, uma vez que a abordagem construcionista não possui o foco em tecnologias específicas, mas no entendimento do funcionamento, tal foco garante maior independência de recursos tecnológicos, garantindo assim a acessibilidade da abordagem, e adequando assim a escolas com carências sócio-econômicas (Raabe et al., 2015).

## **2.7. Desafios de implantação do Pensamento Computacional na Educação Básica**

Os benefícios da inclusão do Pensamento Computacional na educação básica já são considerados por diversos profissionais de educação. Contudo, essa inclusão enfrenta alguns desafios para que seja efetivamente uma realidade nas escolas brasileiras.

Em termos de currículo escolar, há certa preocupação nos efeitos da introdução de uma nova disciplina no currículo escolar, no que diz respeito à motivação e aprendizagem dos estudantes. Além disso, é um desafio trabalhar o Pensamento Computacional junto às disciplinas já existentes no currículo escolar e ir avançando a abordagem com o passar dos anos da educação básica. Questiona-se também o quê e quando ensinar. Para tal, afirma-se a necessidade de definição de diretrizes curriculares para ensino do Pensamento Computacional em escolas brasileiras, enfatizando quais conceitos devem ser introduzidos e em quais anos escolares. Por fim, é necessário não apenas cobrir a manipulação de recursos digitais, mas também e principalmente os fundamentos de Computação enquanto ciência (França e Tesdeco, 2015).

Em termos de formação de professores, há a preocupação de formar professores para introduzir o Pensamento Computacional adequadamente. Importante ressaltar que outras formações poderiam beneficiar-se de uma formação em pensamento computacional por este propiciar uma abordagem interdisciplinar (França e Tesdeco, 2015).

Em termos de processo ensino-aprendizagem, há o desafio de definir qual abordagem é mais adequada a cada público-alvo e peculiaridades e garantir a promoção de mecanismos para uma aprendizagem significativa. O processo de ensino-aprendizagem passa por conceber a idéia,

desenvolvê-la e avaliar a experiência adquirida. Assim, a participação ativa dos estudantes pode trazer um retorno sobre os métodos aplicados, podendo ser realizadas mudanças no emprego da metodologia abordada e a melhoria contínua, a fim de se obter bons resultados de aprendizagem (França e Tesdeco, 2015).

Em termos de aprendizado de computação, os desafios são variados. Muitos métodos tradicionais de ensino não consideram particularidades de aprendizado dos estudantes ou a forma de estudo estimulada pode ser não ser adequada a todos os estudantes. Estudantes que possuem experiência prévia em programação podem ter maior facilidade no aprendizado em disciplinas de programação. Habilidade matemática é indicativo de bom desempenho em disciplinas introdutórias em cursos de ciência da computação. Observações obtidas por pesquisadores e educadores podem orientar experiências de ensino de computação a serem empregadas nos níveis fundamental e médio no Brasil (França e Tesdeco, 2015).

Por fim, o grande desafio de não utilização de novas tecnologias para reproduzir os sistemas tradicionais de ensino, mas fomentar novos espaços de aprendizagem, proporcionando melhorias para professores e alunos e, por conseguinte, melhorar a qualidade da educação (Farias et al., 2015).

Pensando nesse aspecto de abrir caminho para novos espaços de aprendizagem, o presente trabalho propõe uma plataforma de cadastro, divulgação e consulta de objetos de aprendizagem que promovam o Pensamento Computacional, de forma que qualquer pessoa que tenha uma iniciativa de realizar uma atividade que trabalhe o assunto Pensamento Computacional possa divulgar e compartilhar sua iniciativa, de forma a estimular que mais e mais pessoas possam ter contato com o pensar computacionalmente, seja por Robótica Pedagógica, por Computação Desplugada, Computação Criativa, entre outros. Tornar essas experiências em sala de aula pública é uma forma de torná-las colaborativas e uma maneira de fortalecer a idéia de inclusão da programação na educação básica. A proposta, em última análise, é contribuir com a comunidade da Informática na Educação no que tange a promover iniciativas de trabalho no desenvolvimento das habilidades contempladas pelo Pensamento Computacional de forma universal, colaborativa e acessível.

## 3. Proposta de Trabalho

### 3.1. Objeto de aprendizagem e SCORM

O trabalho traz como proposta uma plataforma, denominada “Programe-se”<sup>1</sup>, cujo objetivo é o cadastro, divulgação e consulta de recursos e práticas do Pensamento Computacional na educação básica. Para tratar o tema com o devido destaque, cada cadastro de recurso foi tratado como um Objeto de Aprendizagem (OA). Objetos de Aprendizagem se inserem no contexto de recursos para o processo ensino/aprendizagem e são definidos como “qualquer recurso para apoiar a aprendizagem, por exemplo, uma animação, um software, uma página web, uma imagem, entre outros”. (IEEE, 2002). Embora a definição seja genérica e não haja uma única para o conceito de OA, três características básicas são bastante relacionadas ao conceito, são elas: reuso, interoperabilidade e acessibilidade (Siqueira, 2005). Adicionalmente, a plataforma possui metadados específicos de pensamento computacional, além de concentrar apenas objetos computacionais relacionados ao Pensamento Computacional.

As diversas formas de criar OA dificulta o seu reuso. Para diminuir essas dificuldades, o processo de desenvolvimento de OA deve seguir normas de padronização e características pedagógicas a fim de garantir o reuso, a interoperabilidade e a acessibilidade. Além disso, o OA deve englobar conteúdos a fim de estruturar o conhecimento, apoiar a identificação e a definição de conceitos e informações pertinentes e possibilitar que esses conceitos e definições sejam disponibilizados de forma coerente e ordenada. O SCORM é um dos padrões adotados na construção de objeto de aprendizagem e tem sido o mais utilizado para o desenvolvimento de projetos de aprendizagem virtual (de Oliveira et al., 2007).

---

<sup>1</sup> <http://warm-cliffs-24709.herokuapp.com/login>

O modelo SCORM (*Sharable Content Object Reference Model*), criado pelo Departamento de Defesa dos Estados Unidos, é uma especificação que utiliza metadados para gerenciar conteúdos de aprendizagem, a fim de garantir um desenvolvimento de conteúdo reutilizável e permitir a criação de uma biblioteca de OA e, conseqüentemente, a recuperação desses objetos. O SCORM agrupa um conjunto de especificações e padrões para criação e uso de OA. Ele é composto dos seguintes modelos: **Modelo de Agregação de Conteúdo** (*Content Aggregation Model* - CAM) que realiza a descrição dos componentes de um pacote de conteúdo, além de definir responsabilidades e requisitos para agregar componentes (como lições, módulos e cursos) e a geração dos pacotes, para elaboração de metadados sobre os conteúdos e inserção de detalhes de sequência e navegação no contexto do pacote; **Ambiente de Execução** (*Run-Time Environment* - RTE) para objetos educacionais baseados na Web: descreve como objetos de conteúdo são iniciados, permitindo a comunicação entre os objetos de conteúdo e os *Learning Management Systems* (LMS) que utilizem o padrão SCORM; **Modelo de Sequenciamento e Navegação** (*Sequencing and Navigation* - SN) para apresentação dinâmica de conteúdo baseada na necessidade do usuário: define métodos para representar o comportamento pretendido em uma experiência de aprendizado, possibilitando a qualquer LMS que utilize o padrão SCORM dar sequência às atividades de aprendizagem de forma consistente.

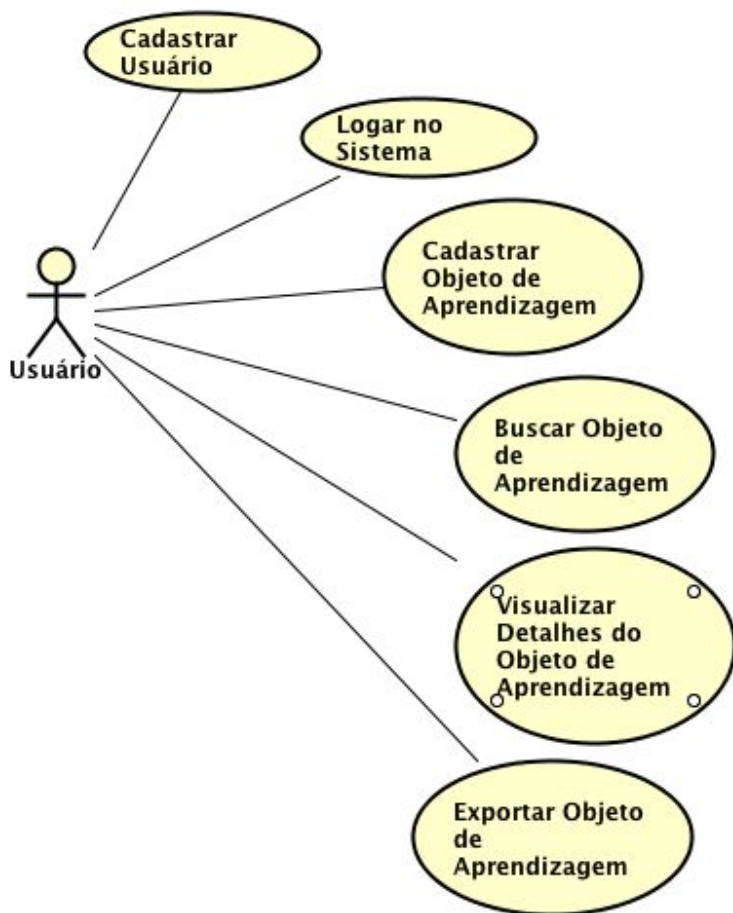
O processo de adequação de um OA ao modelo SCORM consiste na adição de metadados e no empacotamento do conteúdo criado OA + metadados, através da criação de um arquivo .zip. Os metadados adicionados descrevem o conteúdo do pacote e pode também prover informações acerca de requerimentos técnicos, restrições de uso etc. A adição de metadados para descrever um recurso permite que ele seja buscado, indexado e recuperado de repositórios LMS. Desta forma, qualquer LMS é capaz de ler o OA e assim tornar possível a sua reutilização.

Para o desenvolvimento da plataforma do presente trabalho, os OA serão gerados pela plataforma no forma de tutoriais, no formato .txt. O formato .txt, assim, como .pdf, .doc e outros, permitem a criação de objeto de aprendizagem a partir de ferramentas como Reload Editor, Easygenerator, iSpring Free, entre outros (de Oliveira et al., 2007).

### **3.2. Requisitos, Modelos e Ferramentas utilizadas**

A plataforma conterá os seguintes Requisitos Funcionais (RF):

- RF 1: Cadastrar Usuário
- RF2: Logar no Sistema
- RF3: Cadastrar objeto de aprendizagem
- RF4: Buscar objeto de aprendizagem
- RF5: Visualizar objeto de aprendizagem
- RF6: Exportar objeto de aprendizagem



**Figura 1: Diagrama de Casos de Uso da plataforma “Programe-se”**

Os detalhamentos de cada Caso de Uso seguem abaixo:

- **RF 1: Caso de Uso Cadastrar Usuário;**
  - Tipo: Primário;
  - Objetivo: Cadastrar o usuário que deseja cadastrar e/ou visualizar objeto de aprendizagem ;
  - Ator(es): Usuário;
  - Pré-Condições: -

- Pós-Condições: Usuário cadastrado na plataforma “Programe-Se”;
- Fluxo principal:
  - 1.Usuário clica em “Cadastre-se” na página de login
  - 2.Usuário insere dados desejados para login, senha, nome completo, email, cidade, estado e um perfil. O perfil poderá ser “Estudante”, “Professor” ou “Outros”. Cada usuário possui apenas um perfil. Outros pode incluir, por exemplo, um pai ou mãe que deseja trabalhar atividades de pensamento computacional com o filho.
  - 3.Usuário clica em “Cadastre-se”;
  - 4.Mensagem de sucesso aparece na tela, caso cadastro seja efetuado;
- Fluxos alternativos e de exceções:
 

Passo Alternativo 4: Se o login escolhido pelo usuário já estiver cadastrado, a tela de cadastro permanece e uma mensagem é lançada, solicitando o usuário escolha outro login.
- Notas de implementação: -
- Histórico de versões: versão 1, 25/11, Livia Costa Pereira

● **RF2: Caso de Uso Logar no Sistema;**

- Tipo: Primário;
- Objetivo: Usuário se logar no sistema ;
- Ator(es): Usuário;
- Pré-Condições: Usuário deve possuir um cadastro no sistema
- Pós-Condições: Usuário logado na plataforma “Programe-Se”;
- Fluxo principal:
  - 1.Usuário insere um login e senha na página de login
  - 2. A página inicial da plataforma é carregada, contendo um botão com a opção de cadastrar um objeto, exibindo a lista de objetos



cadastrados, exibindo o título, assuntos pedagógicos, conceitos computacionais e formas pedagógicas abordadas em cada objeto e um botão para visualizar mais detalhes do objeto.

- Fluxos alternativos e de exceções:

Passo Alternativo 2: Se o login e/ou senha inseridos pelo usuário não estiverem cadastrado, a tela de login permanece e uma mensagem de erro é lançada, informando ao usuário o erro.

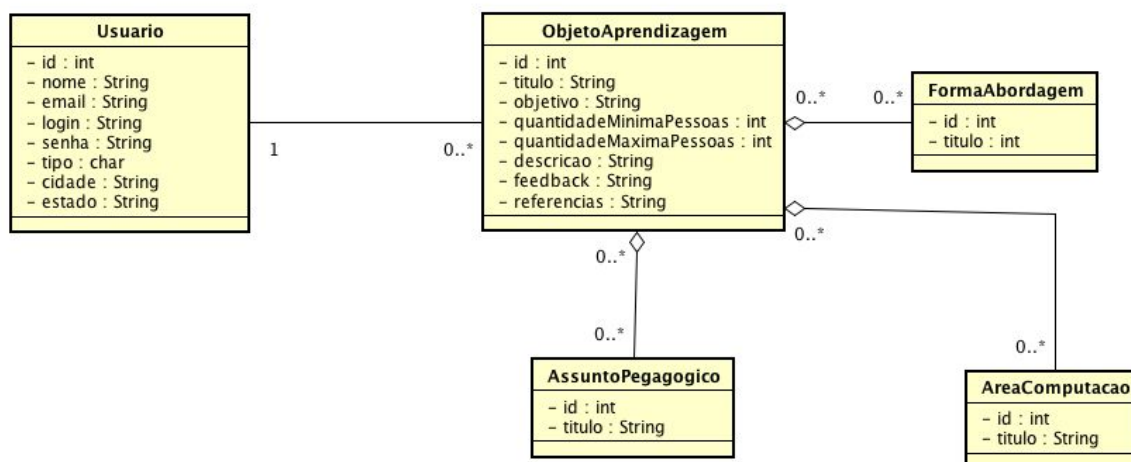
- Notas de implementação: -
- Histórico de versões: versão 1, 25/11, Livia Costa Pereira

- **RF3: Caso de Uso Cadastrar objeto de aprendizagem;**

- Tipo: Primário;
- Objetivo: Cadastrar um objeto de aprendizagem ;
- Ator(es): Usuário;
- Pré-Condições: Usuário deve estar logado na plataforma
- Pós-Condições: objeto de aprendizagem cadastrado na plataforma “Programe-Se”;
- Fluxo principal:
  - 1.Usuário clica em “Cadastrar objeto de aprendizagem” na página de login
  - 2. Sistema carrega formulário de cadastro
  - 3.Usuário insere dados do Objeto, são eles: título, assuntos pedagógicos (podendo ser mais de um), conceitos de pensamento computacional, abordagem, objetivo, quantidade mínima de pessoas, quantidade máxima de pessoas, descrição, *feedback* (da experiência observada no público onde o objeto de aprendizado foi aplicado)e referências dos objetos de aprendizado.
- 3.Usuário clica em “Salvar”;
- 4.Mensagem de sucesso aparece na tela, caso cadastro seja efetuado;

- Fluxos alternativos e de exceções: Se algum elemento obrigatório não for cadastrado
  - 1. Tela de preenchimento permanece, avisando o preenchimento dos campos requeridos.
- Histórico de versões: versão 1, 25/11, Livia Costa Pereira
- **RF4: Caso de Uso Buscar objeto de aprendizagem**
  - Tipo: Primário;
  - Objetivo: Buscar um objeto de aprendizagem ;
  - Ator(es): Usuário;
  - Pré-Condições: Usuário deve estar logado na plataforma
  - Pós-Condições: objeto de aprendizagem cadastrado na plataforma “Programe-Se”;
  - Fluxo principal:
    - 1.Usuário digita palavra chave no campo “Buscar” em “Cadastrar objeto de aprendizagem” na página principal, que filtrará por qualquer campo dos objetos visualizáveis nesta tela (título, assuntos pedagógicos, conceitos computacionais e formas de abordagem).
  - Fluxos alternativos e de exceções: -
  - Histórico de versões: versão 1, 25/11, Livia Costa Pereira
- **RF5: Caso de Uso Visualizar Detalhes do objeto de aprendizagem**
  - Tipo: Primário;
  - Objetivo: Visualizar todos os detalhes de um objeto de aprendizagem cadastrado ;
  - Ator(es): Usuário;
  - Pré-Condições: Usuário deve estar logado na plataforma
  - Pós-Condições: Dados do objeto de aprendizagem disponíveis em tela;
  - Fluxo principal:

- 1.Usuário clica no botão “Mais detalhes” do objeto desejado na tabela de objetos na página principal
  - 2. Detalhes do objeto são exibidos em tela, bem como a opção de voltar para a tela de busca
- Fluxos alternativos e de exceções: -
- Histórico de versões: versão 1, 25/11, Livia Costa Pereira
- **RF6: Caso de Uso Exportar objeto de aprendizagem;**
  - Tipo: Primário;
  - Objetivo: Salvar o objeto de aprendizagem selecionado no formato .txt;
  - Ator(es): Usuário;
  - Pré-Condições: O objeto de aprendizagem deve estar cadastrado no sistema, o usuário deve possuir um login e senha cadastrado
  - Pós-Condições: objeto de aprendizagem estará salvo na pasta Downloads da máquina do Usuário;
  - Fluxo principal:
    - 1.Usuário de loga no sistema
    - 2.Usuário filtra através da busca pelo objeto que deseja exportar
    - 3.Usuário clica em mais detalhes;
    - 4.Usuário visualiza tela com todos os detalhes do objeto de aprendizagem e clica em "Exportar para .txt";
  - Fluxos alternativos e de exceções: -
  - Notas de implementação:
  - Histórico de versões: versão 1, 25/11, Livia Costa Pereira



**Figura 2 - Modelo de classes da plataforma Programa-se**

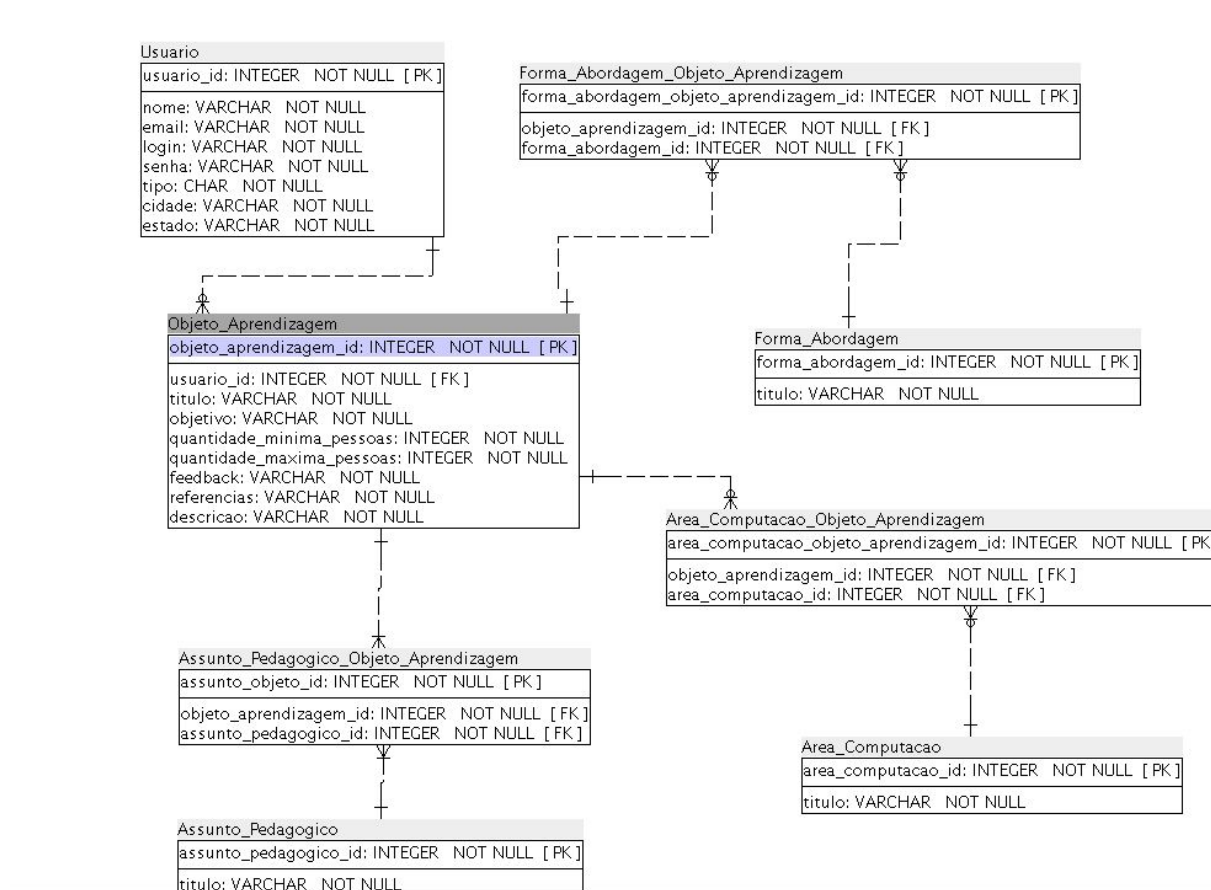
O modelo de classe, conforme consta na figura 2, contém as classes Usuario, ObjetoAprendizado, FormaAbordagem, AssuntoPedagogico, AreaComputacao.

A classe Usuario consiste na entidade usuário da plataforma, de forma que todo o usuário precisa se cadastrar na plataforma para ter acesso aos objeto de aprendizagem. Cada usuário possui um id gerado sequencialmente, um nome, um email, um login, uma senha, um tipo (que pode ser Estudante, Professor ou Outros), uma cidade e um estado.

A classe ObjetoAprendizado consiste na entidade objeto de aprendizagem criado através da plataforma e possui um id gerado sequencialmente, um título, um objetivo, uma quantidade mínima e uma quantidade máxima de pessoas requeridas para executar a atividade proposta no objeto, uma descrição da prática em si, um *feedback* do público cujo o objeto foi aplicado (na percepção do autor do objeto) e as referências necessárias para a execução do objeto, bem como qualquer outra que o autor julgar necessária.

As classes FormaAbordagem, AreaComputacao e AssuntoPedagogico compõem a classe ObjetoAprendizado. FormaAbordagem diz respeito à(s) forma(s) de abordar o OA. Por exemplo, um objeto pode ser abordado através de ‘Jogo’ e ‘Competição’ utilizando ‘Scratch’, ou em ‘Jogo’ e ‘Cooperação’ utilizando o ‘Arduino’. A classe AreaComputacao diz respeito aos conceitos computacionais abordados no objeto. Por exemplo, um objeto pode englobar conceitos

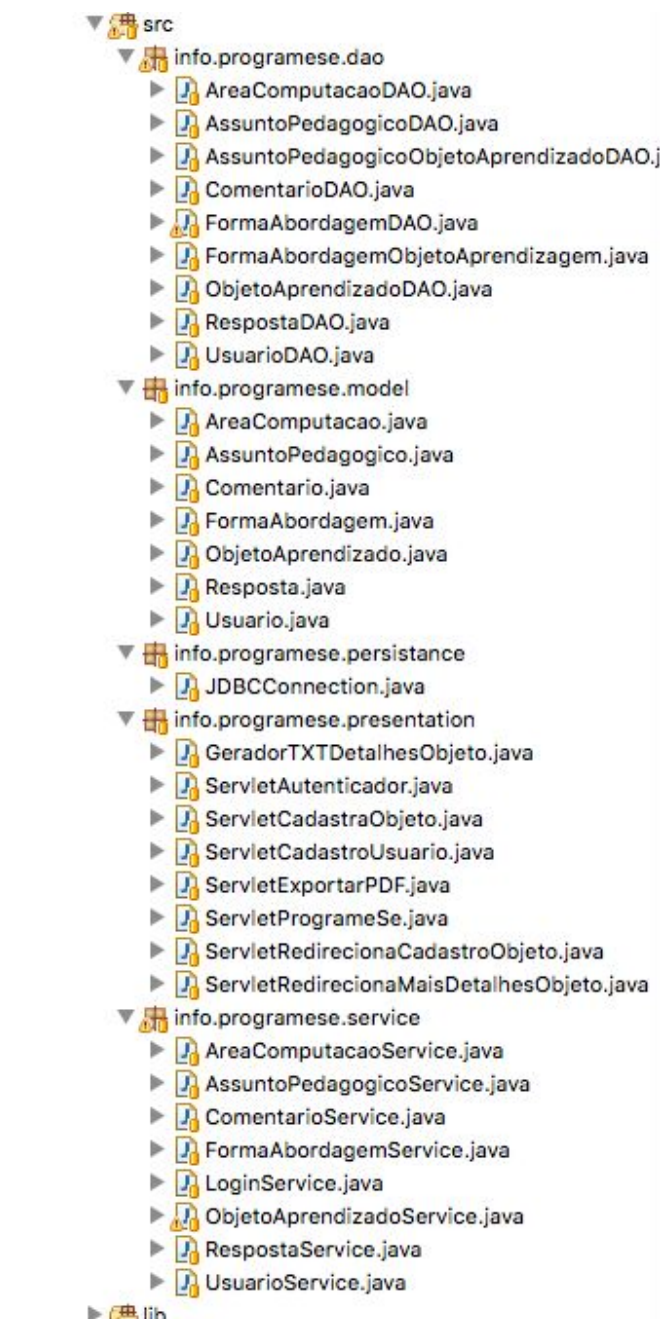
como ‘loop’, ‘variáveis’, ‘orientação a objetos’, entre outros. Por fim, a classe AssuntoPedagogico diz respeito aos assuntos tradicionais abordados na educação básica trabalhados no objeto de aprendizagem. Por exemplo, um objeto interdisciplinar pode englobar assuntos como ‘ciência’, ‘história’ e ‘matemática’.



**Figura 3: Modelo Lógico da plataforma Progame-se**

Como os relacionamentos de ObjetoAprendizado com FormaAbordagem, AreaComputacao e AssuntoPedagogico tinham a cardinalidade n:n, foi criada uma tabela própria para cada um dos três relacionamentos, conforme figura 3. Quanto às ferramentas utilizadas no processo, o desenvolvimento do modelo de classes foi no software Astah Community e o modelo lógico foi realizado no software SQL Power Architect. Foi utilizado o SGBD PostgreSQL e a ferramenta pg-Admin III para criação e gerenciamento das tabelas. Para o desenvolvimento do software, foi utilizada a IDE Eclipse e a linguagem de programação utilizada no desenvolvimento foi Java Web. A arquitetura empregada foi *Model View Controller* através dos

padrões Java Enterprise Edition e a API de conexão com o banco de dados foi JDBC. Por fim, as páginas foram construídas utilizando o framework Bootstrap, a plataforma foi hospedada no Heroku<sup>2</sup> e o projeto foi incluído no [github](https://github.com/livaper/Programe_Se)<sup>3</sup> com o intuito de ser um software *open source*.



**Figura 4: Classes da plataforma Programe-se Criadas no padrão MV**

<sup>2</sup> <http://warm-cliffs-24709.herokuapp.com/login>

<sup>3</sup> [https://github.com/livaper/Programe\\_Se](https://github.com/livaper/Programe_Se)

## 4. Um *overview* sobre a plataforma

Nesta seção, faremos um passeio pelas funcionalidades: cadastro de usuário, login, cadastro de um objeto de aprendizagem, busca e consulta de um objeto e exportação do objeto para o formato .txt .

Ao entrar na plataforma a tela de Login e cadastro de usuário aparece, como segue figura 5.

A imagem mostra a interface de login e cadastro da plataforma Progame-se. No topo, há uma imagem de um cérebro humano com circuitos digitais azuis. Abaixo, o nome "Progame-Se" é exibido em uma fonte preta. O formulário contém duas seções: "Login" com um campo de texto para "Digite o nome de usuario" e "Senha" com um campo de texto para "Digite a Senha". Ambos os campos têm uma borda cinza e um ícone de olho para alternar a visibilidade da senha. Na base do formulário, há dois botões azuis: "Entrar" à esquerda e "Cadastre-se" à direita.

**Figura 5:** Tela de Login e Cadastro de novo Usuário da plataforma Progame-se

Clicando em “Cadastre-se” a tela de cadastro abre, conforme segue figura 6. Todos os campos são obrigatórios para o botão “Cadastre-se”



## Programe-Se

---

**Login**

**Senha**

**Nome**

**Email**

**Cidade**

**Estado**


**Perfil**

Cadastre-se

Figura 6: Tela de Cadastro da plataforma Programe-se



Inserindo os dados do Usuário, e clicando em “Cadastre-se”, um novo usuário é cadastrado, conforme figura 7 e 8 .



The image shows a user registration form for a platform called "Progame-Se". At the top left is a blue brain icon with circuit-like patterns. Below it, the title "Progame-Se" is displayed in a large, bold, black font. The form consists of several labeled input fields: "Login" with the text "felippe", "Senha" with five dots, "Nome" with "Felippe Nardi da Rocha", "Email" with "felippe@nardi.me", "Cidade" with "Rio de Janeiro", "Estado" with "RJ", and "Perfil" with a dropdown menu showing "Estudante". A blue button labeled "Cadastre-se" is positioned at the bottom right of the form.

**Progame-Se**

**Login**

felippe

**Senha**

.....

**Nome**

Felippe Nardi da Rocha

**Email**

felippe@nardi.me

**Cidade**

Rio de Janeiro

**Estado**


RJ

**Perfil**

Estudante

**Cadastre-se**

**Figura 7: Cadastro de Usuário na plataforma Progame-se**



## Programe-Se

Cadastro realizado com sucesso!

---

**Login**

**Senha**

**Entrar** **Cadastre-se**

**Figura 8: Usuário Cadastrado e com dados inseridos para se logar**


Após, a tela inicial é carregada com todos os OA cadastrados, conforme Figura 9.



**Figura 9: Tela Principal da plataforma Progame-se**

O usuário clica em “Cadastrar objeto de aprendizagem”, e é redirecionado para a tela de cadastro conforme figura 10.

# Programe Se



## Cadastrar Objeto de Aprendizado

**Título**

Prática de Construção e Programação de um Robô para perfazer caminho delimitado

**Assunto Pedagógico**

☐ Português ☒ Matemática ☐ Geografia ☐ História ☐ Biologia ☐ Ciências ☐ Sociologia ☐ Filosofia ☐ Artes Visuais ☐ Música ☐ Física ☐ Química ☐ Inglês

**Conceitos de Computação**

☒ Instrução procedural ☐ orientação a objetos ☒ ciclo ☒ sequência ☐ execução em paralelo ☐ eventos ☐ condições ☐ operadores ☐ prática iterativa e incremental ☐ abstração ☐ modularização ☐ estrutura de dados ☐ vetores ☐ árvores ☐ pilha ☐ fila

**Formas de Abordagem**

☒ Jogo ☐ Programação em Blocos ☐ Computação Desplugada ☐ Mecatrônica ☐ Robótica Pedagógica ☐ Competição ☒ Cooperação ☐ Individualização ☐ Scratch ☐ Kodu ☐ Snap ☐ Arduino ☒ LEGO Mindstorm ☐ LEGO Wedo ☐ Robomind

**Objetivo**

Trazer os conceitos de ângulo geométricos alinhando com a introdução do pensamento computacional, além de estimular o trabalho em equipe

**Quantidade Mínima de Pessoas**

5

**Quantidade Máxima de Pessoas**

30

**Descrição**

A prática consiste na montagem e programação do Robô Mindstorm EV3, de forma que ele percorra uma trajetória com obstáculos feitos por garrafas de iogurte recicladas. A idéia é trabalhar em equipes de 5, estimulando o trabalho cooperativo, bem como trabalhar o quanto o robô deve rotacionar em cada trecho da trajetória de forma a não colidir com os obstáculos.

**Figura 10: Tela de Cadastro da plataforma Programe-se com dados do objeto preenchido**

**Objetivo**

Trazer os conceitos de ângulo geométricos alinhando com a introdução do pensamento computacional, além de estimular o trabalho em equipe

**Quantidade Mínima de Pessoas** **Quantidade Máxima de Pessoas**

5 30

**Descrição**

A prática consiste na montagem e programação do Robô Mindstorm EV3, de forma que ele percorra uma trajetória com obstáculos feitos por garrafas de iogurte recicladas. A idéia é trabalhar em equipes de 5, estimulando o trabalho cooperativo, bem como trabalhar o quanto o robô deve rotacionar em cada trecho da trajetória, de forma a não esbarrar no obstáculo.

**Feedback**

A turma alvo do trabalho ficou extremamente envolvida com o projeto ( em um grupo total de 18 alunos, apenas 2 não estavam participando ativamente). Ao final da prática, os alunos conseguiram todos juntos programar o robô para que este percorresse a trajetória, sem que derrubasse o obstáculo.

**Referências**

Coloque aqui as referências que julgar necessário para os alunos acompanharem a prática, bem como as utilizadas na elaboração do material

Salvar

**Figura 11: Tela de Cadastro com dados do objeto preenchido**

Após clicar em “Salvar” segue-se uma tela confirmando o cadastro do objeto (figura 12) , e ao clicar em “Voltar” na mesma, o usuário é redirecionado para a página principal conforme figura 13.

**Programe-Se**

Seja bem vindo(a): Felippe Nardi da Rocha

Cadastro de Objeto de Aprendizagem Realizado Com Sucesso! :)

Voltar

**Figura 12: Tela de confirmação do cadastro do objeto**

# Programe-Se

Seja bem vindo(a): Felipe Nardi da Rocha



## Objetos de Aprendizagem em Pensamento Computacional

Cadastrar Objeto de Aprendizagem +

### Buscar um objeto

Título	Assuntos Pedagógicos	Conceitos Computacionais	Formas de Abordagem	
Construção de Robô para ensino do Filo dos Animais Invertebrados Artrópodes	Ciências	Instrução procedural	Programação em Blocos LEGO Mindstorm	Mais Detalhes
Prática de Construção e Programação de um Robô para perfazer caminho delimitado	Matemática	Instrução procedural ciclo sequência	Jogo Cooperação LEGO Mindstorm	Mais Detalhes

Figura 13: Tela principal com novo objeto cadastrado

Ao clicar em “Mais Detalhes” de um objeto, o usuário poderá ver os metadados do objeto completo, além de poder exportá-lo para .txt conforme figura 14 e 15

# Programe-Se



## Prática de Construção e Programação de um Robô para perfazer caminho delimitado

Autor(a): Felipe Nardi da Rocha

Assunto Pedagógico:

Matemática

Conceitos de Computação:

instrução procedural ciclo sequência

Formas de Abordagem:

Jogo Cooperação LEGO Mindstorm ®

Objetivo:

Trazer os conceitos de ângulo geométricos alinhando com a introdução do pensamento computacional, além de estimular o trabalho em equipe

Quantidade Mínima de Pessoas: 5

Quantidade Máxima de Pessoas: 30

**Figura 14: Detalhes do objeto cadastrado**

**Descrição:**

A prática consiste na montagem e programação do Robô Mindstorm EV3, de forma que ele percorra uma trajetória com obstáculos feitos por garrafas de iogurte recicladas. A idéia é trabalhar em equipes de 5, estimulando o trabalho cooperativo, bem como trabalhar o quanto o robô deve rotacionar em cada trecho da trajetória, de forma a não esbarrar no obstáculo.

**Feedback:**

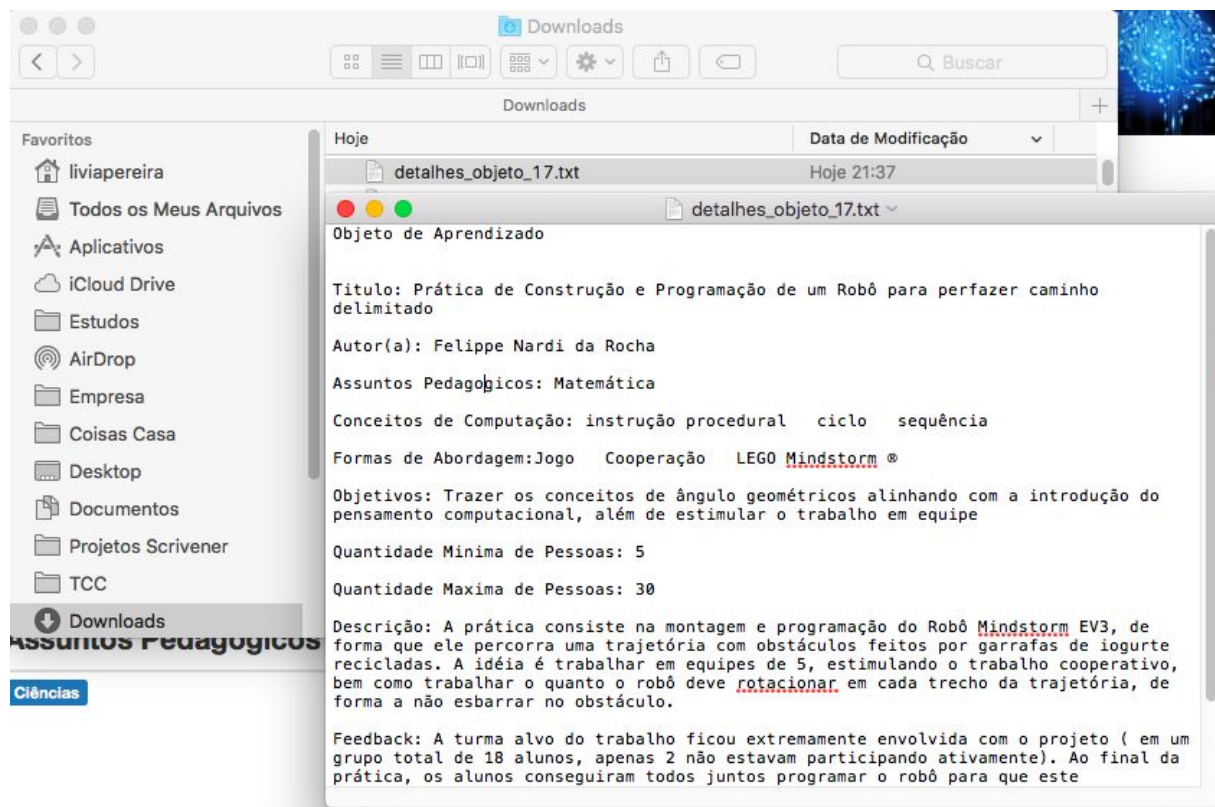
A turma alvo do trabalho ficou extremamente envolvida com o projeto ( em um grupo total de 18 alunos, apenas 2 não estavam participando ativamente). Ao final da prática, os alunos conseguiram todos juntos programar o robô para que este percorresse a trajetória, sem que derrubasse o obstáculo.

**Referências:**

Exportar para .txt

**Figura 15: Detalhes do objeto cadastrado**

Ao selecionar o botão “Exportar para .txt”, um arquivo contendo o detalhamento do objeto de aprendizagem no formato .txt é salvo no computador do usuário, na pasta “Downloads”, conforme figura 14, e o usuário é redirecionado para a página principal .




**Figura 16: objeto de aprendizagem salvo no formato .txt**

Por fim, na tela inicial, o usuário pode utilizar o campo de busca para filtrar um objeto. O campo filtra simultaneamente título do objeto, formas de abordagem, assuntos pedagógicos e conceitos computacionais, conforme figuras 17 e 18.



# Programe-Se

Seja bem vindo(a): Lis



## Objetos de Aprendizado em Pensamento Computacional

Cadastrar Objeto de Aprendizado +

### Buscar um objeto

Titulo	Assuntos Pedagógicos	Conceitos Computacionais	Formas de Abordagem
Construção de Robô para ensino do Filo dos Animais Invertebrados Artrópodes	Ciências	instrução procedural	Programação em Blocos LEGO Mindstorm
			Mais Detalhes

Figura 17: Filtro de objeto de aprendizagem

# Programe-Se

Seja bem vindo(a): Lis



## Objetos de Aprendizado em Pensamento Computacional

Cadastrar Objeto de Aprendizado +

### Buscar um objeto

Titulo	Assuntos Pedagógicos	Conceitos Computacionais	Formas de Abordagem
Prática de Construção e Programação de um Robô para perfazer caminho delimitado	Matemática	instrução procedural ciclo sequência	Jogo Cooperação LEGO Mindstorm
			Mais Detalhes

Figura 18: Filtro de objeto de aprendizagem

## 5. Conclusão

### 5.1. Considerações Finais

Ao contrário do que se pode ter em mente ao primeiro contato com a palavra, o Pensamento Computacional envolve muito mais que pensar algoritmicamente. Embora não exista uma definição unânime para a palavra, o pensamento computacional envolve dimensões cognitivas e sociais, que incluem a formulação de problemas de forma a permitir a utilização de um computador e outras ferramentas para a resolução, organização e análise de dados de forma lógica, representação de dados através de abstrações, aquisição de confiança em lidar com a complexidade, tolerância a ambiguidades e incertezas, capacidade em lidar com problemas abertos, capacidade de comunicar e trabalhar com outras pessoas, entre outros. Sendo o pensamento computacional um conceito com a proposta de alto desenvolvimento cognitivo e social e considerando o alto nível de insatisfação da comunidade acadêmica com o ensino básico tradicional, muitas iniciativas são observadas no intuito de incluir o pensamento computacional como uma disciplina curricular ao longo de todo o ensino básico. A abordagem do pensamento computacional na educação básica traz benefício como melhora da aprendizagem em outras disciplinas, aumento de concentração e foco, melhora na comunicação e expressão de idéias e argumentos, entre outros. Além disso, a inserção do pensamento computacional na educação básica possui um poder de transformação social, por trazer constantes desafios através da resolução de problemas e também pela alta demanda do mercado para profissionais de TI em muitos países, para aqueles que desejam seguir na área. O pensamento computacional pode ser abordado através de jogos, que podem ser competições, cooperações ou individualizações; através da Robótica Pedagógica, na utilização da plataforma Arduino, dos kits LEGO® Mindstorm ou Wedo, entre outros; pela computação desplugada, que não utiliza tecnologias no ensino do pensamento computacional, por programação em blocos como a linguagem Scratch, Kodu, entre outros.

Como forma de divulgação de iniciativas pelo Brasil de trabalhos com o Pensamento Computacional, o presente trabalho apresentou uma plataforma onde é possível a inserção e consulta de objeto de aprendizagem específicos para o trabalho do Pensamento Computacional. A plataforma tem a proposta de estimular cada vez mais professores de todas as áreas do conhecimento a buscar a interdisciplinaridade através do pensamento computacional, onde em uma atividade é possível reunir computação e matérias tradicionais da educação básica. Os objetos educacionais podem ser salvos no formato .txt de forma que podem ser transformados em objetos de aprendizagem no padrão SCORM, a fim de que ocorra melhor leitura por parte de LMS e consequentemente maior reuso, acessibilidade e interoperabilidade.

## **5.2. Limitações do Projeto**

A proposta inicial seria de a ferramenta gerar um objeto de aprendizagem no formato SCORM, porém a proposta se mostrou inviável devido ao tempo restante para a conclusão do trabalho, por conta de dificuldades em relação a integração da aplicação com a API do SCORM. Além disso, o cadastro dos objetos é feito manualmente e a busca pelos objetos é sintática. Por fim, falta uma avaliação da contribuição da plataforma para a divulgação de objetos de aprendizagem

## **5.3. Possibilidades de Trabalhos Futuros**

Como possível desdobramentos futuros, sugere-se a integração da aplicação com a XAPI (considerada a sucessora do SCORM) a fim da geração do objeto de aprendizagem no respectivo formato ser gerado, bem como a adição de novas funcionalidades, como a interação entre os usuários através de mensagens, divulgação de projetos, entre outros. Adicionalmente, sugere-se as seguintes melhorias: cadastro de objetos serem realizados por indexadores automáticos, busca semântica, avaliação da contribuição da plataforma, disponibilização da plataforma através de aplicativo *mobile*. Por fim, o projeto foi incluído no github<sup>4</sup> a fim de abrir para colaboração de qualquer programador interessado em colaborar com o crescimento do projeto .

---

<sup>4</sup> [https://github.com/livaper/Programe\\_Se](https://github.com/livaper/Programe_Se)

## 6. Referências Bibliográficas

- Aguilhar, L. “Ensino de Programação ganha força na Inglaterra: Aulas de linguagem de programação são algumas das novidades do novo currículo escolar”. *Estado de S. Paulo*, <http://link.estadao.com.br/noticias/geral,ensino-de-programacao-ganha-forca-na-inglaterra,10000033370> , 09 Set. 2013. Acessado em Junho de 2016.
- Arantes, F. L.; Ferreira, J. M. L. S. . Uma dinâmica para ensino de conceitos fundamentais de programação. In: I Workshop de Ensino em Pensamento Computacional, Algoritmos e Programação, workshop do CBIE & LACLO 2015, 2015, Maceió, AL. Anais dos Workshops do IV Congresso Brasileiro de Informática na Educação (CBIE 2015), 2015. p. 1218-1225.
- Araújo, A. L. S. O., Andrade, W. L., Guerrero, D. D. S. (2015) “Pensamento Computacional sob a visão dos profissionais da computação: uma discussão sobre conceitos e habilidades”, <http://www.br-ie.org/pub/index.php/wcbie/article/view/6329/4438> . Junho de 2016.
- Bastos, N. ; Carvalho, F. A. H. ; Adamatti, D. F. . Desenvolvimento de habilidades de lógica em estudantes do ensino médio: Uma proposta fundamentada na neurociência. *Revista Brasileira de Informática na Educação*, v. 24, p. 53-65, 2016.
- BRASIL, Parâmetros Curriculares Nacionais (Ensino Médio), (2000). Disponível em : <http://portal.mec.gov.br/seb/arquivos/pdf/blegais.pdf> . Acessado em Novembro de 2016.
- BRASIL, Plano Nacional de Educação 2014-2014, Disponível em: [http://www.planalto.gov.br/CCIVIL\\_03/\\_Ato2011-2014/2014/Lei/L13005.htm](http://www.planalto.gov.br/CCIVIL_03/_Ato2011-2014/2014/Lei/L13005.htm) . Acessado Setembro de 2016.
- Brito, A. L. S., Madeira, C. A. G. (2015) “XP & Skills: gamificando o processo de ensino de introdução a programação”, <http://www.br-ie.org/pub/index.php/wcbie/article/view/6235/4360> . Junho de 2016.

- Cardoso, S., Antonello, R. S. (2015) “Interdisciplinaridade, programação visual e robótica educacional: relato de experiência sobre o ensino inicial de programação”, <http://www.br-ie.org/pub/index.php/wcbie/article/view/6282/4396> . Junho de 2016.
- Carvalho, R.M., Rosa, P.A., Machado, J.V.S., Ribeiro Júnior, J.G., Costa, G.C.B., (2015).“Ferramenta para Auxílio na Aprendizagem de Lógica de Programação em Sistemas Informatizados”, <http://br-ie.org/pub/index.php/wcbie/article/view/6243/4368> . Agosto de 2016.
- CSTA. (2011) Computational Thinking Task Force, Computational Think Flyer. Disponível em: <http://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf>, Acessado Novembro de 2016.
- CSUnplugged. (2015) Terceira Edição. Disponível em [http://csunplugged.org/wp-content/uploads/2015/03/CSUnplugged\\_OS\\_2015\\_v3.1.pdf](http://csunplugged.org/wp-content/uploads/2015/03/CSUnplugged_OS_2015_v3.1.pdf) . Acesso em Outubro de 2016.
- Farias, A. B. ; Andrade, W. L. ; Alencar, R. A. . Pensamento Computacional em Sala de Aula: Desafios, Possibilidades e a Formação Docente. In: I Workshop de Ensino em Pensamento Computacional, Algoritmos e Programação (WAlgProg 2015), 2015, Maceió, AL. Anais dos Workshops do IV Congresso Brasileiro de Informática na Educação (CBIE 2015). Porto Alegre, RS: Sociedade Brasileira de Computação - SBC, 2015. p. 1226-1235.
- Ferreira, C. N., Denardi, R. M., Pozzebon, R. C. B., Voss, G. B. (2015). “Proposta de um Jogo Educacional para o Ensino de Estrutura de Dados”. <http://www.br-ie.org/pub/index.php/wcbie/article/view/6271/4390> . Junho de 2016.
- França, R. S., Tesdeco, P. C. A. R. (2015) “Desafios e oportunidades ao ensino do pensamento computacional na educação básica no Brasil”, <http://www.br-ie.org/pub/index.php/wcbie/article/view/6331> , Junho de 2016.
- Galdino, C. ; Silva Neto, S. R. ; Costa, E. B. . Planejando um serious game para a prática de Programação. In: IV Congresso Brasileiro de Informática na Educação - CBIE 2015 / X Conferência Latino-Americana de Objetos e Tecnologias de Aprendizagem - LACLO 2015, (Evento: I Workshop de Jogos Educativos Digitais Interdisciplinares (JEDI), 2015,

Maceió. Anais dos Workshops do Congresso Brasileiro de Informática na Educação 2015, 2015.

- Heringer, C. Jovem que foi gerente do tráfico no Alemão aos 16 anos agora trabalha em multinacional. Jornal Extra, Rio de Janeiro, 23 de Agosto de 2014. Disponível em: <http://extra.globo.com/casos-de-policial/jovem-que-foi-gerente-do-trafico-no-alemao-aos-16-anos-agora-trabalha-em-multinacional-13704350.html#ixzz4QITGmmLf> . Acessado em Junho de 2016.
- IEEE-LTSC. (2002) Std1484.12.1 IEEE Learning Technology Standard Committee (LTSC) Standard for Learning Object Metadata (LOM).
- Linn, Susan. "Consuming kids: The hostile takeover of childhood." (2004).
- Morin, E. (2001). Os sete Saberes Necessários à Educação do Futuro 3a. ed. - São Paulo - Cortez; Brasília, DF: UNESCO.
- De Oliveira, E. R.; Nelson, M. A. V. ; Ishitani, L. . Ciclo de vida de objetos de aprendizagem baseado no padrão SCORM. In: XVIII Simpósio Brasileiro de Informática na Educação, 2007, São Paulo - SP. Anais do XVIII Simpósio Brasileiro de Informática na Educação. Porto Alegre - RS: SBC, 2007. p. 307-317.
- PAPERT, S. Mindstorms : children, computers, and powerful ideas. New York: Basic Books, 1980.
- Pimentel, C. ; Revoredo, K. ; Sampaio, F. F. . Mecatrônica Educacional Apoiando o Aprendizado de Conceitos de Física e Matemática: Um estudo de caso. In: XXI Workshop de Informática na Escola (WIE 2015), 2015. XXI Workshop de Informática na Escola (WIE 2015).
- Piva Jr., Dilermando; Freitas, R. L. (2011). Estratégias para melhorar os processos de abstração na disciplina de Algoritmos. In XIX Workshop de Educação em Computação (WEI 2011), Anais do XXXI CSBC 2011.
- Qin, H. (2009). Teaching computational thinking through bioinformatics to biology students. In Procs. of SIGCSE, pages 188–191, Chattanooga, TN, EUA.
- Raabe, André Luís Alice; VIEIRA, M. V. ; SANTANA, A. L. M. ; GONCALVES, F. A. ; BATHKE, J. . Recomendações para Introdução do Pensamento Computacional na

Educação Básica. In: 4º DesafIE - Workshop de Desafios da Computação Aplicada à Educação, 2015, Recife. Anais do Congresso Anual da SBC. Porto Alegre: SBC, 2015. v. 1. p. 15-25.

- Ramos, José Luiz. Pensamento Computacional na Escola, no currículo e na aprendizagem. [23 de junho de 2014]. Entrevista concedida a José Moura de Carvalho. Disponível em <https://www.youtube.com/watch?v=BaQqwxTTbQg> . Acessado em Setembro de 2016.
- Rodrigues, R. S. ; Andrade, W. L. ; Guerrero, D. D. S. ; Campos, L. M. R. S. . Análise dos efeitos do Pensamento Computacional nas habilidades de estudantes no ensino básico: um estudo sob a perspectiva da programação de computadores. In: IV Congresso Brasileiro de Informática na Educação (CBIE 2015), 2015, Maceió, AL. Anais do XXVI Simpósio Brasileiro de Informática na Educação (SBIE 2015). Porto Alegre, RS: Sociedade Brasileira de Computação - SBC, 2015. p. 121-130.
- Siqueira, S. W. M. EDUCO: Modelando Conteúdo Educacional. PUC-Rio, 2005. 138p. Tese de Doutorado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.
- USA NATIONAL RESEARCH COUNCIL (NRC). Report of a Workshop of Pedagogical Aspects of Computational Thinking. Washington, D.C.: The National Academies Press. 2011. Disponível em: <https://www.nap.edu/catalog/13170/report-of-a-workshop-on-the-pedagogical-aspects-of-computational-thinking>. Acesso em: Outubro, 2016.
- Wing, J. Computational thinking. Communications of ACM, v. 49, n. 3, p. 33-36, 2006.
- Zanetti, H. A. P., Oliveira, C. L. V. (2015) “Prática de ensino de Programação de Computadores com Robótica Pedagógica e Aplicação de Pensamento Computacional”, <http://www.br-ie.org/pub/index.php/wcbie/article/view/6268> . Acessado em Junho de 2016.