

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DO TRIÂNGULO MINEIRO - CAMPUS UBERABA

FLAMARION ASSIS JERONIMO INÁCIO

**ENSINO DE ALGORITMO E LÓGICA DE PROGRAMAÇÃO:
MODELO CONSTRUTIVISTA AUXILIADO PELO SCRATCH**

UBERABA – MG
2016

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DO TRIÂNGULO MINEIRO - CAMPUS UBERABA

FLAMARION ASSIS JERONIMO INÁCIO

**ENSINO DE ALGORITMO E LÓGICA DE PROGRAMAÇÃO:
MODELO CONSTRUTIVISTA AUXILIADO PELO SCRATCH**

Dissertação apresentada ao Programa de Pós-Graduação nível *Stricto Sensu* - Mestrado Profissional em Educação Tecnológica do Instituto Federal de Educação, Ciência e Tecnologia do Triângulo Mineiro – IFTM – Campus Uberaba - Linha de Pesquisa: II - Tecnologia da Informação e Comunicação (TIC's), Inovação Tecnológica e Mudanças Educacionais, para obtenção do Título de Mestre em Educação Tecnológica.

Orientador:

Prof. Dr. Hugo Leonardo Pereira Rufino

l1e	<p>Inácio, Flamarion Assis Jeronimo Ensino de algoritmos e lógica de programação: modelo construtivista auxiliado pelo Scratch / Flamarion Assis Jeronimo Inácio – 2016. 114 f. : il.</p> <p>Orientador: Prof. Dr. Hugo Leonardo Pereira Rufino. Dissertação (Mestrado Profissional em Educação Tecnológica). Instituto Federal do Triângulo Mineiro- Campus Uberaba- MG, 2016.</p> <p>1. Educação. 2. Algoritmo. 3. Construtivista. I. Inácio, Flamarion Assis Jerônimo. II. Rufino, Hugo Leonardo Pereira.</p> <p style="text-align: right;">CDD 005.1</p> <p>1. Educação. 2. Algoritmo. 3. Construtivismo I. Ensino de Algoritmo e Lógica de Programação: Modelo Construtivista auxiliada pelo <i>Scratch</i>.</p>
-----	--

FLAMARION ASSIS JERONIMO INÁCIO

**ENSINO DE ALGORITMO E LÓGICA DE PROGRAMAÇÃO: MODELO
CONSTRUTIVISTA AUXILIADO PELO SCRATCH**

Data de aprovação 02/05/2016

MEMBROS COMPONENTES DA BANCA EXAMINADORA:

Presidente e Orientador:

Prof. Dr. Hugo Leonardo Pereira Rufino
IFTM - Instituto Federal de Educação, Ciência e Tecnologia do
Triângulo Mineiro - Campus Avançado Uberaba Parque
Tecnológico – MG.

Membro Titular:

Prof. Dr. Alexandre Cardoso
UFU – Universidade Federal de Uberlândia – Faculdade de
Engenharia Elétrica – MG.

Membro Titular:

Prof. Dr. Humberto Marcondes Estevam
IFTM - Instituto Federal de Educação, Ciência e Tecnologia do
Triângulo Mineiro - Campus Uberaba – MG.

Membro Suplente:

Profa. Dra. Paula Teixeira Nakamoto
IFTM - Instituto Federal de Educação, Ciência e Tecnologia do
Triângulo Mineiro - Campus Avançado Uberaba Parque
Tecnológico – MG.

Membro Suplente:

Prof. Dr. Daniel Cintra Cugler
IFTM - Instituto Federal de Educação, Ciência e Tecnologia do
Triângulo Mineiro - Campus Patrocínio – MG.

LOCAL: Auditório do IFTM - *Campus* Avançado Parque Tecnológico/MG - Endereço: Av.
Dr. Florestan Fernandes, 131 - Univerdecidade, Uberaba – MG

INVESTIGADOR

Flamarion Assis Jeronimo Inácio

Professor:

IFTM – Instituto Federal de Educação, Ciência e Tecnologia do Triângulo Mineiro.
Campus Patrocínio - MG.

ORIENTADOR

Prof. Dr. Hugo Leonardo Pereira Rufino

Professor:

IFTM - Instituto Federal de Educação, Ciência e Tecnologia do Triângulo Mineiro.
Campus Avançado Uberaba Parque Tecnológico – MG.

CONTATO:

Avenida Líria Terezinha Lassi Capuano, 255 – Bairro Chácara das Rosas.

Patrocínio – MG – CEP 38740-000 – Brasil.

flamarion@iftm.edu.br

(34) 99100-9570 / 3515-2100

SUMÁRIO

LISTA DE SIGLAS & ABREVIACOES	8
LISTA DE ILUSTRAOES	9
LISTA DE QUADROS	10
LISTA DE GRFICO	11
RESUMO	12
ABSTRACT	13
INTRODUO	14
1 - METODOLOGIAS UTILIZADAS PARA O ENSINO DE ALP	20
1.1 BREVE HISTRICO DOS PROCESSOS DE ENSINO E APRENDIZAGEM	21
1.1.1 METODOLOGIA TRADICIONAL	26
1.1.2 METODOLOGIA BASEADA NO SISTEMA PERSONALIZADO DE ENSINO ...	29
1.1.3 METODOLOGIAS DE ENSINO DE ALP ASSISTIDA POR TICs	31
1.2 COMPARATIVO ENTRE AS METODOLOGIAS APRESENTADAS	37
1.3 OUTRAS PROPOSTAS DE ENSINO DE ALP	39
1.3.1 ERMC2 – Entender, Revisar, Melhorar, Complementar e Construir	39
1.3.2 METODOLOGIAS QUE UTILIZAM FERRAMENTAS WEB PARA APOIO AO ENSINO TRADICIONAL.....	41
1.3.3 MTODOS PARA CONSTRUO DE ALGORITMOS	45
1.3.4 SISTEMA PARA O ENSINO-APRENDIZAGEM DE ALGORITMOS UTILIZANDO UM COMPANHEIRO DE APRENDIZAGEM COLABORATIVO	46
1.3.5 COMPARATIVO DAS OUTRAS PROPOSTAS DE ENSINO DE ALP	49
2 ESTUDO DE CASO – A DISCIPLINA DE ALP EM UMA INSTITUIO DE ENSINO SUPERIOR E PBLICA	51
2.1 ANLISE QUALI-QUANTITATIVA	52
3 PROPOSTA DE UTILIZAO DA FERRAMENTA COMPUTACIONAL <i>SCRATCH</i> AUXILIANDO NO ENSINO DE ALP	57
3.1 A FERRAMENTA COMPUTACIONAL CONSTRUTIVISTA <i>SCRATCH</i>	58
3.2 A PROPOSTA METODOLGICA DO <i>SCRATCH</i>	59
3.3 TRABALHOS CORRELATOS – UTILIZAO DA FERRAMENTA <i>SCRATCH</i> 61	
3.4 <i>SCRATCH</i> – DETALHAMENTO DE SUA UTILIZAO NA DISCIPLINA	63
3.4.1 DETALHAMENTO DA 2ª SEMANA	64
3.4.2 DETALHAMENTO DA 3ª SEMANA	72
3.4.3 DETALHAMENTO DA 4ª SEMANA	73

4	RESULTADOS DA APLICAÇÃO DA FERRAMENTA <i>SCRATCH</i> NA DISCIPLINA DE ALP	74
4.1	RESULTADO FINAL DA DISCIPLINA: APROVADOS, REPROVADOS POR NOTA E REPROVADOS POR FREQUÊNCIA	75
4.2	RESULTADOS DA APLICAÇÃO DOS QUESTIONÁRIOS: ALUNO, PROFESSOR E COORDENADOR DE CURSO	76
4.3	ANÁLISE CRÍTICA – RESULTADOS OBTIDOS PELA UTILIZAÇÃO DA FERRAMENTA <i>SCRATCH</i> NA DISCIPLINA DE ALP	82
4.3.1	ANÁLISE CRÍTICA – ÍNDICES DE APROVAÇÃO E REPROVAÇÃO	82
4.3.2	ANÁLISE CRÍTICA – Questionário dos alunos.....	83
4.3.3	ANÁLISE CRÍTICA – Questionário do professor.....	86
4.3.4	ANÁLISE CRÍTICA – Questionário do coordenador.....	88
	CONCLUSÕES	90
	REFERÊNCIAS	93
	ANEXO A – PLANO DE AULA DO PROFESSOR	100
	APENDICE A – QUESTIONÁRIO APLICADO AOS ALUNOS	105
	APENDICE B – QUESTIONÁRIO APLICADO AO PROFESSOR.....	108
	APÊNDICE C – QUESTIONÁRIO APLICADO AO COORDENADOR	111

LISTA DE SIGLAS & ABREVIACOES

ADS	Anlise e Desenvolvimento de Sistemas.
ALP	Algoritmo e Lgica de Programaco.
CAPES	Coordenao de Aperfeioamento de Pessoal de Nvel Superior
EAD	Ensino  Distncia.
ERMC²	Entender, Revisar, Melhorar, Complementar, Construir.
Ideb	ndice de Desenvolvimento da Educao Bsica
IESB	Instituto de Educao Superior de Braslia
IFTM	Instituto Federal de Educao, Cincia e Tecnologia do Tringulo Mineiro.
Intercom	Sociedade Brasileira de Estudos Interdisciplinares da Comunicao
iVProg	Programao Visual e Interativa na Internet
MBR	Modelo Baseado em Restrioes.
MIT	<i>Massachusetts Institute of Technology.</i>
PPC	Projeto Pedaggico do Curso
PSE	Sistema Personalizado de Ensino
PSI	<i>Personalized System of Instruction</i>
SCIELO	<i>Scientific Electronic Library Online</i>
SENAC	Servio Nacional de Aprendizagem Comercial
SENAI	Servio Nacional de Aprendizagem Industrial
SERPRO	Servio Federal de Processamento de Dados
SISU	Sistema de Seleo Unificada
STT's	Solicitaoes Tcnicas e Tecnolgicas.
TIC's	Tecnologias de Informao e Comunicao.
UCB	Universidade Catlica de Braslia
UniCEUB	Centro Universitrio de Braslia
WEB	<i>Websites</i>

LISTA DE ILUSTRAÇÕES

Figura 1	Evolução da formação do conhecimento humano	23
Figura 2	Paradigmas de Linguagens de Programação	26
Figura 3	Fluxograma da metodologia Tradicional	28
Figura 4	Informática na Educação: Instrucionismo	32
Figura 5	Imagem do <i>Software SuperLogo</i> – uma das versões do Logo	36
Figura 6	Ciclo de avaliação formativa	42
Figura 7	Sequência de questões estruturadas para a ferramenta <i>WEB</i>	43
Figura 8	Fluxograma funcional da ferramenta <i>WEB</i>	44
Figura 9	Fluxograma com a visão do aluno da ferramenta <i>WEB</i>	44
Figura 10	Sequência de Interfaces da ferramenta <i>WEB</i>	45
Figura 11	Arquitetura de funcionamento do <i>AlgoLC</i>	48
Figura 12	Utilização do Jogo <i>Light Bot</i>	65
Figura 13	Versão <i>on-line</i> e <i>offline</i> do software <i>Scratch</i> .	67
Figura 14	Tela inicial de abertura do <i>Scratch</i> .	68
Figura 15	Plano cartesiano de representação da Tela Simulador do Ecrã.	69
Figura 16	Movimentação do <i>Scratch</i> para a posição (-150,100).	69
Figura 17	Tela de desenho do Plano de Fundo do <i>Scratch</i> .	70
Figura 18	Tela de Edição do personagem <i>Scratch</i> .	71
Figura 19	Tela de indicação do ângulo de movimentação do <i>Scratch</i> .	71
Figura 20	Interação do <i>Scratch</i> através de balões – utilizando comando “diga”.	72

LISTA DE QUADROS

Quadro 1	Número de alunos matriculados, aprovados, reprovados na disciplina de ALG, entre os anos 2010 e 2015.	38
Quadro 2	Número de alunos matriculados, aprovados, reprovados na disciplina de ALG, entre os anos 2010 e 2015.	50
Quadro 3	Número de alunos matriculados, aprovados, reprovados na disciplina de ALG, entre os anos 2010 e 2015.	54
Quadro 4	Professores que ministraram a disciplina de ALP de 2010/2 a 2015/1	55
Quadro 5	Plano de aula para utilização da disciplina <i>Scratch</i>	64
Quadro 6	Exemplos de problemas apresentados aos alunos	72
Quadro 7	Número de alunos matriculados, aprovados, reprovados por nota e reprovados por frequência na disciplina de ALG, em 2015/2.	76
Quadro 8	Resultado do questionário aplicado aos alunos ao final da disciplina, respostas de múltiplas escolhas.	76
Quadro 9	Resultado do questionário aplicado aos alunos, ao final da disciplina, respostas dissertativas.	77
Quadro 10	Resultado do questionário aplicado ao professor, ao final da disciplina.	78
Quadro 11	Resultado do questionário aplicado ao coordenador do curso	80
Quadro 12	Resumo percentual das informações coletadas do questionário - aluno	83

LISTA DE GRÁFICO

Gráfico 1	Curvas de alunos aprovados, reprovados por frequência e reprovados por nota.	82
------------------	--	----

INÁCIO, F. A. J. **Ensino de Algoritmos e Lógica de Programação: Modelo Construtivista auxiliado pelo Scratch**. Uberaba: IFTM - Instituto Federal de Educação, Ciência e Tecnologia do Triângulo Mineiro – Campus Uberaba, 2016. Dissertação Mestrado Profissional *Strictu Senso* – Área de concentração: Educação, Linha de Pesquisa: Tecnologia da Informação e Comunicação (TICs), Inovação Tecnológica e Mudanças Educacionais). Orientador: Prof. Dr. Hugo Leonardo Pereira Rufino.

RESUMO

Os cursos da área de Informação e Comunicação¹ possuem, em suas matrizes curriculares, disciplinas que exigem dos alunos, logo nos primeiros períodos, cognição desenvolvida para a interpretação e resolução de problemas relacionados ao raciocínio lógico. Nos cursos abrangidos pela subárea de Informática, a disciplina de ALP (Algoritmo e Lógica de Programação) é o primeiro desafio enfrentado pelos alunos, testando suas habilidades cognitivas na criação de algoritmos dentro de uma linguagem de programação. Muitos alunos que ingressam em cursos técnicos e superiores desta área, possuem grandes dificuldades na realização de atividades que necessitam do raciocínio lógico. Uma das consequências desta dificuldade é o aumento dos índices de retenção e evasão na disciplina de ALP. Este grande índice de retenção e evasão produz desmotivação na realização da disciplina, tornando essencial a metodologia de ensino utilizada pelo professor para garantir que os objetivos da disciplina sejam cumpridos. Este trabalho tem por finalidade apresentar os resultados referentes às pesquisas realizadas sobre as metodologias de ensino, atualmente utilizadas na disciplina de ALP, descrevendo aspectos positivos e negativos de cada uma delas. Posteriormente, estes resultados serão comparados com as metodologias utilizadas na disciplina de ALP de uma instituição pública de ensino, coletando os impactos causados pelas metodologias utilizadas neste cenário. Conforme descrito no pensamento construtivista de Jean Piaget (1988), é importante que o docente, ao tentar desenvolver a cognição nos alunos, leve em consideração a existência/modificação das estruturas mentais dos alunos, passando por todas as fases necessárias ao desenvolvimento cognitivo. A utilização de ferramentas de tecnologia da informação pode potencializar o aprendizado dos alunos, criando ambientes mais atrativos e estimulantes para que o aluno desenvolva a cognição. Desta forma, propõe-se, aplicar e coletar os resultados da utilização da ferramenta computacional *Scratch*, uma evolução do Logo², como apoio ao ensino da lógica de programação na disciplina de ALP. Alicerçada no Construcionismo de Seymour Papert (1986), que propõe a utilização do computador como um instrumento a ser ensinado, foi apresentado aos alunos; mais uma ferramenta de tecnologia da informação que favorece o desenvolvimento cognitivo, motivando alunos e professores, durante a difícil tarefa de desenvolver a cognição dentro da disciplina de ALP.

Palavras chave: Educação. Algoritmo. Construtivismo

¹ Área definida pelo Catálogo de Cursos Superiores de Tecnologia – MEC, 2010.

² “Logo” é uma linguagem de programação interpretada, voltada para crianças, jovens e adultos.

INÁCIO, F. A. J. **Teaching ALP: Constructivist Model using Scratch**. Uberaba: IFTM - Instituto Federal de Educação, Ciência e Tecnologia do Triângulo Mineiro – Campus Uberaba, 2016. Thesis (Master's Degree Strictu Sense - Concentration area: Education, Research Line: Information and Communication Technology (ICT), Technological Innovation and Education Removals). Oriented by: Teacher. Dr. Hugo Leonardo Pereira Rufino.

ABSTRACT

The courses of the area "Information and Communication"³ have in their curriculum matrices disciplines that require the students, in the early periods, cognition developed for interpretation/problem solving related to logical reasoning. In the courses covered by sub-area of Information Technology, the discipline of ALP - Algorithm and Logic Programming - is the first challenge faced by students, testing their cognitive skills in creating algorithms in a programming language. Many students enrolled in technical courses and higher this area have great difficulty in performing activities that require logical thinking. One consequence of this difficulty is increased retention and dropout rates in ALP discipline. This great retention and dropout rate produces demotivation in achieving discipline becoming essential teaching methodology used by the teacher to ensure that the course objectives are met. This study aims to present the results of the research on teaching methodologies currently used in the discipline of ALP, describing positive and negative aspects of each. Subsequently, these results will be compared with the methodologies used in ALP discipline of a public education institution, collecting the impacts caused by the methodologies used in this scenario. As described in constructivist thought of Jean Piaget, it is important that teachers, when trying to develop cognition leads us students into account the existence / modification of the mental structures of the students through all the stages necessary for cognitive development. The use of information technology tools can enhance student learning by creating more attractive and stimulating environment for the student to develop cognition. Thus, we propose, apply and collect the results of using the computational tool Scratch, an evolution of the "Logo"⁴ to support the programming logic of teaching in the discipline of ALP. Founded on Constructionism Seymour Papert, proposing to use the computer as a tool to be taught, presented to students more an information technology tool that promotes cognitive development, motivating students and teachers during the difficult task of developing cognition within the ALP discipline.

Key words: Education. Algorithm. Constructivism.

³ Area defined by the Higher Course Catalog Technology - MEC, 2010.

⁴ **Logo** is an interpreted programming language, aimed at children, youth and adults.

INTRODUÇÃO

Há um tempo em que é preciso abandonar as roupas usadas, que já tem a forma do nosso corpo, e esquecer os nossos caminhos, que nos levam sempre aos mesmos lugares. É o tempo da travessia: e, se não ousarmos fazê-la, teremos ficado, para sempre, à margem de nós mesmos (**Fernando Pessoa**).

Dentro do exercício da docência, desde 2003, em instituições renomadas de ensino como, por exemplo, SENAC (Serviço Nacional de Aprendizagem Comercial), SENAI (Serviço Nacional de Aprendizagem Industrial), Instituto Federal de Educação Ciência e Tecnologia e Universidade Particular, houve a oportunidade, em conjunto com as equipes de profissionais de cada uma destas instituições, de realizar, além da ministração de aulas, assessorias e STT's (Solicitações Técnicas e Tecnológicas) nas indústrias e empresas do Alto Paranaíba, em Patos de Minas - MG, Patrocínio - MG e cidades vizinhas. A ministração de aulas, nas modalidades de aprendizagem, qualificação, ensino técnico e ensino superior, possibilitou conhecer o perfil dos alunos, nas mais variadas modalidades de ensino. Em todas as instituições citadas, foram trabalhadas disciplinas iniciais dos cursos da área de Informação e Comunicação, Engenharia e áreas Tecnológicas. Estas disciplinas tinham como característica básica a presença de conteúdos que exigiam como pré-requisito o desenvolvimento cognitivo, relacionado ao raciocínio lógico apurado. A finalidade principal destas disciplinas, além de apresentar a sintaxe e semântica de uma linguagem de programação específica, era desenvolver o raciocínio lógico dos alunos, visando o entendimento e a criação de algoritmos para a resolução de problemas dentro disciplina.

Analisando o cenário discente nestas instituições e comparando com os trabalhos de Maciente e Araújo (2011) sobre os cursos de níveis técnicos e superiores nas escolas do Brasil, percebeu-se que, grande parte dos alunos que iniciam os cursos da área de Ciências exatas e da terra, além de cursos voltados para a área tecnológica, desiste, logo nos primeiros períodos. Santos & Costa (2006) apresentam várias situações que explicam os altos índices de evasão e retenção nas disciplinas introdutórias de programação e desenvolvimento de algoritmos, foco desta dissertação.

Santos & Costa (2006) afirmam que a disciplina de Programação, integrante da formação básica em Ciência da Computação é exigência da estrutura curricular, ditadas pelas Diretrizes Curriculares do MEC. O conteúdo engloba o ensino de linguagens de programação: conceitos, princípios, modelos de programação, classificação e pesquisas de dados. Por isso, a compreensão do raciocínio lógico influencia no alto índice de reprovação nas disciplinas de Algoritmos e Programação, chegando até à desistência do aluno em relação ao curso. “[...] a didática ou falta de metodologia de ensino dificulta o aprendizado dos novos e diversos conceitos” (SANTOS & COSTA, 2006, p. 2).

Dessa forma, a disciplina de Algoritmo e Lógica de Programação, sendo essencial à produção dos conceitos e princípios da formação básica dos cursos da subárea de Informática, é uma das primeiras disciplinas inseridas nas matrizes curriculares destes cursos. Rapkiewicz

et al (2006) atribui a disciplina de ALP à responsabilidade de abordar os princípios da Lógica de Programação, com o objetivo de desenvolver a capacidade de análise e resolução de problemas, transcrevendo-os em forma de algoritmos computacionais. Devido à grande importância da disciplina de ALP no contexto dos cursos da área de Informação e Comunicação, é importante dedicar uma atenção especial à disciplina, pois na maioria das matrizes curriculares, existe o desafio encontrado pelos alunos, ao ingressarem em curso da área de Informação e Comunicação e Áreas Tecnológicas. Testando, assim, seu desenvolvimento cognitivo na formação do conhecimento, por meio do raciocínio lógico. Muitas vezes, os alunos não têm dificuldade com a semântica ou sintaxe da linguagem de programação, mas possuem grandes dificuldades na estruturação lógica, necessária à formação dos algoritmos.

Consultando sites que possuem artigos disponíveis, on-line, no portal da CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) e SCIELO (*Scientific Electronic Library Online*), verificou-se que existem muitos artigos escritos com a criação de ambientes de aprendizagem para Física, Matemática, entre outras unidades curriculares, presentes em pesquisas como: Forte & Kirner, (2009); Sales *et al.* (2008); Aviz Junior (2007) e Martins; Oliveira; Guimarães (2014). Porém, poucos autores dissertam sobre a criação de estratégias para o desenvolvimento de pensamento cognitivo, voltado ao crescimento do raciocínio lógico significativo, (FALCKEMBACH, ARAÚJO, 2006). Fonseca (1995) ressalta que, variadas pesquisas sobre as dificuldades de aprendizagem, por serem “complexas e pouco consistentes” (FONSECA, 1995), não chegam a um conteúdo, visto que não apresentam respostas práticas de como ensinar e fazer com que o aluno se interesse pela investigação.

Visando apresentar novas estratégias que estimulem a criação/modificação das estruturas mentais dos alunos na disciplina de ALP, partindo do epistemólogo Jean Piaget (1988), que afirma que, à medida que as crianças se desenvolvem intelectualmente, criam esquemas mentais para a resolução de problemas das situações cotidianas.

Foi proposta ao professor da disciplina, a utilização de ferramentas computacionais construtivistas, que exercitassem o desenvolvimento cognitivo para a interpretação/resolução de problemas relacionados ao raciocínio lógico. A utilização destas ferramentas computacionais introduziria a disciplina de ALP, possibilitando aos alunos conhecerem ferramentas que estimulem seu desenvolvimento cognitivo, anteriormente aos conteúdos constantes no ementário da disciplina. Suscitar o desenvolvimento de novos esquemas lógicos, a partir de esquemas atuais, consolidados em cada indivíduo, foi um desafio. A

criação de estratégias de ensino que produzam desenvolvimento cognitivo, conforme situações que circundam o indivíduo, levando-o, de forma ativa, a desenvolver o raciocínio lógico, é uma excelente forma de possibilitar aos alunos ingressantes na disciplina de ALP, uma oportunidade de conseguirem desenvolver seus estudos de forma agradável.

Porém, a criação de estratégias de ensino, omitindo-se a figura do professor, não seria eficiente, conforme Piaget (1988). O autor declara que alguns pesquisadores eram adeptos que não houvesse necessidade de professores em sala de aula, deixando para as crianças, a escolha do que aprender em cada momento. Outros, receavam que isso não trouxesse resultados. Piaget (1988) percebe que o educador continua indispensável, no sentido de animar e criar dispositivos capazes de levar os alunos à reflexão e busca de soluções. “[...] o que se deseja é que o professor [...] estimule a pesquisa e o esforço, ao invés de se contentar com a transmissão de soluções prontas” (PIAGET, 1988).

Dessa forma, é importante que o professor acompanhe o andamento destas estratégias de ensino, orientando o aluno no desenvolvimento de sua cognição. Conforme Nunes (apud GARCIA, 2006), outro fator importante a destacar é que o professor, quando faz uso dos objetos de aprendizagem, visando o desenvolvimento do raciocínio lógico pode planejar suas aulas, fazendo uso deles, conseguindo maior flexibilidade para se adaptar ao ritmo e ao interesse dos alunos, mantendo, assim, seus objetivos de ensino. Uma das vantagens em utilizar objetos de aprendizagem na forma de ferramentas computacionais é que, estes propiciam a curiosidade para resolver problemas, ampliam conhecimentos e despertam interesses.

Analisando todo o contexto apresentado (importância da disciplina de ALP, altos índices de evasão e retenção, dificuldade no raciocínio lógico, proposição de utilização de ferramentas computacionais construtivistas) e comparando os trabalhos de Paulo Freire e Piaget (apud MARTIN, 2007) nota-se que, apesar das várias estratégias criadas pelos professores, a forma de aprendizagem utilizada pelas escolas, durante o ensino fundamental, ainda não desenvolve, de forma eficiente, o raciocínio lógico nos alunos. Piaget (apud TERRA, 2014) separa o desenvolvimento humano em quatro estágios, sendo que no último estágio, de 12 anos em diante, afirma que o aluno possui desenvolvimento intelectual, maturação biológica, para formar esquemas conceituais abstratos e, por meio deles, executar operações mentais dentro de princípios da Lógica Formal. Porém, a forma como o conhecimento é levado a estes alunos, criticada por ambos os autores, não ajuda a desenvolver um indivíduo com pensamento ativo; pelo contrário, a Educação proposta aos alunos é uma “educação de repetição”. Percebeu-se que, nos últimos anos, este modelo de ensino sofreu

várias alterações, sendo que muitas propostas foram realizadas a partir da essência deste método. Talvez, o método puro não exista mais (LEÃO, 1999). Nesse sentido, a base do ensino, nas escolas atuais, ainda tem muito a melhorar, visto os problemas enfrentados pelos alunos ingressantes, na disciplina de ALP, nos ensinos técnicos e superiores.

Este trabalho se justifica, devido aos altos índices de evasão e retenção, apresentados nas disciplinas introdutórias de Programação e desenvolvimento de algoritmos, dos cursos da área de Ciências Exatas e da Terra, além dos cursos Tecnológicos. Conforme apresentado por Santos & Costa (2006), são vários os motivos causadores da evasão e retenção. Logo, este trabalho estuda o cenário e propõe estratégias, visando melhorar o cenário observado. A grande dificuldade que os alunos ingressantes em cursos regulares, tanto a nível técnico quanto superiores, demonstra, no cumprimento das unidades curriculares, que exigem raciocínio lógico, instiga a proposição de novas estratégias educacionais.

O objetivo geral deste trabalho consiste em realizar um estudo dos aspectos positivos e negativos das principais metodologias utilizadas no ensino de ALP, fazendo um comparativo com as metodologias de ensino utilizadas por uma instituição de ensino superior e pública, durante a realização da disciplina de ALP das turmas de Análise de Desenvolvimento de Sistemas de 2010 até 2014. Após a análise, aplicar e coletar os resultados da utilização de uma ferramenta computacional construtivista, o *Scratch*, na disciplina de ALP, da instituição, visando melhorar o aprendizado na disciplina, conforme proposto por Jean Piaget (1988) e Seymour Papert (1986). Acredita-se que será possível promover nos alunos um melhor desenvolvimento do pensamento cognitivo, voltado ao raciocínio lógico, necessário à construção de algoritmos. Este trabalho tem por objetivos: 1. Realizar um estudo de caso do ensino de ALP, de uma instituição de ensino superior e pública. 2. Propor a utilização de uma ferramenta computacional construtivista que auxilie o aluno no desenvolvimento da lógica de programação e construção de algoritmos, durante a disciplina de ALP. 3. Motivar alunos e professores no ensino/aprendizagem da disciplina de ALP.

A estrutura desta dissertação está disposta de forma a facilitar o entendimento do estudo realizado. No capítulo I, foi apresentado ao leitor o resultado de pesquisas de caráter descritivo e exploratório em livros de leitura corrente, livros de referência e publicações periódicas, visando identificar as principais metodologias de ensino de ALP utilizadas em instituições públicas e privadas (GIL, 2010). No capítulo II, será apresentado o resultado de um estudo de caso sobre o ensino de ALP, em uma instituição de ensino superior e pública. Será apresentada, neste capítulo, a análise da coleta de dados, na forma quali-quantitativa (GIL, 2010). Como sugestão, para melhorar o desempenho e motivação dos alunos na

disciplina de ALP da instituição de ensino superior e pública do capítulo II, será utilizada a ferramenta computacional construtivista *Scratch*. Dessa forma, no capítulo III, apresentar-se-á a ferramenta computacional *Scratch* e como se realizou, na instituição de ensino, citada anteriormente. O capítulo IV tem por objetivo apresentar os resultados da aplicação da ferramenta computacional *Scratch*, assim como a análise crítica dos questionários aplicados aos alunos, ao professor e ao coordenador. Por fim, na conclusão, serão expostas as discussões finais a respeito de todo o trabalho realizado. Também serão apresentadas sugestões para a instituição, sobre o futuro da disciplina de ALP, na instituição de ensino estudada.

1 - METODOLOGIAS UTILIZADAS PARA O ENSINO DE ALP

A principal meta da educação é criar homens que sejam capazes de fazer coisas novas, não simplesmente repetir o que outras gerações já fizeram. Homens que sejam criadores, inventores, descobridores. A segunda meta da educação é formar mentes que estejam em condições de criticar, verificar e não aceitar tudo que a elas se propõe (**Jean Piaget**).

Foram apresentados, neste capítulo, os pressupostos teóricos das metodologias de ensino de ALP, mais utilizadas em livros, artigos e revistas, da área de Informação e Comunicação. Também, são apresentadas algumas propostas interessantes (ERMC², Ferramentas WEB, Companheiro de aprendizagem e outros), de mudanças destas metodologias de ensino existentes, com o objetivo de aumentar o nível de aprendizado dos alunos.

1.1 BREVE HISTÓRICO DOS PROCESSOS DE ENSINO E APRENDIZAGEM

Quando se trata do processo de ensino-aprendizagem e das metodologias existentes de ensino, de qualquer área de conhecimento, é importante lembrar que vários fatores contribuem para a escolha da metodologia utilizada, neste processo, por parte do docente. Para aqueles que desejam aprofundar no estudo destas metodologias de ensino, torna-se imprescindível entender que existem diversas teorias que explicam a formação do conhecimento humano. A forma com que o professor define a criação do conhecimento dos alunos afeta, drasticamente, na escolha do método utilizado. Todorov *et al.* (2009), afirmam que:

A prática de reunir crianças, adolescentes, jovens e adultos ao redor de um mestre, conhecedor dos segredos do mundo e das práticas mais apropriadas para ensiná-los, remonta às civilizações mais antigas da história humana. O ensino na antiga Roma, por exemplo, principalmente após as propostas reformadoras do educador Marco Fábio Quintiliano, já apresentava muitas das características que ainda hoje podemos encontrar em nossas práticas educacionais: a criança deve começar a frequentar a escola o mais cedo possível; [...] deve-se instruir os alunos simultaneamente em diversas disciplinas.

Conforme apresentado pelos autores acima citados, para o povo da época de Fábio Quintiliano, a figura do mestre era entendida, pela comunidade romana, como a melhor metodologia de ensino. Um mestre, detentor de todo o conhecimento, era o único capaz de repassar o conhecimento aos aprendizes. Marco Fábio Quintiliano, entendendo que esta é a melhor definição da formação do conhecimento, propôs uma série de mudanças no processo de ensino/aprendizagem (TODOROV *et al.*, 2009). Desta forma, tem-se que as diversas metodologias de ensino existentes na atualidade, possuem suas origens nas principais teorias de formação do conhecimento, existentes na história da humanidade. Não é foco desta dissertação, aprofundar nas teorias de formação do conhecimento. A seguir, são apresentadas as principais características das correntes filosóficas que produziram os conceitos fundamentais da formação do conhecimento para o desenvolvimento das metodológicas de ensino, utilizadas pelos docentes, para o ensino de Algoritmo e Lógica de Programação, também apresentadas nesta dissertação.

Desde o surgimento da filosofia, definida pelo amor ao saber e, particularmente, pela investigação das causas e dos seus efeitos (FERREIRA, 2014), passando pelos grandes filósofos, como Tales de Mileto, Anaximandro de Mileto, Heráclito de Éfeso e, tantos outros. A teoria do saber é discutida, firmando-se correntes filosóficas, que davam ao ser humano, uma nova perspectiva da realidade e da formação do conhecimento. Conforme relatado por Xenofonte e por Platão, a vida de Sócrates trouxe uma renovação na cultura empreendida pelos sofistas⁵, surgindo novas formas de ensinar e aprender (SANTOS 2014). Sócrates possuía uma técnica de ensino chamada maiêutica (parto de ideias), que, por meio de questionamentos, ensinava os indivíduos a pensar de forma crítica, duvidando até mesmo de suas próprias ideias. Após a morte de Sócrates, seus contemporâneos, Platão e Aristóteles, continuaram a desenvolver seus pensamentos e o estudo da Metafísica ganhou grandes proporções (VESCE, 2015). Com Aristóteles, dá-se o nascimento da Lógica, com a construção de uma sofisticada teoria de argumentos, cujo núcleo é a caracterização e análise dos chamados silogismos (SANTOS 2014). O estudo dos silogismos faz parte do estudo da Lógica até os dias atuais. O processo de ensino-aprendizagem começou a tomar novas proporções, vista a intensa necessidade de definição, característica do contexto histórico.

Durante a Idade Moderna, surgem dois movimentos filosóficos que debatem entre si a origem da formação do conhecimento. De um lado, o Racionalismo, defendido por René Descartes (1596-1650), uma corrente filosófica que enfatiza o papel da razão como fundamento para o conhecimento da realidade. Por outro lado, o Empirismo, de David Hume (1711-1776) e John Locke (1632-1704), uma corrente filosófica que não admite a existência de nenhuma substância, considerando a realidade como composta exclusivamente de fenômenos e das percepções e ideias que se formam destes (SANTOS, 2014). Essas correntes filosóficas serviram de base para as principais técnicas de ensino-aprendizagem existentes, incluindo o método tradicional de ensino, que será apresentado na próxima seção. É característica desta metodologia, que o docente realize uma série de experimentos demonstrativos, deduzindo que, por meio destas experimentações e repetições de seus exemplos, os alunos desenvolvam sua cognição.

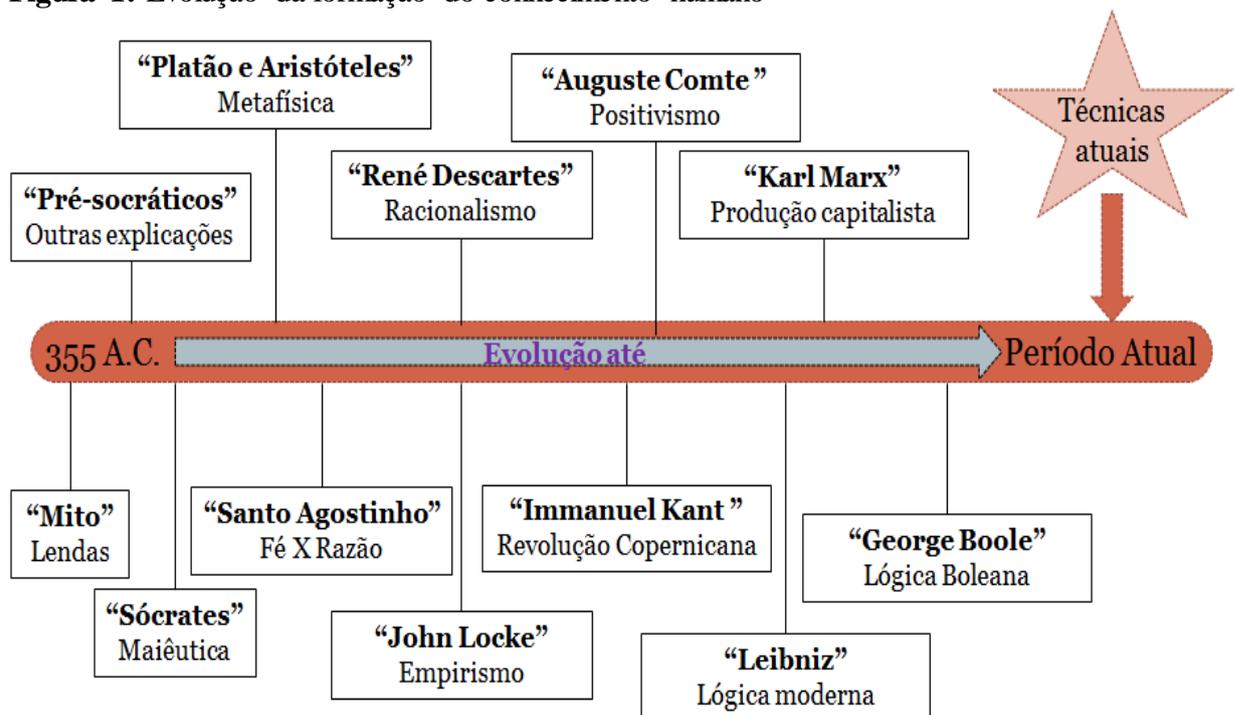
Aproximadamente, entre os séculos XIV e XIX (período em que surgiu o Iluminismo), a sociedade passou por inúmeras transformações, no campo social, político, religioso, cultural e econômico, devido ao surgimento das cidades, fábricas, máquinas a vapor, novas religiões e novas formas de governo. Auguste Comte (1798-1857), considerado o criador da Sociologia,

⁵ Mestres da retórica e oratória, que ensinavam sua arte aos cidadãos interessados em dominar melhor a técnica do discurso, visando o domínio político.

e pai do Positivismo, (corrente filosófica que possui como objetivo reorganizar o conhecimento humano), buscou a ideia de que tudo o que se refere ao saber humano pode ser sistematizado segundo os princípios adotados como critério de verdade, para as ciências exatas e biológicas (SANTOS 2014). A partir da proposta assumida por Comte, acontece a divisão das ciências, conforme a generalidade e a especificidade de cada uma, por meio do estudo das partes que constituem as coisas, com o intuito de explicar o todo, de forma mais eficiente.

A partir do século XX, diversos autores dissertaram sobre a estrutura de formação do conhecimento, nas várias áreas do conhecimento. Para os leitores mais interessados, a **Figura 1** mostra um pequeno resumo dos principais autores que discutem a evolução do conhecimento apresentando novas possibilidades de ensino-aprendizagem. A **Figura 1** demonstra o processo de divisão das ciências, no século XX.

Figura 1: Evolução da formação do conhecimento humano



Fonte: Elaborada pelo autor.

Uma nova forma de ensino-aprendizagem surge, por causa da divisão das ciências, o que afeta, drasticamente, o ensino, pois as pessoas se tornam muito especializadas em uma única área, perdendo a visão do todo. É importante salientar que, para o entendimento da lógica de construção dos algoritmos, uma das metodologias de ensino utilizadas por alguns docentes (CAMPOS, 2010), é a apresentação de problemas, de forma fracionada, aos alunos, mas, caso o docente não se atente, o aluno pode produzir algoritmos que não atendam ao todo do programa desejado.

A Lógica moderna iniciou-se com os trabalhos de Leibniz (1646-1716), que pretendia dotar a Metafísica de um instrumento suficientemente poderoso, capaz de alcançar o mesmo grau de rigor da matemática. Segundo Fonseca Filho (2007),

A ideia de Leibniz sustentava-se em dois conceitos intimamente relacionados: o de um simbolismo universal e o de um cálculo de raciocínio (isto é, um método quase mecânico de raciocínio). Isso, para a História da Computação, tem um particular interesse, pois esse *calculus ratiocinator* de Leibniz contém o embrião da *machina ratiocinatrix*, a máquina de raciocinar buscada por Turing e depois pelos pesquisadores dentro do campo da Inteligência Artificial. Leibniz, assim como Boole, Turing, e outros perceberam a possibilidade da mecanização do cálculo aritmético. O próprio Leibniz, e Pascal um pouco antes, procuraram construir uma máquina de calcular. Nota-se, portanto que o mesmo impulso intelectual que o levou ao desenvolvimento da Lógica Matemática o conduziu à busca da mecanização dos processos de raciocínio.

O inglês George Boole (1815-1864) desenvolveu, com sucesso, o primeiro sistema formal para raciocínio lógico. Este novo sistema, agora baseado na Lógica formal, foi precursor da criação da Lógica computacional e garantiu uma nova forma de projetar o conhecimento, supervalorizando o ensino-aprendizagem, focado no desenvolvimento do raciocínio lógico. Esta forma de pensamento concebeu os computadores atuais. O ensino na disciplina de ALP deve ter como objetivo, a criação de algoritmos mais eficientes, ou seja, quanto mais próximo à criação destes algoritmos e da leitura que os compiladores fazem nas linguagens de alto nível, melhores serão os algoritmos. Conhecer a forma de concepção da lógica computacional é o primeiro passo para alcançar essa eficiência.

Por meio da apresentação deste panorama de ensino-aprendizagem, mostrando as principais correntes filosóficas que dissertaram sobre o processo de formação do conhecimento, é possível discutir as metodologias atuais de ensino-aprendizagem de programação, entendendo suas origens e o impacto delas na sociedade moderna.

Depois da formalização de um sistema de raciocínio lógico, proposto por George Boole (1815-1864), foi possível sistematizar a execução de operações formais, acarretando, anos mais tarde, no surgimento do computador. Diversas linguagens de programação surgiram paralelamente à sua evolução, com a finalidade de aperfeiçoar as operações realizadas pelos computadores. As linguagens de programação atuais podem ser divididas em vários paradigmas, sendo que a metodologia de ensino utilizada pelos professores se baseia no paradigma escolhido. Cabe ao professor, a escolha da metodologia adequada, de acordo com o paradigma adotado, visando potencializar o ensino, durante a disciplina de ALP. Logo, conforme apresentada pelo professor Buffoni (2003), as linguagens de programação podem ser divididas em linguagens:

- **Máquina** – é considerada a linguagem mais próxima do processamento real dos computadores. É formada de *STRINGS* de números, definindo as operações realizadas pelo hardware do computador. São comandos ligados diretamente às ações realizadas na memória, no processador e nos dispositivos de entrada e saída. Por serem formadas de *STRINGS* de números, essas linguagens possuem como característica, a comunicação direta com a máquina. A semântica e sintaxe dos *STRINGS* estão diretamente relacionadas ao tipo de máquina utilizada.
- **Assembly** – é uma linguagem mista, pois possui parte de sua sintaxe e semântica próxima do usuário e outra próxima da máquina. É formada de mnemônicos acompanhados por registradores, códigos para armazenamento das informações e execução das instruções. É uma linguagem particular de cada modelo de processador (computador), que codifica as instruções em linguagem de máquina e facilita um pouco o trabalho do programador. É também considerada uma linguagem de baixo nível, mas ainda hoje é utilizada.
- **Alto nível** – são linguagens que estão mais próximas do programador (mais amigáveis) do que do dispositivo a ser programado. Geralmente, utilizam sintaxe e semântica próximas das linguagens do ser humano. Dessa forma, são mais intuitivas que as linguagens de baixo nível.

A linguagem de programação mais utilizada, no ensino de ALP para iniciantes, é a linguagem de alto nível, pois se aproxima mais da linguagem humana, facilitando o processo ensino-aprendizagem. Por sua vez, a linguagem de Alto Nível pode ser classificada em:

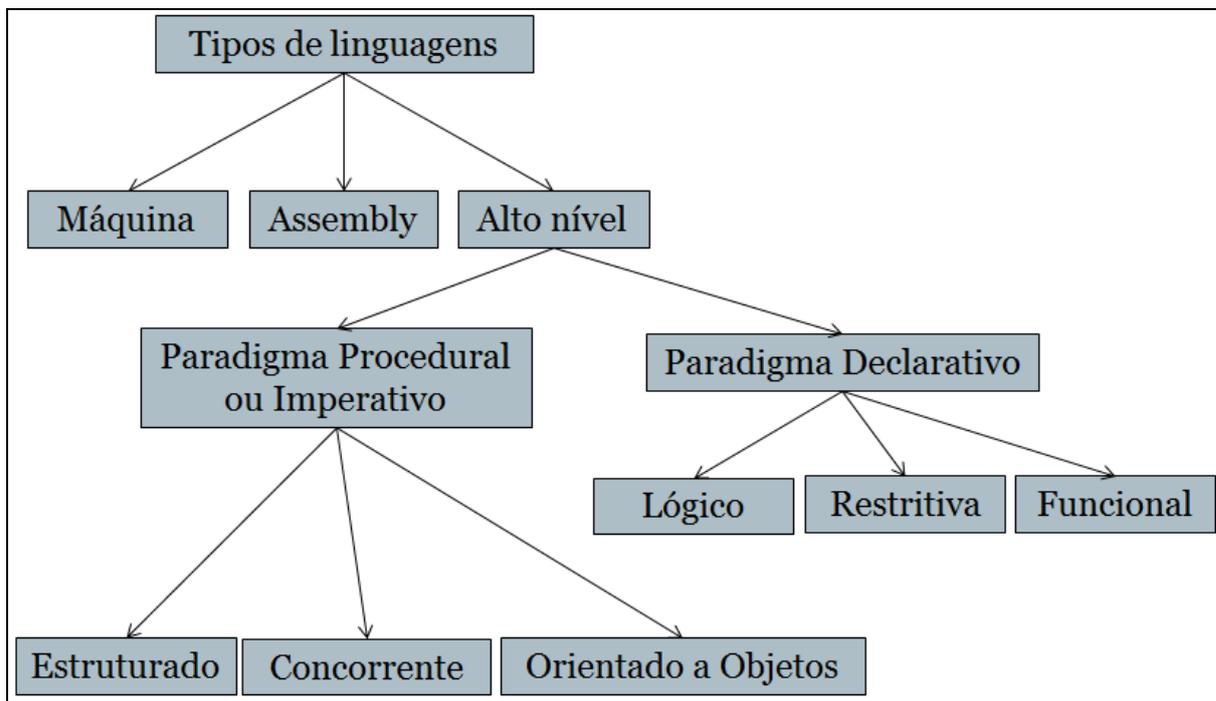
- Paradigma Imperativo ou procedural: São construídos por meio de rotinas e sub-rotinas, que enfatizam as mudanças de estados. Podem ser subdivididas em: a) Estrutura de blocos (programas criados por meio de instruções, b) estruturas de dados e c) sub-rotinas. Qualquer programa pode ser reduzido a três tipos de estruturas: sequência, decisão e interação. **Orientação a objetos** descreve linguagens que suportam a interação entre objetos, concorrente e **computação distribuída** suporta mais de uma rotina, executando independentemente.
- Paradigma declarativo: abordagem centrada em descrever o que o programa deve fazer e, menos, em como seus procedimentos devem funcionar. Ela é subdividida em:
 - Funcional: Um programa é uma função (ou grupo de funções), tipicamente constituída de outras funções mais simples), lógica (faz

forte uso da lógica matemática para a resolução de problemas, geralmente, por meio de valores ou objetos e relações entre os mesmos, formulando, assim, proposições e regras)

- Restritiva (utiliza de restrições na definição de relações entre as variáveis).

O diagrama da **Figura 2** representa o resumo dos paradigmas de linguagens de programação apresentados.

Figura 2: Paradigmas de Linguagens de Programação



Fonte: Souza (1995).

A partir deste breve histórico, dividiram-se as metodologias de ensino-aprendizagem de Algoritmo e Lógica de Programação em: **Tradicional**, baseada em **PSE** e **Assistidas** por TIC's (Tecnologia da Informação e Comunicação).

1.1.1 METODOLOGIA TRADICIONAL

Também chamada de linha pedagógica de ensino tradicional, é a mais utilizada nas escolas públicas e privada do Brasil, para ensino do ALP. Também é a mais citada em livros didáticos, como um processo de ensino-aprendizagem da disciplina de ALP. A linha tradicional de ensino teve a sua origem no século XVIII, a partir do Iluminismo (período da história chamado de época das luzes, onde surgiram duas correntes filosóficas citadas na seção anterior: racionalismo e empirismo). Apesar de o modelo original ter passado por várias

mudanças e evoluções, desde a sua concepção, o modelo tradicional, usado, atualmente ainda, possui muito de suas características originais. O objetivo principal de criação desta metodologia, é a universalização do acesso do indivíduo ao conhecimento, de forma a levar este conhecimento ao máximo de pessoas possíveis. As escolas que adotam a linha tradicional acreditam que a formação de um aluno crítico e criativo depende justamente da bagagem de informação adquirida e do domínio dos conhecimentos consolidados. (LEÃO, 1999).

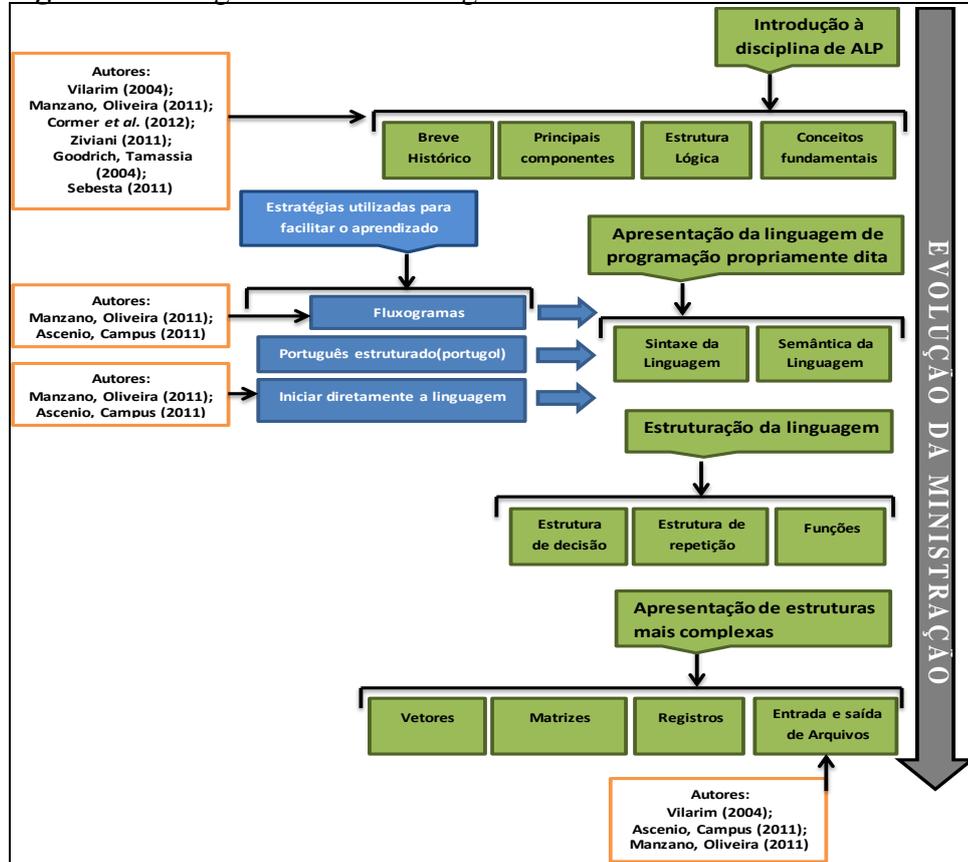
Neste método, não há lugar para o aluno atuar, agir ou reagir, de forma individual. Não existem atividades práticas que permitem aos alunos inquirir, criar e construir seu próprio conhecimento (MOCARZEL & VIEIRA, 2016). Geralmente, as aulas são expositivas, com muita teoria e exercícios sistematizados para a memorização dos passos que definem as estruturas dos principais algoritmos. Dentro desta metodologia, acredita-se que quanto mais exercícios aplicados aos alunos, melhor é a formação de conhecimento destes. Porém, estes exercícios são construídos a partir dos exemplos repassados pelo professor, em sala de aula. (LEÃO, 1999).

As avaliações são periódicas, por meio de provas, visando medir a quantidade de informação que o aluno conseguiu absorver, ou a sequência de passos que o aluno conseguiu armazenar, reproduzindo estas informações na construção de algoritmos. (OSAKA, 2009).

Alguns autores, Vilarim (2004); Manzano & Oliveira (2011); Cormen *et al.* (2012); Ziviani (2011); Goodrich & Tamassia (2004) e Sebesta (2011), optam por iniciar seus livros com uma introdução à computação, onde são apresentados: breve histórico da computação, principais componentes, estrutura lógica do computador e, por último, conceitos importantes para a construção de algoritmos, como por exemplo, tipos de linguagens, tipos de dados, entre outros. Após esta introdução, iniciam o ensino de construção dos algoritmos. Nesse ponto, é apresentada uma proposta de sintaxe e semântica para formação do algoritmo. Geralmente, visando o melhor entendimento do aluno, os professores trabalham com uma linguagem de programação, chamada de português estruturado (portugol) ou também, chamada de pseudocódigo. Com a utilização do português estruturado, pretende-se facilitar a visão do aluno na escrita do algoritmo (sintaxe), comparando-o com a linguagem materna de comunicação. Cormen *et al.* (2012), optam por já iniciarem com a sintaxe e semântica, passíveis de serem executadas por compiladores, como C, Java, Pascal entre outras. Estes professores acreditam que se perde tempo migrando os algoritmos do português estruturado para a linguagem de programação e que o aluno, realmente, desenvolverá seus programas. Para facilitar a visualização de construção dos algoritmos, autores como, Manzano (2011) e Oliveira (2012) utilizaram uma estrutura chamada de fluxogramas, para auxiliar o aluno a

estruturar os algoritmos antes da escrita do pseudocódigo. A **Figura 3** apresenta o fluxograma de resumo da metodologia tradicional de ensino.

Figura 3: Fluxograma da metodologia tradicional.



Fonte: Próprio autor

Após estas etapas, foram apresentados aos alunos, alguns elementos importantes para a estrutura dos algoritmos, como tipos de variáveis, estruturas de decisão, repetição e funções. A cada etapa, foi apresentada uma série de exercícios, visando fixação das estruturas apresentadas. A finalidade dos exercícios de fixação não é desenvolver o senso crítico, mas levar o aluno a repetir a estratégia de programação, repassada pelo professor, até que o aluno memorize a sequência lógica, necessária para sua execução. A partir da memorização da sequência lógica realizada pelo aluno para a resolução do problema, acredita-se que o mesmo consiga aplicar ou rearranjar esta estrutura em outros problemas (OSAKA, 2009). Para finalizar, são apresentadas estruturas mais complexas, como laços, vetores, matrizes, registros. Por último, alguns autores, como, por exemplo, Vilarim (2004); e Manzano & Oliveira (2011) trabalham, ainda, com estruturas de entrada e saída de arquivos. Como as aulas são expositivas, estas são apresentadas para um grupo de aproximadamente 30 alunos ou mais, e as avaliações verificam como esses alunos absorvem e conseguem aplicar as estruturas recebidas na resolução de problemas semelhantes.

1.1.2 METODOLOGIA BASEADA NO SISTEMA PERSONALIZADO DE ENSINO

A metodologia de ensino, baseada no Sistema Personalizado de Ensino PSI (*Personalized System of Instruction*), no Brasil, popularizado como PSE (Sistema Personalizado de Ensino) é uma metodologia ou sistema de ensino, idealizado pelos professores Fred Keller, Carolina Martuscelli Bori, John Gilmour Sherman e Rodolpho Azzi, em 1963, e embasada em princípios da Análise do Comportamento (KELLER, 1968 apud ROCHA *et al.*, 2010). É a partir da proposta de divisão do conhecimento de *saber como* e *saber sobre*, que se fundamenta a metodologia de ensino baseada em PSE.

Rocha *et al.* (2010) descrevem sua implantação e seu crescimento:

No Brasil, foi implantado inicialmente na Universidade de Brasília em 1964, e amplamente utilizado, principalmente nos anos 70. Em 1979 havia, mundialmente, cerca de 5.000 cursos conhecidos baseados no PSI, um periódico específico para o assunto (*Journal of Personalized Instruction*) e, de 1973 a 1979, manteve-se em funcionamento o Center for Personalized Instruction na Georgetown University.

Trata-se de uma metodologia que propõe o acompanhamento individual do aluno. Além de propostas de estratégias de ensino, conforme o desenvolvimento individual. A grande flexibilidade do método é uma de suas grandes características, pois permite ao professor a inserção de recursos educacionais diversos na formação do aluno, de acordo com seu nível de aprendizado (ROCHA *et al.*, 2010). Esta metodologia não foi criada exclusivamente para o ensino de Algoritmo e Lógica de Programação, porém, devido aos seus excelentes resultados quando aplicado a outras disciplinas, sua metodologia foi utilizada para o ensino de ALP. Em 2009, por meio do artigo intitulado “Sistema Personalizado de Ensino, Educação à Distância e Aprendizagem Centrada no Aluno” (TODOROV *et al.*, 2009). Foi apresentada uma versão do método que o associa ao ambiente virtual de aprendizagem, usando ferramentas do ambiente de ensino à distância. Nesse método de ensino, o professor não é visto como o centro crítico de transmissão da informação. O papel do professor diferencia-se do modelo tradicional, sendo sua função acompanhar, aprimorar, treinar e gerenciar (TODOROV *et al.*, 2009).

Para que o PSI alcance seus objetivos e os resultados esperados, algumas características devem ser mantidas (ROCHA *et al.*, 2010):

- **Fácil disponibilização de materiais e exercícios:** devem ser disponibilizados aos alunos, com facilidade de acesso, os vários instrumentos de estudo personalizados pelo professor, como por exemplo, apostilas, exercícios, roteiros de estudos, entre outros. Para facilitar o acesso, o professor pode utilizar ferramentas de TIC's (Tecnologias de Informação e de Comunicação) para auxiliá-lo no processo.

Também, é importante para o professor, manter o alto grau de detalhamento e utilizar estratégias que chamem a atenção dos alunos, como: imagens, animações, vídeos, áudios, entre outros recursos.

- **Organização em níveis:** organização em níveis sequenciais, permitindo um aprendizado gradual. Os níveis são estruturados de forma que um nível anterior é pré-requisito ao nível posterior. A cada nível é acrescido um novo conceito, aumentando à complexidade de forma gradual.
- **Mobilização de professores e monitores:** Nesta metodologia, professores e monitores têm contato direto com os alunos, identificando as dificuldades individuais e propondo ações que os estimulem a superar as dificuldades.
- **Disponibilização permanente de orientação e monitoramento:** uma vez que os materiais didáticos e exercícios são de fácil acesso e disponíveis ao aluno a qualquer momento, as dificuldades e dúvidas, também, podem surgir, na mesma frequência. Dessa forma, o monitoramento deve ser constante, utilizando todos os recursos para diminuir ao máximo o tempo de resposta, para dificuldades ou dúvidas dos alunos.
- **Encontros periódicos de estudos assistidos:** nestes encontros presenciais, que devem acontecer de forma periódica, conforme a necessidade de cada aluno, deve haver a troca de experiências entre aluno, professor e monitor. A fim de promover a interação entre alunos, também, devem ser promovidos encontros entre os alunos, assistidos pelos professores e monitores.
- **Avaliação de cada nível:** as avaliações ocorrerão de forma individual, à medida que cada aluno sintá-se preparado para sua realização. Assim como o material de estudo deve ficar à disposição do aluno, as avaliações, também, estarão disponíveis, a qualquer momento. Os alunos que apresentarem proficiência naquele nível serão promovidos para o próximo nível.

Diferentemente dos outros modelos de avaliação tradicional, em que o aluno é considerado aprovado com 50% de aprendizado, podendo este percentual chegar a 70%, no PSE, para a promoção do aluno ao próximo nível, o mesmo deve alcançar 90% de aprendizado.

Esta metodologia é muito utilizada em cursos da área de Informação e Comunicação, promovida na modalidade EAD (Ensino à Distância).

1.1.3 METODOLOGIAS DE ENSINO DE ALP ASSISTIDA POR TICs

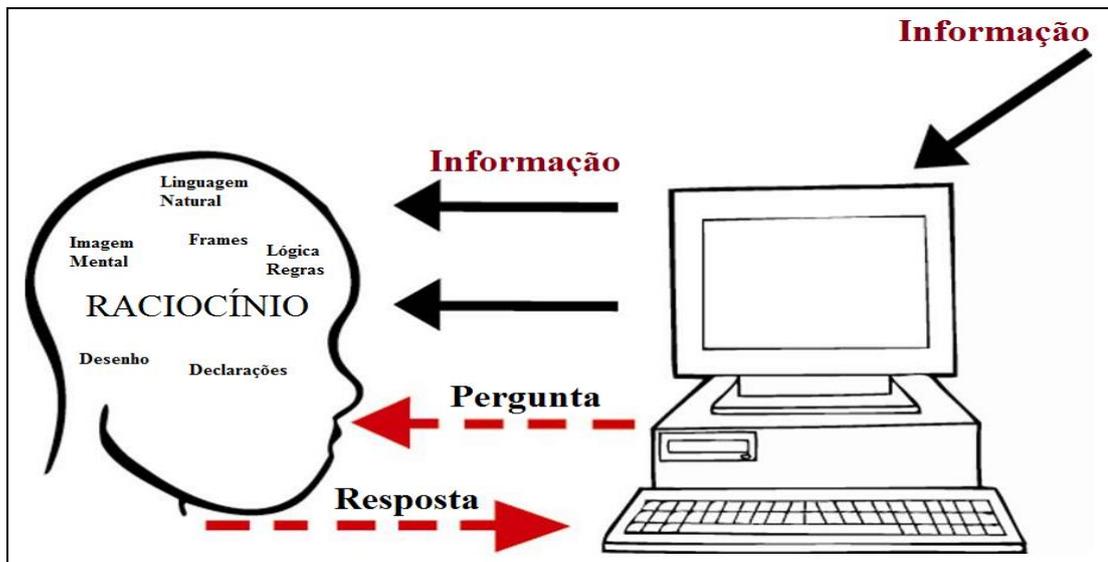
O surgimento de computadores pessoais transformou o mundo atual numa velocidade nunca vista na história da humanidade. Com suas redes de comunicação globais, como a Internet, diminuindo distâncias e acelerando a transmissão das informações, os computadores pessoais colocaram a humanidade frente a uma nova onda de transformações. As luzes que se acendem são de uma era em que *bits* valem mais que átomos e que bens materiais não são mais garantia de poder e riqueza (MARIN DA SILVA, 2006).

O computador afetou diretamente a sociedade, abrindo novas discussões sobre a formação e transmissão do conhecimento: “Sociedade da informação”, “Teoria da Complexidade”, “Sociedade digital” e muitas outras tendências (KOHN & MORAES, 2007). Como discutido anteriormente, analisando os processos históricos, percebe-se que mudanças nos paradigmas que explicam a criação do conhecimento afetam, também, as metodologias de ensino-aprendizagem. A mudança na sociedade, devido à evolução dos computadores, fez com que as metodologias de ensino assistidas pelas TIC’s surgissem como uma nova tendência no processo ensino-aprendizagem, sustentada no computador como meio de ensino, ou seja, o próprio computador dando suporte no ensino da criação de seus próprios algoritmos de programação. Pode-se dividir as práticas pedagógicas de ensino assistidas por TIC’s em Instrucionismo, Tecnicismo e Construcionismo (VALENTE, 2011).

1.1.3.1 INSTRUCIONISMO

De acordo com Valente (2011), entende-se o Instrucionismo como a utilização do computador como máquina de ensinar. Por meio desta teoria, o computador é usado como agente transmissor de informação aos alunos. Neste paradigma, o computador é programado com uma série de informações, sendo repassadas aos alunos, conforme a programação pré-estipulada. As formas mais comuns de repasses de informações são por meio de tutoriais, exercícios de múltipla escolha, manuais, jogos, simulação, vídeos, entre outros. É o paradigma que transfere para o computador a tarefa de ensinar, ou reforça as atividades realizadas em sala de aula. O ensino Instrucionista é o meio mais utilizado nas escolas como modelo de ensino-aprendizagem, assistido por TIC’s. A **Figura 4** demonstra como o paradigma Instrucionista funciona:

Figura 4: Informática na Educação: Instrucionismo



Fonte: Valente (2011)

Valente (2011) entende que este modelo de ensino diverge do modelo apresentado por Jean Piaget e não deve ser confundido com o Construcionismo. Para ele:

Embora, nesse caso o paradigma pedagógico ainda seja o Instrucionista, esse uso do computador tem sido caracterizado, erroneamente, como Construtivista, no sentido piagetiano, ou seja, para propiciar a construção do conhecimento na "cabeça" do aluno. Como se o conhecimento fosse construído através tijolos (informação) que devem ser justapostos e sobrepostos na construção de uma parede. Nesse caso, o computador tem a finalidade de facilitar a construção dessa "parede", fornecendo "tijolos" do tamanho mais adequado, em pequenas doses e de acordo com a capacidade individual de cada aluno.

Para o Instrucionismo:

- Aprender é acumular informações;
- Aprende-se por meio da repetição, memorização e imitação;
- Ensinar é transmitir informação e colocar o aprendiz em contato com modelos da cultura;
- O papel do professor é transmitir informações, apresentar modelos, corrigir equívocos por meio de modelos pré-definidos no computador;
- O papel do aluno é memorizar informações e repetir modelos que foram apresentados pelo computador, conforme pré-configuração do professor;
- O erro é visto como uma distorção da realidade, inadequação, portanto é algo a ser corrigido e evitado;
- A avaliação é realizada, aferindo e quantificando a informação acumulada, repassada pelo professor por meio das TIC's.

1.1.3.2 TECNICISMO

Modelo de ensino baseado na proposta pedagógica de Burrhus Frederic Skinner (1904-1990), psicólogo behaviorista norte-americano, que acredita que a educação deve ser planejada passo a passo para a obtenção dos resultados desejados. Considera que o comportamento é apenas um conjunto de reações aos estímulos externos. Para Skinner apud Ferrari (2008, p.1):

A educação é o estabelecimento de comportamentos que serão vantajosos para o indivíduo e para outros em algum tempo futuro. Quando houver domínio sobre a ciência do comportamento, ela será a única alternativa para a sociedade planejada.

O tecnicismo planeja a educação de modo que ela possa organizar, de forma racional, as ideias, minimizando interferências subjetivas que possam por em risco a eficiência dessas ideias (ALTOÉ & PENATI, 2005).

Para o Tecnicismo:

- Aprender é mudar o comportamento de forma relativamente permanente;
- Aprender é condicionar o comportamento, com esquemas de reforço do comportamento desejado;
- Ensinar é estruturar esquemas de reforço dos comportamentos desejados – recompensa e punição;
- O papel do professor é aplicar esquemas de reforço, montados por técnicos. Estes esquemas de reforço são inseridos em programas, jogos e outros aplicativos, por meio de repetição;
- O papel do aluno é agir para ser reforçado, aluno ativo possui um condicionamento operante;
- O erro é visto como um comportamento indesejado, que deve ser reforçado negativamente (punido) ou ignorado;
- A avaliação é a aferição e quantificação das respostas emitidas (comportamento).

1.1.3.3 CONSTRUCIONISMO

O Construcionismo é uma corrente teórica, fundamentada no Construtivismo (ALTOÉ & PENATI, 2005). O Construtivismo é uma teoria que afirma que o desenvolvimento da inteligência humana é determinado pelas ações entre o sujeito e o meio. Os principais autores que dissertaram sobre o construtivismo foram: Jean Piaget (1896-1980) e Lev Semenovitch Vygotsky (1896-1934). Levando em consideração a produção teórica inicial de Piaget, existe entre ele e Vygotsky uma aproximação básica entre suas teorias: a ação do meio social e

cultural na explicação do desenvolvimento do pensamento infantil (MONTROYA, 2013). Para Montoya (2013):

Apesar de partirem do mesmo princípio a diferença aparece no interesse de Piaget pelos processos de centração – descentração e de Vygotsky pelos processos de interiorização da linguagem. [...] Piaget é o conceito de egocentrismo como processo de centração, o que não teria sido levado em conta por Vygotsky na sua crítica ao primeiro. [...] Segundo Piaget, o conceito de relação social segundo Vygotsky seria amplo demais, pois este não estaria considerando a diferença fundamental entre o caráter centrado ou descentrado das relações entre os indivíduos. [...] Piaget destaca processos e mecanismos psicológicos fundamentais que poderiam diferenciar as teorias de ambos. É, justamente, no que diz respeito ao uso dos mecanismos explicativos da formação dos conceitos que as fronteiras entre esses autores se marcam com maior nitidez: a evolução dos conceitos para Vygotsky obedeceria à generalização perceptiva e não à generalização operatória, como postula Piaget, o que permite revelar opções epistemológicas sobre a origem do pensamento e do conhecimento.

O Construcionismo é uma reconstrução teórica, realizada a partir do construtivismo piagetiano, feita por Seymour Papert (1986). O Construcionismo se propõe a explicar as relações entre aprendiz-computador para produzir o máximo de aprendizagem, com o mínimo de ensino. O Construcionismo busca alcançar formas de aprendizagem fortes, que valorizem a construção mental do sujeito, apoiada em suas próprias construções no mundo. Como qualquer construtor, a criança se apropria, para seu próprio uso, de materiais que ela encontra e, mais significativamente, de modelos e metáforas, sugeridos pela cultura que a rodeia (PAPERT, 1986 apud PINEL, 2004).

Diferentemente do Instrucionismo, no qual o computador é visto como uma máquina que ensina e repassa instruções, no Construcionismo, o computador passa a ser visto como uma máquina a ser ensinada e o aluno, ainda pode refletir sobre seus erros, acompanhado pelo professor, durante o processo de aprendizagem. Jean Piaget, em sua obra intitulada “Epistemologia Genética”, considera a existência de estruturas mentais, desde a fase inicial do desenvolvimento cognitivo (conhecimento hereditário – genótipo). Para ele, durante o processo de desenvolvimento cognitivo, existe a maturação destas estruturas ou criação de novas estruturas por meio de dois processos denominados assimilação e acomodação. A maturação das estruturas é influenciada pelas interações com o meio externo, almejando a equilíbrio das estruturas (conhecimento adquirido – fenótipo). Seymour Papert (1986) propõe a utilização do computador como ferramenta, para produzir o desenvolvimento destes esquemas.

No início dos anos 1960, Seymour Papert afiliou-se ao MIT (Massachusetts Institute of Technology), onde, em conjunto com Marvin Minsky, fundou o Laboratório de Inteligência Artificial e co-autorou seu trabalho fundamental "Perceptrons", em 1970. Ele é

autor de "Mindstorms: Children Computers and Powerful Ideas" (1980) e "The Children's Machine: Rethinking School in the Age of the Computer" (1992)⁶.

A fim de demonstrar todas as suas ideias, Seymour Papert apresentou uma linguagem de programação chamada **Logo**. De acordo com sua proposta, a linguagem de programação **Logo** é de fácil compreensão e manipulação por crianças ou por pessoas leigas em computação e sem domínio em Matemática. Segundo Seymour Papert (apud PINEL, 2004):

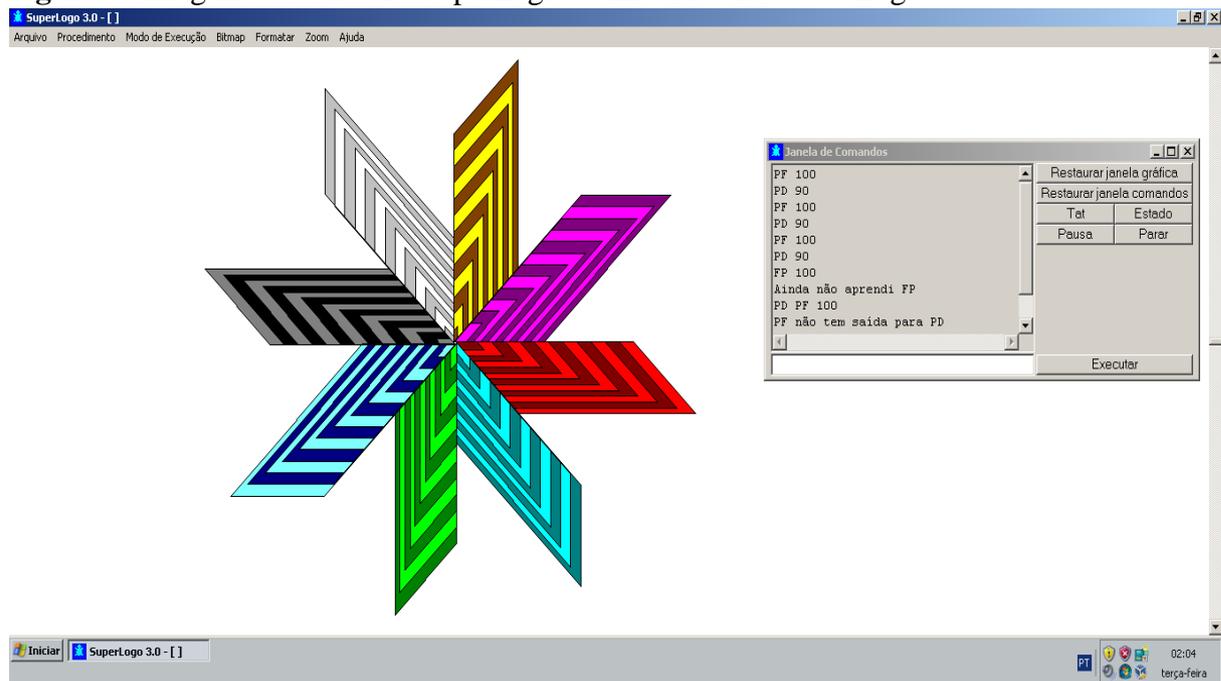
Minha meta tornou-se lutar para criar um ambiente no qual todas as crianças – seja qual for sua cultura, gênero ou personalidade – poderiam aprender Álgebra, Geometria, Ortografia e História de maneira mais semelhante à aprendizagem informal da criança pequena, pré-escolar, ou da criança excepcional, do que ao processo educacional seguido nas escolas.

1.1.3.3.1 A LINGUAGEM DE PROGRAMAÇÃO LOGO

A palavra **Logo** foi usada como referência a um termo grego que significa "pensamento, raciocínio e discurso", ou também, "razão, cálculo e linguagem". **Logo** é uma linguagem de programação simples e estruturada, voltada à educação, que tem como objetivo permitir que uma pessoa conheça, com o seu uso, conceitos lógicos e matemáticos por meio da exploração de atividades espaciais que auxiliam o usuário a formalizar seus raciocínios cognitivos (GREGOLIN, 2008). Esta linguagem popularizou-se nas escolas de todo o mundo em que se desenvolveram projetos de integração das TIC's na educação, sobretudo durante os anos 1980. No **Logo** se pode trabalhar de duas formas: **Modo Direto** ou pilotagem: a tartaruga, que representa o cursor dentro do ambiente Logo, executa imediatamente cada instrução que se digita no teclado. E **Modo Programa**: consiste em ampliar o vocabulário da tartaruga. Novos comandos podem ser obtidos, por meio do comando APRENDA, com execução similar aos comandos primitivos (GREGOLIN, 2008). Na **Figura 5**, pode-se ver uma imagem do software **SuperLogo**.

⁶ Biografia de Seymour Papert. Disponível em: <<http://goo.gl/fQH3qO>>, Acesso: 24 maio 2005.

Figura 5: Imagem do Software SuperLogo – uma das versões do Logo



Fonte: Elaborada pelo autor.

Para o Construcionismo:

- Aprender é construir relações, internamente, a partir das interações que o sujeito tem com o mundo, mediado pela cultura;
- O aluno aprende por meio do levantamento de hipóteses, testes, reelaboração das hipóteses, novo teste, num processo recursivo;
- Ensinar é facilitar, por meio da criação de um ambiente, cuja tônica seja a proposição de desafios, desequilíbrios e questionamentos que ponham em cheque as hipóteses do aluno, auxiliando-o na reflexão sobre o contexto, no qual está inserido e ajudando-o na sistematização dos resultados;
- São vários os papéis do professor: ter uma concepção do sujeito como totalidade: um ser experimental, pessoal e, em relação; ter o aprendiz como referência, compreendido em suas relações com o mundo, que também faz mediação entre ele e os outros homens;
- Mediar as relações interpessoais; introduzir o aluno numa heurística que lhe permita encontrar as soluções, ao invés de somente apresentá-las; provocar o aluno a pensar sobre o que está fazendo, indagando sobre os planos e suas hipóteses;

- O papel do aluno é agir sobre as situações e desafios, levantando hipóteses e testando-as, por fim refletindo sobre os resultados. O aluno é considerado um ser ativo;
- O erro é visto como resultado inesperado. Condição *sine qua non* para a aprendizagem, portanto, algo a ser considerado e refletido;
- A avaliação é o acompanhamento das hipóteses do aprendiz, de seu nível cognitivo, dos conceitos que domina e das estratégias que utiliza, com o objetivo de encaminhar o próximo passo no processo.

1.2 COMPARATIVO ENTRE AS METODOLOGIAS APRESENTADAS

O **quadro 1**, abaixo, apresenta um comparativo das metodologias de ensino, já vistas anteriormente:

Quadro 1: Comparativo das metodologias de ensino atuais

Metodologia tradicional	Metodologia PSE	Metodologia assistida por TIC		
<ul style="list-style-type: none"> - Origem no Iluminismo; - Proposta de ensino baseada na universalização do acesso ao conhecimento; - Aluno crítico e criativo consegue-se através da quantidade de informação adquirida e do domínio dos conhecimentos consolidados; - Não existe individualismo; - Não existem atividades práticas que permitem aos alunos inquirir, criar e construir seu próprio conhecimento; - Aulas são expositivas, com muita teoria e exercícios de repetição; - Memorização é o ponto forte; - Avaliações periódicas visam medir a quantidade de informação adquirida. 	<ul style="list-style-type: none"> - Origem embasada em princípios da Análise do Comportamento; - Proposta de ensino baseada na divisão do conhecimento de - <i>saber como</i>” e - <i>saber sobre</i>”; - Acompanhamento individual do aluno; - Flexibilidade ⇒ grandes características; - Utilizada no EAD; - Professor ⇒ não é visto como o centro crítico de transmissão da informação ⇒ acompanhar, aprimorar, treinar e gerenciar; - Características necessárias para a metodologia: <ul style="list-style-type: none"> ➢ Fácil disponibilização de materiais didáticos e exercícios; ➢ Organização em níveis; ➢ Permanente orientação / monitoramento; 	<ul style="list-style-type: none"> - Origem com o surgimento da sociedade da informação; - Proposta de ensino baseada na utilização do computador como meio de ensino; - Dividido em: 		
		<p>Instrucionismo:</p> <ul style="list-style-type: none"> - Computador ⇒ transmissor de informação - Aprender: repetição, memorização, e imitação. - Aprender é acumular Informações; - Professor: transmitir informações; - Erro ⇒ distorção da realidade; - Avaliação ⇒ medição da quantidade da informação acumulada. 	<p>Tecnicismo:</p> <ul style="list-style-type: none"> - Ensinar ⇒ estruturar esquemas através de recompensa e punição; - Aprender ⇒ mudar o comportamento de forma relativamente permanente; - Aprender ⇒ condicionar o comportamento com esquemas desejados; - Professor ⇒ aplica esquemas de reforço através de repetição; - O erro ⇒ comportamento indesejado ⇒ deve ser punido ou ignorado; - Avaliação ⇒ aferição e quantificação das respostas emitidas. 	<p>Construcionismo:</p> <ul style="list-style-type: none"> - Ensinar ⇒ construir relações internamente a partir das interações que o sujeito tem com o mundo; - Aprender ⇒ levantamento de hipóteses, testes, reelaboração das hipóteses, através de um processo recursivo; - Professor ⇒ concepção do sujeito como totalidade; - O erro ⇒ resultado inesperado ⇒ algo a ser considerado e refletido - Avaliação ⇒ acompanhamento das hipóteses do aprendiz.
<p>Fonte: Elaborada pelo autor.</p>				

1.3 OUTRAS PROPOSTAS DE ENSINO DE ALP

Abaixo são apresentadas propostas de melhorias nos métodos de ensino de ALP da seção anterior.

1.3.1 ERMIC2 – Entender, Revisar, Melhorar, Complementar e Construir

É uma proposta de modificação da metodologia tradicional de ensino de ALP, proposta pelo professor Ricardo Luiz B. L. Campos, Mestre em Gestão do Conhecimento e Tecnologia da Informação pela UCB (Universidade Católica de Brasília), em 2006 e graduação em Licenciatura em Ciências - Habilitação em Matemática pelo UniCEUB (Centro Universitário de Brasília) em 1985. Atualmente, administrador do banco de dados do SERPRO (Serviço Federal de Processamento de Dados) e professor celetista da IESB (Instituto de Educação Superior de Brasília) nas disciplinas de Lógica de Programação do curso de ADS e Algoritmos e Estrutura de Dados, no curso de Jogos Digitais e Gestão da informação⁷. Conforme apresentado por Campos (2010) em pesquisa realizada em quatro instituições de ensino superior, verificou-se que, mais de 60% dos alunos da disciplina de Lógica de programação ou similares, eram reprovados a cada semestre. Do quantitativo de alunos reprovados, 65% já tinham sido reprovados mais de uma vez na disciplina. Raabe, Santiago & Dazzi (apud CAMPOS, 2010) afirmam:

[...] a lógica de programação é uma das mais importantes disciplinas para os alunos que ingressam em um curso superior de Tecnologia da Informação – TI. Pois os conhecimentos adquiridos nela vão influenciar diretamente no desempenho das demais disciplinas correlatas durante o restante do curso. Sendo uma das disciplinas com maior índice de reprovação em Instituições de Ensino Superior, é importante identificar quais os motivos do alto índice de repetência, bem como identificar as alternativas possíveis para reduzi-lo.

Campos (2010) atribui a dificuldade para a construção da resolução de problemas de algoritmos a dois pontos: o nível de detalhamento (passo a passo) que o algoritmo criado deve ter e a dificuldade que os alunos têm em associar os procedimentos estabelecidos na linguagem natural, com os procedimentos na linguagem de programação. Assim, propõe uma metodologia que engloba: avaliação de uma solução existente; manutenção de uma solução existente, de forma que esta solução seja corretiva ou evolutiva; otimização de uma solução existente e a criação de uma nova solução. A partir destas premissas, ele define a metodologia ERMIC² (Entender, Revisar, Melhorar, Complementar, Construir).

⁷ Informação extraída do currículo Lates de Ricardo Luiz B. L. Campos, disponível no site <<http://goo.gl/TIVV5v>> Acesso em 01 maio 2015.

Campos (2010) diz que, para tornar o método de ensino-aprendizagem, os professores, geralmente, utilizam três estratégias: Ferramentas computacionais de auxílio no processo de ensino-aprendizagem, Estratégias de ensino-aprendizagem e Ferramentas e Estratégias, conjuntamente. Para Baeza-Yates (apud CAMPOS, 2010):

[...] uma metodologia para o ensino de lógica de programação deve permitir, ao aluno, o entendimento das técnicas básicas de construção de algoritmos e a habilidade para analisar algoritmos, além do entendimento do problema de acordo com a sua complexidade. Tudo isto irá permitir que o programa, resultado da codificação de um algoritmo, seja bom: faça o que é esperado (precisão) e continue fazendo sempre (confiabilidade); manipule precisa e corretamente todos os dados (potência) sem quaisquer processamentos desnecessários (eficiência); qualquer alteração seja facilmente realizada (manutenibilidade).

A partir destas premissas, o professor Campos (2010) apresenta o ERM², que possui, conforme o próprio nome propõe, cinco passos:

- **Entender:** A finalidade é a apresentação de exercícios, em um mínimo de 10, visando fornecer ao discente, a habilidade para interpretar um algoritmo, independente de quem seja o autor. A utilização de fluxogramas, nesta etapa, ajuda o aluno no cumprimento da finalidade desta etapa.
- **Revisar:** Esta etapa objetiva habilitar os alunos a indicar as ações do algoritmo que, se executadas, possam alterar o resultado final esperado. Este nível será alcançado pelo aluno que conseguir realizar no mínimo 10 exercícios, nos níveis básico, intermediário e avançado, possibilitando-o identificar, entender e propor correções de erros em programas. Estas informações são suficientes para que a manutenção corretiva seja feita no algoritmo.
- **Melhorar:** Por meio de proposições de alterações visando correção de erros ou melhoria do desempenho do algoritmo, o aluno conseguiria alcançar o objetivo da etapa “Melhorar”. Conforme Campos (2010), esta é a etapa mais difícil do processo, pois possui duas partes distintas: construir um algoritmo que produza o resultado final esperado (potência); e identificar um conjunto de ações que prejudicam o desempenho do algoritmo, propondo soluções para substituí-las (eficiência). Também são necessários, no mínimo 10 exercícios, nos níveis básico, intermediário e avançado, para o aluno alcançar esta etapa.
- **Complementar:** nesta etapa, o aluno irá propor um conjunto de ações necessárias para o novo resultado final esperado seja alcançado. A manutenção evolutiva e/ou adaptativa, que é o objetivo desta etapa, poderá ser trabalhada, primeiramente, em um único ponto da lógica existente, e, posteriormente, em mais de um local do

algoritmo. O aluno que realizar o mínimo de 14 algoritmos em cada um dos níveis (básico, intermediário e avançado) terá alcançado esta etapa.

- **Construir:** é a fase idêntica à metodologia tradicional de ensino apresentada anteriormente. Uma quantidade mínima de 10 exercícios, nos níveis básico, intermediário e avançado, cumprirá esta etapa.

De acordo com Campos (2010), o autor aplicou a metodologia durante quatro semestres, em alunos voluntários, totalizando 48 horas/aula. Embora, segundo o autor, o resultado fora não conclusivo, ele acredita que o resultado possa ser considerado satisfatório, uma vez que a quantidade de novatos reprovados fora, em média, de dois alunos, ou seja, dezoito por cento (18%), inferior aos trinta e sete por cento (37%) revelados pela pesquisa. E a quantidade de alunos que já tinham cursado a disciplina foi, em média, de menos de um (1) aluno, ou seja, aproximadamente, oito por cento (8%), inferior aos vinte e três por cento (23%). Caso o leitor queira conhecer mais sobre a metodologia ERM2C², o autor Campos (2010) possui uma página de internet⁸ específica sobre a metodologia.

1.3.2 METODOLOGIAS QUE UTILIZAM FERRAMENTAS WEB PARA APOIO AO ENSINO TRADICIONAL

Existem diversos artigos que versam sobre a criação de ferramentas WEB para auxiliar no aprendizado de ALP, como forma de complementação do conteúdo ministrado de forma tradicional (por exemplo, pode-se citar Giraffa *et al.* (2003); Santos & Costa (2005); Kamiya & Brandão (2009); Tori (2002); Ferrandin & Stephani (2005) e Galhardo (2004)). Para exemplificar este tipo de modalidade, escolheu-se a ferramenta *WEB* de ensino de programação, orientada a objetos, proposta por Mariane Fogaça Galhardo. Sua proposta é baseada na metodologia de acompanhamento da aprendizagem, por meio da *WEB* de Zaina *et al.* (2004).

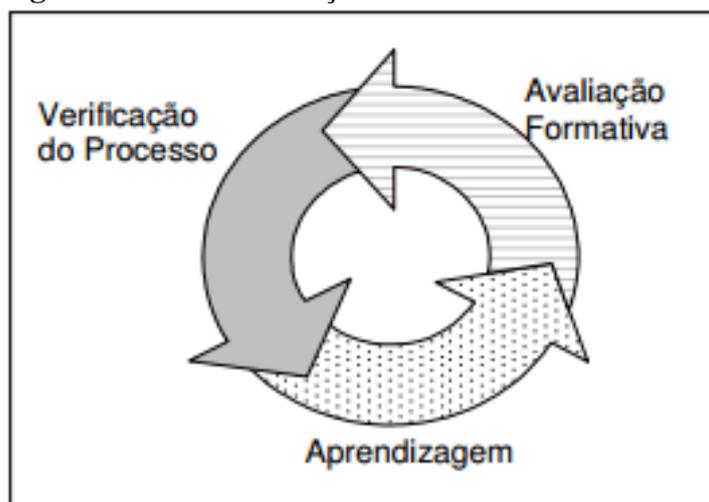
Conforme Moran, Gonzalez e McCormick (apud Zaina *et al.*, 2004),

A utilização da tecnologia como apoio ao ensino presencial tem sido amplamente adotada principalmente dentro do contexto da WEB. Adequar os meios tradicionais de ensino ao ensino on-line é uma tarefa árdua e tem sido tema de estudo para muitos pesquisadores da área educacional e de tecnologia. Há um grande desenvolvimento de infra-estruturas tecnológicas que deem suporte ao uso de novas tecnologias de ensino, porém o uso desenfreado e desorganizado da tecnologia pode ter um resultado catastrófico; é preciso associar a tecnologia às metodologias pedagógicas adaptando-as à realidade virtual para se obter sucesso no processo de aprendizagem.

⁸ Disponível em: < <http://www.erm2c.com.br/> > Acesso:16 jun 2016.

Para Zaina *et al* (2004), as ferramentas *WEB* são uma alternativa que docentes e alunos possuem, durante o processo de ensino aprendizagem. Por este ambiente *WEB*, o docente pode realizar o acompanhamento da aprendizagem do aluno passo a passo, conforme seu desenvolvimento. A metodologia proposta pela autora permite que o teste de avaliação da aprendizagem seja conduzido, de acordo com a resposta produzida pelo aluno, proporcionando ao docente, formas de construir um teste baseado na interação do aluno com o professor. Desta forma, o docente pode visualizar o caminho que o aluno percorrerá durante a resolução das questões propostas, analisando pontos críticos da aprendizagem do aluno e, onde o aluno encontra-se no processo de aprendizagem. No modelo proposto por Zaina *et al.* (2004), a avaliação sugerida é a avaliação formativa, que deve ser aplicada em períodos curtos, a fim de que o acompanhamento seja mais preciso. A **Figura 6** apresenta um diagrama sugerido pela autora, para melhor explicar o processo de avaliações formativas.

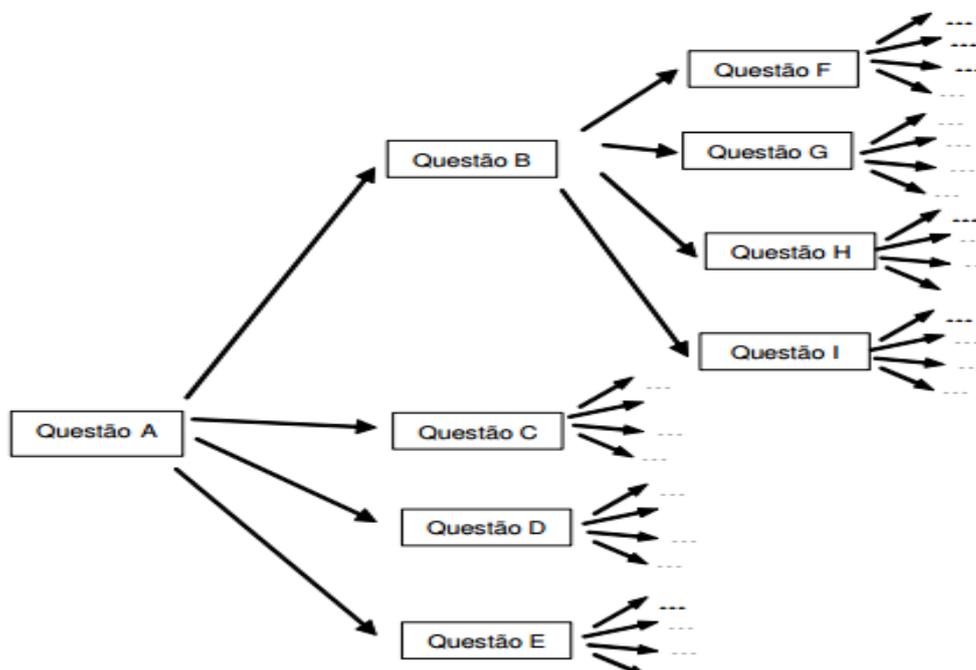
Figura 6: Ciclo de avaliação formativa



Fonte: Extraído de GALHARDO (2004)

Cada resposta do aluno, durante a avaliação formativa, vincula-o a uma sequência diferente de exercícios, conforme estabelecido no modelo da **Figura 7**.

Figura 7: Sequência de questões estruturadas para a ferramenta WEB



Fonte: Extraído de Galhardo (2004).

Para a autora (GALHARDO, 2004), este tipo de montagem se justifica:

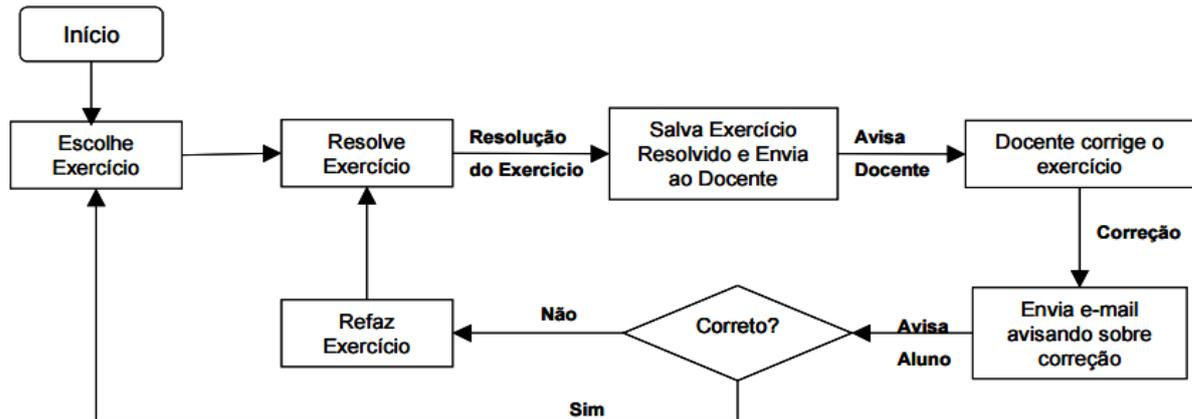
[...] a dependência entre as questões que estão diretamente envolvidas a resposta selecionada pelo aluno. Desta maneira, o docente consegue extrair os conhecimentos adquiridos pelo estudante, traçando assim, o caminho que o mesmo percorreu para resolver um conjunto de questões. É importante destacar que os múltiplos caminhos devem ser mapeados pelo docente; conforme a resposta escolhida pelo aluno a direção do teste será variada.

Baseado na proposta de Zaina *et al.* (2004), Galhardo (2004) propõe a criação de uma ferramenta WEB, onde o professor poderá acompanhar o aluno, durante o ensino dos conceitos fundamentais do paradigma de programação, orientado a objetos. Galhardo (2004) afirma que os cursos de ALP iniciam seu estudo por uma linguagem de programação imperativa, por ser mais próxima da linguagem formal do aluno. Algumas propostas foram sugeridas para o ensino de programação, começando pelo paradigma orientado a objetos (GALHARDO, 2004; LEITE & JÚNIOR, 2002), mas os resultados não foram satisfatórios. Quando os alunos têm que fazer a transição da linguagem de programação imperativa para orientada a objetos, eles sentem dificuldades em transpor as barreiras que diferenciam os estilos de programação.

Segundo Burton (apud GALHARDO, 2004), o paradigma orientado a objetos utiliza premissas da linguagem imperativa e, caso seja invertida a ordem no processo de aprendizagem, seria necessário explorar os aspectos estruturais imperativos primeiro, para depois, introduzir os conceitos da orientação a objetos propriamente ditos.

Na ferramenta *WEB* proposta por Galhardo (2004), é possível ao professor cadastrar exercícios, dividindo-os por assunto e inserindo uma descrição de cada um deles. Durante o processo de cadastro dos exercícios, o professor também define o próximo exercício que cada aluno deverá resolver, caso o aluno acerte o exercício proposto. A sequência funcional é apresentada na **Figura 8**.

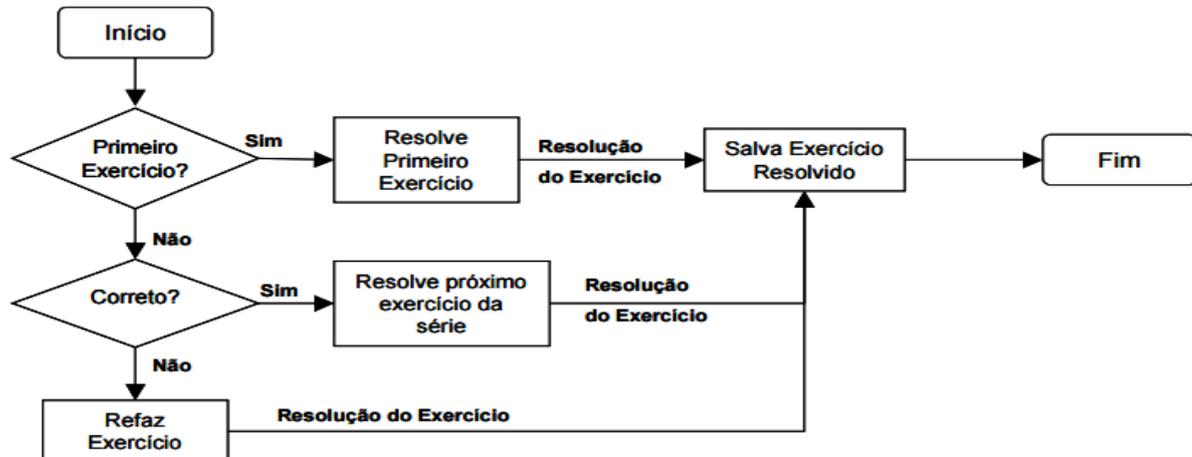
Figura 8: Fluxograma funcional da ferramenta *WEB*



Fonte: Extraído de Galhardo (2004)

Após a resolução, são armazenadas informações relevantes para o professor, como por exemplo, data e hora da resolução do exercício. A ferramenta também proporciona ao aluno saber se o exercício a ser resolvido é o primeiro da série. A **Figura 9** define a apresentação da ferramenta na visão do aluno.

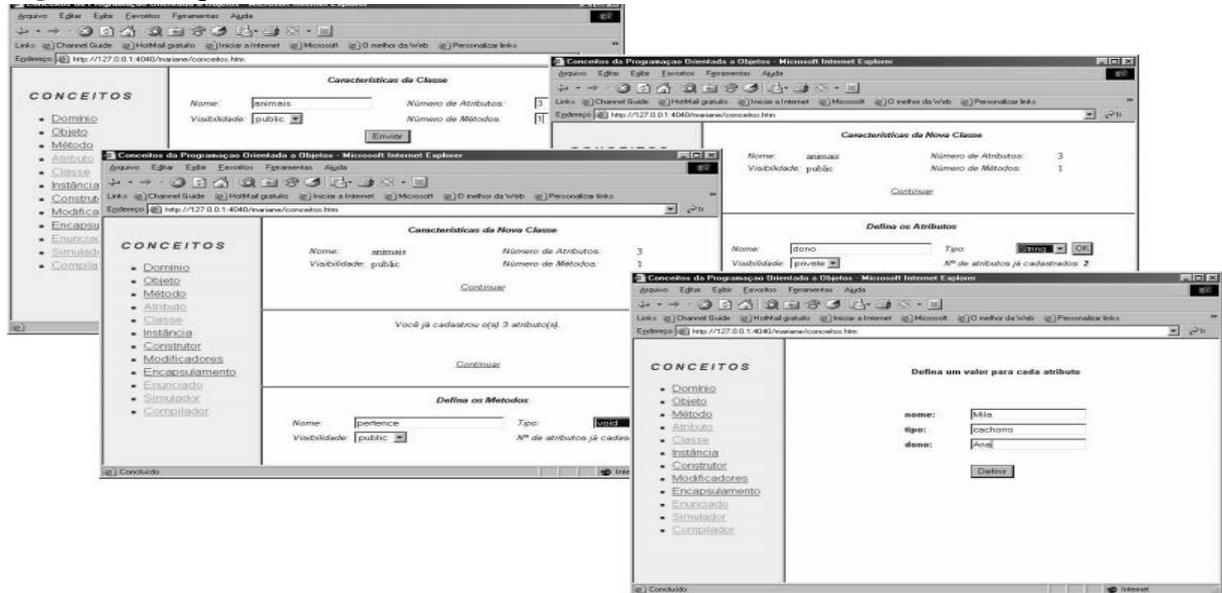
Figura 9: Fluxograma com a visão do aluno da ferramenta *WEB*



Fonte: Extraído de Galhardo (2004).

Após a resolução das questões, a ferramenta possui uma seção onde são apresentadas as respostas para o professor, possibilitando ao professor acrescentar observações a cada questão. Algumas telas da ferramenta são apresentadas na **Figura 10**.

Figura 10: Sequência de Interfaces da ferramenta WEB



Fonte: Extraído de Galhardo (2004)

1.3.3 MÉTODOS PARA CONSTRUÇÃO DE ALGORITMOS

Vinícios (2010) apresenta uma proposta de criação de algoritmos muito utilizada por professores. Também, baseado no modelo tradicional, ele compara a criação de algoritmos como uma receita de bolo, apresentando alguns passos válidos para a criação de qualquer algoritmo. Para auxiliar no processo de ensino-aprendizagem de sua proposta, foi criado um site no gvnsino.com.br⁹, com vídeos de aplicação do método proposto.

Segundo Vinícios (2010):

Qualquer tipo de algoritmo que formos fazer (e isso inclui receitas de bolos ou rotas para se chegar a algum lugar) deve seguir um método para que possa ser bem feito. Entretanto, o método que veremos serve melhor para algoritmos computacionais, ou seja, algoritmos que serão utilizados em programas de computador.

Para o autor, o método proposto possui os seguintes passos, definidos na ordem apresentada abaixo:

- **Compreender completamente o problema:** O aluno não deve tentar resolver o problema antes do completo entendimento. A ideia é não perder tempo, resolvendo um problema que ainda não entendeu claramente o objetivo.
- **Definir os dados de entrada:** Neste ponto, o aluno deve-se fazer as seguintes perguntas: quais as informações que serão recebidas por este programa? Quais são as informações que preciso obter para resolver o problema?

⁹ Site que disponibiliza cursos *on-line* em diversas áreas.

- **Definir o processamento, ou seja, cálculos e ações:** O que o programa precisa fazer com as variáveis recebidas como entradas do algoritmo? Ele irá fazer um cálculo? Irá tomar uma decisão? Definir estas ações é definir o que o programa processará.
- **Definir os dados de saída:** O aluno deve fazer as seguintes perguntas: Qual vai ser o resultado? Como ele será exibido? Com cores? Com letras grandes? Com letras maiúsculas ou minúsculas? Ou deverá aparecer alguma outra coisa (uma imagem, por exemplo)?
- **Construir o algoritmo:** Depois de concluir todas as etapas anteriores, o aluno deve construir o algoritmo. Neste passo, pode-se utilizar o fluxograma a princípio e, em seguida, com o fluxograma pronto, colocar o problema em pseudocódigo.
- **Testar o algoritmo realizando Testes de Mesa e Simulações:** Após a construção do algoritmo, é necessário testá-lo. Assim, como uma receita de bolo, saberá se funcionou se experimentar o bolo. Existem duas formas de testar algoritmos: o teste de mesa, que é um teste feito em um papel e o teste simulado, que é feito, usando um programa de computador que simula o comportamento do algoritmo. Para finalizar seu texto, o autor Vinícios (2010) faz o seguinte comentário:

Podemos comparar todas estas etapas para construção de um algoritmo à confecção de um bolo. Vejamos: 1. Análise e compreensão do problema: que bolo quero fazer? Para quem quero fazer? Quando deverá ficar pronto? 2. Analisar quais são os itens de entrada do bolo – ou seja, quais são os ingredientes? Em que quantidades? 3. Analisar, então, qual é o processamento que será feito (as misturas, quantidades, tempos, temperatura de assar, etc.). 4. Então, imaginar o resultado: nesse caso, o bolo pronto, talvez colocado em uma tigela, ou em um prato? Com recheio ou não? É o resultado. 5. Irá, finalmente, construir o bolo, isto é, confeccioná-lo, de acordo com o planejamento feito acima (algoritmo)! 6. Experimentar o bolo! Será que ficou bom ou algo pode ser melhorado?

1.3.4 SISTEMA PARA O ENSINO-APRENDIZAGEM DE ALGORITMOS UTILIZANDO UM COMPANHEIRO DE APRENDIZAGEM COLABORATIVO

Esta estratégia de ensino de ALP, apresentada por Petry (2005), apresenta um modelo computacional para suporte ao ensino e aprendizagem de algoritmos, por meio de um companheiro de aprendizagem que utiliza a MBR (Modelagem Baseada em Restrições) como forma de representação do conhecimento e raciocínio. Conforme descrito por Petry (2005):

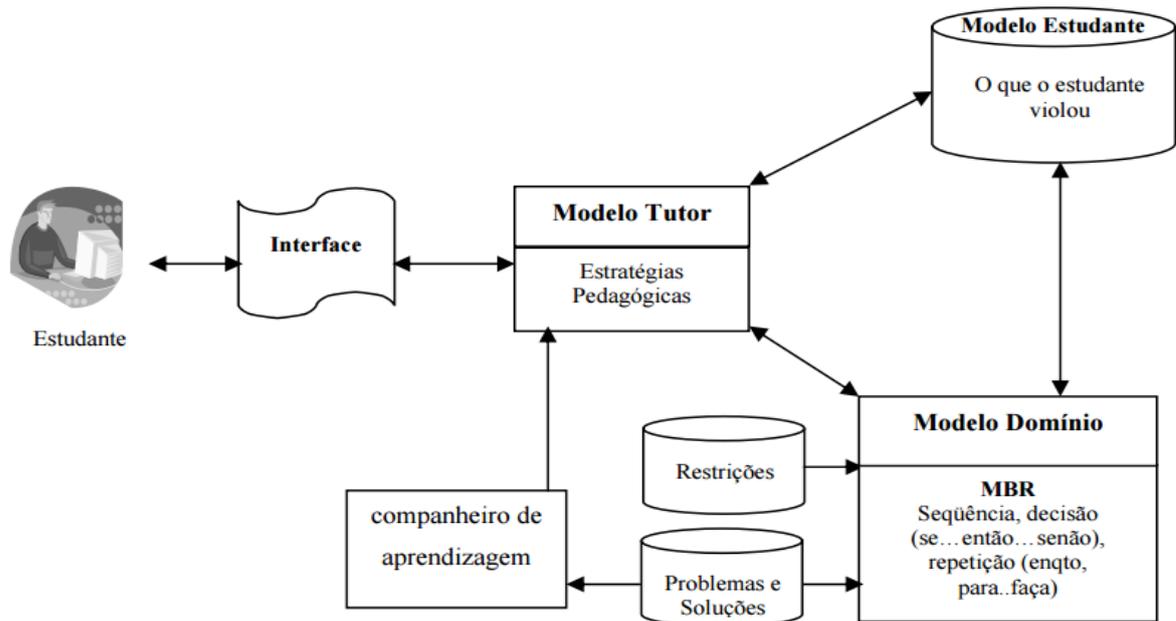
Este sistema utiliza um companheiro de aprendizagem que auxilia os alunos no desenvolvimento das possíveis soluções dos problemas a partir da identificação dos seus erros. O uso deste ambiente possibilita ao professor acompanhar e propor situações desafiadoras ao raciocínio do aluno através de relatórios que são gerados após as tentativas de solução do problema pelo aluno.

Nobre (apud PETRY, 2005) apresenta algumas dificuldades vivenciadas pelo professor, durante a disciplina ALP, para reconhecer as habilidades inatas de seus alunos, apresentar técnicas de soluções de problemas, trabalhar a capacidade de abstração do aluno, promover a cooperação e colaboração entre os alunos. O modelo proposto por Petry (2005) visa sanar estas dificuldades vivenciadas pelo professor. Os Sistemas Companheiros de Aprendizagem são considerados uma evolução dos Sistemas Tutores Inteligentes, que prevêem, além dos módulos tradicionais, a existência de pares virtuais com comportamento humano, que apoiam os estudantes, durante o processo de ensino-aprendizagem (CHAN & BASKIN, 1988 apud PETRY, 2005).

A abordagem MBR tem como fundamento uma teoria de aprendizagem que se baseia na identificação dos erros. Conforme a autora, este modelo é interessante, pois identifica os erros em programas desenvolvidos por alunos que não conseguem identificar sozinhos seus próprios erros. Isto se deve ao fato destes alunos, iniciantes em programação, não possuírem o conhecimento completo de uma determinada linguagem de programação e, portanto, não são capazes de detectar os próprios erros.

O sistema proposto por Petry (2005) foi nomeado de AlgoLC, sendo capaz de fazer análises sobre o que o aluno sabe e como ele está progredindo no desenvolvimento de um algoritmo. O sistema possui o princípio de funcionamento, como na **Figura 11**.

Figura 11: Arquitetura de funcionamento do AlgoLC



Fonte: Petry (2005).

O sistema armazena em um banco de dados, informações relevantes ao processo de aprendizagem do aluno, tais como: número de vezes em que o Companheiro de Aprendizagem interferiu no desenvolvimento dos problemas, número de acertos no desenvolvimento dos problemas propostos, número de vezes em que ocorreram violações das restrições na resolução dos problemas e quais as restrições que foram mais violadas pelo aluno. O sistema é dividido em três modelos principais:

Modelo Tutor: possui as estratégias pedagógicas que tratam do conhecimento sobre o ensino daquele domínio. Dentro de um Sistema companheiro de aprendizagem, o tutor tem o papel de coordenar as atividades de aprendizagem: apresentação dos exercícios propostos, o controle de pré-requisitos e a interação entre estudante e companheiro de aprendizagem.

Modelo Domínio: contém as informações corretas que o aluno deverá aprender, ou seja, é a base de conhecimento do sistema. O sistema contém um conjunto de restrições e a solução ideal das restrições.

Modelo Estudante: consiste nas características gerais do estudante (nome, histórico, etc.) e também se refere ao conhecimento correto ou incorreto do estudante, ou seja, quais restrições o estudante violou ou não.

Para Petry (2005),

O AlgoLC atendeu os objetivos do presente trabalho, que apresentou uma ferramenta de apoio ao ensino e aprendizagem de algoritmos que utiliza um companheiro de aprendizagem colaborativo. Esta ferramenta permite um suporte adequado ao professor na identificação de dúvidas e erros dos alunos na construção de algoritmos.

1.3.5 COMPARATIVO DAS OUTRAS PROPOSTAS DE ENSINO DE ALP

O **Quadro 2** apresenta uma comparação de outras propostas de ensino de ALP, apresentadas nas seções anteriores.

Quadro 2: Comparativo das outras metodologias de ensino de ALP

ERMC ²	Utilização de ferramentas WEB	Método para construção de Algoritmos	Utilização de um companheiro de aprendizagem
<ul style="list-style-type: none"> - Modificação do modelo tradicional sistematizando passos a serem realizados para construção do Algoritmo; - Utiliza os seguintes passos: Entender, Revisar, Melhorar, Complementar, Construir; <ul style="list-style-type: none"> ➤ Entender ⇒ Saber interpretar o algoritmo; ➤ Revisar ⇒ identificar ações que modifiquem o resultado esperado; ➤ Melhorar ⇒ identificar formas de melhorar o desempenho; ➤ Complementar ⇒ realização de manutenção evolutiva e adaptativa; ➤ Construir ⇒ Realizar a construção do algoritmo no modelo tradicional. 	<ul style="list-style-type: none"> - Modificação do modelo tradicional utilizando ferramentas WEB no processo de ensino; - O professor faz o acompanhamento do aluno através de ferramentas WEB; - Para o acompanhamento do aluno são criados ambientes WEB específicos para aplicação de sequências de exercícios estruturados; - A avaliação deve seguir um ciclo bem definido: avaliação formativa verificação do processo aprendizagem ⇒ avaliação formativa; - As próximas etapas da página WEB são escolhidas conforme as respostas que o aluno insere em cada etapa. 	<ul style="list-style-type: none"> - Modificação do modelo tradicional que compara a criação de qualquer algoritmo com uma receita de bolo, com passos pré-fixados; - Passos para criação de qualquer algoritmo: <ul style="list-style-type: none"> ➤ Compreender completamente o problema; ➤ Definir os dados de entrada; ➤ Definir o processamento, ou seja, cálculos e ações; ➤ Definir os dados de saída; ➤ Construir o algoritmo, com a utilização de fluxogramas para facilitar a criação; ➤ Testar o algoritmo realizando Testes de Mesa e Simulações. 	<ul style="list-style-type: none"> - Modificação do modelo tradicional onde o ensino e aprendizagem de algoritmos utiliza um personagem como companheiro de aprendizagem; - Utiliza a Modelagem Baseada em Restrições (MBR); - A finalidade do companheiro é reconhecer as habilidades inatas de seus alunos; - Trabalha a capacidade de abstração do aluno; - Característica: Alunos iniciantes não conseguem interpretar/encontrar seus erros sozinhos, assim o companheiro de aprendizagem orienta-o para alcançar estas habilidades.
<p>Fonte: Elaborada pelo autor.</p>			

2 ESTUDO DE CASO – A DISCIPLINA DE ALP EM UMA INSTITUIÇÃO DE ENSINO SUPERIOR E PÚBLICA

O estudo de caso é uma investigação que se assume como particularística, isto é, que se debruça deliberadamente sobre uma situação específica que se supõe ser única ou especial, pelo menos em certos aspectos, procurando descobrir a que há nela de mais essencial e característico e, desse modo, contribuir para a compreensão global de um certo fenómeno de interesse (Ponte, 2006).

Conforme proposto por Gil (2010), para a realização do estudo de caso sobre o ensino de programação, explorou-se uma situação da vida real, preservando as características coletadas do objeto estudado. A escolha da disciplina de ALP, do curso de Análise e Desenvolvimento de Sistema de uma instituição de ensino superior e pública, foi realizada devido às suas características, a saber: disciplina presente em cursos da área de Informática e Comunicação, Engenharia e áreas Tecnológicas, necessidade do desenvolvimento do raciocínio lógico para aprovação, altos índices de reprovação e retenção e pelo objetivo do curso, disciplina na qual o aluno deve criar linhas de comandos seguindo uma determinada estrutura lógica, imposta pelas regras de uma linguagem de programação pré-definida. Outro fator importante a ser destacado, é que a disciplina de ALP, da instituição escolhida como objeto deste estudo, está localizada no primeiro período do curso de ADS (Análise e Desenvolvimento de Sistemas), sendo um dos primeiros desafios enfrentados pelos alunos no desenvolvimento cognitivo voltado ao raciocínio lógico. Pelo fato de ser um dos primeiros desafios dos alunos na tarefa de programar, geralmente, os alunos não têm vícios, advindos de outras unidades curriculares.

A seguir, é apresentada uma análise quali-quantitativa da metodologia de ensino utilizada pela instituição. A finalidade da análise quantitativa é expressar, em forma de grandezas matemáticas, a eficácia da metodologia, enquanto a análise qualitativa comprova a importância da metodologia no processo de ensino-aprendizagem (GIL, 2010). Em nenhuma hipótese, foram questionadas a eficiência ou a competência da Instituição. A finalidade desta análise é realizar uma avaliação pedagógica do método de ensino de ALP e sua eficiência no processo de aprendizagem dos alunos. Por se tratar de um processo complexo, a análise puramente quantitativa desconsideraria o perfil da turma e características importantes do comportamento dos alunos, professores e coordenador. Por outro lado, a análise puramente qualitativa, deixaria de lado os índices importantes que são considerados pelas instituições governamentais e gestão da instituição. Logo, justifica-se a escolha da análise quali-quantitativa visando unir o melhor das duas estratégias.

2.1 ANÁLISE QUALI-QUANTITATIVA

Para a realização do estudo de caso proposto, foi realizada uma entrevista inicial com os coordenadores do curso (durante o período analisado, a instituição possuía 3 coordenadores) e, após análise da ementa, bibliografia básica e complementar, atividades diárias dos professores, verificou-se que a instituição utilizou, durante o período de 2010/2

(início do curso na instituição) a 2015/1, o método tradicional de ensino na disciplina de ALP. O método tradicional de ensino, neste caso, teve as seguintes características:

- A disciplina iniciou-se com a apresentação da estrutura do computador e uma breve explicação de seu funcionamento;
- Em seguida, por meio de montagem de fluxogramas, apresentou-se aos alunos a estrutura de programas de computadores. Nesta disciplina, utilizou-se uma linguagem genérica para a criação de algoritmos, um pseudocódigo, o português estruturado (ferramenta computacional de apoio ao ensino de ALP – VISUALG, conforme Morgado de Souza (2009));
- Por meio de exemplos, o professor apresentou aos alunos as diversas formas e estruturas que um programa possui para resolução de um problema específico;
- Para cada conteúdo, foram apresentados exemplos de estruturas de programação aos alunos e, em seguida, foi repassada uma série de exercícios de fixação, conforme sugestões dos livros didáticos propostos na ementa. Por meio dos exercícios de fixação, aumentou-se, gradualmente, a dificuldade, prevendo que, a partir dos exemplos apresentados, os alunos seriam capazes de criar seus próprios algoritmos.
- Para finalizar a disciplina, foram estudados os conceitos básicos de uma linguagem de programação estruturada específica (sintaxe, semântica, análise léxica e tipos de dados). Dentro desta etapa, foi realizada a migração do pseudocódigo para esta linguagem de programação estruturada;
- O método consistiu, basicamente, de exemplos, seguidos de exercícios de fixação (repetição), utilizando-se a seguinte estratégia: mais exercícios, mais repetição, mais aprendizado.

Além das informações obtidas, também foram coletadas as seguintes informações, a partir do Projeto Pedagógico do Curso e taletas de notas das turmas:

- Abertura do curso em agosto de 2010;
- Processos seletivos semestrais;

Observam-se dois pontos interessantes, no **Quadro 3**, abaixo, um no início do curso e outro, a partir de 2012/2. O primeiro ponto, no início do curso, tem uma característica única: todos os alunos são novatos, não se têm alunos repetentes na turma. Assim, nota-se a pequena quantidade de alunos reprovados por nota. Observa-se uma queda gradativa de alunos aprovados até 2012/1, e um número crescente de alunos reprovados até o mesmo período. Uma explicação possível para o fenômeno é que os alunos retidos, no semestre anterior, são rematriculados na turma do período seguinte, aumentando o número de alunos na turma.

Visando diminuir esta crescente de alunos reprovados, foram realizadas várias mudanças no curso, a partir de 2012/1, sendo que, em 2012/2, foi realizada uma mudança na matriz curricular do curso. Este é o segundo ponto importante, pois se inicia uma nova crescente dos alunos aprovados e uma decrescente dos alunos reprovados por nota. Conforme relatado pelo coordenador de curso, havia problemas em relação à matriz curricular do curso, detectados pelos professores, prejudicando o bom andamento da disciplina de ALP. O conjunto das alterações planejadas pela instituição provocou novamente um aumento crescente no número de alunos aprovados na disciplina, chegando ao número máximo de 74% de alunos aprovados em 2015/1. Também se observa que o número de alunos reprovados por falta permanece basicamente constante, durante todo o período analisado, sofrendo poucas variações ao longo de todas as turmas. As modificações propostas não conseguiram atingir de maneira eficiente este grupo de alunos, formado por alunos novatos e repetentes. Conforme relatos dos coordenadores, são vários os motivos apresentados pelos alunos para justificar a desistência durante a realização da disciplina, dentre os quais se podem citar: problemas familiares, mudança de cidade, dificuldades no trabalho, falta de tempo e reprovação em disciplinas. Um dos motivos citados pelo coordenador que se destaca nesta dissertação é a dificuldade apresentada no raciocínio lógico e/ou receio em conseguir realizar as atividades que exigem este raciocínio.

O **Quadro 3** refere-se às quantidades de alunos matriculados, aprovados, reprovados por nota e reprovados por frequência, com os percentuais de cada item.

Quadro 3: Número de alunos matriculados, aprovados, reprovados na disciplina de ALG entre os anos 2010 e 2015								
Ano	Semestre	Alunos Matriculados	Aprovados	%	Reprovados por nota	%	Reprovados por falta	%
2010	2	30	17	57%	2	7%	11	37%
2011	1	34	17	50%	5	15%	12	35%
2011	2	35	18	51%	6	17%	11	31%
2012	1	41	13	32%	17	41%	11	27%
2012	2	53	17	32%	15	28%	21	40%
2013	1	53	25	47%	7	13%	18	34%
2013	2	43	16	37%	11	26%	16	37%
2014	1	41	19	46%	9	22%	13	32%
2014	2	28	13	46%	2	7%	13	46%
2015	1	31	23	74%	1	3%	7	23%

Fonte: Elaborada pelo autor.

A partir da coleta dos dados e sua observação, pôde-se verificar, no **Quadro 3**, acima apresentado, que o percentual de aprovações aumentou após 2012/2, visto as várias estratégias de ensino-aprendizagem utilizadas pelo coordenador do curso. As modificações realizadas no Projeto Pedagógico do Curso, visando aumentar a quantidade de aprovações, causaram impacto direto no índice de aprovações das turmas, aumentado gradualmente, até atingir o índice de 74%.

Em 2012/2, PPC (Projeto Pedagógico do Curso) foi atualizado, conforme relato do coordenador, a modificação visava diminuir a quantidade de reprovações e evasões do curso. A disciplina de ALP foi diretamente afetada, visto que apresentava os maiores índices de reprovações/retenções. A carga horária foi modificada, uma parte do conteúdo foi trabalhada em outra disciplina, a ementa foi melhorada, visando uma melhor organização didática do conteúdo e bibliografia atualizada.

Em 2012/1, foi disponibilizado aos alunos novas possibilidades de aprendizado, como por exemplo: horário de atendimento individualizado com o professor, criação de monitoria,

estímulos para a criação de grupos de estudo e criação de ambiente virtual, utilizando o Moodle para aulas de reforço.

Conforme entrevista com o coordenador do curso, em 2013/1 a turma foi dividida em duas, visando atendimento mais individualizado para os alunos.

O **Quadro 4** apresenta os professores que ministravam aulas na disciplina. Conforme entrevista com o coordenador do curso, também foi modificado o quadro de professores, numa tentativa de melhor interação com a turma:

Quadro 4: Professores que ministraram a disciplina de ALP de 2010/2 a 2015/1.	
Período	Professor
De 2010/2 a 2011/2	A
De 2012/1 a 2012/2	B
De 2013/1 a 2013/2	B e C
Em 2014/1	C e D
De 2014/2 a 2015/1	E

Fonte: Elaborada pelo autor.

Conforme entrevista com o coordenador, a metodologia de ensino de ALP não foi alterada, desde o início do curso. Com a alteração do PPC, a ementa foi ajustada, a bibliografia atualizada, porém os livros sugeridos continuavam a abordar a metodologia tradicional.

Em 2011/1, ocorreu a troca de professores, o que ocasionou uma diminuição na quantidade de aprovações. A mudança de professor, com o objetivo de melhorar a interação com os alunos, não afetou de forma positiva o número de aprovações.

A disponibilidade de diversas estratégias, a partir de 2012/1, buscando um atendimento mais individualizado, como ocorreu por meio de horários de atendimento, criação de duas turmas em 2013/1 e outras estratégias, produziu uma mudança significativa na quantidade de alunos aprovados, chegando ao pico de 74% de aprovações em 2015/1.

Durante as pesquisas realizadas no capítulo 1 desta dissertação, observou-se que, em alguns casos, as instituições de ensino propõem a utilização de objetos de aprendizagem para aumentar o quantitativo de alunos aprovados (GARCIA, 2006), ou propõem estratégias diferentes de exposição de aulas, como a utilização de lousas digitais (SIMONETTO *et al.*, 2012), porém continuam utilizando o método clássico de ensino.

3 PROPOSTA DE UTILIZAÇÃO DA FERRAMENTA COMPUTACIONAL *SCRATCH* AUXILIANDO NO ENSINO DE ALP

Programar um computador significa nada mais, nada menos, do que se comunicar com ele em uma linguagem que tanto a máquina e quanto o usuário possam se entender (**Papert**).

A finalidade deste capítulo é descrever a utilização da ferramenta computacional *Scratch* como apoio ao ensino da disciplina de ALP de uma instituição de ensino superior e pública que utiliza a metodologia de ensino tradicional. Inicialmente será apresentada uma breve explanação sobre a ferramenta computacional *Scratch*, suas origens, a linguagem de programação utilizada pela ferramenta e alguns detalhes do órgão que atualmente é responsável pela manutenção da ferramenta. Em seguida, será descrita a metodologia de ensino que a ferramenta construtivista *Scratch* utiliza, detalhando informações importantes desta metodologia. É importante esclarecer que a proposta inicial para utilização da ferramenta computacional *Scratch* era a aplicação da ferramenta na disciplina pelo próprio autor desta dissertação, porém, após a distribuição de disciplinas para o segundo semestre de 2015, na instituição de ensino definida para a aplicação da ferramenta, esta foi alocada para outro professor do quadro permanente da instituição. Este professor havia ministrado a disciplina nos dois semestres anteriores. Apesar da situação adversa, o professor que assumiu a disciplina aceitou realizar a inserção da ferramenta computacional *Scratch* durante a ministração de suas aulas. Foi disponibilizado ao autor desta dissertação três semanas para a apresentação da ferramenta computacional *Scratch*, de forma que os alunos e professor pudessem conhecer a ferramenta computacional como apoio aos seus estudos. Durante este período também foi realizada a aplicação de exercícios para diagnosticar e iniciar um trabalho de desenvolvimento dos conceitos de algoritmos e lógica de programação, conforme proposto pelos autores construtivistas.

Ao final da disciplina foi aplicado questionários aos alunos e professores coletando a opinião dos mesmos em relação à utilização da ferramenta. Também, conforme realizado no capítulo II: foram coletadas informações quantitativas e qualitativas para comparações com os anos anteriores. Esta discussão foi detalhada neste capítulo IV.

3.1 A FERRAMENTA COMPUTACIONAL CONSTRUTIVISTA SCRATCH

O *Scratch* é uma linguagem de programação criada em 2003 pelo *Media Lab* do MIT (*Massachusetts Institute of Technology*), por meio de um projeto do grupo *Lifelong Kindergarten*. A finalidade de criação do *Scratch* é possibilitar a qualquer indivíduo programar suas próprias histórias interativas, jogos e animações e compartilhar suas criações com outros membros da comunidade online¹⁰.

Baseado na linguagem de programação **Logo** (PAPERT, 1980, apud ROBERTO LIMA, 2009) e do *Etoys* (KAY 2010; STEINMETZ, 2002 apud ROBERTO LIMA, 2009)

¹⁰ Site oficial do Scratch - <<https://scratch.mit.edu/>>.

apresentados no capítulo 1, o *Scratch* é uma linguagem de programação muito simples e intuitiva, o que a torna ideal para ser usada por principiantes, jovens ou adultos, que queiram iniciar-se no mundo da programação de computadores, ganhando gosto e asas para voos mais altos noutras linguagens mais poderosas e profissionais (FREDERICO, 2009).

Seymour Papert, o idealizador da linguagem de programação **Logo** trabalhou com Jean Piaget durante cinco anos, tratando sobre a natureza do pensamento e como as crianças o desenvolvem. Gregolin (2008) nos diz que a linguagem **Logo** surgiu:

A partir da reflexão simultânea de como as crianças pensam e como os computadores poderiam pensar. Papert planejou uma linguagem computacional que deveria ser apropriada para crianças, mas que não fosse uma linguagem de brinquedo. Deveria ter o poder das linguagens profissionais e também constituir em fáceis vias de acesso para principiantes sem domínio da matemática.

Atualmente, o *Scratch* possui uma página de *internet*, mantida pelo MIT⁶ no que é mantido um repositório de projetos, jogos, comunidades, vídeos e inclusive um editor on-line, a partir do qual, qualquer usuário possa criar seus protótipos sem necessitar instalar o aplicativo em seu computador.

3.2 A PROPOSTA METODOLÓGICA DO SCRATCH

O objetivo principal do *Scratch* é iniciar a programação para leigos em linguagens de programação. Sua principal característica é a simplicidade de utilização, assemelhando a jogos de LEGO¹¹. O software *Scratch* possibilita uma aprendizagem baseada no conceito de *design*, enfatizando a concepção, ou seja, valoriza a criação dos objetos e não, simplesmente, a utilização ou interação. A possibilidade de personalização (criação de algo que é pessoalmente significante), colaboração (relacionamentos interpessoais) e reflexão (possibilidade de agir sobre o próprio “erro”) tornam o *software* extremamente eficiente para atividades que valorizem o desenvolvimento cognitivo. (AMARAL & FRANÇA, 2013).

Assim como o **Logo**, a proposta metodológica do *Scratch* foi criada, baseada no construtivismo Piagetiano, corrente teórica que se fundamenta em explicar como a inteligência humana se desenvolve, a partir do pressuposto de que o desenvolvimento da inteligência humana é determinado pelas ações mútuas entre o indivíduo e o meio em que vive (ARGENTO, 2010). Piaget (1975) separa o processo cognitivo inteligente em dois conceitos: aprendizagem e desenvolvimento. Aprendizagem referindo-se à aquisição de uma resposta particular, por meio da interação com o meio (experiência). Desenvolvimento

¹¹ Grupo LEGO é uma empresa privada com sede em Billund, na Dinamarca, que produz jogos de montagem de objetos. Disponível em: <<http://www.intercom.org.br/papers/nacionais/2014/resumos/R9-2400-1.pdf>>. Acesso em 8 jun 2016.

referindo-se a uma aprendizagem de fato, sendo este o responsável pela formação do conhecimento. Para ele, o desenvolvimento pode ser descrito em quatro estágios ou fases de transição: Sensório-Motor (0-2anos), Pré-operatório (2-7anos), Operatório Concreto (7-12 anos) e Operatório Formal-Lógico (12-16 anos). Com o Scratch é possível criar atividades para trabalhar com as crianças, jovens e adultos, em cada uma das etapas do desenvolvimento, segundo Piaget.

Sendo uma evolução do **Logo**, o *Scratch* possui muitas de suas características como, por exemplo, a cada instrução, o usuário pode solicitar a sua execução, de forma que o personagem tartaruga no Logo ou o gato no *Scratch* execute a instrução de forma visual para o usuário. Porém, a grande evolução do *Scratch* em relação ao **Logo**, é que, enquanto no **Logo**, o usuário tem que se preocupar com a sintaxe, pois os comandos são digitados, no *Scratch*, basta arrastar os blocos de instruções pré-definidos, não havendo erros de sintaxe. Consequentemente, o aluno pode se preocupar com a montagem lógica do programa, deixando de lado os problemas relacionados com a sintaxe.

A partir do construtivismo Piagetiano, Seymour Papert (1980) sugeriu a interação do computador com os alunos, considerando o computador como uma máquina de aprender, diferentemente das outras propostas de ensino existentes. Definindo uma nova corrente teórica, chamada Construcionismo, conforme apresentado no capítulo II. Por meio dos projetos desenvolvidos com a ferramenta *Scratch*, é possível aos alunos, adquirir conhecimento sobre sequência, evento, paralelismo, loop, condicionais, operadores de dados entre outras funções consideradas pontos fundamentais de uma linguagem de programação. (AMARAL, FRANÇA, 2013).

Partnership for the 21st Century (apud AMARAL & FRANÇA, 2013) identificam nove tipos de habilidades de aprendizagem, consideradas essenciais para o século XXI e que são incentivadas pelo uso do *Scratch*. Estas habilidades são divididas em três áreas chaves: **Habilidades de Informação e Comunicação**: os alunos aprendem a selecionar, criar e gerenciar múltiplas formas de mídia, incluindo texto, imagens, animações e gravações de áudio;

Habilidade de pensar e resolver problemas: à medida que os alunos aprendem a programar, eles aprimoram seu raciocínio crítico e pensamento sistemático, no desenvolvimento de suas soluções, em contextos significativos;

Habilidades Interpessoais e auto direcionáveis: Os objetos visuais presentes no *Scratch* e código modular suportam a colaboração entre os alunos possibilitando que os mesmos trabalhem em conjunto.

Apesar de possuir diversas vantagens na sua utilização para processo de ensino-aprendizagem, devem-se tomar determinados cuidados ao propor a ferramenta *Scratch* como plataforma de ensino para a disciplina de ALP. Por se tratar de uma ferramenta que se apresenta visivelmente concebida para instigar crianças para sua utilização, os alunos jovens e adultos podem querer responsabilizar a ferramenta por não conseguirem atingir os objetivos propostos pelo professor ou se sentirem desmotivados pela utilização do mesmo, afirmando ser um “jogo de criança” (AURELIANO & TEDESCO, 2013).

3.3 TRABALHOS CORRELATOS – UTILIZAÇÃO DA FERRAMENTA *SCRATCH*

São várias as propostas de utilização da ferramenta *Scratch*, visando produzir a melhoria do processo ensino-aprendizagem de programação em instituições de ensino (RAMOS DOS SANTOS, 2014; SCAICO *et al*, 2013; PEREIRA *et al*, 2012; NETO, 2013; CARNEIRO *et al*, 2013).

SCAICO *et al* (2013) propõe a inserção da ferramenta *Scratch* nas turmas da Educação Básica, conforme proposta realizada por alguns países da Europa e Estados Unidos. A iniciação de programação em jovens faz com que estes deixem de ser somente consumidores de tecnologia, para que também desenvolvam habilidades para serem produtores de novas tecnologias. O autor acredita que, por meio da programação, é possível melhorar habilidades nos estudantes em fase escolar, como por exemplo, o raciocínio lógico. SCAICO *et al* (2012^a) afirmam em seu artigo que:

Aprender uma linguagem de programação é uma tarefa desafiadora. Todavia, tornar o ensino de programação mais acessível para um maior número de indivíduos é algo importante porque é capaz de estimular muitas capacidades cognitivas e para que aquele que aprende possa aplicar as técnicas utilizadas na programação na resolução de diversos outros tipos de problemas [...] A principal dificuldade encontrada pelos programadores iniciantes é a questão da sintaxe, pois eles precisam compreender a linguagem que o computador entende e também o idioma, já que nem todas as sintaxes das linguagens de programação possuem a clareza suficiente para que um novato entenda a funcionalidade e o uso dos seus comandos.

Para a aplicação da utilização do *Scratch*, foram escolhidas escolas públicas do estado do Paraná, em 2011. Estas foram divididas em quatro turmas, contendo cada uma delas oito alunos, com professores diferentes para cada uma delas. Anteriormente à aplicação do *Scratch*, os estudantes participaram de diversas atividades, com temas relacionados à área da tecnologia. Foram feitas por meio de competições, com a finalidade de instigar os alunos com relação às várias possibilidades relacionadas à tecnologia. Após cada aula teórica do *Scratch*, foram ministradas aulas práticas, desenvolvendo capacidades relacionadas à descoberta de erros nos códigos e trabalho colaborativo. Um ambiente colaborativo foi estabelecido para

que os alunos das diferentes turmas pudessem trocar experiências entre alunos e professores, semanalmente. O projeto desenvolvido demonstrou que, apesar das inúmeras deficiências que os estudantes possuíam e que estavam relacionadas à escrita, leitura e fundamentação lógico-matemática, foi possível ensinar e aprender os conceitos básicos de programação, ao passo que, se aprendia também sobre o mundo, por meio da própria programação (SCAICO *et al*, 2013).

Por sua vez, Ramos (2014) realizou um trabalho de utilização da ferramenta computacional *Scratch* em conjunto com a estratégia *Gamification*, uma estratégia de interação entre pessoas, usada essencialmente nas empresas, produzindo estímulos positivos e negativos ao público com quem lida para se alcançar um determinado objetivo. O autor desenvolveu um conjunto de jogos utilizando o *Scratch* para utilizar a estratégia *Gamification*. Os jogos foram usados por cinco turmas do 8º ano de escolaridade, alunos com idade aproximada de 13 anos. Após a aplicação dos jogos, as principais conclusões atingidas pelo autor foram: a junção da *Gamification* com o *Scratch* foi uma solução possível para ter bons resultados acadêmicos de forma descontraída. As notas finais da *Gamification* situaram-se entre os 73% e 92%; e as notas com o *Scratch* situaram-se entre os 85% e os 90%. E, para desenvolver as competências pessoais de cada indivíduo, nomeadamente, a competitividade positiva foi motivada por meio de um espírito de competição saudável e amigável, tendo melhorado o comportamento na sala de aula, em relação ao período de observação (RAMOS DOS SANTOS, 2014).

Nestes dois casos apresentados, os autores optaram pelo trabalho com alunos dentro do Ensino Básico, nas idades onde é realizada, conforme Jean Piaget, o desenvolvimento da cognição voltada ao raciocínio lógico. Porém, muitos alunos que ingressam no ensino superior ou ensino Técnico de curso da área de Ciências Exatas e da Terra e cursos tecnológicos, chegam ao início de seus estudos com grandes dificuldades no raciocínio lógico. Por sua vez, o autor Carneiro *et al* (2013) propôs a implantação da ferramenta computacional *Scratch* no ensino superior, na Universidade Estadual de Feira de Santana. Seu objetivo foi facilitar o aprendizado de programação de computadores para os estudantes do curso de Engenharia da Computação, visando diminuir os índices de retenção nas disciplinas iniciais de programação. O trabalho foi aplicado em forma de oficina tendo uma carga horária total de 4 horas. Vinte e seis alunos participaram, sendo conduzida por seis estudantes do curso de Engenharia da Computação, sendo dois tutores e quatro monitores. Contou, ainda com um professor, para observação e avaliação da metodologia aplicada, bem como a resposta dos estudantes à oficina. O objetivo principal do autor foi promover a motivação nos alunos

ensinando-os a programar brincando. Após a proposição de problemas, os alunos buscariam as resoluções, utilizando as ferramentas disponíveis no *Scratch*. Como resultado obtido pela aplicação da experiência, os autores declararam que, apesar de, inicialmente, a maioria dos alunos não possuem conceitos de programação, intuitivamente, eles já se mostraram capazes de criar cenários e objetos para o jogo proposto. Foi observado que, apesar de o jogo proposto ser um desafio mais geral, cada estudante estava livre para desenvolver a solução e o desenvolvimento de cada jogo. A grande maioria dos alunos conseguiu concluir todas as metas da oficina. Alguns foram além e introduziram funcionalidades ao jogo que não foram propostas inicialmente, o que foi considerado como um resultado positivo e reflexo direto da utilização da metodologia ativa de aprendizagem. Outro efeito também observado pelos autores foi a redução nas taxas de reprovação dos estudantes ingressantes no curso de Engenharia de Computação na disciplina de Algoritmos e Programação I, demonstrados por dados estatísticos. Porém, os autores ressaltam que a diminuição nos índices de evasão e retenção não foram somente pela utilização da ferramenta computacional *Scratch*, fatores como mudança de professores ministrantes da disciplina, mudança de metodologia aplicada, modificações nos trabalhos propostos para os estudantes e, até mesmo, as características inerentes aos alunos ingressantes, influenciaram nos índices obtidos (CARNEIRO *et al*, 2013).

3.4 SCRATCH – DETALHAMENTO DE SUA UTILIZAÇÃO NA DISCIPLINA

Conforme trabalhos correlatos apresentados na seção anterior e em conjunto com o professor da disciplina, foi preparado um plano de aula que previa a utilização da ferramenta *Scratch* para acontecer na segunda, terceira e quarta semana do curso de ALP (3 semanas com carga horária semanal de 8 aulas com 45 minutos cada aula, totalizando 24 aulas). A disciplina de ALP possui, dentro da matriz curricular do curso de ADS (Análise e Desenvolvimento de Sistemas), uma carga horária total de oito aulas semanais que aconteceram durante um total de 20 semanas, no segundo semestre letivo de 2015. A disciplina possui uma carga horária total de 120 horas, sendo que 18 horas foram separadas para trabalhar exclusivamente com a Lógica computacional, utilizando a ferramenta *Scratch*. Após a introdução à Lógica computacional com a utilização direta da ferramenta *Scratch*, a ferramenta ficaria disponibilizada aos alunos, de forma que paralelamente à ministração da disciplina, os alunos poderiam utilizá-la como apoio aos estudos na disciplina de ALP.

A proposta de utilização do *Scratch*, durante as três semanas, foi aplicada conforme detalhada no **Quadro 5**, abordados semanalmente. Posteriormente, será detalhada cada uma das semanas citadas no **Quadro 5**.

Quadro 5: Plano de aula para utilização da disciplina <i>Scratch</i> .		
Semana	Carga horária	Conteúdo previsto
2º	8 aulas – 6 horas	Introdução à lógica de programação Tipos de linguagens Linguagem estruturada e sequencial – Exemplo Light Bot Apresentação da metodologia do <i>Scratch</i> Apresentação da ferramenta computacional <i>Scratch</i> : comandos, barras de ferramentas, modos de salvamento, modos de utilização, personagem, simulador do Ecrã, execução do programa entre outras funcionalidades do software.
3º	8 aulas – 6 horas	Apresentação de problemas e aplicação da lógica na resolução dos problemas. Aplicação do <i>Scratch</i> na resolução dos problemas.
4º	8 aulas – 6 horas	Criação de Jogos, exercícios e desafios utilizando a ferramenta computacional <i>Scratch</i> .
Fonte: Elaborada pelo próprio autor, a partir da proposta desenvolvida com o professor da disciplina.		

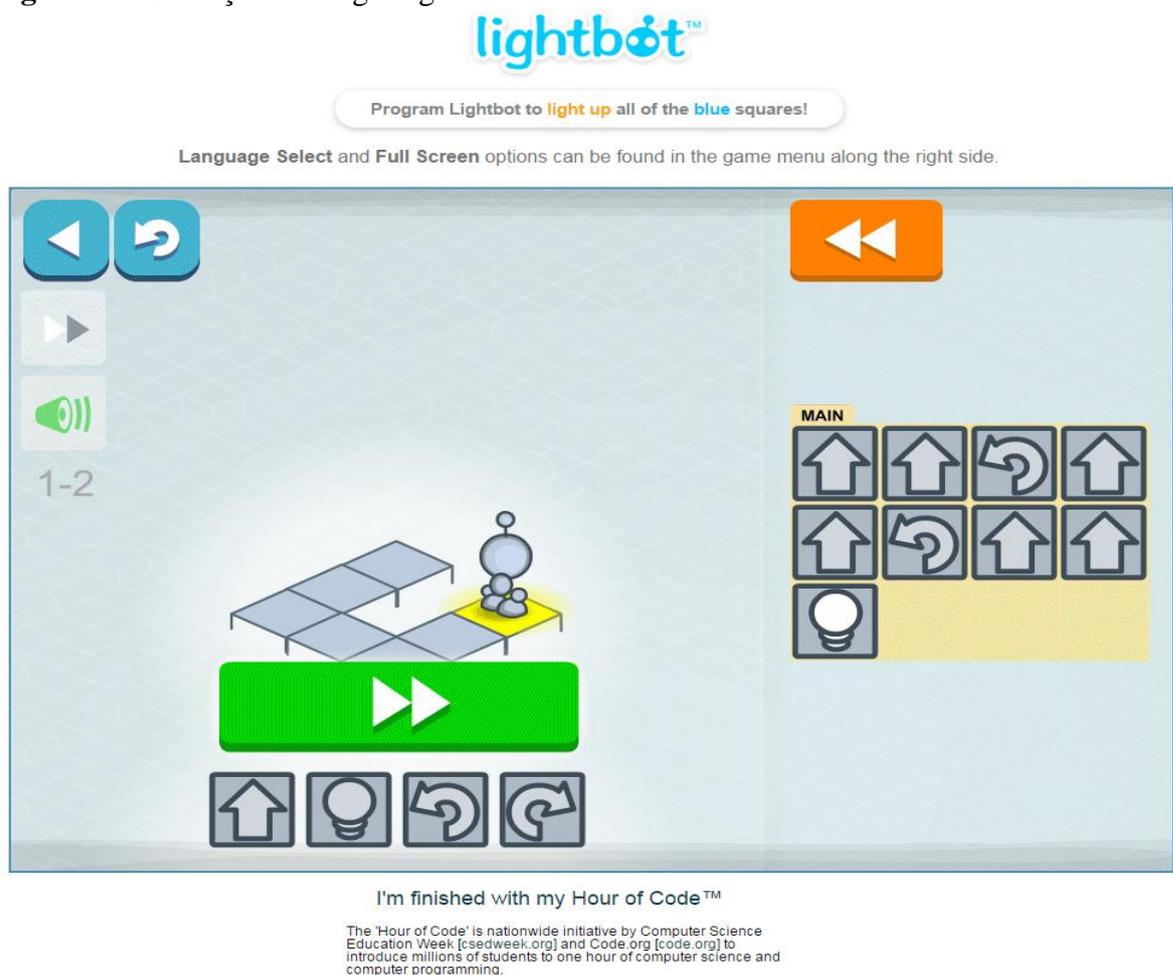
3.4.1 DETALHAMENTO DA 2ª SEMANA

Nesta segunda semana, como a disciplina de ALP é a primeira disciplina que os alunos possuem contato com programação, apesar de a turma possuir alunos de dependência, foi separado um momento para apresentação de conceitos importantes e necessários à utilização da ferramenta computacional *Scratch*. Dessa forma, usando quadro, giz, datashow e pequenas discussões, os conceitos fundamentais da introdução à lógica de programação, tipos de linguagens e a proposta metodológica de utilização do *Scratch* foram apresentados na seção anterior.

Durante a ministração da utilização da ferramenta *Scratch*, percebeu-se que, enquanto os novatos estavam muito preocupados com os conteúdos da disciplina e seus objetivos, os alunos veteranos questionavam quando se iniciaria a linguagem de programação estruturada propriamente dita, para já começarem a programar. Neste momento, visando demonstrar a

importância da lógica de programação para a resolução de problemas e construção dos algoritmos, usou-se o jogo *Light Bot*. Este jogo possui a mesma proposta metodológica do *Scratch* (utilização de blocos pré-definidos para montagem dos algoritmos), porém, em forma de um jogo onde cada fase possui um objetivo pré-definido de movimentação do personagem, neste caso um robô. A diferença entre a ferramenta *Scratch* e o jogo *Light Bot* é que, com a ferramenta *Scratch* não existe fases, o aluno tem a total liberdade de criar seus próprios jogos. Na **Figura 12** é apresentado o jogo *Light Bot*.

Figura 12: Utilização do Jogo *Light Bot*



Fonte: Imagem retida do Site <https://lightbot.com/>

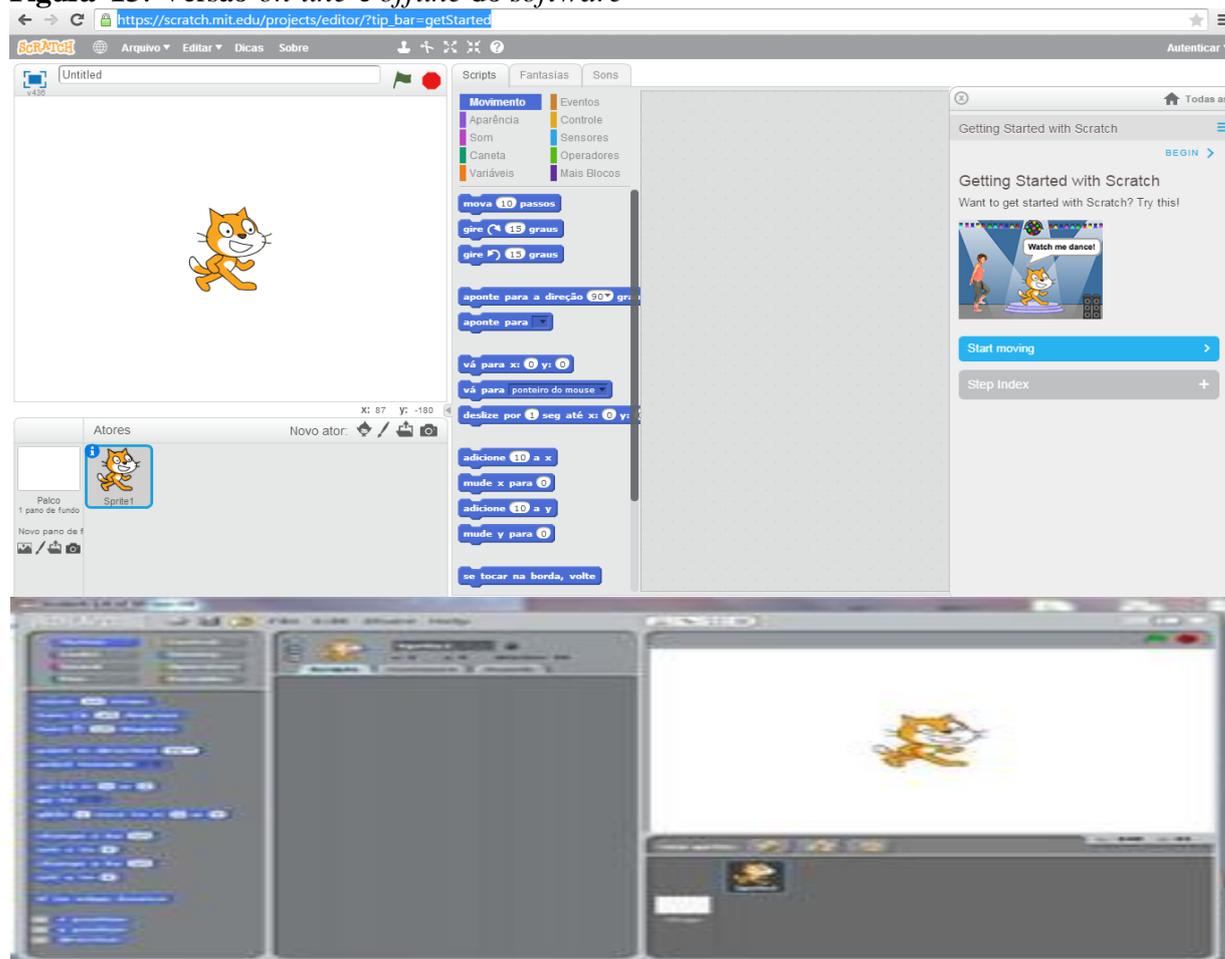
Para surpresa deste autor, os alunos se sentiram muito motivados e extremamente interessados com a utilização do jogo *Light Bot*, sendo que, quase todos os alunos finalizaram a primeira etapa do jogo disponibilizado no link <https://lightbot.com/hour-of-code-2015-flash.html>, em sala de aula, utilizando, com facilidade, os comando como: repetição, execução, movimentação, rotação entre outros.

Em seguida, foi realizada a apresentação da ferramenta computacional *Scratch*: os principais comandos, as barras de ferramentas, os modos de salvamento, os modos de utilização, o personagem, o simulador do Ecrã, a execução do programa, entre outras funcionalidades do software, conforme apresentados na **Figura 14**.

O software *Scratch* pode ser obtido, de forma gratuita, no site oficial do *Scratch*¹², sendo que neste capítulo apresentou-se a versão 2.0. Quando se opta pela instalação do *Scratch* em um sistema operacional, pode-se criar projetos de forma *off-line*.

A versão apresentada aos alunos funciona em *Mac*, *Windows* e em algumas versões Linux de 32 bits (**Debian**, **Ubuntu**). Também é possível executar esta mesma versão do *software on-line*¹³. Para a aplicação em sala de aula, visto a facilidade de utilização, optou-se pela versão *on-line* da ferramenta. Dessa forma, o usuário não precisa instalar o programa, porém, necessitará estar a todo momento com acesso à *internet* estabelecida. Abaixo, na **Figura 13**, é apresentada a tela inicial de cada uma das versões.

Figura 13: Versão *on-line* e *offline* do software



Fonte: Print Screen da tela do aplicativo Scratch versão 2.0

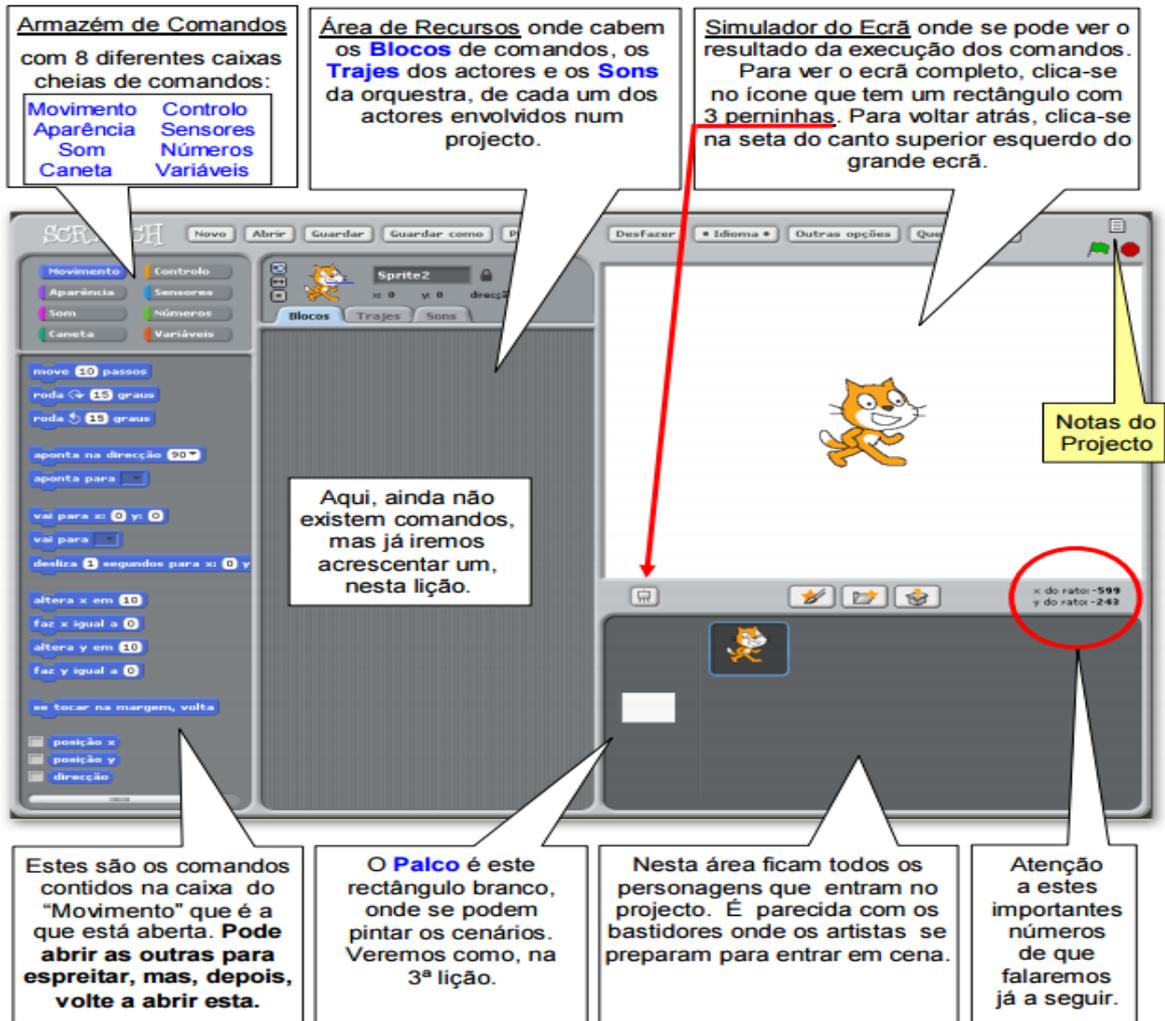
¹² Site para download do programa Scratch: <<https://goo.gl/ZM6tX>>

¹³ Site para execução on-line do software Scratch: <<https://goo.gl/7Oiqqj>>

Apresentou-se o Ambiente de Programação *Scratch*, versão *on-line*, e a forma de divisão de sua interface: (FREDERICO, 2009).

- **Scripts (Armazém de comandos):** com 10 grupos de comandos. Cada grupo contém funções pré-definidas que podem ser arrastadas para a área de recursos. Abaixo dos grupos, são apresentados os comandos, conforme o grupo escolhido. Na **Figura 13**, está selecionado o grupo de movimento, com comandos para movimentação do *Scratch*.
- **Área de Recursos:** Nesta área, são montados os programas. Cada bloco, advindo do armazém de comandos, é encaixado uns aos outros, de forma a criar um grupo de comandos, que serão executados no simulador, um duplo clique no bloco executa o comando, no simulador do Ecrã.
- **O simulador do Ecrã** apresenta a execução dos comandos inseridos na área de recursos.
- **O palco** é outra parte importante do ambiente, pois possui um retângulo onde se podem pintar o cenário e, também, por esta área, se pode visualizar os personagens que entram no projeto. Na **Figura 14**, é apresentado um resumo, contendo as atribuições de cada área.

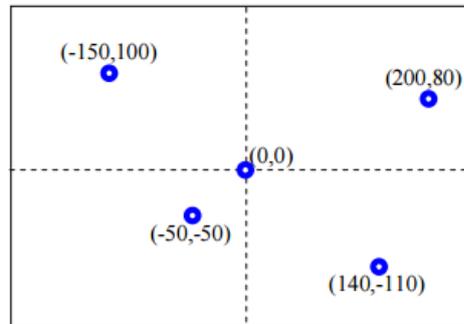
Figura 14: Tela inicial de abertura do Scratch



Fonte: Frederico (2009).

Realizaram-se pequenos testes de movimentação do personagem no palco visando fornecer aos alunos o entendimento de como funciona a movimentação do *Scratch* pelo simulador do Ecrã. Foi informado que é necessário visualizar esta área com o plano cartesiano, onde o ponto central do simulador do **Ecrã** é o ponto (0,0), conforme mostrado na **Figura 15**.

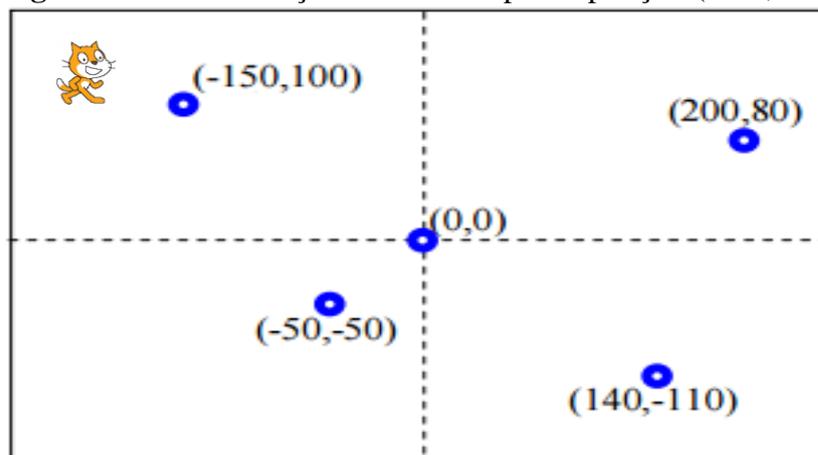
Figura 15: Plano cartesiano de representação da Tela Simulador do Ecrã



Fonte: Frederico (2009).

Para exemplificar, pode-se arrastar o comando **vai para x: y:** do Armazém de Comandos, grupo: Movimentos, para a Área de Recurso. Preenchendo o comando com “vai para x:-150 y:100”, o *Scratch* movimentou para a posição mostrada na **Figura 16**:

Figura 16: Movimentação do *Scratch* para a posição (-150,100)

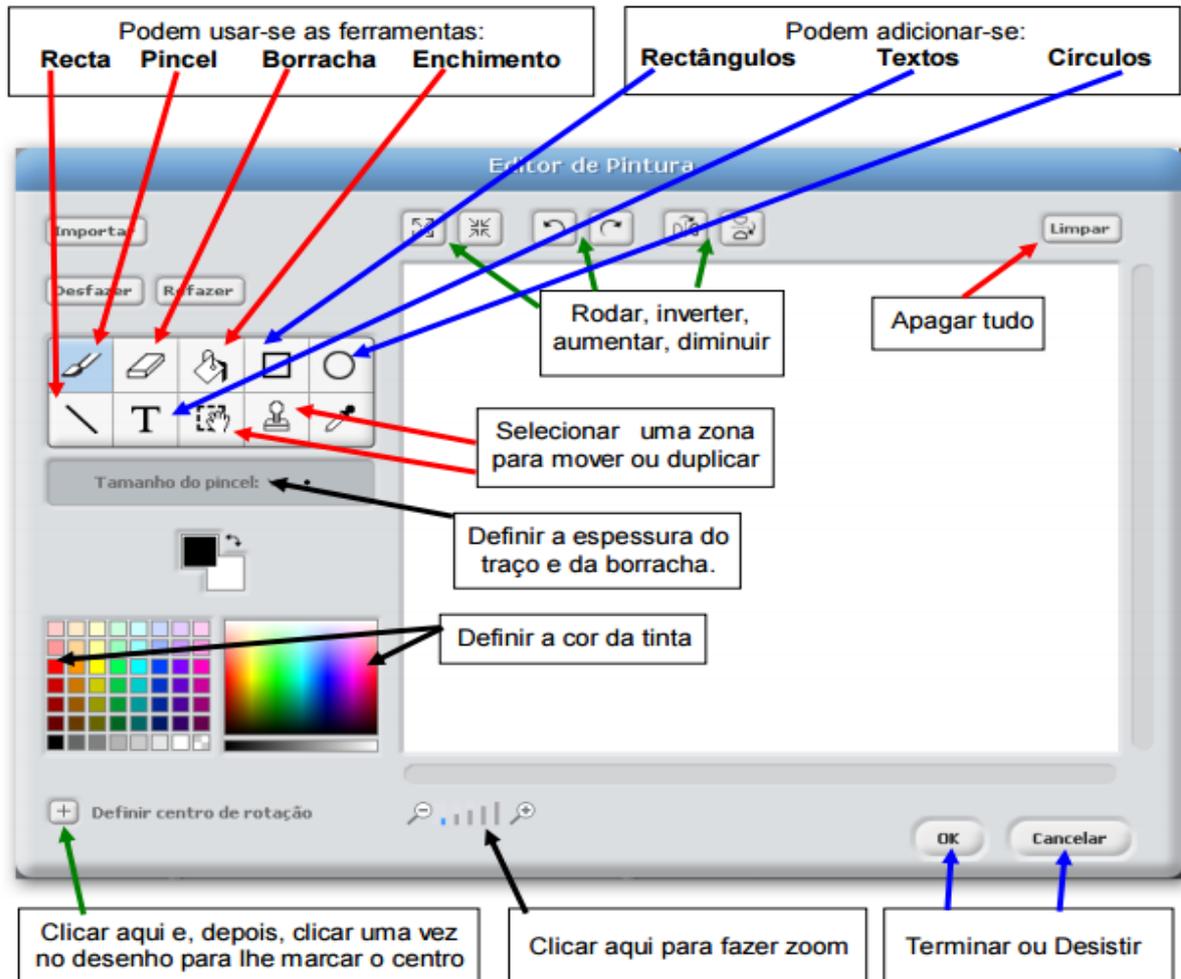


Fonte: Frederico (2009).

Quando se leva um bloco para área de recurso e o mesmo é encaixado em outro, para separá-los basta arrastar o bloco inferior. Também é possível encaixar um bloco entre um grupo de blocos, mas deve-se lembrar de que para retirá-lo novamente deve-se primeiro movimentar todos os blocos abaixo dele para em seguida removê-lo.

Para modificar o fundo do Painel do Ecrã, é possível usar o recurso de desenho. Para isto basta clicar no retângulo em branco do Palco e será selecionado o plano de fundo. Pela aba Plano de fundo, é possível desenhar, usando as ferramentas do *Scratch*. Este é um editor de imagens simples e baseado no **Paint** do *Windows*. Clicando no *Scratch*, ele, o ambiente volta para a edição do Gato. A **Figura 17** apresenta as principais funções de edição do plano de fundo.

Figura 17: Tela de desenho do Plano de Fundo do *Scratch*



Fonte: Frederico (2009)

No *Scratch*, também é possível criar seu próprio personagem. Na parte superior da área de recursos, há três abas: Blocos, Trajes e sons. A 1ª aba abre o Editor de Imagens para que se desenhe o personagem. A 2ª aba permite que se vá ao disco rígido buscar um previamente desenhado. A 3ª aba traz um personagem pré-desenhado qualquer. A **Figura 18** apresenta a tela com mais detalhes sobre a criação de personagens no *Scratch*. Para finalizar, basta clicar em Trajes e o novo personagem será exposto na tela.

Figura 18: Tela de Edição do personagem *Scratch*



Terminado o desenho, deve clicar-se no botão "OK" para voltar ao ambiente de programação e, então:

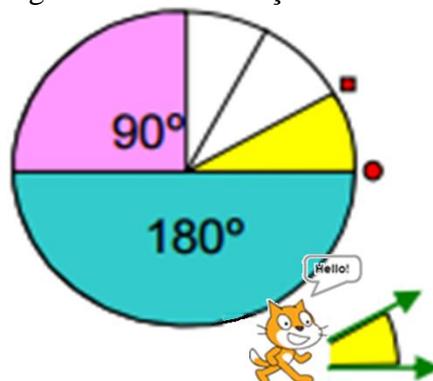
- Clicar no campo do nome para escrever "Joana"
- Clicar no separador "Trajes" para fazer o 2º traje
- Clicar no campo do nome para escrever "Vermelha"
- Clicar no botão "Copiar" do 1º traje para fazer o 2º
- Clicar no campo do nome para escrever "Verde"
- Clicar no botão "Editar" do 2º traje para entrar na folha de desenho e mudar-lhe a cor para verde (ou outra) com o "Enchimento" Não deve mudar-se o centro.
- Este botãozinho serve para apagar um traje.

Numeração automática

Fonte: Frederico (2009).

Para rotacionar o personagem, deve-se utilizar o comando "gire" no sentido horário e anti-horário. Os valores de rotação são inseridos no comando na unidade de medida GRAUS. A **Figura 19** apresenta a explicação detalhada do ângulo de movimentação do *Scratch*.

Figura 19: Tela de indicação do ângulo de movimentação do *Scratch*



Fonte: Frederico (2009).

Foram apresentados aos alunos, os comandos de aparência, som, caneta, eventos, controle, sensores, operadores por meio de exemplos simples, em que pequenos algoritmos foram criados junto com eles. A utilização de todos os blocos do programa seguiu o mesmo padrão dos comandos apresentados anteriormente, encaixando cada bloco na posição desejada e, em seguida, solicitando a execução do bloco. A **Figura 20** demonstra esta interação do comando "diga" – *Hello!*

Figura 20: Interação do *Scratch* através de balões – utilizando comando “diga”



Fonte: Frederico (2009).

3.4.2 DETALHAMENTO DA 3ª SEMANA

Conhecidas a metodologia, comandos e forma de utilização da ferramenta *Scratch*, foi proposto aos alunos resolverem problemas diversos utilizando os blocos de comando do *Scratch*. O nível de dificuldade dos problemas propostos foi aumentando gradativamente, conforme os alunos demonstravam que novas habilidades foram adquiridas. O detalhamento dos problemas propostos é apresentado no **Quadro 6**:

Quadro 6: Exemplos de problemas apresentados aos alunos

Nível do problema	Exemplos de problemas propostos
1	<ul style="list-style-type: none"> - movimentar o gato para esquerda e direita; - movimentar o gato para esquerda e, em seguida, rotacioná-lo para retorno à origem; - alterar o personagem; - mudar o palco; - inserir um novo palco com desenho de uma rua e, em seguida, movimentar o gato;
2	<ul style="list-style-type: none"> - Inserir dois personagens e realizar interação entre eles; - fazer realizar embaixadinha com uma bola; - fazer o personagem desenhar um quadrado na tela com a caneta; - fazer o personagem desenhar diferentes formas geométricas com a caneta;
3	<ul style="list-style-type: none"> - Fazer o personagem movimentar na tela infinitamente e, quando tocar na borda, inverter a direção; - fazer surgir uma bola em uma das partes superior da tela aleatoriamente e movimentá-la na vertical até a parte inferior da tela; - quando a bola do exercício anterior chegar à parte inferior, reaparecer na parte superior novamente;
4	<ul style="list-style-type: none"> - movimentar uma barra na horizontal por meio dos controles: direita e esquerda do teclado; - movimentar uma barra na vertical por meio dos controles: direita e esquerda do teclado; - fazer aparecer uma bola aleatoriamente na parte superior e descer na vertical de forma infinita, desde que toque a barra movimentada pelo usuário.

Fonte: próprio autor, a partir dos exercícios propostos aos alunos na 3ª semana de ensino da ferramenta *Scratch*

Para cada problema apresentado, o professor agia como um mediador, dando dicas aos alunos, mas deixando que eles mesmos realizassem a proposta de resolução de cada problema. Durante a resolução dos problemas, foi comum surgir propostas diferentes que produziam o mesmo resultado final. Após a proposta de cada problema apresentada aos alunos, era apresentada na lousa, a todos os alunos, a resolução do colega que utilizou menos blocos de comandos para a resolução, resultando em uma pequena competição ao final da semana.

3.4.3 DETALHAMENTO DA 4ª SEMANA

Para a última semana de utilização da ferramenta computacional *Scratch*, foi proposta aos alunos, a criação de um jogo com as seguintes definições:

- Possibilitar ao jogador iniciar o jogo conforme seu comando inicial, definido pelo desenvolvedor do algoritmo;
- Devem ser gerados, aleatoriamente, objetos, na parte superior da tela;
- Estes objetivos devem movimentar-se para a parte inferior da tela em 5 segundos, gradualmente;
- Deve existir uma barra na parte inferior da tela, movimentando-se somente na horizontal, conforme o controle do jogador;
- Enquanto os objetos toquem na barra, o jogo continua gerando, aleatoriamente, novos objetos na tela;
- Caso algum objeto alcance a borda inferior da tela antes do contato com a barra, o jogo encerra-se, apresentando uma mensagem de finalização e questionando ao usuário se o jogo se encerra ou reinicia.

Todos os alunos assumiram, de prontidão, a realização do desafio. Lembrando que o menor algoritmo para a resolução seria apresentado na tela e discutido com todos os alunos.

Após esta semana, o professor assumiu a disciplina e continuou a ministração da mesma, conforme plano de aula (Apêndice C).

4 RESULTADOS DA APLICAÇÃO DA FERRAMENTA *SCRATCH* NA DISCIPLINA DE ALP

Ensinar não é transferir conhecimento, mas criar as possibilidades para a sua própria produção ou a sua construção (**Paulo Freire.**).

Visando buscar resultados mais efetivos a respeito da utilização da ferramenta Scratch como apoio ao ensino de ALP, foram coletados dados que tornassem possíveis a realização de uma análise quali-quantitativa. Conforme expresso por Gil (2010), o pesquisador deve tomar o cuidado de, apesar do ensino da utilização da ferramenta Scratch, durante as três semanas previstas no plano de aula, terem sido realizadas pelo próprio pesquisador, que os resultados obtidos não fiquem subordinados à subjetividade do mesmo.

Gil (2010) afirma que, “Com efeito, nos estudos de caso os dados podem ser obtidos mediante análise de documentos, entrevistas, depoimentos pessoais, observação espontânea, observação participante e análise de artefatos físicos”. Dessa forma, escolheu-se como elementos para compor a coleta de dados, os artefatos físicos (taleta de notas da disciplina contendo índice de aprovação, índice de reprovação por nota, índice de reprovação por frequência e índice de alunos evadidos), entrevistas com alunos por meio de questionários aplicados ao final da disciplina, entrevista com o professor, também por questionário ao final da disciplina e entrevista com o coordenador com questionário aplicado ao final da disciplina.

Para a fase de análise de dados, foram comparados os dados atuais com os dados do histórico, fruto do capítulo 2. Apesar dos dados da comparação não poderem ser conclusivos a fim de informar se a utilização do software produziu mais ou menos conhecimento nos indivíduos das turmas em estudo, deixou-se a comparação, neste capítulo, para fins de contribuição no levantamento de dados para a instituição estudada. Ressalta-se a não conclusividade visto que, cada turma possui perfis diferentes para o aprendizado. Conforme apresentado pela psicóloga Sandra Vaz de Lima (2008), são vários os fatores que influenciam a aprendizagem dentre os quais se podem citar: carências afetivas; deficientes condições habitacionais, sanitárias, de higiene e de nutrição; pobreza da estimulação precoce; privações lúdicas, psicomotoras, simbólicas e culturais; nível elevado de ansiedade; relações interfamiliares; e muitas outras que vão além do perfil do professor ou das metodologias de ensino existentes, entrando nos campos social, econômico e psicológico de cada indivíduo, conforme a complexidade do ser humano e da formação individual do seu conhecimento.

4.1 RESULTADO FINAL DA DISCIPLINA: APROVADOS, REPROVADOS POR NOTA E REPROVADOS POR FREQUÊNCIA

Após a aplicação do software *Scratch* durante as três semanas, conforme previsão do plano de aula (Anexo A), a disciplina seguiu seu andamento normal, conforme sugerido pelo professor da disciplina. A ferramenta *Scratch* passou a ser apoio ao ensino da disciplina de ALP. Ao final da disciplina foram coletadas as informações do **Quadro 7**.

Quadro 7: Número de alunos matriculados, aprovados, reprovados por nota e reprovados por frequência na disciplina de ALG em 2015/2

Ano	Semestre	Alunos Matriculados	Aprovados	%	Reprovados por nota	%	Reprovados por falta	%
2015	2	30	10	33%	10	33%	10	33%

Fonte: Elaborada pelo autor.

4.2 RESULTADOS DA APLICAÇÃO DOS QUESTIONÁRIOS: ALUNO, PROFESSOR E COORDENADOR DE CURSO

Após a realização da última avaliação do curso, foi aplicado um questionário aos alunos presentes para levantamento de informações da utilização da ferramenta computacional construtivista *Scratch* (Modelo está no Apêndice A). Neste encontro, estavam presentes 13 alunos. Os resultados das perguntas de múltiplas escolhas do questionário estão detalhados no **Quadro 8**.

Quadro 8: Resultado do questionário aplicado aos alunos ao final da disciplina, respostas de múltiplas escolhas

Pergunta 1	A quanto tempo concluiu o Ensino Médio?									
Resultado	<u>9</u>	Menos de um ano	<u>3</u>	De 1 a 3 anos	<u>0</u>	De 3 a 5 anos	<u>1</u>	Mais de 5 anos		
Pergunta 2	Possui outro curso superior? Se SIM, terminou à:									
Resultado	<u>0</u>	Sim	<u>13</u>	Não	-	Menos de 1 ano	-	De 1 a 3 anos	-	Mais de 3 anos.
Pergunta 3	Você possui computador em casa?									
Resultado	<u>11</u>	Sim	<u>2</u>	Não						
Pergunta 4	Você trabalha?									
Resultado	<u>7</u>	Sim	<u>6</u>	Não						
Pergunta 5	É a primeira vez que faz a disciplina de Algoritmo e Lógica de Programação no IFTM? Se SIM, Você já possuía conhecimento de programação antes de cursar a disciplina? Se SIM, quando você estudou programação qual a linguagem e software que foram utilizados?									
Resultado	<u>13</u>	Sim	<u>0</u>	Não	<u>7</u>	Sim	<u>6</u>	Não	- PHP E DEV C++ - C e Java - VISUALG - DEV C++, JAVA, PHP, VISUALG - HTML, C/C++ E C#	
Pergunta 6	Seu tempo de dedicação semanal, fora de sala de aula, para estudar à disciplina de ALP é de									
Resultado	<u>7</u>	Menos de 1 hora	<u>3</u>	Menos de 1 hora	<u>1</u>	De 2 a 3 horas	<u>1</u>	Mais de 3 horas		
Pergunta 7	Você tentou criar outros programas utilizando a linguagem C, não propostos pelo professor durante a realização da disciplina de ALP?									
Resultado	<u>8</u>	Sim	<u>5</u>	Não						
Pergunta 9	Você conhecia a ferramenta <i>Scratch</i> de programação? Se SIM, durante quanto tempo utilizou a ferramenta?									

Resultado	<u>2</u>	Sim	<u>11</u>	Não	<u>1</u>	Menos de 1 ano	<u>1</u>	De 1 a 3 anos	<u>0</u>	Mais de 3 anos
Pergunta 10	O que você achou da utilização do <i>Scratch</i> na disciplina de ALP?									
Resultado	Não gostei e não entendi sua utilidade		Não gostei, mas entendi sua utilizada		Não gostei, mas entendi sua utilizada		<u>2</u>	Gostei mas achei difícil de usar	<u>11</u>	Gostei e achei fácil de usar
Pergunta 11	Você gostou de utilizar a ferramenta <i>Scratch</i> para criação de jogos?									
Resultado	<u>13</u>	Sim	<u>0</u>	Não						
Pergunta 12	Seu tempo de dedicação semanal, fora de sala de aula, para usar os softwares propostos pelo professor que auxiliam o estudo de ALP, (<i>Lightbot</i> e <i>Scratch</i>) foi de:									
Resultado	<u>10</u>	Menos de 1 hora	<u>3</u>	De 1 a 2 horas		De 2 a 3 horas		Mais de 3 horas		
Pergunta 13	Você chegou a utilizar o tutorial da ferramenta <i>Scratch</i> ou outros disponíveis <i>on-line</i> nos exercícios propostos pelo professor?									
Resultado	<u>8</u>	Sim	<u>7</u>	Não						
Pergunta 14	A disponibilidade <i>on-line</i> da ferramenta <i>Scratch</i> motivou-o a usá-lo?									
Resultado	<u>8</u>	Sim	<u>7</u>	Não						
Pergunta 15	Você tentou realizar outras tarefas ou exercícios não propostos pelo professor quando utilizou a ferramenta <i>Scratch</i>?									
Resultado	<u>9</u>	Sim	<u>4</u>	Não						
Pergunta 16	O que achou da utilização da ferramenta <i>Scratch</i> para programação?									
Resultado	<u>0</u>	Muito fácil	<u>5</u>	Fácil	<u>7</u>	Mediano	<u>1</u>	Difícil	<u>0</u>	Muito difícil

Fonte: Elaborada pelo autor.

As respostas às questões dissertativas do questionário, aplicadas aos alunos, estão detalhadas no **Quadro 9**.

Quadro 9: Resultado do questionário aplicado aos alunos ao final da disciplina, respostas dissertativas.

Pergunta 8	Quais as principais dificuldades encontradas durante a disciplina de ALP?
Respostas	<ul style="list-style-type: none"> - Dificuldade de entender; - Interpretar Códigos (2 alunos); - Exercícios são grandes e complexos; - Sem muita dificuldade, tive falta de tempo, mas gosto da matéria; - Aprender vetor e matriz, pois não tinha um conhecimento prévio; - Falta de tempo; - Entender a matéria, mas já estou melhorando.
Pergunta 17	Quais as principais dificuldades encontradas durante a utilização da ferramenta <i>Scratch</i>?

Respostas	<ul style="list-style-type: none"> - Repetição; - Alguns comandos não seguiam como deveriam, mesmo que fosse igual do professor; - Nenhum (2 alunos); - Entender a Lógica de rotação; - migrar para outro programa ou plataforma para programação; - As ferramentas ainda não conhecidas e a interface também eram desconhecidas; - A utilização de ângulos (2 alunos); - Minha grande dificuldade foi na parte Lógica pois nunca havia estudado.
Pergunta 18	Gostaria de comentar ou deixar sua opinião a respeito da disciplina de ALP ou da utilização de ferramentas de apoio utilizadas na disciplina de ALP (<i>Light Bot, Scratch</i> e outros)?
Respostas	<ul style="list-style-type: none"> - Ferramenta muito boa para ter um pouco de ideia de como é a matéria de ALP; - É notório que a utilização dessas ferramentas auxilia muito no aprendizado e deveria se tornar a grade da disciplina; - Eu achei muito interessante, inclusive ainda utilizo, ajudou a prática da lógica e a ajuda na hora de resolver outros problemas de programação; - Disciplina aplicada com muito empenho; - A programação tem que ser mais efetivada para casos atuais; - Bons para o desempenho da lógica; - A disciplina é boa mais tem que praticar para não prejudicar a si mesmo;

Fonte: Elaborado pelo autor.

Também foi aplicado um questionário ao professor da disciplina, possibilitando-nos ter uma visão do educador do processo, em relação à utilização da ferramenta computacional *Scratch* (Modelo está no Apêndice B). O resultado do questionário aplicado ao professor está detalhado no **Quadro 10**.

Quadro 10: Resultado do questionário aplicado ao professor, ao final da disciplina.									
Pergunta 1	A quanto tempo concluiu o curso de formação superior?								
Resposta	<input type="checkbox"/>	Menos de um ano	<input type="checkbox"/>	De 1 a 3 anos	<input type="checkbox"/>	De 3 a 5 anos	<input checked="" type="checkbox"/>	Mais de 5 anos	<input type="checkbox"/>
Pergunta 2	Possui outro curso superior? Se SIM, terminou à:								
Resposta	<input type="checkbox"/>	Sim	<input checked="" type="checkbox"/>	Não	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pergunta 3	Qual a modalidade da sua formação superior?								
Resposta	<input checked="" type="checkbox"/>	Bacharelado	<input type="checkbox"/>	Licenciatura	<input type="checkbox"/>	Tecnológico	<input type="checkbox"/>	Outro	<input type="checkbox"/>
Pergunta 4	Acredita que a carga horária da disciplina de ALP é suficiente para a ministração de seus conteúdos?								

Resposta	<input checked="" type="checkbox"/>	Sim		Não					
Pergunta 5	Você acredita que a maioria dos alunos que iniciam a disciplina de ALP possui o desenvolvimento lógico necessário à realização da disciplina de ALP?								
Resposta		Sim	<input checked="" type="checkbox"/>	Não					
Pergunta 6	Você acha que, em média, a maioria dos alunos disponibilizam quantas horas semanais ao estudo da disciplina de ALP?								
Resposta		Menos de 1 hora	<input checked="" type="checkbox"/>	Menos de 1 hora		De 2 a 3 horas		Mais de 3 horas	
Pergunta 7	Você acha que, em média, para o bom andamento da disciplina, quantas horas os alunos deveriam se dedicar à disciplina?								
Resposta		Menos de 1 hora		Menos de 1 hora		De 2 a 3 horas	<input checked="" type="checkbox"/>	Mais de 3 horas	
Pergunta 8	Você achou que os alunos desta turma estavam motivados para a realização da disciplina?								
Resposta	<input checked="" type="checkbox"/>	Sim		Não					
Pergunta 9	Quais as principais dificuldades você encontrou para ministração da disciplina de ALP?								
Resposta	<u>Alunos possuem fraca compreensão/interpretação de texto e raciocínio lógico, essenciais para algoritmos.</u>								
Pergunta 10	Você conhecia a ferramenta <i>Scratch</i> de programação?								
Resposta		Sim	<input checked="" type="checkbox"/>	Não					
Pergunta 11	O que você achou da utilização do <i>Scratch</i> na disciplina de ALP?								
Resposta		Não gostei e acho que os alunos também não		Não gostei, mas acho que os alunos gostaram de utilizá-la.		Gostei mas acho que os alunos não gostaram de utilizá-la	<input checked="" type="checkbox"/>	Gostei e acho que os alunos também	
Pergunta 12	Você conhece a metodologia construtivista de ensino utilizada pelo <i>Scratch</i>?								
Resposta		Sim	<input checked="" type="checkbox"/>	Não					
Pergunta 13	Você usa ferramentas computacionais de apoio ao ensino de ALP?								
Resposta	<input checked="" type="checkbox"/>	Sim		Não		Quais:	Visualg e Devepp		
Pergunta 14	Quais as principais dificuldades encontradas pelos alunos no desenvolvimento da disciplina de ALP?								
Resposta	<u>Dificuldade de compreensão dos conceitos, dificuldade de ligar um conceito ao outro para resolver problemas.</u>								

Pergunta 15	Gostaria de comentar ou deixar sua opinião a respeito da disciplina de ALP ou da utilização de ferramentas de apoio utilizadas na disciplina de ALP (<i>Light Bot e Scratch</i>)?
Resposta	<u>Achei muito interessante a utilização do Scratch e demais técnicas no início da disciplina, pois os alunos já tiveram um bom contato inicial com os conceitos de lógica de programação de uma maneira simples e divertida, menos pesada que a linguagem de programação final da disciplina.</u>
Fonte: Elaborada pelo autor.	

Para finalizar, foi aplicado um último questionário ao coordenador do curso. O resultado do questionário está detalhado no **Quadro 11**.

Quadro 11: Resultado do questionário aplicado ao coordenador do curso.										
Pergunta 1	Há quanto tempo concluiu o curso de formação superior?									
Resposta	<input type="checkbox"/>	Menos de um ano	<input type="checkbox"/>	De 1 a 3 anos	<input type="checkbox"/>	De 3 a 5 anos	<input checked="" type="checkbox"/>	Mais de 5 anos	<input type="checkbox"/>	<input type="checkbox"/>
Pergunta 2	Sua área de formação é na área de Informação e Comunicação? Há quanto tempo é coordenador do curso?									
Resposta	<input checked="" type="checkbox"/>	Sim	<input type="checkbox"/>	Não	<input checked="" type="checkbox"/>	De 1 a 3 anos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pergunta 3	Qual a modalidade da sua formação superior?									
Resposta	<input checked="" type="checkbox"/>	Bacharelado	<input type="checkbox"/>	Licenciatura	<input type="checkbox"/>	Tecnológico	<input checked="" type="checkbox"/>	Doutor em Ciência da Computação	<input type="checkbox"/>	<input type="checkbox"/>
Pergunta 4	Acredita que a carga horária da disciplina de ALP é suficiente para a ministração de seus conteúdos?									
Resposta	<input checked="" type="checkbox"/>	Sim	<input type="checkbox"/>	Não	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pergunta 5	Você acredita que a maioria dos alunos que iniciam a disciplina de ALP possui o desenvolvimento lógico necessário à realização da disciplina de ALP?									
Resposta	<input type="checkbox"/>	Sim	<input checked="" type="checkbox"/>	Não	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pergunta 6	Quais as estratégias realizadas pela instituição para ajudar os alunos, fora de sala de aula, no desenvolvimento da realização do curso?									
Resposta	<u>Disponibilização de horário de atendimento do professor que ministra a disciplina aos discentes que possuem dificuldades em assimilar o conteúdo. Além disso, há um discente monitor que está à disposição dos alunos, em horário extra-aula, para sanar dúvidas.</u>									
Pergunta 7	Você acredita que o perfil da turma que entrou em 2015/2 é diferente das anteriores? Justifique.									
Resposta	<input checked="" type="checkbox"/>	Sim	<input type="checkbox"/>	Não	<u>Segundo relato de professores que ministraram aulas para esta turma, os alunos que ingressaram neste semestre apresentaram dificuldades acima da média dos semestres anteriores. Um dos fatores que influenciam a grande dificuldade desses alunos é o</u>					

					fato de ser comum ter ingressos que tenham sido chamados após a 5ª chamada do SISU (Sistema de Seleção Unificada). Quanto mais distante da primeira chamada ocorrer o ingresso do aluno, normalmente maior é sua dificuldade. Em 2015/2, o último ingresso neste curso foi chamado na 7ª chamada do SISU. Além disso, nesta turma houve dois alunos que necessitaram de atendimento especial. Estes alunos foram acompanhados pelo NAPNE (Núcleo de Apoio a Portadores de Necessidades Especiais).
Pergunta 8	Você achou que os alunos desta turma estavam motivados para a realização do curso?				
Resposta	<input checked="" type="checkbox"/>	Sim		<input type="checkbox"/>	Não
Pergunta 9	Você conhecia a ferramenta <i>Scratch</i> de programação? Se SIM, durante quanto tempo utilizou a ferramenta?				
Resposta	<input checked="" type="checkbox"/>	Sim		<input type="checkbox"/>	Não
Pergunta 10	Você conhece a metodologia construtivista de ensino?				
Resposta	<input checked="" type="checkbox"/>	Sim		<input type="checkbox"/>	Não
Pergunta 11	Você acredita que é importante a utilização de ferramentas computacionais de apoio ao ensino de ALP?				
Resposta	<input checked="" type="checkbox"/>	Sim		<input type="checkbox"/>	Não
Pergunta 12	Em sua opinião, quais as principais dificuldades encontradas pelos alunos no desenvolvimento da disciplina de ALP?				
Resposta	<p>O perfil dos alunos que se matriculam neste curso costuma ser semelhante na maioria das instituições de ensino. Alunos acreditam que desde o primeiro dia em sala de aula trabalharão somente com computadores e que a maioria das aulas são práticas. No entanto, assim que o curso é iniciado, se deparam com muitas disciplinas teóricas, com alta carga de lógica e matemática. Neste momento, muitos alunos percebem que não tem afinidade com o curso, resultando em abandono do curso (a maior parcela de evasão ocorre entre os primeiros e terceiros períodos). Também há alunos que percebem que não tem o perfil desejado para o curso, mas, mesmo assim, continuam a cursá-lo, apenas para adquirir um diploma de curso superior. Nestes casos existe alto índice de reprovação. Portanto, diante deste cenário, ressalto que a resposta da questão 8 deste questionário refere-se à maioria dos alunos da turma, no entanto, há sempre parcela considerável de alunos desmotivados por conta de ingressarem no curso sem ter o perfil adequado.</p>				
Pergunta 13	Gostaria de comentar ou deixar sua opinião a respeito da disciplina de ALP ou da utilização de ferramentas de apoio que podem ser utilizadas na disciplina (<i>Light Bot e Scratch</i>)?				
Resposta	<p>Em conversa com alguns alunos que cursaram a disciplina de ALP em 2015/2, obtive <i>feedback</i> informando que a aplicação da ferramenta de apoio na disciplina foi positiva e auxiliou os alunos na compreensão de como funciona a lógica de programação. Acredito que o emprego desta categoria de ferramentas deverá facilitar a sedimentação do conhecimento abordado na disciplina, conhecimento este que demonstra um novo paradigma para solução de problemas a alunos que nunca tiveram contato com lógica de programação.</p>				
Fonte: Elaborado pelo autor.					

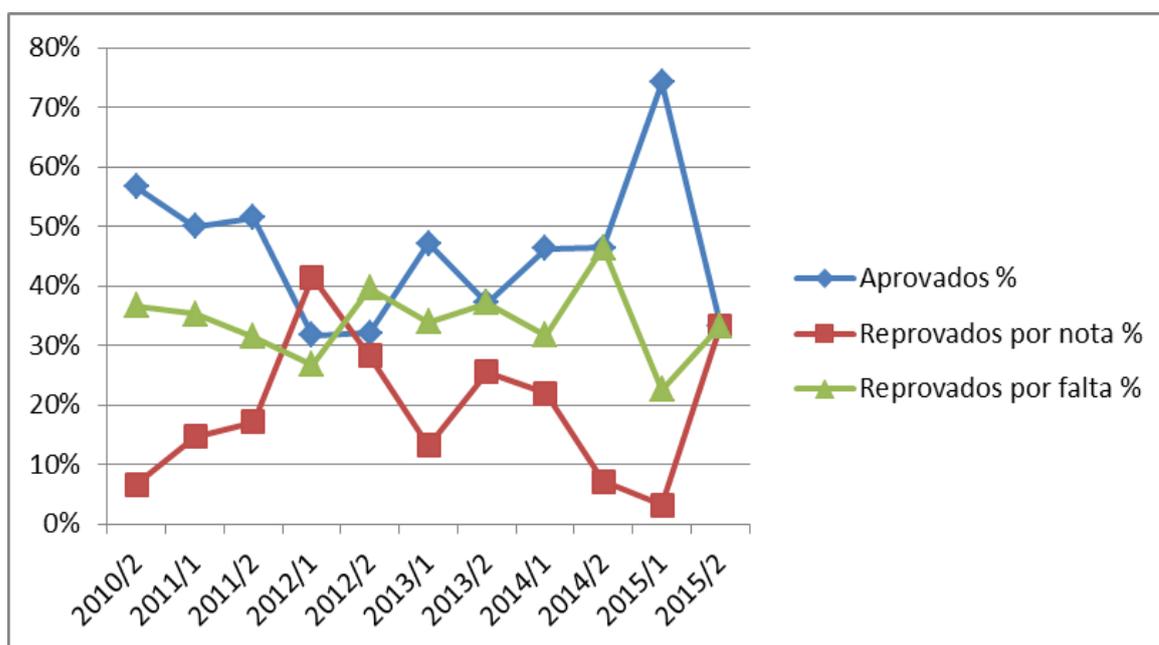
4.3 ANÁLISE CRÍTICA – RESULTADOS OBTIDOS PELA UTILIZAÇÃO DA FERRAMENTA SCRATCH NA DISCIPLINA DE ALP

A fim de facilitar a análise crítica dos resultados obtidos pela utilização da ferramenta *Scratch* na disciplina de ALP, conforme apresentado anteriormente por Gil (2010), dividiu-se a análise crítica, conforme coleta de dados apresentada na seção anterior.

4.3.1 ANÁLISE CRÍTICA – ÍNDICES DE APROVAÇÃO E REPROVAÇÃO

Para iniciar a análise crítica dos dados obtidos pela utilização da ferramenta computacional *Scratch*, acrescentaram-se as informações extraídas da taleta de notas finais da disciplina de ALP, conforme capítulo 4.1, às informações contidas no **Quadro 3**, referentes ao histórico de aprovações e reprovações na disciplina. A partir da união destas duas informações foi criado o **Gráfico 1**, que demonstra as curvas de alunos aprovados, reprovados por nota e reprovados por frequência na disciplina.

Gráfico 1: Curvas de alunos aprovados, reprovados por frequência e reprovados por nota.



Fonte: Próprio autor, a partir da união dos dados contidos nos quadros 3 e 7.

Existe um ponto de convergência no momento da aplicação da ferramenta computacional *Scratch*, criando um ponto interessante no **Gráfico 1**, pois as três curvas possuem o mesmo valor, 33,33%.

Observa-se que a curva de alunos reprovados por falta permaneceu praticamente constante (oscilando entre 25% a 40%), durante todo o período de análise, demonstrando que as estratégias propostas pelos professores e coordenação não conseguiram atingir

drasticamente este grupo. Porém, as curvas de alunos aprovados e reprovados por nota sofreram grandes variações (aprovados de 30 a 75% e reprovados por nota de 2% a 45%), demonstrando que a várias estratégias propostas afetaram bastante estas duas variáveis, ora de forma positiva, ora de forma negativa. À medida que se inseriu atividades que visavam diminuir ou aumentar estes índices de reprovação e aprovação respectivamente, essas turmas responderam de forma inesperada, conforme as características únicas das turmas.

Verifica-se, em 2015/2, uma queda brusca do número de aprovações e um aumento drástico do número de reprovações por nota, gerando uma situação inesperada com a aplicação da ferramenta computacional *Scratch*. Conforme apresentado posteriormente, acredita-se que estas variações ocorreram por vários motivos: não conhecimento da metodologia construtivista pelo professor, perfil dos alunos da turma divergindo do perfil inicial do curso, grande quantidade de alunos repetentes que estavam acostumados a usar a metodologia tradicional, baixo tempo de dedicação à disciplina entre outros.

4.3.2 ANÁLISE CRÍTICA – Questionário dos alunos

O questionário do aluno foi dividido em três partes, a fim de buscar informações específicas acerca do aluno, da disciplina de ALP e da utilização da ferramenta *Scratch* como apoio ao ensino de ALP. Apesar do foco da dissertação ser a utilização da ferramenta *Scratch*, Lima (2008) afirma que são muitos os fatores que influenciam no processo de ensino-aprendizagem (carências afetivas; deficientes condições habitacionais, sanitárias, de higiene e de nutrição entre outras citadas anteriormente). Assim, foram definidas, dentre as situações apresentadas por Lima (2008), algumas situações importantes, conforme o perfil da instituição e informações coletadas do coordenador de curso, que poderiam influenciar na utilização da ferramenta *Scratch* como ferramenta de apoio ao ensino-aprendizado dos alunos, melhorando consideravelmente a análise qualitativa dos resultados obtidos (citamos tempo de conclusão do ensino médio, se possui outra graduação, se possui computador em casa, se trabalha, entre outras inseridas no questionário do aluno). As informações coletas do questionário do aluno estão resumidas no **Quadro 12**.

Quadro 12: Resumo percentual das informações coletadas do questionário aluno	
Dados básicos do Aluno	<ul style="list-style-type: none"> - 54% dos alunos trabalham; - 85% dos alunos tem computador em casa; - 69% dos alunos formaram a menos de 1 ano, 23% dos alunos formaram de 1 a

	<p>3 anos e 8% dos alunos formaram a mais de 5 anos;</p> <ul style="list-style-type: none"> - 100% dos alunos realiza o primeiro curso de nível superior.
Dados básicos da Disciplina de ALP	<ul style="list-style-type: none"> - 100% dos alunos são novatos na disciplina de ALP no IFTM, mas 54% dos alunos já estudaram programação em outras instituições; - 54% dos alunos dedicaram-se menos de 1 hora ao estudo de ALP, 23% de 1 a 2 horas, 8% de 2 a 3 horas e 8% mais de 3 horas; - 61% dos alunos tentaram criar outros programas além dos solicitados pelo professor durante a disciplina de ALP.
Dados básicos da utilização da ferramenta computacional <i>Scratch</i>	<ul style="list-style-type: none"> - 85% dos alunos não conheciam a ferramenta <i>Scratch</i>; - 85% dos alunos gostaram e acharam fácil a utilização do <i>Scratch</i>; - 100% dos alunos gostaram do desafio de criação de jogos usando o <i>Scratch</i>; - 77% dos alunos dedicaram-se a utilização da ferramenta <i>Scratch</i> menos de 1 hora semanal fora de sala de aula; - 62% utilizaram o tutorial do <i>Scratch</i> para esclarecer dúvidas sobre o programa; - 69% dos alunos tentaram fazer atividades no <i>Scratch</i> além das atividades propostas pelo professor em sala.
Fonte: Criado pelo próprio autor, a partir dos questionários dos alunos.	

É importante comentar que apesar de 43% dos alunos da turma terem respondido o questionário de forma anônima, sendo a pesquisa realizada com os alunos presentes após a última avaliação, pode-se prever que grande parte dos alunos reprovados por frequência não participaram do questionário, visto que, na última avaliação estes já estavam reprovados. Conforme **Quadro 7**, a disciplina de ALP, em 2015/2, teve 33% dos alunos reprovados por frequência, isto posto, acredita-se que grande parte desses alunos abandonou os estudos antes da aplicação do questionário.

Conforme informações obtidas, a partir do **Quadro 11**, destaca-se que, apesar da instituição estudada ser uma instituição pública, 85% dos alunos possui computador em casa, tendo a possibilidade de realizar as atividades de programação repassadas pelo professor, em sua própria casa. De acordo com informações obtidas do coordenador, a instituição disponibiliza laboratório de informática para os alunos que não possuem acesso à internet fora do horário de aula.

Conforme apresentado também pelo coordenador, o perfil das turmas mudou desde sua concepção, pois as primeiras turmas do curso de ADS possuíam alunos que, em sua maioria, haviam terminado o Ensino Médio a mais de cinco anos e trabalhavam em áreas afins do curso. Conforme perfil apresentado no questionário, à maioria dos alunos é recém-formada.

Observa-se que 54% dos alunos trabalham, fazendo com que o tempo de dedicação aos estudos, fora de sala de aula, fique limitado. Por meio da questão 6 do questionário, verifica-se que o tempo de estudo dos alunos é limitado, pois 54% dos alunos responderam que se dedicam, semanalmente, 1 hora ou menos à disciplina. Também, durante a utilização do *Scratch*, 77% dos alunos informaram que se dedicaram menos de 1 hora à utilização da ferramenta. É importante ressaltar que o professor, conforme consta em seu formulário, sabia da situação da turma e informou aos alunos que o tempo de dedicação ideal à disciplina é de mais de 3 horas.

Apesar do índice de alunos aprovados ter diminuído, observa-se que os alunos se sentiram extremamente motivados a utilizarem a ferramenta computacional *Scratch*. Conforme relatado por um aluno do curso:

Eu achei muito interessante, inclusive ainda utilizo, ajudou a prática da lógica e a ajuda na hora de resolver outros problemas de programação.

A ferramenta se mostrou um excelente meio para ensinar aos alunos a prática de criação de algoritmos e no desenvolvimento da lógica de programação. Jesus Medeiros *et al* (apud SÁ, TEIXEIRA E FERNANDES, 2007) mostram que o uso de jogos digitais nas atividades de ensino possibilita oferecer ao aprendiz momentos lúdicos e interativos como etapas do processo de aprendizagem. A possibilidade de utilização do *Scratch*, para a criação de jogos, estimulou os alunos na realização de atividades que exigiam muito raciocínio lógico e manipulação/criação de algoritmos, diferentemente do modo tradicional de ensino que muitas vezes desanima os alunos.

É importante destacar que 85% dos alunos gostaram da utilização da ferramenta computacional *Scratch* e acharam-na fácil de usar. Ao ensinar Algoritmos e Lógica de programação aos alunos envolvendo ao mesmo tempo sintaxe, semântica e lógica de programação é imposto aos alunos da disciplina de ALP uma carga de ensino muito grande, em um só momento. A ferramenta computacional *Scratch* possibilita trabalhar estes itens separadamente. Primeiro trabalhando a lógica de programação para posteriormente ensinar o aluno uma linguagem de programação mais robusta.

Conforme relatado no capítulo anterior, notou-se que no início da disciplina, quando foram iniciados os trabalhos com a ferramenta computacional *Scratch* os alunos estavam ansiosos e nervosos com a disciplina de ALP. Dois cenários foram criados entre alunos novatos e veteranos que foram sanados esclarecendo o objetivo da disciplina e da utilização da ferramenta *Scratch*. Muitas vezes, não fica claro para os alunos o objetivo da disciplina, o que pode gerar nos alunos dúvidas sobre o processo de ensino realizado pelo professor.

4.3.3 ANÁLISE CRÍTICA – Questionário do professor

Da mesma forma que se dividiu o questionário do aluno em três partes visando buscar dados relevantes ao aprendizado do aluno, também se realizou a divisão do questionário do professor em três partes consideradas importantes no processo de ensino. Estas são descritas a seguir.

A **primeira parte** do questionário possui informações referentes à formação do professor. No caso estudado, por meio da análise do questionário e conversa direta com o professor na fase de preparação do plano de aula, observou-se que, assim como na maioria das instituições de ensino, os docentes que ministram ALP, devido à importância da disciplina na conjuntura global do curso, finalizaram seus estudos a mais de cinco anos possuindo uma considerável experiência docente. Hoje, no Brasil, de acordo com Moreira *et al* (apud PIMENTA E ANASTASIOU, 2008),

Na maioria das instituições de ensino superior, incluindo as universidades, embora seus professores possuam experiência significativa e mesmo anos de estudo em suas áreas específicas, predomina o despreparo e até um desconhecimento científico do que seja o processo de ensino e aprendizagem, pelo qual passam a ser responsáveis a partir do instante em que ingressam na sala de aula.

Assim, a maioria dos professores que ministram aulas nos cursos superiores ou técnicos possui bacharelado, ou seja, possuem formação específica para atuar como profissionais na área específica de sua formação. Assim, a prática docente é baseada na experiência profissional individual ou na forma como ele recebeu os conhecimentos durante sua graduação. Muitos docentes, também possuem Mestrado e/ou Doutorado, visando aperfeiçoar a prática docente, mas conforme afirma Silva e Alencar (2013),

De acordo com a Lei de Diretrizes e Bases da Educação (BRASIL, 1996), o magistério na educação superior deve ser exercido, prioritariamente, por pessoas com títulos de mestre ou doutor. No entanto, os cursos de pós-graduação *stricto sensu* usualmente oferecem mais disciplinas que visam à formação do pesquisador, com número muito limitado de matérias voltadas para a docência. E isso acontece a despeito de o Plano Nacional de Pós-Graduação incluir, entre os objetivos da pós-graduação, a capacitação de docentes para as instituições de educação superior (BRASIL, 2004).

No caso estudado, o docente também possui formação específica para a área profissional e mestrado, porém a situação reportada por Silva e Alencar (2013) é confirmada pela questão doze. As metodologias de ensino construtivistas são uma forma de ensino adotada por muitos educadores na atualidade, pois modificam a forma de ensino-aprendizado dos alunos, saindo da passividade para a proatividade, construindo seu próprio conhecimento. Em cursos de Tecnologia, foco deste estudo, é muito importante a experiência profissional para a formação dos alunos, mas esta deve ser sustentada por técnicas de ensino que auxiliem o processo de ensino-aprendizado.

A **segunda parte** do questionário trata sobre informações relevantes à disciplina de ALP. Dessa forma, o professor, pela sua experiência docente, acredita que a carga horária da disciplina é suficiente para alcançar os objetivos da disciplina. O professor conseguiu realizar um exame diagnóstico da turma, verificando que, assim como confirma os questionários dos alunos e professor, que a maioria dos alunos (54% dos alunos no questionário dos alunos) se dedica menos de 1 hora semanal para o estudo dos conteúdos de ALP. Ele acredita que este tempo não é suficiente para o bom aprendizado durante a disciplina (para o professor são necessários mais de 3 horas semanais). Também identificou, assim como Júnior *et al* (2005), que, apesar da motivação em realizar a disciplina, a turma possuía os problemas mais citados na literatura para aprendizado de linguagem de programação: dificuldade de interpretação do problema, dificuldade de identificar os pré-requisitos necessários para o desenvolvimento das competências de construção de algoritmos e programação, dificuldade em aplicar suas habilidades prévias, criando fonte de medo e frustração e dificuldade com raciocínio lógico necessário à criação dos algoritmos.

A **terceira parte** do questionário se refere à aplicação do *Scratch* na disciplina. É importante ter em mente que as informações levantadas na primeira parte e segunda parte do questionário afetam diretamente a utilização da ferramenta computacional construtivista *Scratch*. Conforme perfil do professor, desconhecedor da ferramenta *Scratch*, mas susceptível à sua utilização em sala de aula, demonstra a possibilidade de recepção às mudanças de estratégias educacionais possibilitando aos alunos a possibilidade de apresentação na disciplina de uma quantidade maior de estratégias de ensino-aprendizagem. Conforme relatado pelo professor e alunos, eles possuem receio e até medo em realizar a disciplina, devido à dificuldade relatada principalmente pelos alunos veteranos. Tanto alunos quanto professores gostaram e se sentiram motivados em utilizar a ferramenta como apoio aos seus estudos. Segundo o professor, pelo seu relato,

Achei muito interessante à utilização do Scratch e demais técnicas no início da disciplina, pois os alunos já tiveram um bom contato inicial com os conceitos de lógica de programação de uma maneira simples e divertida, menos pesada que a linguagem de programação final da disciplina.

É importante criar diversas estratégias para o aprendizado do aluno, para motivá-lo de forma simples e divertida, fazendo com que os alunos, conforme apresentado por Júnior *et al* (2005), tenham menos problemas durante a realização da disciplina de ALP.

4.3.4 ANÁLISE CRÍTICA – Questionário do coordenador

Como verificado nos questionários anteriores, verifica-se que o coordenador possui bastante conhecimento na área, pois possui titulação de Doutor e é coordenador do curso a mais de 1 ano. Apesar de possuir o título de Bacharelado, conhece as metodologias de ensino construtivistas, inclusive a ferramenta *Scratch*.

Considera que a carga horária da disciplina é suficiente para a ministração dos conteúdos exigidos para a disciplina. E que são realizadas diversas estratégias para que o aluno consiga adquirir as habilidades e competências solicitadas pela disciplina de ALP.

Porém, o coordenador, por meio de seu questionário, informa que esta turma possui um perfil diferente das turmas anteriores, declarando que:

Segundo relato de professores que ministraram aulas para esta turma, os alunos que ingressaram neste semestre apresentaram dificuldades acima da média dos semestres anteriores.

Conforme o coordenador, um dos fatores que influenciaram nesta diferença de perfil é o processo seletivo escolhido pela instituição. Nesta instituição, o processo seletivo é por meio do SISU (Sistema de Seleção Unificada). É comum, nesta instituição, o ingresso de alunos advindos de chamadas posteriores à 5ª chamada do SISU (fato que ocorreu no primeiro semestre de 2015). Quanto mais distante da primeira chamada, maior é a dificuldade apresentada pelo aluno, no desenvolvimento do raciocínio lógico. Também é importante observar que o SISU utiliza a mesma nota (nota do ano anterior) para selecionar os alunos que ingressarão nas turmas do segundo semestre das instituições que o utilizam como processo seletivo. Logo, os alunos com as maiores notas foram aproveitados pelas instituições no primeiro semestre, ficando para o segundo semestre os alunos com maiores deficiências no raciocínio lógico. Em 2015/2, o último ingresso neste curso foi pela 7ª chamada do SISU. Dessa forma, esta turma possui uma média maior de alunos com baixo índice de desenvolvimento cognitivo necessário para resolução de problemas que necessitam do

raciocínio lógico, comparado às turmas anteriores, além de possuir alunos que necessitavam de acompanhamento do NAPNE (Núcleo de Assistência às Pessoas com Necessidades Específicas) da instituição. Apesar da motivação da turma em realizar a disciplina, percebeu-se que o perfil da turma afeta diretamente na estratégia de ensino escolhida para o ensino-aprendizado da turma.

Apesar da dificuldade da turma, os alunos reportaram ao coordenador que:

A aplicação da ferramenta de apoio na disciplina foi positiva e auxiliou os alunos na compreensão de como funciona a lógica de programação. Acredito que o emprego desta categoria de ferramentas deverá facilitar a sedimentação do conhecimento abordado na disciplina, conhecimento este que demonstra um novo paradigma para solução de problemas a alunos que nunca tiveram contato com lógica de programação.

CONCLUSÕES

Após a realização de todo o trabalho, desde as pesquisas realizadas sobre o ensino de ALP, trabalhos correlatos até a análise crítica das informações contidas nos questionários aplicados aos alunos, professor e coordenador, pode-se notar que a disciplina de ALP realmente merece atenção, devido as suas particularidades e características únicas apresentadas. A disciplina exige do aluno o desenvolvimento de algoritmo dentro de uma linguagem de programação, além disso, necessita que o aluno tenha desenvolvido sua cognição para interpretação/resolução de problemas relacionados ao raciocínio lógico. Assim, a disciplina imprime ao aluno um nível de exigência considerado alto, além de grande tempo de dedicação aos estudos. Conjuntamente com as dificuldades apresentadas pelos alunos, existe uma diversidade de metodologias e estratégias de ensino possíveis de serem aplicados à disciplina.

Verifica-se que diversas estratégias são propostas, visando melhor os índices de alunos aprovados, porém, apesar dos esforços apresentados nesta dissertação, situações inesperadas acontecem, conforme o estudo de caso proposto para esta dissertação. A turma em estudo apresentou um perfil único dentro do histórico de ministração de aulas para a disciplina na instituição. Apesar da utilização do *Scratch* apresentar bons resultados na motivação dos alunos e no rendimento dos alunos durante o aprendizado de ALP, demonstrado no capítulo 3 - trabalhos correlatos, sua aplicação na instituição de ensino superior, foco deste trabalho, não apresentou resultados positivos durante a análise quantitativa, considerando o parâmetro índice de aprovação dos alunos comparado ao histórico apresentado no capítulo 2.

Apesar da utilização do *Scratch* como uma ferramenta de apoio ao ensino ser considerado por muitos autores uma excelente mudança no paradigma de ensino-aprendizagem, gerando ótimos benefícios aos alunos, na turma estudada, houve a surpresa que, ao final de sua utilização, o número de alunos aprovados diminuiu.

Durante a aplicação da proposta de ensino utilizando o *Scratch*, foram levantadas as seguintes situações:

- **Choque da metodologia Tradicional com a metodologia Construcionista:** Quando alunos e professores estão acostumados a um paradigma de ensino/aprendizagem e faz-se uma alteração deste paradigma, produz-se uma inquietação de alunos e professores. Toda mudança traz um desconforto inicial.
- **O perfil único da turma:** cada turma possui um perfil complexo, único e depende, diretamente, do processo seletivo de entrada dos alunos. No caso da instituição estudada é o SISU (Sistema de Seleção Unificada, havendo a necessidade de um levantamento diagnóstico da mesma para definir a melhor metodologia de ensino a

ser aplicada. O coordenador de curso afirma em seu questionário, que o perfil da turma que utilizou o *Scratch* era diferente das turmas anteriores.

- **Motivação na utilização da ferramenta *Scratch*:** Apesar do pouco tempo em que se trabalhou diretamente com a ferramenta *Scratch*, verifica-se que a maioria dos alunos reportou em seus questionários somente problemas com o desconhecimento de alguns comandos da ferramenta *Scratch*. Durante sua utilização, os alunos não reportaram grandes problemas com a construção dos algoritmos necessários à resolução dos problemas apresentados. Foram trabalhados códigos grandes e complexos no *Scratch*, principalmente durante a fase de desafio na quarta semana. Poucos alunos tiveram dificuldades com a criação do jogo e com a montagem de algoritmos.
- **Dificuldades com o retorno à disciplina:** Por meio dos questionários dos alunos, eles expressaram que, durante a realização da disciplina, tiveram problemas com a formação de algoritmos, interpretação e desenvolvimento do raciocínio lógico para a criação dos mesmos, durante a utilização da linguagem estruturada, apresentada na disciplina posterior à utilização da ferramenta *Scratch*.

Apesar de não ter aumentado o índice de aprovação da turma em estudo, comparada com os índices das turmas anteriores, verifica-se que a ferramenta *Scratch* trouxe benefícios aos alunos como motivação, facilidade de criação de algoritmos dentro da ferramenta computacional *Scratch* e facilidade de interpretação dos problemas dentro do *Scratch*. Acredita-se que isto se deve ao fato de que no *Scratch* não existe preocupação com a sintaxe e semântica, no momento de criação dos algoritmos, deixando o aluno focado somente no raciocínio lógico para a criação dos problemas propostos. Notou-se, também, uma boa interação entre os colegas e motivação do professor no trabalho do raciocínio lógico, com os alunos.

REFERÊNCIAS

ALTOÉ, Anair; PENATI, Marisa Morales. O Construtivismo e o Construcionismo Fundamentando a Ação docente. **Revista Educação e Novas Tecnologias**. Maringá, 2005. p. 55-57. Disponível em: <<http://goo.gl/tiBPKT>>. Acesso: 20 maio 2015.

AMARAL, Haroldo José Costa do; FRANÇA, Rozelma Soares de. **Proposta Metodológica de Ensino e Avaliação para o Desenvolvimento do Pensamento Computacional com o Uso do Scratch**. II Congresso Brasileiro de Informática na Educação – CBIE, XIX Workshop de Informática na Escola – WIE. 2013. Disponível em: <<http://goo.gl/3jiMXN>>. Acesso: 8 abr. 2015.

ARGENTO, Heloisa. **Teoria Construtivista**. 2010. Disponível em: <<http://goo.gl/GtbUBD>> Acesso: 20 maio 2015.

AURELIANO, Viviane Cristina Oliveira; TEDESCO, Patrícia Cabral de Azevedo Restelli. **Utilizando o Scratch para Apoiar o Processo de Ensino-aprendizagem de Programação para Iniciantes na EAD**. Laboratório de Banco de Dados, UFMG. 2013. Disponível em: <<http://goo.gl/tFKDTp>> Acesso: 5 abr 2015.

AVIZ JUNIOR, Ademar Alves de. **A aprendizagem de algoritmos: uma experiência no curso de Tecnologia em Informática do CEFET- PA**. Belém. Dissertações em Educação, em Ciências e Matemáticas (Mestrado) - PPGECM/IEMCI. 2007. Disponível em: <<http://goo.gl/DQGBwa>> Acesso: 21 maio 2015.

BUFFONI, Saete. **Apostila de Algoritmo Estruturado**. FIAA-Faculdades Integradas Anglo-Americano. Curso de Sistemas de Informação. 4ª Edição. 2003. Disponível em: <<http://goo.gl/OE7HfG>> Acesso: 5 abr 2015.

CAMPOS, Ricardo Luiz B. L. **ERMC2: Uma proposta de metodologia para melhoria do ensino-aprendizado de lógica de programação**. XXX Congresso da Sociedade Brasileira de Computação. Belo Horizonte/MG, Brasil. SBC, 2010. Disponível em: <<http://goo.gl/zQRoy8>> Acesso: 5 abr 2015.

CAMPOS, Marcio; RAPKIEWICZ, Cleli Elena. **Uma abordagem pedagógica para iniciação ao estudo de algoritmos**. Faculdades Integradas de Bennett. Rio de Janeiro, 2004. Disponível em: <<http://goo.gl/plS2Kr>>. Acesso: 25 fev 2016.

CARNEIRO, Douglas Almeida et al. **Utilização do Scratch como ferramenta de auxílio à aprendizagem de programação**. COBENGE – XLI Congresso Brasileiro de Educação em Engenharia. Gramado/RS, 2013. Disponível em: <https://www.researchgate.net/publication/260834707_Utilizacao_do_Scratch_como_Ferramenta_de_Auxilio_a_Aprendizagem_de_Programacao> Acesso: 5 abr 2015.

CORMEN, Thomas H et al. **Algoritmos**. Teoria e Prática. 3. Edição. Tradução da 3. Edição Americana, 2012. Elsevier, Campus.

FALCKEMBACH, Gilse A. Morgental; ARAUJO, Fabrício Viero de. Aprendizagem de Algoritmos: dificuldades na resolução de problemas. **Anais SULCOMP**, Vol. 2, n. 2, 2006. Disponível em: <<http://goo.gl/WTYC2z>> Acesso: 1 maio 2015.

FERRANDIN, Mauri; STEPHANI, Simone Lillian. Ferramenta para o ensino de Programação via Internet. **Periódicos UNESC**. 2005. Anais do SULCOMP. Congresso Sul Brasileiro de Computação. Vol. 1. N. 1. Disponível em: <<http://goo.gl/W0mLKL>> Acesso: 1 maio 2015.

FERRARI, Márcio. B. F. Skinner. O cientista do comportamento e do aprendizado. **Revista Nova Escola**. Pesquisadores da Educação. 2008. Disponível em: <<http://goo.gl/IHE4c6>> Acesso: 22 maio 2015.

FERREIRA, Aurélio Buarque de Holanda. **Dicionário Aurélio da Língua Portuguesa**. Disponível em: <<http://goo.gl/KT58WO>>. Acesso: 20 dez 2014.

FONSECA FILHO, Clézio. **História da computação**. O caminho do pensamento e da tecnologia. Editora ediPUCRS. Porto Alegre, 2007. Disponível em: <<http://goo.gl/TNUAUu>>. Acesso: 20 dez 2014.

FORTE, Cleberson E.; KIRNER, Claudio. **Usando Realidade Aumentada no Desenvolvimento de Ferramenta para Aprendizagem de Física e Matemática**. 6. Workshop de realidade Virtual e Aumentada. UNISANTA, 2009. Disponível em: <<http://goo.gl/7CDAZy>> Acesso: 22 maio 2015.

FREDERICO, Fernando. **Explorando o Scratch**. Guia prático de utilização do Scratch. 2009. Disponível em: <<http://goo.gl/HvLV8b>> Acesso: 20 maio 2015.

FREIRE, Paulo. **Carta de Paulo Freire aos professores**. Estud. Av. São Paulo, v. 15, n. 42, 2001. Disponível em: <<http://goo.gl/RWO6Nq>>. Acesso: 5 maio 2014.

FREIRE, Paulo. **Pedagogia do Oprimido**. 43. ed. Rio de Janeiro: Paz e Terra, 2005.

GALHARDO, Mariane Fogaça. **Metodologia e Ferramenta para Ensino da Programação Orientada a Objetos**. COBENGE, 2004. Congresso Brasileiro de ensino de Engenharia. Brasília. Disponível em: <<http://goo.gl/UOr4oh>> Acesso: 5 maio 2014.

GARCIA, Simone Carboni. **Objetos de Aprendizagem: Investindo na Mediação Digital do Conhecimento**. Círculo de Encontros Linguísticos do Sul. Encontro de Pelotas. Rio grande do Sul. Pelotas, 2006. Disponível em <<http://goo.gl/8gGrCr>> Acesso: 5 maio 2014.

GIL, Antônio Carlos. **Como elaborar projetos de pesquisa**. 5. ed. São Paulo: Atlas, 2010.

GIRAFFA, Lucia M. M.; MARCZAK, Sabrina S; ALMEIDA, Gláucio. **O ensino de algoritmos e programação mediado por um ambiente na Web**. Laboratório de Banco de dados. UFMG, 2003. Disponível em: <<http://goo.gl/Tw8I1R>> Acesso: 5 maio 2014.

GOODRICH, Michael T.; TAMASSIA Roberto. **Projeto de Algoritmos**. Fundamentos, análise e exemplos da Internet. 1. Ed. Bookman, 2004.

GREGOLIN, Vanderlei Rodrigues. Linguagem Logo: Explorando Conceitos Matemáticos. **Revista Tecnologias na Educação**. 2008. Disponível em: <<http://goo.gl/qfL3q>>. Acesso: 5 maio 2015.

JESUS MEDEIROS, Tainá; SILVA, Thiago Reis da; ARANHA, Eduardo Henrique da Silva. Ensino de Programação utilizando Jogos Digitais: uma revisão sistemática da literatura. **Revista Novas Tecnologias na Educação**. vol.11. n. 3. CINTED-UFRGS, 2013. Disponível em: <<http://goo.gl/9qdTac>> Acesso: 28 fev 2016.

KAMIYA, Reginaldo Rideaki; BRANDÃO, Leônidas de Oliveira. **iVProg** - um sistema para introdução à Programação através de um modelo visual na Internet. Simpósio Brasileiro de Informática na Educação, 2009. Disponível em: <https://goo.gl/jkPyaa>. Acesso: 5 maio 2014.

KOHN, Karen; Moraes, CLÁUDIA Herte de. **O impacto das novas tecnologias na sociedade**: conceitos e características da Sociedade da Informação e da Sociedade Digital. Intercom. Sociedade Brasileira de Estudos Interdisciplinares da Comunicação. XXX Congresso Brasileiro de Ciências da Comunicação. Santos/SP, 2007. Disponível em: <<http://goo.gl/pzZQSD>> Acesso: 5 maio 2014.

LEAO, Denise Maria Maciel. Paradigmas contemporâneos de Educação: Escola Tradicional e escola Construtivista. **Cadernos de Pesquisa**. n. 107, 1999. Disponível em: <<http://goo.gl/UD0fwu>>. Acesso: 25 fev. 2016.

LEITE, Mário; JÚNIOR, Nelson Abu Sanra Rahal. Programação Orientada à Objetos: Uma abordagem didática. **Revista de Informação e Tecnologia INFOTEC**. Unicamp. Disponível em: <<http://goo.gl/OxpSs9>>. Acesso: 28 fev. 2016.

LIMA, Sandra Vaz de. **Fatores que interferem na Aprendizagem**. Publicada em 25 de fevereiro de 2008; Disponível em: <<http://goo.gl/0I9dq7>>. Acesso: 28 fev 2016.

MACIENTE, Aguinaldo Nogueira; ARAÚJO, Thiago Costa. A demanda por engenheiros e profissionais afins no mercado de trabalho formal. Ipea – Instituto de Pesquisa Econômica Aplicada. **Revista Radar** n. 12 ed. especial. 2011. Disponível em: <<http://goo.gl/8M705Z>>. Acesso: 5 maio 2014.

MANZANO, José Augusto N. G.; OLIVEIRA, Jayr Figueiredo de. **Algoritmos** - Lógica para Desenvolvimento de Programação de Computadores - Programar independente da linguagem de programação. 25. ed. revisada. Editora Érica, 2011.

MARIN DA SILVA, Cláudia. **As Novas Tecnologias de Informação e Comunicação e a Emergência da Sociedade Informacional**. UFSM – Universidade Federal de Santa Maria. 2006. Disponível em: <<http://goo.gl/UfGwpK>>. Acesso: 20 dez 2014.

MARTIN, Daniele Fortes. **A aprendizagem em Paulo Freire e Piaget**. Faculdades de Ciências. Campus Bauru. Trabalho de Conclusão de Curso de Pedagogia da Faculdade de Ciências da UNESP, 2007. Disponível em: <<http://goo.gl/ubFwjU>>. Acesso: 4 Jul 2014.

MARTINS, Valéria Farinazzo; OLIVEIRA, Alisson José Gregório de; GUIMARÃES, Marcelo de Paiva. Implementação de um laboratório de realidade virtual de baixo custo: estudo de caso de montagem de um laboratório para o ensino de Matemática. **Revista Brasileira de computação Aplicada**, 2013. Disponível em: <<http://goo.gl/4vyjn8>>. Acesso: 5 maio 2014.

MEDEIROS, Marco; PEREIRA, Priscilla de Sousa; MENEZES, José Wally Mendonça. **Análise do Scratch como Ferramenta de Auxílio ao Ensino de Programação de Computadores**. COBENGE – XL Congresso Brasileiro de Educação em Engenharia. Belém/PA, 2012. Disponível em: <<http://goo.gl/9m3FOA>>. Acesso: 5 maio 2014.

MESQUITA, Silvana. Os resultados do Ideb no cotidiano escolar. **Revista Scielo**. vol. 20. n. 76. Rio de Janeiro, 2012. Disponível em: <<http://goo.gl/zBIeeJ>>. Acesso: 28 fev 2016.

MOCARZEL, Cleicy; VIEIRA, Marcelo. **Avaliação Educacional**: um estudo das perspectivas tradicional e construtivista na visão docente. 2016. Disponível em: <<http://goo.gl/gmG0qS>> Acesso: 28 fev 2016.

MONTOYA, Adrian Oscar Dongo. Resposta de Piaget a Vygotsky: convergências e divergências teóricas. **Revista Educação & Realidade**. N. 38. Porto Alegre, 2013. Disponível em: <<http://goo.gl/msCSEp>> Acesso: 20 maio 2015.

MORGADO DE SOUZA, Cláudio. VisuAlg. Ferramenta de apoio ao ensino de programação. **Revista TECCEN**. Vol. 2. n. 2. 2009.

NETO, Valter dos Santos Mendonça. **A utilização da Ferramenta Scratch como ensino na aprendizagem de lógica de programação**. II Congresso Brasileiro de Informática na Educação – CBIE. Unicamp, Campinas/SP. 2013. Disponível em: <<http://www.br-ie.org/pub/index.php/wcbie/article/view/2675/2329>>. Acesso: 5 maio 2014.

OLIVEIRA, J. F; MANZANO, J. A. N. G. **Estudo Dirigido de Algoritmos**. São Paulo: Érica, 2012.

OSAKA, Ana. Linhas Pedagógicas: veja como elas funcionam e qual tem a ver mais com seu filho. **Revista online UOL Educação**. São Paulo, 2009. Disponível em: <<http://goo.gl/E8JiDX>>. Acesso: 28 fev 2016.

PEREIRA, Priscilla de Sousa; MEDEIROS, Marcos; MENEZES, José Wally Mendonça. **Análise do Scratch como ferramenta de auxílio no ensino de programação de computadores**. COBENGE – XL Congresso Brasileiro de Educação em Engenharia. Belém/PA, 2012. Disponível em: <<http://www.abenge.org.br/CobengeAnteriores/2012/artigos/104281.pdf>> Acesso: 5 maio 2014.

PETRY, Patrícia Gerent. **Um Sistema para o Ensino e Aprendizagem de Algoritmos Utilizando um Companheiro de Aprendizagem Colaborativo**. Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação. Florianópolis/SC, 2005. Disponível em: <<https://goo.gl/TfxNn6>> Acesso: 5 maio 2014.

PIAGET, Jean. **Para Onde Vai a Educação?** 9. ed. Rio de Janeiro: José Olympio, 1988.

_____. BARBEL, Inhelder. **A Psicologia da criança**. Tradução de Octavio Mendes Cajado. Rio de Janeiro: Bertrand Brasil, 2003.

_____. Aprendizagem e Conhecimento. In: **Aprendizagem e conhecimento**. Rio de Janeiro: Freitas Bastos, 1974.

_____. A epistemologia Genética. In: **Os pensadores**. 2. ed. São Paulo: Abril Cultural, 1983.

PINEL, Hiram; Seymour Papert: O Construcionismo. Psicologia Educacional: **Alguns Textos Esparsos**. Belo Horizonte/MG. NUEX-PSI. 1. Ed, 2004.

RAMOS DOS SANTOS, Bruno Filipe. **Uso do Scratch no processo de ensino e aprendizagem das Tecnologias da Informação e comunicação no 8º ano de escolaridade**. Relatório de estágio do Mestrado, Ensino em Informática. Universidade do Minho, Instituto de Educação, 2014.

RAPKIEWICZ, Cleli Elena et al. Estratégias Pedagógicas no Ensino de Algoritmos e Lógica de Programação associadas ao uso de Jogos Educacionais. **Revista Novas Tecnologias da Educação**. vol. 4. n. 2. 2006.

RISSI, M. C.; MARCONDES, M. A. S. **Estudo sobre reprovação e retenção nos cursos de graduação**. 2009 - 2011. Londrina/PR. Disponível em: <<http://goo.gl/I4Ydzs>>. Acesso: 1 maio 2014.

ROBERTO LIMA, Márcio. **Construcionismo de Papert e Ensino-Aprendizagem de Programação de Computadores no Ensino Superior**. Universidade Federal de São João Del Rei. Departamento de Educação. 2009. Disponível em: <<http://goo.gl/aBnQ9o>> .Acesso: 20 dez 2014.

ROCHA, Paulo Santana et al. Ensino e Aprendizagem de Programação: análise da aplicação de proposta metodológica baseada no sistema personalizado de ensino. **Revista Novas Tecnologias na Educação**. CINTED-UFRGS, 2010. Disponível em: <<http://goo.gl/eiENVL>> Acesso: 2 maio 2015.

SALES, Gilvandenys Leite; et al. Atividades de modelagem exploratória aplicada ao ensino de física moderna com a utilização do objeto de aprendizagem pato quântico. **Revista Brasileira de ensino de Física**. vol. 30, n. 3. São Paulo, 2008. Disponível em: <<http://goo.gl/bD4pUh>>. Acesso: 19 maio 2015.

SANTIAGO, Rafael de; DAZZI, Rudimar Luís Scaranto. **Ferramenta de apoio ao ensino de algoritmos**. XXIV Seminário de Computação. Departamento de Ciência da Computação da FURB, 2004. Disponível em: <<http://goo.gl/viY64n>>. Acesso: 25 fev 2016.

SANTOS, Fábio, **Pré-Vestibular Popular da UFF na engenharia**. Disponível em:<<http://goo.gl/Cgac8q>>. Acesso: 20 dez 2014.

SANTOS, Rodrigo Pereira dos; COSTA, Heitor Augustus Xavier. **TBC-AED e TBC-AED/WEB: Um desafio no ensino de algoritmos, estruturas de dados e programação**. 2005. Disponível em: <<http://goo.gl/1u5gQj>>. Acesso: 5 maio 2014.

_____; Rodrigo Pereira dos; COSTA, Heitor Augustus Xavier. **Análise de Metodologias e Ambientes de Ensino para Algoritmos, Estrutura de dados e Programação aos iniciantes em Computação e Informática**. Universidade Federal de Lavras. Programa de Engenharia de sistemas de computação, 2006. Disponível em: <<http://goo.gl/q10AMK>>. Acesso: 25 fev 2016.

SCAICO, Pasqueline Dantas et al. Ensino de Programação no Ensino Médio: uma abordagem orientada ao design com a linguagem Scratch. **Revista Brasileira de informática na Educação**, vol. 21, n. 2. 2013.

SEBESTA, Robert W. **Conceitos de Linguagens de Programação**. 9. Ed. Bookmam, 2011.

SIMONETTO, E. O.; FILHO, W. P.; ABEGG, I. **Uma abordagem diferenciada no ensino de algoritmos através da utilização de uma lousa digital**. 2012. Disponível em: <<http://goo.gl/T6ik8R>>. Acesso:5 maio 2014.

SILVA, Dayse Souza da. ALENCAR, Eunice Maria Lima Soriano de. A docência da educação superior na área da saúde: formação, satisfação e práticas pedagógicas. **Revista Educação em Destaque**. v. 4. n. 1. 2013.

SOUZA, Jairo Francisco de. **Linguagens de Programação – conceitos e técnicas**. 1995. Disponível em: <<http://goo.gl/5aLN0A>>. Acesso: 20 maio 2015.

TEIXEIRA, Elisiane Alves. **Problemas de Aprendizagem e Psicopedagogia**. 2013. Disponível em <<http://goo.gl/OYizqE>> Acesso: 5 maio 2014

TERRA, Márcia Regina, **O desenvolvimento humano na teoria de Piaget**. Disponível em: <<http://goo.gl/Ld1f1v>>. Acesso: 5 maio 2014.

TODOROV, João Claudio; MOREIRA, Márcio Borges; MARTONE, Ricardo Corrêa. Sistema personalizado de ensino, educação à distância e aprendizagem centrada no aluno. **Revista Psicologia: Teoria e Prática**. v. 25, n. 3. 2009. Disponível em: <<http://goo.gl/nNw3Cs>>. Acesso: 7 maio 2015.

TORI, Romero. **Aprendendo Lógica de Programação via WEB**. 2002. Disponível em: <<http://goo.gl/76Wqup>> .Acesso: 7 maio 2015.

VALENTE, José Armando. Informática na educação: Instrucionismo x Construcionismo. **Revista Educação Pública**. Seção Tecnologia. 2011. Disponível em: <<http://goo.gl/Jn3R93>>. Acesso: 7 maio 2015.

VESCE, Gabriela e. Possolli. **Metafísica**. Disponível em: <<http://goo.gl/gM0jiI>>. Acesso: 20 dez 2014.

VILARIM, Gilvan. 2004. **Algoritmos Programação para Iniciantes**. Rio de Janeiro: Ciência Moderna, 2004.

VINICIOS, Gabriel. **Método para construção de Algoritmo**. 2010. Disponível em: <<http://goo.gl/5CyxPN>> Acesso: 20 dez 2014.

XEXEO, José Antonio Moreira. **Ensino de Algoritmo e Programação**: uma experiência no Ensino Médio. XXV Congresso da Sociedade Brasileira de Computação. XIII WEI. Unisinos – São Leopoldo/RS, 2005.

ZAINA, Luciana Aparecida Martinez; RUGGIERO, Wilson Vicente; BRESSAN, Graça. Metodologia para acompanhamento da aprendizagem através da *Web*. **Revista Brasileira de Informática na Educação**. v. 12. n. 1, 2004. Disponível em:< <http://goo.gl/FWZ4g1>>. Acesso: 2 maio 2015.

ZIVIANI, Nivio. **Projeto de Algoritmos com implementações em Pascal e C**. 3. ed. Revista e ampliada. Cengage – Learning, 2011.

1. Identificação

Curso:	TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - INGRESSO A		
Período letivo:	2º SEM/2015	Turno:	NOTURNO
Unidade curricular:	ALGORITMO E LÓGICA DE PROGRAMAÇÃO		
Unidade:	CAMPUS PATROCÍNIO		
Tipo:	TEÓRICA/PRÁTIC		
Professor(es)	<i>(PRINCIPAL)</i>		
Carga horária:	108.00		
Duração aula:	45 minutos		

2. Ementa

Abordar os conceitos básicos de algoritmos, linguagem algorítmica, operadores, estruturas e modularização de programas. Estes conceitos serão implementados através de uma linguagem de programação estruturada.

3. Objetivos

- Analisar e distinguir, por meio de máximas de programação e da metodologia para desenvolvimento de algoritmos, os processos lógicos necessários para a resolução de problemas escritos em algoritmo (pseudo código) baseado em Português Estruturado (Portugol);
- Distinguir as estruturas dos comandos e suas sintaxes de forma a resolver problemas lógicos com aplicações práticas;
 - Avaliar processos lógicos e criticar suas aplicações, culminando na verbalização de sugestões de melhoria quando cabíveis;
 - Aplicar os algoritmos em pseudo linguagem;
 - Utilizar lógica matemática para expressar raciocínio e construir algoritmos de maneira formal.
 - Desenvolver programas em linguagem estruturada. Utilizar linguagens de programação de algoritmos.
 - Conhecer ambientes de programação.
 - Aplicar a lógica de programação em programas computacionais.

4. Conteúdo Programático

- Lógica no cotidiano
Conceituação de algoritmos
Formas de Representação de Algoritmos
Pseudocódigo
- Tipos Primitivos
 - Expressões aritméticas / relacionais / lógicas
 - Operação de atribuição
 - Estruturas condicionais
 - Estruturas de repetição
- Linguagem C
- Bibliotecas
 - Variáveis e constantes
 - Instruções de entrada e saída
 - Estruturas condicionais e de repetição
 - Matrizes e vetores
 - Funções

PLANO DE ENSINO

5. Metodologia

Aulas expositivas dialogadas
Aulas práticas em laboratório
Exercícios teóricos e práticos

6. Recursos Didáticos

Projektor Multimídia
Quadro branco
Softwares de desenvolvimento

7. Avaliação (critérios, valores, procedimentos, recuperação)

Provas - 70 pontos (3 provas)
Exercícios - 15 Pontos (diversas listas e atividades)
Trabalhos - 15 pontos (seminários ou projetos práticos)

A recuperação dos conteúdos se dará ao longo do semestre através do atendimento individual agendado e programas de monitoria quando disponíveis

A recuperação dos pontos será dada ao final do semestre através de uma prova abordando todos os conteúdos da disciplina, com um valor total de 100 pontos

8. Referências

Básica:

OLIVEIRA, U. Programando Em C: fundamentos: v.1. Rio de Janeiro: Ciência Moderna, [2000].
SEBESTA, R.W. Conceitos de Linguagens de Programação. 9. ed. São Paulo: Bookman, 2011.
THOMAS, H. Cormem; CHARLES, E. Leiserson; RONALD, L. Rivest. Algoritmos: teoria e Prática. 2. ed. São Paulo: Campus, 2002.

Complementar:

BARRY, P; GRIFFITHS, D. Use a cabeça: programação. Rio de Janeiro: Altabooks, [2000].
JON KLEINBERG, Éva Tardos. Algorithm design. USA: Addison-wesley, 2006. ISBN 0321295358.
LUIS, Joyanes Aguilar. Fundamentos de Programação Algoritmos: estruturas de dados e objetos. 3. ed. São Paulo: [s.n.], 2008.
MIZRAHI, V.V. Treinamento em linguagem C. São Paulo: Prentice Hall, [2000].
OLIVEIRA, U. Programando em C: a biblioteca padrão de c: v. 2. Rio de Janeiro: Ciência Moderna, [2000].
SCHILD, H. CC : completo e total. São Paulo: Makron Books, [2000].

Cronograma das aulas				
SEMANA	Nº AULAS PREVISTAS	DESCRIÇÃO	C. H. TEÓRICA	C.H. PRÁTICA
1	8	Recepção aos alunos novatos Atividades integradoras	06:00	
2	8	Apresentação do plano de ensino da disciplina; Conceitos de lógica; Conceitos de algoritmos e suas formas de representação	06:00	

PLANO DE ENSINO

SEMANA	Nº AULAS PREVISTAS	DESCRIÇÃO	C. H. TEÓRICA	C.H. PRÁTICA
3	8	conceitos de algoritmos e lógica de programação	06:00	
4	8	tipos primitivos expressões aritméticas , relacionais e lógicas variáveis e constantes	06:00	
5	8	operação de atribuição operações de entrada e saída algoritmos simples	06:00	
6	8	resolução de exercícios de algoritmos introdução à linguagem C	06:00	
7	6	variáveis, constantes, instruções de entrada e saída em linguagem C, revisão para prova	04:30	
8	8	prova 1 vista de prova 1	06:00	
9	8	estruturas condicionais em pseudocódigo; exercícios com estruturas condicionais	06:00	
10	8	estruturas condicionais em linguagem C exercícios com estruturas condicionais em C	06:00	
11	8	finalização estruturas condicionais; revisão para prova 2	03:00	03:00
12	8	prova 2; vista de prova 2; correção prova 2; elucidação de dúvidas	06:00	
13	8	Semana Nacional da Ciência e Tecnologia		06:00
14	8	estruturas de repetição em pseudocódigo	06:00	
15	8	estruturas de repetição em linguagem C	03:00	03:00
16	8	finalização estruturas de repetição; vetores e matrizes. introdução	03:00	03:00
17	8	vetores e matrizes em linguagem C; aplicações de vetores e matrizes; exercícios com vetores e matrizes	03:00	03:00
18	8	funções, introdução, sintaxe, exemplos, funções em C++	06:00	
19	8	aplicação de funções em linguagem C desenvolvimento de trabalho em duplas revisão para prova 3	03:00	03:00
20	8	prova 3 vista de prova 3 fechamento de notas	03:00	03:00

PLANO DE ENSINO

SEMANA	Nº AULAS PREVISTAS	DESCRIÇÃO	C. H. TEÓRICA	C.H. PRÁTICA
21	8	revisão de conteúdos, finalização semestre	06:00	
21	166	Totais em horas	100,50	24,00



**MESTRADO PROFISSIONAL EM EDUCAÇÃO
TECNOLÓGICA**
CONVITE PARA REALIZAÇÃO DE QUESTIONÁRIO SOBRE O ENSINO NA
DISCIPLINA DE ALP E A UTILIZAÇÃO DE FERRAMENTAS CONSTRUTIVISTAS –
SCRATCH

Olá,

Meu nome é Flamarion Assis Jeronimo Inácio, sou aluno do Mestrado Profissional em Educação Tecnológica do IFTM – Câmpus Uberaba, sob a orientação do Prof. Dr. Hugo Leonardo Pereira Rufino. Convidamos você a responder algumas questões sobre as Ferramentas Construtivistas no Apoio ao Ensino da Disciplina de Algoritmo e Lógica de Programação – ALP, mais precisamente sobre a Ferramenta Scratch utilizada no início da disciplina de ALP.

As respostas de cada participante serão tratadas de forma confidencial. Portanto, marque a resposta que mais expresse a sua opinião.

Agradeço sua colaboração e aceite do convite.

Pergunta 1 – Há quanto tempo concluiu o Ensino Médio?

Menos de 1 ano de 1 a 3 anos de 3 a 5 anos mais de 5 anos

Pergunta 2 – Possui outro curso superior?

SIM NÃO

Se **SIM**, terminou à Menos de 1 ano de 1 a 3 anos mais de 3 anos. **Qual curso:** _____

Pergunta 3 – Você possui computador em casa?

SIM NÃO

Pergunta 4 – Você trabalha?

SIM NÃO

Pergunta 6 – Seu tempo de dedicação semanal, fora de sala de aula, para estudar a disciplina de ALP é de

Menos de 1 hora de 1 a 2 horas de 2 a 3 horas mais de 3 horas

Pergunta 7 – Você tentou criar outros programas utilizando a linguagem C, não propostos pelo professor durante a realização da disciplina de ALP?

SIM NÃO

Pergunta 8 – Quais as principais dificuldades encontradas durante a disciplina de ALP?

Pergunta 9 – Você conhecia a ferramenta *Scratch* de programação?

SIM NÃO

Se **SIM**, durante quanto tempo utilizou a ferramenta?

Menos de 1 ano de 1 a 3 anos mais de 3 anos só conhecia, mas não usou

Pergunta 10 – O que você achou da utilização do *Scratch* na disciplina de ALP?

não gostei e não entendi sua utilidade não gostei mas entendi sua utilização
 gostei mas é difícil de usar gostei e achei fácil de usar

Pergunta 11 – Você gostou de utilizar a ferramenta *Scratch* para criação de jogos?

SIM NÃO

Pergunta 12 – Seu tempo de dedicação semanal, fora de sala de aula, para usar os softwares propostos pelo professor que auxiliam o estudo de ALP, (*Lightbot* e *Scratch*) foi de:

<input type="checkbox"/> Menos de 1 hora <input type="checkbox"/> de 1 a 2 horas <input type="checkbox"/> de 2 a 3 horas <input type="checkbox"/> mais de 3 horas
Pergunta 13 – Você chegou a utilizar o tutorial da ferramenta <u>Scratch</u> ou outros disponíveis <i>on-line</i> nos exercícios propostos pelo professor? <input type="checkbox"/> SIM <input type="checkbox"/> NÃO
Pergunta 14 – A disponibilidade <i>on-line</i> da ferramenta <u>Scratch</u> motivou-o a usá-lo? <input type="checkbox"/> SIM <input type="checkbox"/> NÃO
Pergunta 15 – Você tentou realizar outras tarefas ou exercícios não propostos pelo professor quando utilizou a ferramenta <u>Scratch</u> ? <input type="checkbox"/> SIM <input type="checkbox"/> NÃO
Pergunta 16 – O que achou da utilização da ferramenta <u>Scratch</u> para programação? <input type="checkbox"/> muito fácil <input type="checkbox"/> fácil <input type="checkbox"/> mediano <input type="checkbox"/> difícil <input type="checkbox"/> muito difícil
Pergunta 17 – Quais as principais dificuldades encontradas durante a utilização da ferramenta Scratch?
Pergunta 18 – Gostaria de comentar ou deixar sua opinião a respeito da disciplina de ALP ou da utilização de ferramentas de apoio utilizadas na disciplina de ALP (<u>Light Bot</u> , <u>Scratch</u> e outros)?

APENDICE B – QUESTIONÁRIO APLICADO AO PROFESSOR



**MESTRADO PROFISSIONAL EM EDUCAÇÃO
TECNOLÓGICA**
CONVITE PARA REALIZAÇÃO DE QUESTIONÁRIO SOBRE O ENSINO NA DISCIPLINA
DE ALP E A UTILIZAÇÃO DE FERRAMENTAS CONSTRUTIVISTAS – SCRATCH

Olá,

Meu nome é Flamarion Assis Jeronimo Inácio, sou aluno do Mestrado Profissional em Educação Tecnológica do IFTM – Câmpus Uberaba, sob a orientação do Prof. Dr. Hugo Leonardo Pereira Rufino e convidamos você a responder algumas questões sobre as Ferramentas Construtivistas no Apoio ao Ensino da Disciplina de Algoritmo e Lógica de Programação – ALP, mais precisamente sobre a Ferramenta *Scratch* utilizada no início da disciplina de ALP.

As respostas serão tratadas de forma confidencial. Portanto, resposta de forma que melhor expresse a sua opinião.

Agradeço sua colaboração e aceite do convite.

Pergunta 1 – Há quanto tempo concluiu o curso de formação superior?

Menos de 1 ano de 1 a 3 anos de 3 a 5 anos mais de 5 anos

Pergunta 2 – Possui outro curso superior?

SIM NÃO

Se SIM, terminou à Menos de 1 ano de 1 a 3 anos mais de 3 anos. Qual curso: _____

Pergunta 3 – Qual a modalidade da sua formação superior?

Bacharelado Licenciatura Tecnológico Outro _____

Possui outra titulação? _____

Pergunta 4 – Acredita que a carga horária da disciplina de ALP é suficiente para a ministração de seus conteúdos?

SIM NÃO

Pergunta 5 – Você acredita que a maioria dos alunos que iniciam a disciplina de ALP possui o desenvolvimento lógico necessário à realização da disciplina de ALP?

SIM NÃO

Pergunta 6 – Você acha que, em média, a maioria dos alunos disponibilizam quantas horas semanais ao estudo da disciplina de ALP?

Menos de 1 hora de 1 a 2 horas de 2 a 3 horas mais de 3 horas

Pergunta 7 – Você acha que, em média, para o bom andamento da disciplina, quantas horas os alunos deveriam se dedicar à disciplina?

Menos de 1 hora de 1 a 2 horas de 2 a 3 horas mais de 3 horas

Pergunta 8 – Você achou que os alunos desta turma estavam motivados para a realização da disciplina?

SIM NÃO

Pergunta 9 – Quais as principais dificuldades você encontrou para ministração da disciplina de ALP?

Pergunta 10 – Você conhecia a ferramenta *Scratch* de programação?

SIM NÃO

Se SIM, durante quanto tempo utilizou a ferramenta?

Menos de 1 ano de 1 a 3 anos mais de 3 anos só conhecia, mas não usou

Pergunta 11 – O que você achou da utilização do *Scratch* na disciplina de ALP?

não gostei e acho que os alunos também não não gostei mas acho que os alunos

<p>gostaram de utilizá-la <input type="checkbox"/> gostei mas acho que os alunos não gostaram de utilizá-la <input type="checkbox"/> gostei e acho que os alunos também</p>
<p>Pergunta 11 – O que você achou da utilização do <i>Scratch</i> na disciplina de ALP? <input type="checkbox"/> não gostei e acho que os alunos também não <input type="checkbox"/> não gostei mas acho que os alunos gostaram de utilizá-la <input type="checkbox"/> gostei mas acho que os alunos não gostaram de utilizá-la <input type="checkbox"/> gostei e acho que os alunos também</p>
<p>Pergunta 12 – Você conhece a metodologia construtivista de ensino utilizada pelo <i>Scratch</i>? <input type="checkbox"/> SIM <input type="checkbox"/> NÃO</p>
<p>Pergunta 13 – Você usa ferramentas computacionais de apoio ao ensino de ALP? <input type="checkbox"/> SIM <input type="checkbox"/> NÃO Se SIM, quais? _____</p>
<p>Pergunta 14 – Quais as principais dificuldades encontradas pelos alunos no desenvolvimento da disciplina de ALP?</p>
<p>Pergunta 15 – Gostaria de comentar ou deixar sua opinião a respeito da disciplina de ALP ou da utilização de ferramentas de apoio utilizadas na disciplina de ALP (<i>Light Bot e Scratch</i>)</p>

APÊNDICE C – QUESTIONÁRIO APLICADO AO COORDENADOR



MESTRADO PROFISSIONAL EM EDUCAÇÃO TECNOLÓGICA

CONVITE PARA REALIZAÇÃO DE QUESTIONÁRIO SOBRE O ENSINO NA DISCIPLINA DE ALP E A UTILIZAÇÃO DE FERRAMENTAS CONSTRUTIVISTAS – SCRATCH

Olá,

Meu nome é Flamarion Assis Jeronimo Inácio, sou aluno do Mestrado Profissional em Educação Tecnológica do IFTM – Câmpus Uberaba, sob a orientação do Prof. Dr. Hugo Leonardo Pereira Rufino. Convidamos você a responder algumas questões sobre as Ferramentas Construtivistas no Apoio ao Ensino da Disciplina de Algoritmo e Lógica de Programação – ALP, mais precisamente sobre a Ferramenta Scratch utilizada no início da disciplina de ALP.

As respostas de cada participante serão tratadas de forma confidencial. Portanto, marque a resposta que mais expresse a sua opinião.

Agradeço sua colaboração e aceite do convite.

Pergunta 1 – Há quanto tempo concluiu o curso de formação superior?

Menos de 1 ano de 1 a 3 anos de 3 a 5 anos mais de 5 anos

Pergunta 2 – Sua área de formação é na área de Informação e Comunicação?

SIM NÃO

Há quanto tempo é coordenador do curso?

Menos de 1 ano de 1 a 3 anos mais de 3 anos

Pergunta 3 – Qual a modalidade da sua formação superior?

Bacharelado licenciatura Tecnológico outro _____

Possui outra titulação? Doutor em Ciência da Computação

Pergunta 4 – Acredita que a carga horária da disciplina de ALP é suficiente para a ministração de seus conteúdos?

SIM NÃO

Pergunta 5 – Você acredita que a maioria dos alunos que iniciam a disciplina de ALP possui o desenvolvimento lógico necessário à realização da disciplina de ALP?

SIM NÃO

Pergunta 6 – Quais as estratégias realizadas pela instituição para ajudar os alunos, fora de sala de aula, no desenvolvimento da realização da disciplina?

Pergunta 7 – Você acredita que o perfil da turma que entrou em 2015/2 é diferente das anteriores?

SIM NÃO

