**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO**
**DEPARTAMENTO DE INFORMÁTICA**
**PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

# NEWPROG - UM AMBIENTE ONLINE PARA CRIANÇAS APRENDEREM PROGRAMAÇÃO DE COMPUTADORES

**CARPEGIERI TOREZANI**

**VITÓRIA**
**2014**

CARPEGIERI TOREZANI

# NEWPROG - UM AMBIENTE ONLINE PARA CRIANÇAS APRENDEREM PROGRAMAÇÃO DE COMPUTADORES

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Informática.
Orientador: Prof. Dr. Orivaldo de Lira Tavares.

VITÓRIA

2014

**CARPEGIERI TOREZANI**

# NEWPROG - UM AMBIENTE ONLINE PARA CRIANÇAS APRENDEREM PROGRAMAÇÃO DE COMPUTADORES

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Departamento de Informática da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do título de Mestre em Informática, na área de concentração de Informática na Educação.

## COMISSÃO EXAMINADORA

_____

Prof. Dr. Orivaldo de Lira Tavares
UFES – Universidade Federal do Estado do Espírito Santo - BR
Orientador

_____

Prof. Dr. Crediné Silva de Menezes
UFRGS – Universidade Federal do Rio Grande do Sul – BR
Examinador Externo

_____

Prof. Dr. Eloi Luiz Favero
UFPA - Universidade Federal do Pará – BR
Examinador Externo

*Aos meus familiares, em especial minha companheira Mireli e minha pequena Lilithy, pelo apoio durante toda esta jornada.*

*Aos meus amigos que contribuíram para a realização de mais esta etapa da minha vida.*

# AGRADECIMENTOS

À minha família por todo apoio, à minha companheira Mireli Herpis pela paciência e crença de que tudo daria certo e a minha filha Lilithy Torezani fonte de inspiração e motivação.

Ao orientador desta pesquisa, Dr. Orivaldo de Lira Tavares, a quem sempre serei grato pelo modo como me acolheu no PPGI, sem receios em me ofertar oportunidades de aprendizado e sem reservas em depositar sua confiança em meu trabalho.

Ao professor Crediné Menezes por suas contribuições que auxiliaram na construção deste trabalho e enriqueceram a pesquisa.

Aos professores componentes da banca examinadora, pelas críticas e contribuições.

Aos amigos e colegas que me ajudaram e viveram comigo esta trajetória, ao pessoal do Laboratório de Informática em Educação – LIED e a todos que conheci durante o mestrado.

A todos muito obrigado!

*Ir mais além*

*Vencer um desafio,*
*Procurar a superação,*
*Escapar por um fio,*
*E tornar-se campeão,*

*Superar-se em cada gesto,*
*Conquistar o infinito,*
*Ir mais além do que o certo,*
*Ultrapassar o mais bonito,*

*Ir além da superação*
*E conquistar o impossível,*
*Ir além da imaginação*
*Para vencer o invencível.*

***Rômulo Raulino***

# RESUMO

Esta dissertação apresenta o desenvolvimento de um ambiente web com recursos digitais para crianças aprenderem programação. Quando as crianças desenvolvem atividades de programação, especialmente planejadas pelo professor e preparadas com os recursos criados nesta dissertação, passam por um processo de desenvolvimento de suas habilidades cognitivas, tais como: Resolução de Problemas, Autonomia, Planejamento, Autorregulação e Espírito Crítico.

Os seguintes recursos digitais foram desenvolvidos: 1) EANewProg - um Editor de Atividades de Programação que permite ao professor editar atividades a serem apresentadas às crianças e 2) NewProg - um ambiente para crianças visualizarem as atividades, editarem e avaliarem soluções. O NewProg também guarda o histórico das atividades (percurso da aprendizagem) desenvolvidas pela criança, de modo que o professor possa acompanhar a aprendizagem e planejar atividades mais adequadas a cada criança.

# ABSTRACT

This dissertation presents the development of a web environment with digital resources for children to learn programming. When children develop programming activities, specially planned by the teacher and prepared with the resources created in this work, undergo a process of development of their cognitive skills such as: Troubleshooting, Autonomy, Planning, Self-Regulation and Critical Spirit.

The following digital resources were developed: 1) EANewProg - an Editor of Programming Activities that allows teachers to edit activities to be presented to children and 2) NewProg - an environment for children to visualize activities, edit and evaluate solutions. The NewProg also stores a history of activities (learning path) developed by the child, so that the teacher can monitor and plan the most suitable learning activities to each child.

# LISTA DE TABELAS

# LISTA DE FIGURAS

**SUMÁRIO**

# 1. INTRODUÇÃO

A falta de domínio de técnicas de resolução de problemas faz com que seja difícil aprender programação. A aprendizagem de programação é uma tarefa de enorme complexidade, e os processos cognitivos envolvidos são pouco conhecidos. Atualmente existem metodologias pedagógicas para o ensino-aprendizagem de programação, mas, mesmo as mais conhecidas, não formalizam procedimentos claros de ensino e aprendizagem, ficando muitos aspectos a serem decididos pelo educador.

A aprendizagem de programação é benéfica para o desenvolvimento de habilidades cognitivas necessárias para a resolução de problemas. De acordo com FESSAKIS et alii (2013), a aprendizagem de programação permite o desenvolvimento de conhecimentos em diversas áreas.

TAVARES et alii (2012) relatam que após tentativas frustradas de construir programas para resolver problemas, o aluno reinicia suas ações em busca de uma solução correta, tentando descobrir e corrigir os erros cometidos, inicialmente, em um processo de tentativa e erro, até que consiga ganhar experiência que lhe permita tomar consciência sobre esse processo e sobre a construção de soluções corretas. Esse processo de tentativa e erro pode ser mais curto ou mais longo, de acordo com a experiência prévia do estudante.

De acordo com NETO et alii (2006), a aprendizagem de programação pressupõe a transposição de alguns patamares de perícia (expertise) até que o sujeito seja um programador experiente. Esses patamares, embora cogitados por muitos pesquisadores, ainda são desconhecidos, por isso, o ensino de programação não pode seguir uma metodologia linear.

## 1.1. Descrição do problema

A aprendizagem de programação é considerada um processo difícil que exige habilidades cognitivas de ordem superior. A falta dessas habilidades antes dos alunos chegarem a Universidade aumenta as dificuldades de aprendizagem. Por

isso, é necessária a realização de atividades que colaborem com o desenvolvimento da capacidade cognitiva dos estudantes.

## 1.2. Justificativa

A falta de habilidades cognitivas de ordem superior tende a dificultar o processo de aprendizagem em diversas áreas. O desenvolvimento de um ambiente para auxiliar o ensino e a aprendizagem de programação nas séries iniciais do ensino fundamental justifica-se pela importância do uso de ferramentas que possam despertar a independência, a autonomia e o espírito crítico das crianças, bem como o uso de ferramentas que tornem fácil a implementação de atividades a serem propostas para as crianças, de modo a desenvolverem habilidades importantes para a programação de computadores. Além disso, pode-se destacar:

- A informática é capaz de unir elementos importantes na aprendizagem, permitindo em uma atividade, desenvolver habilidades como resolução de problemas, planejamento, autorregulação;
- Com o aumento de alunos em uma mesma turma, o educador precisa de ferramentas que o auxilie na avaliação individual, permitindo mudanças de estratégias na busca de uma aprendizagem eficaz;
- Acompanhamento evolucional de cada aprendiz, ao guardar seu histórico de atividades, permite ao educador a construção de novas estratégias para o crescimento cognitivo do aprendiz.

## 1.3. Objetivos

Seguem os objetivos deste trabalho.

### 1.3.1 OBJETIVOS GERAIS

O objetivo geral desta dissertação é o desenvolvimento de um ambiente web que contenha recursos digitais que possam ser usados para a aprendizagem de programação por crianças, de modo a colaborarem com o desenvolvimento de

habilidades cognitivas de ordem superior, tais como: Resolução de Problemas, Autonomia, Planejamento, Autorregulação e Espírito Crítico. Para isso, este trabalho apresenta a especificação, o projeto e a prototipação de um ambiente na web, chamado NewProg, que permite às crianças desenvolverem atividades de programação, previamente preparadas por seus professores. O NewProg possui vários recursos digitais que permitem: registrar o percurso de aprendizagem de cada criança (um registro das atividades da criança durante cada sessão), de modo que o professor possa acompanhar a evolução cognitiva de cada criança; agentes que desafiam a criança a encontrar novas soluções para cada atividade, para fazerem as crianças refletirem sobre as soluções propostas e buscarem novas soluções possíveis.

Outro objetivo é desenvolver um Ambiente de Edição de Atividades, chamado EANewProg, que possa ser usado pelos professores para a edição de novas atividades a serem realizadas pelas crianças no NewProg.

### 1.3.2. OBJETIVOS ESPECÍFICOS

- Auxiliar os educadores no processo de acompanhamento da aprendizagem de seus aprendizes;
- Oferecer aos educadores ferramentas que permitam a construção de atividades para o aprendizado de programação;
- Possibilitar acompanhamento individualizado das construções feitas pelos aprendizes, a fim de verificar se o processo usado pela criança é coerente.

## 1.4. Metodologia

De maneira a alcançar os objetivos almejados, houve a necessidade de uma abordagem com o uso do conceito de pesquisa etnográfica e da teoria da atividade, acompanhando e vivenciando o dia-a-dia da escola.

### 1.4.1. INVESTIGAÇÃO ETNOGRÁFICA

Segundo MATTOS (2001), etnografia é também conhecida como: pesquisa social, observação participativa, pesquisa interpretativa, pesquisa analítica, pesquisa hermenêutica. Compreende o estudo, pela observação direta e por um período de tempo, das formas costumeiras de viver de um grupo particular de pessoas: um grupo de pessoas associadas de alguma maneira, uma unidade social representativa para estudo, seja ela formada por poucos ou muitos elementos. Por exemplo: uma vila, uma escola, um hospital, etc.

MATTOS (2001) ressalta que, a etnografia é um processo guiado preponderantemente pelo senso questionador do etnógrafo. Deste modo, a utilização de técnicas e procedimentos etnográficos, não segue padrões rígidos ou pré-determinados, mas sim, o senso que o etnógrafo desenvolve a partir do trabalho de campo no contexto social da pesquisa. Estas técnicas, muitas vezes, têm que ser formuladas ou criadas para atenderem à realidade do trabalho de campo. Nesta perspectiva, o processo de pesquisa será determinado explícita ou implicitamente pelas questões propostas pelo pesquisador.

Segundo CAMPANA (2009), a etnografia (Grafia vem do grego graf(o) significa escrever sobre, escrever sobre um tipo particular - um etn(o) ou uma sociedade em particular) estuda os padrões mais previsíveis do pensamento e comportamento humanos mostrados em sua rotina diária; estuda ainda os fatos e/ou eventos menos previsíveis ou manifestados particularmente em determinado contexto interativo entre as pessoas ou grupos. Na etnografia, são observados os modos como esses grupos sociais ou pessoas conduzem suas vidas com o objetivo de "revelar" o significado cotidiano, nos quais as pessoas agem. O objetivo é documentar, monitorar e encontrar o significado da ação (MATTOS, 2001).

Para GEERTZ apud MATTOS (2001), praticar etnografia não é somente estabelecer relações, selecionar informantes, transcrever textos, levantar genealogias, mapear campos, manter um diário, o que a define é o tipo de esforço intelectual que ele representa: um risco elaborado para uma "descrição densa”.

Etnografia é a escrita do visível. A descrição etnográfica depende das qualidades de observação, de sensibilidade ao outro, do conhecimento sobre o contexto estudado, da inteligência e da imaginação científica do etnógrafo (MATTOS, 2001).

No contexto deste trabalho a abordagem etnográfica significa levantar as necessidades, interesses e motivações dos professores por atividades que possam levar seus alunos, das séries iniciais da educação fundamental, a desenvolverem habilidades cognitivas importantes para a programação de computadores.

### 1.4.2. TEORIA DA ATIVIDADE

A teoria da atividade fornece o método de compreensão e análise de um fenômeno, de modo a encontrar padrões e fazer inferências por meio de interações, delineando e apresentando fenômenos.

A Teoria da Atividade tem origem em três vertentes: a filosofia clássica Alemã dos séculos XVIII e XIX (de Kant a Hegel); os escritos de Marx e Engels, que elaboraram o conceito de atividade; e a psicologia Soviética, fundada por Vygostky, Leont'ev e Lúria. O termo "Teoria da Atividade" surgiu durante as décadas de 1920 e 1930, dentro da escola histórico-cultural soviética de psicologia (NARDI, 1996 e KAPTELININ, 1997). A Teoria da Atividade, num sentido amplo, pode ser definida como uma estrutura filosófica e interdisciplinar para estudar diferentes formas de práticas humanas de processos de desenvolvimento, tanto no nível individual como no nível social (MARTINS, 1999).

Segundo NARDI apud MARTINS (1999) e KAPTELININ (1997) a Teoria da Atividade é constituída por um conjunto de princípios que integram um sistema conceitual geral. Esses princípios são:

- **Princípio da unidade entre consciência e atividade:** Considerado o princípio fundamental da Teoria da Atividade, em que consciência e atividade são concebidas de forma integrada. Na qual a consciência significa a mente humana como um todo, e a atividade, a interação humana com sua realidade objetiva. Esse princípio diz que a mente humana emerge e existe como um componente especial da interação humana com o seu ambiente. A mente é

um órgão especial que aparece no processo de evolução para ajudar organismos a sobreviverem. Assim, esse princípio pode ser analisado e entendido somente dentro do contexto da atividade humana.

- **Princípio da orientação a objetos:** Faz a abordagem da Teoria da Atividade no que se refere ao ambiente em que pessoas interagem. As pessoas procuram viver num ambiente que é significativo para elas. Este ambiente consiste de entidades que combinam todos os tipos de características objetivas, incluindo aquelas determinadas culturalmente, que por sua vez determinam as formas como as pessoas agem sobre essas entidades.
- **Princípio da estrutura hierárquica da atividade:** a Teoria da Atividade diferencia os procedimentos humanos em vários níveis (atividade, ação e operação), levando em conta os objetivos para os quais esses procedimentos são orientados. Numa situação real, frequentemente essa distinção é necessária para prever o comportamento humano. Para essa finalidade, é de importância crítica para a diferenciação entre motivos, metas e condições, que estão associados à atividade, ação e operação, respectivamente.
- **Princípio da internalização-externalização:** Apresenta os mecanismos básicos da origem dos processos mentais. Esse princípio declara que processos mentais são derivados das ações externas através do curso da internalização. Internalização é o processo de absorção de informações (nas suas diversas formas) realizado pela mente humana, que ocorre a partir do contato com o ambiente em que a pessoa está inserida. A externalização é o processo inverso da internalização, manifestado através de atos, de tal forma que eles possam ser verificados e corrigidos se necessário.
- **Princípio da mediação:** A atividade humana é mediada por um número de ferramentas, tanto externas (por exemplo: um machado ou um computador) como internas (por exemplo: uma heurística ou um conceito). As ferramentas são “veículos” da experiência social e do conhecimento cultural.
- **Princípio do desenvolvimento:** Entender um fenômeno significa conhecer como ele se desenvolveu até sua forma atual, pois ao longo do tempo sofre alterações. Compreender essas alterações auxiliará no entendimento do seu estado atual. Esses princípios não são ideias isoladas, eles estão intimamente ligados. A natureza da Teoria da Atividade é manifestada nesse conjunto de princípios.

## 1.5 Questões de investigação

Durante a pesquisa desta dissertação, foram levantadas diversas questões a serem analisadas e solucionadas, destacamos:

- Quais atividades de programação são úteis para crianças desenvolverem habilidades cognitivas?
- Como deve ser a interface de um ambiente que permita que crianças elaborem programas de computadores?
- Quais recursos digitais devem ser apresentados para crianças programarem?
- Como motivar as crianças a encontrarem soluções diferentes e melhores?
- Quais recursos digitais devem ser apresentados aos professores que queiram propor atividades de programação para crianças?
- Como deve ser a interface de um ambiente que permita a edição de atividades de programação para crianças?
- Como será o acompanhamento das atividades solucionadas?
- Quais as plataformas, compatíveis com esse ambiente, são necessárias?
- Quais funções inteligentes são necessárias no ambiente?

## 1.6 Requisitos básicos de uma linguagem de programação

Uma linguagem de programação precisa permitir a descrição de:

- Sequências de instruções;
- Testes de condições;
- Recursões ou Repetições de instruções.

Na linguagem de programação usada para as crianças resolverem as atividades do NewProg, usamos apenas sequências de instruções. Como os cenários usados atualmente são estáticos, os testes das condições são feitos pela própria criança, ao selecionar o caminho por onde o avatar passará. Quando for possível usarmos cenários dinâmicos em que a configuração deles possa se alterar dinamicamente, será necessário o uso de testes de condições no programa que controla o avatar,

para que ele possa selecionar um caminho que solucione o problema definido em cada atividade.

As repetições de instruções também não são usadas na linguagem atual do NewProg, uma vez que a criança pode definir os caminhos que resolvam os problemas que lhe são apresentados usando apenas sequências de instruções.

Portanto, esta dissertação trata apenas de uma parte inicial da programação de computadores. Entretanto, é possível beneficiar o desenvolvimento cognitivo de crianças das séries iniciais do ensino fundamental (no Brasil), propondo a elas atividades que possam ser solucionadas com a descrição de soluções que usem apenas sequências de instruções, conforme se apresenta nesta dissertação.

## 2. PROFESSORES, FERRAMENTAS E APRENDIZAGEM

Segundo SICA et alii (2007), os professores da educação básica têm sobre si uma grande carga de responsabilidades. São responsáveis pelo processo de alfabetização, que abrirá inumeráveis possibilidades de construção de conhecimentos no desenvolvimento das crianças e adolescentes. São responsáveis pelo ensino de conceitos fundamentais em todas as áreas, a partir do desenvolvimento do raciocínio operacional concreto e formal. Além disso, em quase todo projeto pedagógico, um dos principais objetivos é o desenvolvimento da autonomia, mesmo que não se compreenda bem o profundo significado dessa dimensão.

De acordo com MORAES (2000), a realidade da pedagogia dos meios modernos, cuja interação professor-aluno-informação deverá levar o indivíduo a aprender a pensar, a aprender a antecipar, a aprender a cultivar o espírito crítico e criativo, para que ele possa sobreviver num mundo onde inúmeras informações estarão disponíveis, e que precisam ser criticamente avaliadas, para serem transformadas em conhecimentos.

O computador, para alguém que está apenas iniciando sua alfabetização informática, é um objeto complexo, do qual esta pessoa apenas pode compreender relações muito simples, mas que através da mediação de um modelo, torna-se um objeto mais simples, e correlativamente há um enriquecimento quantitativo e qualitativo das relações percebidas, e também uma maior capacidade de atribuir importância relativa a essas relações percebidas. A percepção inicial, nebulosa, cede lugar a uma discriminação esquemática das relações relevantes. Escolhas (digitais) e comparações (analógicas) estão presentes em um movimento cíclico em torno desse processo de construção da rede de significados que possibilita compreender o que é o computador. A compreensão das relações relevantes torna o objeto mais nítido, um feixe de relações significativas (TENÓRIO, 2003).

Ainda segundo TENÓRIO, alguns conteúdos escolares têm sua importância aumentada em decorrência dos impactos da informática na sociedade e no ensino. É o caso das noções de algoritmo. A noção de algoritmo se refere à decomposição de uma tarefa em uma sequência ordenada de tarefas simples, elementares, as

quais realizadas passo-a-passo, conduzem à solução de um determinado problema – o problema para o qual o algoritmo foi construído.

Atualmente o computador representa mais que uma máquina de ensinar. Seu uso na educação, segundo VALENTE (1993), pode promover a aprendizagem do educando e ajudar na construção do processo de conceituação e no desenvolvimento de habilidades importantes para que ele participe da sociedade do conhecimento. Para que isso ocorra, é preciso levar em consideração aspectos pedagógicos e sociais ao se inserir a informática nos processos educacionais.

> A verdadeira função educacional não deve ser a de ensinar, mas sim a de criar condições de aprendizagem. O professor precisa deixar de ser o repassador de conhecimento e passar a ser o criador de ambiente de aprendizagem e o facilitador do processo de desenvolvimento intelectual do aluno (VALENTE, 1993).

O uso de novas tecnologias que promovam a aprendizagem, principalmente as computacionais, requer interação, participação, colaboração e a consciência de que o conhecimento une o resultado de um processo de construção e de que o indivíduo é o autor de seu próprio conhecimento. De acordo com MORAN (1998), as tecnologias são extensões da mente humana.

AZEVEDO (2007) destaca, o uso da informática na educação tem progressivamente sido aplicado no processo ensino-aprendizagem, mediante o desenvolvimento de atividades lúdicas apoiadas em softwares educacionais, capazes de tornar a prática educacional e a relação professor-aluno mais prazerosa.

No entanto, PAPERT (1994) alerta que atividades lúdicas não constituem apenas momentos de diversão; isso porque existem estratégias a serem observadas durante as brincadeiras. Na opinião de VYGOTSKY *apud* AZEVEDO (2007) as atividades lúdicas não estão simplesmente ligadas ao prazer:

> [...] definir o brinquedo como uma atividade que dá prazer à criança é incorreto por duas razões. Primeiro, muitas atividades dão à criança experiências de prazer muito mais intensas. E segundo, existem jogos nos quais a própria atividade não é agradável [...].

## 2.1 Papel da Informática na Educação

De acordo com VALENTE (1993), o uso da informática na educação já tem uma longa história, na qual podemos encontrar casos de sucesso e de fracasso. Abordagens mais recentes buscam utilizar o computador como ferramenta de apoio ao aluno para a construção de seu próprio conhecimento, ao invés de fazer do computador uma máquina que ajuda a ensinar (MONTEIRO *apud* CAMPANA, 2009). O desenvolvimento de ambientes para apoiar educação tem buscado fornecer recursos que não transmitam informações apenas, mas façam com que o aprendiz desenvolva as habilidades necessárias para aplicar seu conhecimento (SANCHES *apud* CAMPANA, 2009).

Segundo PAPERT *apud* LA TAILLE (1990), “Se o computador e a programação se tornarem parte do cotidiano das crianças, o intervalo conservação-combinação certamente se fechará e poderia chegar a se inverter: as crianças podem aprender a serem sistemáticas antes de aprender a serem quantitativas”.

CHAVES *apud* LA TAILLE (1990), “Defende a tese de que toda criança deveria aprender a programar porque esse aprendizado, além de útil por si mesmo, traz embutido à aprendizagem de uma série de conceitos, habilidades e atitudes que são importantes – eu diria até essenciais - para seu desenvolvimento intelectual e cognitivo”.

Opinião parecida é emitida por M. Pair, Directeur des Lycées do Ministério Francês da Educação: “A informática traz uma nova dimensão à formação do espírito, porque ela favorece a passagem do “fazer” ao “conceber”, do estágio das operações concretas àquele das operações formais, que é um objetivo essencial do ensino, e, particularmente, do ensino ginasial” (Colooque, Informatique et Enseignement CHAVES *apud* LA TAILLE, 1990).

Devido ao potencial da informática na educação de criar cenários educacionais, com novas perspectivas pedagógicas, diversos pesquisadores na área de informática na educação estão propondo e avaliando formas de se trabalhar com essas ferramentas, de modo que o aprendizado seja realizado de forma mais interativa, amigável e com resultados mais efetivos.

BOSSUET apud LA TAILLE (1990) destaca que "A linguagem Logo não contém a noção de erro no sentido que existe em outras linguagens de programação. O sistema nunca insulta uma criança que escreveu uma regra de sintaxe não reconhecida, nunca lhe envia mensagens do tipo *sintax error*, mas pede-lhe educadamente que seja precisa no que ela quis dizer. A linguagem empregada, portanto, não é “culpabilizante”.

Segundo PAPERT apud LA TAILLE (1990), "A escola ensina que errar é mau; a última coisa que alguém deseja fazer é examinar esses erros, deter-se neles ou mesmo pensá-los. A criança fica contente em poder usar a capacidade do computador para apagar esses erros sem deixar vestígios para que ninguém os veja".

De acordo com ALMEIDA (2012), o computador, embora nascido de uma dada civilização e para solucionar dados problemas, hoje é um patrimônio transcultural. A absorção crítica de sua utilização na educação deve ser procedida de análises das questões mais radicais que afligem esta dimensão da cultura brasileira. Como tarefa dos educadores, cumpre desenvolver uma pedagogia do uso crítico da informática na educação.

## 2.2 Computador e aprendizagem

De acordo com MARQUES et alii (1986), à primeira vista, as vantagens e limitações originárias da utilização do computador em educação estão vinculadas apenas à forma como o mesmo é utilizado. No entanto, esta utilização é determinada em grande parte pela filosofia de educação dos educadores que vão empregar o computador como instrumento didático no processo de ensino-aprendizagem. Em outras palavras, o que muitos veem como vantagem pode ser considerada por outros como uma séria limitação ou mesmo um emprego errôneo do instrumento.

Segundo MATTOS et alii (1986), nessa perspectiva filosófica, podem ser diferenciadas atualmente duas posições sobre a forma mais adequada de utilizar o computador em educação:

- Uma das posições afirma que o computador deve ser usado só como um recurso de aprendizagem. Nesta abordagem, dispondo do instrumental necessário, em geral linguagens de programação, o aluno dirige seu próprio aprendizado. Por exemplo, usando a linguagem Logo, que permite, através de instruções simbólicas, o desenho de figuras geométricas na tela, ele pode descobrir sozinho conceitos geométricos, como abertura de ângulo e outros;
- A outra posição defende o uso do computador como instrumento didático, fornecendo ao aluno programas educativos estruturados que visam cumprir um determinado objetivo, vinculado ou não ao currículo.

LA TAILLE et alii (1986), destaca que o uso do computador como instrumento de ensino traz a vantagem de possibilitar a introdução de praticamente qualquer área do currículo, em qualquer momento do processo ensino-aprendizagem. Além disto, o computador por características que lhe são próprias apresenta algumas vantagens sobre outros instrumentos didáticos em muitas situações de ensino.

LA TAILLE et alii (1986) ressalva, não se deve perder de vista que utilizar o computador apenas como recurso de aprendizagem pode representar uma subutilização de um recurso extremamente rico e versátil. Isto, porque, embora possa produzir benefícios, como a construção do próprio aprendizado, o desenvolvimento do raciocínio lógico etc., esta forma de utilização está, por ora, limitada as áreas específicas de aprendizagem, como matemática, geometria ou linguagens de programação. Estas últimas exigem uma maturidade ainda maior do aluno em relação ao raciocínio lógico.

### 2.2.1 VANTAGENS EM RELAÇÃO A OUTROS INSTRUMENTOS

De acordo com MARQUES et alii (1986), entre as características do computador que lhe conferem vantagens sobre os demais instrumentos, podemos citar as seguintes:

- É um recurso audiovisual, superior aos demais por ser interativo. Neste sentido, pode solicitar e responder às intervenções do aluno, evitando que

este permaneça passivo e, consequentemente, que se disperse para outros aspectos não relevantes da situação;

- O computador possui a vantagem de poder obedecer ao ritmo próprio de cada aluno, por exemplo, repetindo uma mesma explicação o número de vezes que o aluno desejar, ou, esperando o tempo que for necessário por uma resposta do aluno;
- Prontidão, com que o aluno recebe o *feedback* às suas intervenções. Desta forma, ao trabalhar com um determinado conteúdo, o aluno tem uma avaliação imediata, sobre o que precisa exercitar mais para um completo domínio do assunto.

## 2.3 Computadores em Sala de Aula

A existência dos computadores nos mais diferentes locais de ação humana é uma realidade incontestável. Essa realidade, aliada ao fato de que a informática é uma novidade e, como tal, exige mudanças, provoca ferrenhas discussões: questiona-se a introdução e as formas de uso dessas modernas máquinas nas mais diversas áreas de atividade humana (COX, 2008).

Ainda de acordo com COX, no campo educacional, a atmosfera não se encontra diferente. Há fervorosos seguidores e ferozes opositores da informática a questionar se os computadores devem ser inseridos no contexto escolar e de que modo. Há aqueles que atribuem às máquinas de processamento o papel “mágico” de salvadoras da educação e há os que acreditam que a inserção delas nas salas de aula mecanizará os efeitos do processo ensino-aprendizagem.

Segundo VALENTE (1998), para a implantação do computador na educação são necessários basicamente quatro ingredientes: o computador, o software educativo, o professor capacitado para usar o computador como meio educacional e o aluno.

Continuando VALENTE afirma, quando o computador ensina o aluno, o computador assume o papel de máquina de ensinar e a abordagem educacional é a instrução auxiliada por computador. Essa abordagem tem suas raízes nos métodos de instrução programada tradicionais, porém, ao invés do papel ou do livro, é usado o

computador. Os softwares que implementam essa abordagem podem ser divididos em duas categorias: tutoriais e exercícios-e-práticas ("drill-and-practice"). Outro tipo de software que ensina são os jogos educacionais e a simulação. Nesse caso, a pedagogia utilizada é a exploração autodirigida ao invés da instrução explícita e direta.

Os computadores são indubitavelmente, velozes e confiáveis depositários de informações. No entanto, para que as informações “frutifiquem” em conhecimentos e/ou competências, os computadores precisam ser criteriosamente explorados no ambiente escolar (COX, 2008).

ALMEIDA (2012) ressalva que, implantar a informática na educação representa introduzir um instrumento perigoso, porque muito potente, numa região da cultura de extrema delicadeza. Trata-se de atuar com poderosos utensílios na formação dos traços culturais de uma sociedade. O desconhecimento das consequências psicológicas, culturais e políticas de seu uso abusivo ou acrítico representa um risco que não pode ser corrido por uma política educacional consistente.

### 2.3.1 FORMAS DE USO

COX (2008) destaca que as maneiras como os computadores podem ser usados no ambiente escolar certamente não se esgotam nas citações a seguir, buscando registrar apenas as propostas voltadas para o trabalho específico em sala de aula:

- Simulação: “Uma simulação educacional é uma ambientação realística na qual o aluno é apresentado a um problema e toma uma série de decisões, executando ações, em seguida, recebe informações sobre como a situação do ambiente se altera em resposta de suas ações. Em outras palavras, a simulação, permite que o aluno verifique o funcionamento de um determinado modelo simplificado da realidade, a partir de suas próprias hipóteses” (EIVAZIAN, 1995).
- Jogos: Jogo digital (ou videojogo ou jogo eletrônico), expressão genérica que se refere a jogos eletrônicos desenhados para serem jogados num computador, num console ou outro dispositivo tecnológico (PIVEC e

KEARNEY apud DIAS, 2009), pode ser definido como jogo onde existe interação entre humano e computador, recorrendo ao uso de tecnologia (GEE, 2003). São indicados para diversão, mas podem ser utilizados na escola, sob a supervisão do educador.

- Comunicação: “Dizemos que dois ou mais computadores estão em rede, quando eles são capazes de compartilhar recursos através de um sistema de comunicação. Os computadores de uma rede podem estar ligados por cabo, por linha telefônica privada ou por satélite. Uma rede pode transmitir textos, valores, sons e imagens e pode partilhar recursos também, como uma impressora” (GENNARI, 1999).
- Ensino a distância: Fazendo uso dos recursos disponibilizados pela informática, quando os computadores são interligados em rede, surge a educação via computador, pois as mensagens podem ser compostas por animações, cores, imagens, o que pode enriquecer muito o material educativo usado e possivelmente favorecer o processo de aprendizagem. (COX, 2008)
- Programas educacionais: são programas desenvolvidos especificamente para as atividades em sala de aula, tais como: histórias, enciclopédias, tutoria, autoria.

### 2.3.2 VANTAGENS DA UTILIZAÇÃO

COX destaca as possíveis vantagens da utilização do uso da informática na escola, sendo elas:

- Provocação de mudanças na educação escolar;
- Desenvolvimento da Linguagem e da Escrita;
- Favorecimento do Desenvolvimento da Cidadania;
- Favorecimento da Interdisciplinaridade;
- Preparação para o Mundo de Trabalho;
- Estímulo para o Aluno participar da Escola;
- Promoção da Interação dos Agentes Escolares;

## 2.4 História da Informática na Educação

De acordo com VALENTE (1998, 1999), o ensino através da informática tem suas raízes no ensino através das máquinas. Esta ideia foi usada por Dr. Sidney Pressey em 1924 que inventou uma máquina para corrigir testes de múltipla escolha. Isso foi posteriormente elaborado por B.F. Skinner que no início de 1950, como professor de Harvard, propôs uma máquina para ensinar usando o conceito de instrução programada. A utilização de computadores na educação é tão remota quanto o advento comercial dos mesmos. Esse tipo de aplicação sempre foi um desafio para os pesquisadores preocupados com a disseminação dos computadores na nossa sociedade. Já em meados da década de 50, quando começaram a serem comercializados os primeiros computadores com capacidade de programação e armazenamento de informação, apareceram às primeiras experiências do seu uso na educação.

# 3. AMBIENTES COMPUTACIONAIS PARA A APRENDIZAGEM DE PROGRAMAÇÃO

Segundo ALMEIDA (2012), “Tudo é educativo. Todos educam. A sociedade é uma grande agência educadora.” A partir destes pressupostos e em nome do espírito educativo de cada um, a educação escolar vê-se invadida por milhões de mestres-educadores, planejadores educacionais, instrumentos pedagógicos e linguagens computacionais com propostas educativas. No debate sobre informática aplicada à educação, a participação da maior parte possível da sociedade é de máxima importância.

Continuando, ALMEIDA, afirma que, formular uma pedagogia informática, não baseada na carência e pressupostos escolares e educacionais, costuma ser o risco que correm exatamente os países terceiro-mundistas! Quando se elabora cuidadosamente esta reflexão e estudos, tende-se a confundir informatização da educação com a utilização da linguagem BASIC, que acima de dar rigor ao pensamento estimula a rigidez do pensar. Um trabalho apressado e sem fundamentação com o BASIC poderá levar a escola a ter que fazer o aluno a desaprender a programar para reaprender a raciocinar, que é muito mais complexo que programar.

## 3.1 Trabalhos correlatos

Ao longo dos anos, foi criada uma infinidade de ferramentas para ajudar no aprendizado de programação para adultos, adolescentes e crianças, na qual cada um tinha uma postura, uma metodologia e uma maneira de ensinar seu público alvo.

A presente seção tem como objetivo avaliar ferramentas utilizadas pelo público infantil, elucidando suas características e deficiências de modo a destacar as características e os pontos chaves do ambiente apresentado nesta dissertação.

### 3.1.1 LINGUAGEM DE PROGRAMAÇÃO LOGO

O sistema Logo se constitui, até o momento, na mais estruturada e abrangente visão e prática de um instrumental informático aplicado à educação. Desenvolvido no *Massachusetts Institute of Technology* (MIT) pelo matemático Seymour Papert, discípulo de Jean Piaget e inspirado em teorias da psicologia genético-evolutiva. O sistema Logo vai desenvolver um trabalho que permitirá à criança programar o computador criativa e espontaneamente, quase sem instruções (ALMEIDA, 2012).

> "A teoria Logo privilegia a apropriação da tecnologia pelo usuário. Fundada na individualização ela propõe micromundos de logiciel e materiais estruturados, no interior dos quais os usuários podem elaborar modelos de pensamentos ou descobrir novos" (BOSSUET apud ALMEIDA, 2012).

Segundo SOARES (2009), Logo é uma linguagem simples e poderosa. Simples, porque é fácil de aprender: pessoas alfabetizadas, de qualquer idade, podem programar em seu primeiro contato com ela (ao contrário de outras linguagens, permite que a pessoa programe sem necessitar que tenha muitos conhecimentos prévios). Poderosa, porque tem recursos sofisticados, que atendem às exigências de programadores experientes. Apesar de ser uma linguagem acessível às crianças, Logo não é uma linguagem infantil. Através dela, as pessoas aprendem explorando, investigando e descobrindo por si mesmas. Inclusive, é possível trabalhar com Logo e Robótica com crianças a partir da quarta série.

#### 3.1.1.1 Comandos Básicos

Segundo MANZANO (2012), os programas escritos em Logo são chamados de procedimentos e para fazer uso desses é necessário seguir algumas regras de escrita:

- Para a definição dos nomes de procedimentos, variáveis e listas de propriedades ocorre a diferenciação entre letras maiúsculas e minúsculas;
- Dentro dos símbolos de colchetes são definidas as palavras. Uma palavra é delimitada apenas por espaços em branco e colchetes. Por exemplo, a definição da operação matemática [1+4] é considerada uma palavra;

- Após um sinal de aspas do lado de fora dos colchetes, uma palavra será delimitada imediatamente por um espaço em branco, colchete ou parênteses;
- Uma linha de instrução pode ser continuada na linha seguinte se seu último caractere for um símbolo til (~);
- Apesar de alguns comandos poderem ser escritos em caracteres maiúsculos faça sempre a entrada em caracteres minúsculos.

#### 3.1.1.2 INTERPRETADOR LOGO

De modo a trabalhar com a linguagem Logo é necessário um interpretador para a mesma, existe uma gama de interpretadores para a linguagem, nesta seção será apresentado o SuperLogo versão 3.0.

O interpretador SuperLogo por ser um software educativo sem fins lucrativos é disponibilizado pelo Universidade Estadual de Campinas, sendo disponibilizado gratuitamente no site: http://www.nied.unicamp.br/?q=content/super-logo-30.

Interpretador SuperLogo é composto por um plano coordenado sem eixos desenhados e uma tartaruga gráfica no centro da tela, na posição (0,0). Sendo indispensável à programação de comandos que façam com que a tartaruga ande e gire, permitindo assim a construção de formas e figuras geométricas.

A tela do interpretador do SuperLogo é composta de duas janelas de trabalho: Sendo uma grande e principal chamada SuperLogo 3.0, posicionada em cima da janela menor chamada Janela de comandos, conforme Figura 1.

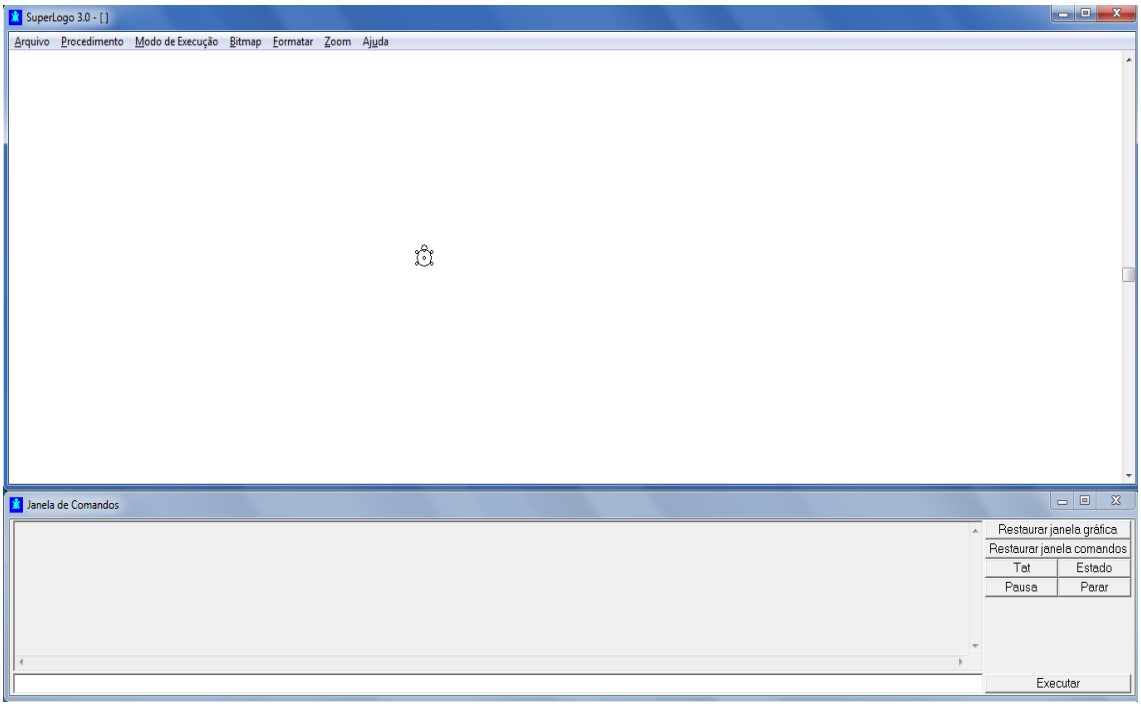


**Figura 1 – Tela do Interpretador do SuperLogo**

### 3.1.1.3 CÓDIGOS LOGO

Nesta seção serão apresentados Códigos Funcionais Logo, expostos na Tabela 1 – Códigos Logo, com imagens das suas execuções interpretadas no SuperLogo 3.0.

**Tabela 1 – Códigos Logo**

| Descrição | Código | Observações |
|---|---|---|
| Quadrado de medidas 100x110, sem repetição. | pf 100 pe 90 pf 100 pe 90 pf 100 pe 90 pf 100 pe 90 | Resultado da execução, vide Figura 2. |
| Quadrado de medidas 100x110, com repetição. | repita 4 [pf 100 pd 90] | Resultado da execução, vide Figura 2. |
| Flor | repita 8 [pd 45 repita 3 [repita 90 [pf 2 pd 2] pd 90]] | Resultado da execução, vide Figura 3. |
| Flor de Lotus | repita 16 [pd 22.5 repita 6 [repita 45 [pf 4 pd 4] pd 90]] | Resultado da execução, vide Figura 4. |

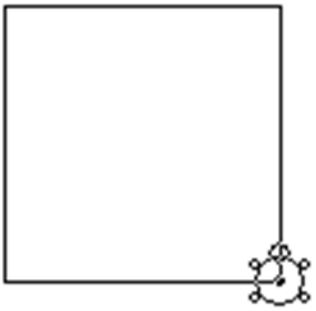

**Figura 2 – Execução código quadrado 100x100**

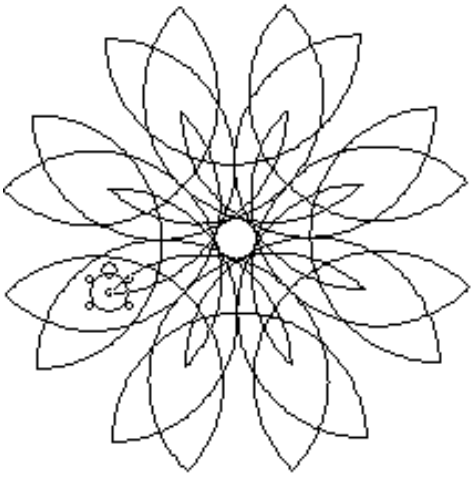

**Figura 3 – Execução código Flor**

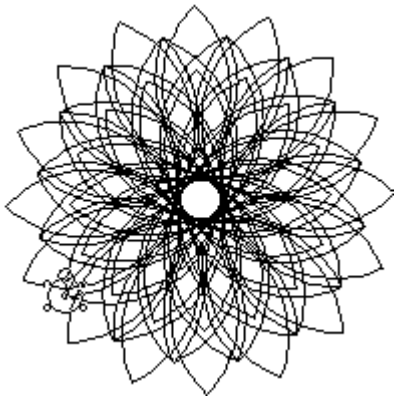

**Figura 4 - Execução código Flor Lotus**

### 3.1.2 SCRATCH

O Scratch é uma ferramenta concebida no Media Laboratory do Massachusetts Institute of Technology (MIT), sendo a mais recente de uma longa linhagem de ferramentas que se iniciou com a criação da linguagem de programação LOGO por Seymour Papert. Inspirada na linguagem LOGO, mas pretendendo ser mais simples

e mais intuitiva uma vez que utiliza a metodologia de “clicar e arrastar” através de blocos, a linguagem de programação Scratch utiliza diversos tipos de mídias, possibilitando a criação de histórias interativas, animações, jogos, músicas e o compartilhamento dessas criações na Internet (ANDRADE et alii, 2013).

Ainda de acordo com ANDRADE, uma preocupação dos autores do Scratch é que a criança não tenha que se preocupar com erros de sintaxe. Desta forma, a atenção da criança volta-se apenas para a lógica necessária para o desenvolvimento da atividade que ela deseja realizar, ou seja, os blocos de programação são concebidos para poderem se encaixar apenas de forma que faça sentido sintaticamente.

> Algumas das potencialidades do Scratch são: liberdade de criação, criatividade, associada a programas abertos e sem limitações do software; comunicação e partilha, associada à aprendizagem, facilitada pelas ferramentas Web que permitem a publicação direta; aprendizagem de conceitos escolares, partindo de projetos livres e não escolarizados; manipulação de media, permitindo a construção de programas que controlam e misturam gráficos, animação, texto, música e som; partilha e colaboração, a página da Internet do Scratch fornece informação, permite a partilha, pode-se experimentar os projetos de outros, reutilizar e adaptar imagens, divulgar as nossas criações e tem como meta desenvolver uma cultura de aprendizagem e partilha em torno do Scratch; integração de objetos do mundo físico, o Scratch pode integrar objetos exteriores de vários tipos (PINTO, 2010).

A figura 5 – Tela Scratch, mostra a tela principal do Scratch. No lado superior esquerdo, encontram-se os grupos de comandos para construção dos programas, sendo divididos em: movimento, aparência, som, caneta, controle, sensores, operadores e variáveis. No lado superior direito responsável pelo feedback, onde os usuários veem seus códigos sendo executados.

O grupo de **Movimento** é responsável pelas funções de movimentação e rotação do objeto que será selecionado (traje). A Aparência serve para substituição dos trajes (imagem gráfica do objeto), para fazer desaparecer e aparecer e para descrever textos que aparecerão ao jogador. Os comandos de **Som** têm a finalidade de importar músicas para as atividades elaboradas. A **Caneta** é responsável pelos traços deixados quando o objeto é movimentado, podendo alterar cor, espessura e tonalidade. O bloco denominado **Controle** possui comandos pré-definidos, responsáveis pelas estruturações lógicas de conexão (conetivos lógicos) entre outros comandos. Os **Sensores** servem para perceber cores, distâncias, posições

no plano cartesiano, e normalmente são combinados com outros comandos. O conjunto formado pelos **Operadores** apresenta os operadores lógicos (maior, menor, igual, e, ou, não), as quatro operações, sorteio de números, as funções trigonométricas e suas inversas, as funções exponenciais na base E e base 10 e suas inversas e a função modular.

A região central da tela é o local onde a “programação” do jogo é feita. Os blocos de comando são conectados como se fossem peças do jogo Lego. Os elementos que compõem essa seção são Comandos, Trajes e Sons. O item **Comandos** é destinado à construção da sintaxe da programação. Nesta seção os blocos de comando acima citados são combinados formando a estrutura principal do jogo. Os **Trajes** são as imagens que o objeto apresenta, podendo ser importadas de pastas do Computador ou construídas no próprio programa Scratch. A seção de **Sons** é responsável pela importação e escolha dos sons que serão utilizados no jogo.

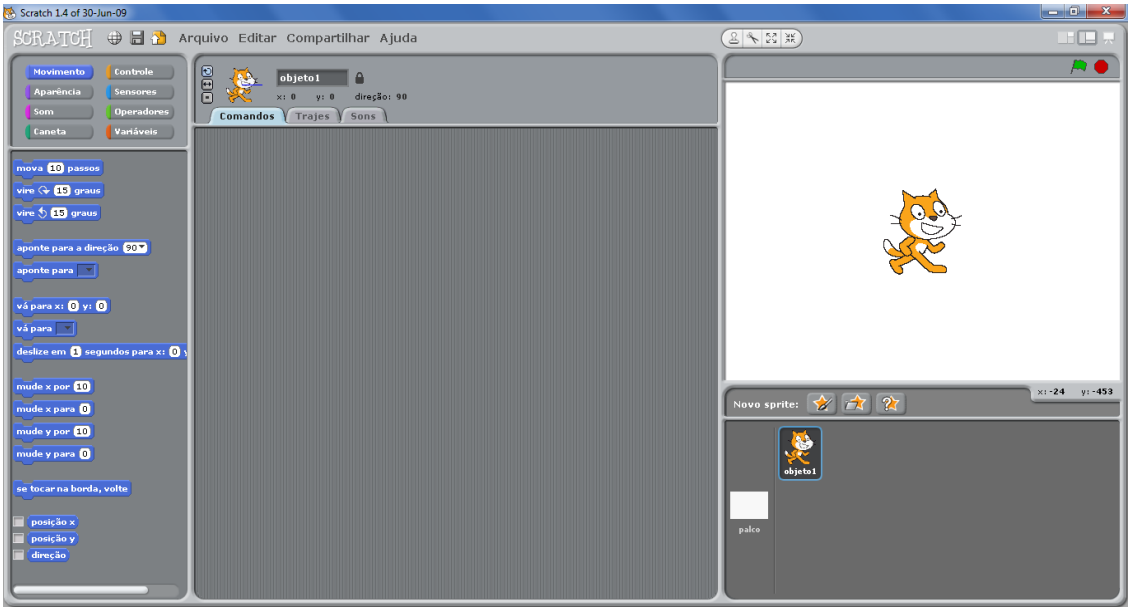


**Figura 5 – Tela Scratch**

As Figuras 6 e 7 mostram códigos construídos no ambiente Scratch, o código apresentado na Figura 6 faz com que o traje repita o movimento de 10 passos e vire 15 graus no sentido horário. O código da Figura 7 repete 10 vezes o procedimento aumentando ou diminuindo o tamanho do traje dependendo do valor escolhido aleatoriamente, quando o valor é maior que 5, o traje aumenta seu tamanho para 200% dizendo “Estou grande!”, quando o valor é menor que 5, o traje diminui seu tamanho para 50% dizendo “Estou pequeno!”.


**Figura 6 – Código Scratch**

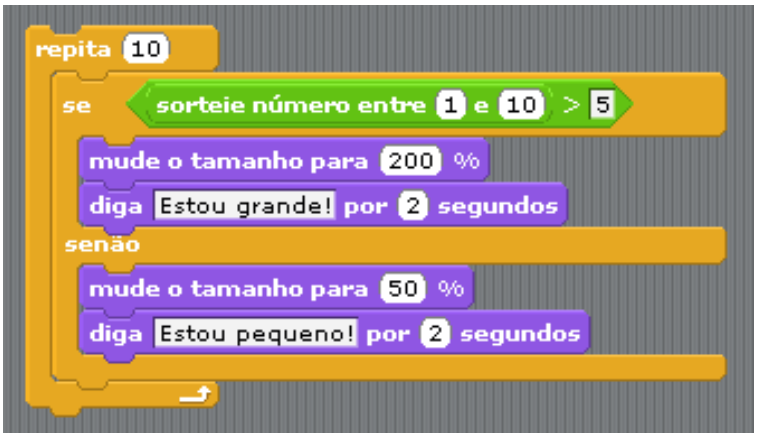


**Figura 7 – Código Scratch 2**

### 3.1.3 CODE COMBAT

O Code Combat é um ambiente lúdico, criado em 2013, baseado na web, que ensina os respectivos usuários a programar em uma linguagem, sendo esta escolhida pelo jogador no início das atividades, desafiando-os a escrever pedaços de “código” para o jogo.

A interface é simples. Ao lado do campo de simulação, há um editor de código. O ambiente oferece orientações sobre como escrever o código, quais funções utilizar e qual a sintaxe correta para que os personagens atinjam cada objetivo e avancem de fase.

Contendo dois módulos: um módulo para programadores principiantes e outro para programadores experientes:

- O módulo principiante: Dirigido aos que estejam aprendendo os primeiros passos na programação e ensina os conceitos básicos.
- O módulo experiente: Sendo um módulo multi-jogador em que cada jogador desenvolve o seu código/programa para derrotar o inimigo, saindo vitorioso o

jogador com código/programa mais eficiente. Aperfeiçoando seus códigos conforme as batalhas com outros jogadores e dependendo das suas vitórias vai subindo e descendo no “ranking”.

A Figura 08 – Menu de Seleção, permite o usuário escolher seu Avatar e a Linguagem de Programação a ser aprendida.


**Figura 8 – Menu de Seleção**

Ao definir seu Avatar e a linguagem de programação a ser aprendida, será carregada a primeira atividade. Onde a missão do usuário é a escrita de códigos na linguagem de programação escolhida, de forma a movimentar o Avatar até a gema (pedra de diamante).

Após a escrita do código, o ambiente permite a execução do mesmo para identificar possíveis falhas de escrita ou lógica. Com o código completado, o usuário deve

submetê-lo de modo a finalizar a missão. A Figura 9 – Atividade Code Combat exibe a tela da primeira atividade do ambiente.

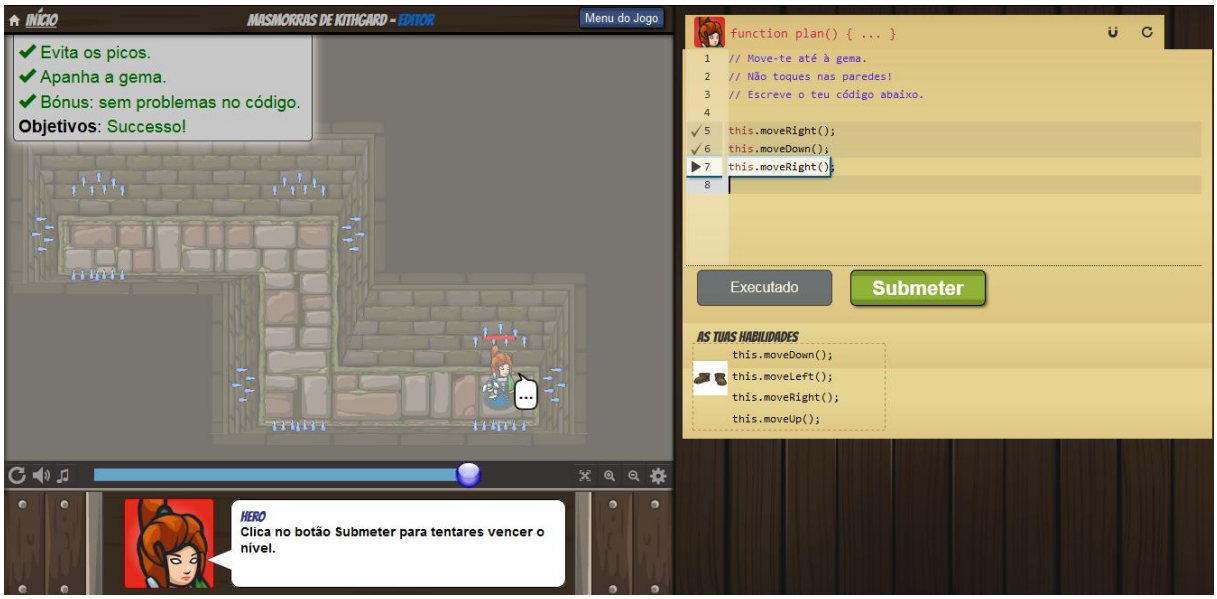


**Figura 9 – Atividade Code Combat**

Para a solução desta atividade o código construído foi:

this.moveRight();

this.moveDown();

this.moveRight();

O ambiente Code Combat permite o aprendizado das seguintes linguagens de programação:

- Python;
- Java Script;
- Coffee Script;
- Clojure;
- Lua;
- Io.

# 4. INTELIGÊNCIA ARTIFICIAL

A inteligência artificial é uma das ciências mais recentes. Teve início após a Segunda Guerra Mundial e, atualmente, abrange uma enorme variedade de subcampos, desde áreas de uso geral, como aprendizado e percepção, até tarefas específicas como jogos de xadrez, demonstração de teoremas matemáticos, criação de poesia e diagnóstico de doenças. A inteligência artificial sistematiza e automatiza tarefas intelectuais e, portanto, é potencialmente relevante para qualquer esfera da atividade intelectual humana. Nesse sentido, ela é um campo universal (RUSSELL; NORVIG, 2004).

## 4.1 Definição

Existem várias maneiras de se definir o campo de inteligência artificial, dentre eles podemos destacar:

- É o estudo de conceitos que permitem os computadores (ou as máquinas) serem inteligentes (WINSTON, 1988).
- É o estudo de como fazer os computadores realizarem coisas que, no momento, as pessoas fazem melhor (RICH & KNIGHT, 1994).
- É uma forma de produção de conhecimento não natural, utilizando-se principalmente meios computacionais para sua produção. É uma área da ciência da computação e engenharia da computação que busca a produção de dispositivos e métodos computacionais que possibilite, por meio de simulações, serem produzidos conhecimentos a partir da solução de problemas por computadores (MANZANO, 2012).

A Inteligência Artificial ao longo do tempo seguiu quatro linhas de pensamento:

- **Sistemas que pensam como seres humanos**: “O novo e interessante esforço para fazer os computadores pensarem... máquinas com mentes, no sentido total e literal.” (HAUGELAND, 1985).
- **Sistemas que atuam como seres humanos**: “A arte de criar máquinas que executam funções que exigem inteligência quando executadas por pessoas.” (KURZWEIL, 1990).

- **Sistemas que pensam racionalmente**: “O estudo das faculdades mentais pelo seu uso de modelos computacionais.” (CHARNIAK; MCDERMOTT, 1985).
- **Sistemas que atuam racionalmente**: “A Inteligência Computacional é o estudo do projeto de agentes inteligentes.” (POOLE et al., 1998).

## 4.2 Agentes Inteligentes

De acordo com RUSSELL e NORVIG (2004), agente é simplesmente algo que age (a palavra agente vem do latino *agere*, que significa “fazer”). No entanto, espera-se que um agente computacional tenha outros atributos que possam distingui-lo de meros "programas", tais como operar com controle autônomo, perceber seu ambiente, persistir por um período de tempo prolongado, adaptar-se às mudanças e ser capaz de assumir metas de outros.

Ainda segundo RUSSELL e NORVIG, agente racional é aquele que age para alcançar o melhor resultado ou, quando há incerteza, o melhor resultado esperado.

De acordo com HAYES-ROTH (1995), agentes inteligentes executam continuamente três funções: percebem as condições dinâmicas do ambiente, agem alterando as condições do ambiente e raciocinam de modo a interpretar percepções, resolver problemas, fazer inferências e determinar ações.

### 4.2.1 AGENTE ANALISADOR DE CAMINHO – AAC

Para a identificação automática do menor caminho nas atividades desenvolvidas, houve a necessidade da construção de um Agente Analisador de Caminho. Esse agente é responsável pela identificação do melhor caminho em cada atividade do Ambiente *NewProg*.

De maneira a usar esse agente, foram analisadas diversas formas de construí-lo. Ao final, optou-se pela utilização do Método Branch-and-Bound.

#### 4.2.1.1 Método Branch-and-Bound

O método ramificar-e-podar (branch-and-bound B&B) é um algoritmo exato de enumeração implícita, ou seja, um método que, embora não teste explicitamente todas as soluções possíveis, garante a melhor solução possível. Para isso, utiliza-se da geração de uma árvore de nós que representam subproblemas do problema principal. Trata-se do conhecido conceito de dividir e conquistar (NEMHAUSER, RINNOY KAN & TODD, 1989), ou seja, otimizar pequenos subconjuntos considerando o resultado total ao invés de tratar integralmente o conjunto total de soluções. O método é composto de duas operações básicas:

- Branching: dividir o problema principal em subproblemas menores de modo a facilitar a análise, eliminando soluções inviáveis, sem comprometer a integridade do campo de soluções.
- Bounding: eliminar soluções de baixa qualidade através de comparações com limitantes. São comumente utilizados dois tipos de limitantes: superior e inferior. Num problema de minimização, o limitante superior é um valor conhecido e viável da função objetivo, não necessariamente o valor ótimo, que tem o papel de servir como parâmetro para avaliar soluções obtidas, ou seja, soluções com valores superiores ao limitante superior são descartadas por se tratarem de soluções piores do que a atualmente conhecida. Por sua vez, o limitante inferior, em um problema de minimização, é uma estimativa da função objetivo tendo-se como base a solução parcial até então obtida. Nota-se que o limitante inferior, nesses casos, é sempre menor ou igual do que o valor da função objetivo, já que seu cálculo é baseado em um subconjunto da solução, enquanto que a função objetivo é calculada considerando-se a solução completa. Assim sendo, é possível eliminar soluções que tenham limitantes inferiores piores do que os atuais limitantes superiores conhecidos.

As divisões ou gerações de ramos da árvore são realizadas recursivamente como mostra a Figura 10:

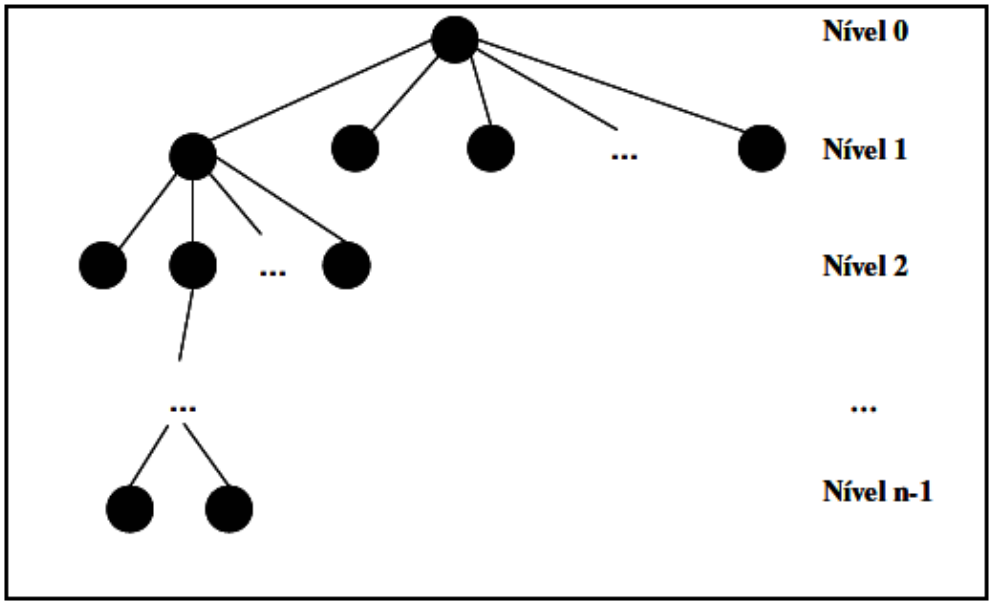


**Figura 10 – Árvore de busca do Branch-and-Bound**

De modo a aperfeiçoar a busca, optou-se pelo Método de Busca Ramificar-e-Podar.

Segundo WINSTON (1988), o esquema ramificar-e-podar funciona do seguinte modo: durante a busca há muitos caminhos incompletos, competindo por maior consideração. O caminho mais curto é estendido em um nível, criando tantos novos caminhos incompletos quantos forem os ramos. Esses novos caminhos são então examinados juntos com os antigos remanescentes e, de novo, o mais curto é estendido. Isto se repete até que o destino seja alcançado por algum caminho. Como o caminho mais curto foi sempre o escolhido para ser estendido, o caminho que primeiro alcançar o destino é, certamente, o ótimo.

WINSTON observa que há uma deficiência na explicação, na forma em que foi dada acima. O último passo alcançar o destino pode ser suficientemente longo para tornar a solução suposta mais extensa do que um ou mais caminhos incompletas. Poderia acontecer que apenas um passo bem pequeno estendesse um dos caminhos incompletos ao ponto de solução. Para ter certeza de que não é assim, é necessária uma condição de finalização melhor. Em vez de finalizar quando um caminho é encontrado, finalizar quando o caminho incompleto mais curto for maior do que o caminho completo mais curto.

De modo a conduzir uma busca ramificar-e-podar, WINSTON (1998) definiu os seguintes passos:

1. Forme uma fila de caminhos parciais. Faça a fila inicial consistir de um caminho de comprimento zero, passo zero, desde o nó raiz para lugar nenhum.
2. Até que a fila esteja vazia ou que o destino tenha sido alcançado, determine se o primeiro caminho na fila alcança o nó destino.
    2.1. Se o primeiro caminho alcançar o nó destino, não faça nada.
    2.2. Se o primeiro caminho não alcançar o nó destino:
        2.2.1. Remova o primeiro caminho da fila.
        2.2.2. Forme novos caminhos a partir do caminho removido com a extensão de um passo.
        2.2.3. Acrescente os novos caminhos à fila.
        2.2.4. Ordene a fila por custo acumulado até aí, com o menor custo na frente.
3. Se o nó destino foi achado, proclame sucesso; senão registre fracasso.

Os passos acima precisam ainda considerar a observação de WINSTON registrada acima: o algoritmo somente deve terminar quando não houver mais caminhos na fila de caminhos parciais menores do que o caminho achado.

### 4.2.2 AGENTE AVALIADOR E MOTIVADOR – AAM

Com o intuito de motivar cada criança a construir novas soluções, foi criado um Agente que monitora cada solução desenvolvida pela criança, comparando-a com o melhor caminho encontrado pelo AAC e apresentando à criança um *feedback* adequado para cada solução construída.

No contexto das atividades do Ambiente *NewProg*, existe a possibilidade de haver mais de uma caminho com a melhor solução, visto que cada passo dado no ambiente tem o mesmo peso. Desse modo, considera-se o custo de cada caminho e não apenas o caminho em si. O AAC encontra o melhor caminho e o custo dele, repassando esse custo para o AAM, de modo que o mesmo compare o custo da solução do usuário com o custo da solução encontrado pelo AAC. Assim é possível saber se a solução encontrada pelo usuário é a melhor possível para a realização da atividade.

Os *feedback* são personalizados de acordo com a solução criada pelo usuário, sendo eles:

- Solução Melhor Caminho = Parabéns, você conseguiu pegar o alimento com o menor caminho!
- Solução Acerto sem Melhor Caminho = Parabéns, você conseguiu, tente pegar o alimento com o menor caminho!
- Solução Errada = Que pena, tente novamente!

# 5. REQUISITOS DO AMBIENTE

O objetivo principal deste capítulo é fazer uma ligação entre a fundamentação e os objetivos traçados. Além dos requisitos teóricos, consideramos os levantamentos feitos com professores, por meio da pesquisa etnográfica.

## 5.1. A origem

Segundo TOREZANI et alii (2013), pesquisas enfatizando o uso da programação com crianças têm crescido surpreendentemente, todavia os ambientes existentes como ToonTalk, Squeak Etoys, Scratch, permitem apenas a criação e execução de atividades, não sendo possível um acompanhamento detalhado da evolução dos alunos.

Desta forma, a evolução das atividades não pode ser vista pelos educadores. Nos ambientes acima, os educadores têm acesso apenas às conclusões das atividades, impossibilitando-os de ver quais as principais dificuldades enfrentadas pelos aprendizes.

A utilização de recursos que estimulam a criatividade dos estudantes, e ao mesmo tempo, permita o acompanhamento detalhado pelos educadores é essencial a uma educação que atenda às exigências deste século.

## 5.2. Requisitos desejáveis no ambiente

A seguir, são apresentados os requisitos funcionais do sistema, assim como uma breve descrição dos mesmos:

- Simplicidade – o ambiente de ser claro e objetivo, de maneira a facilitar a interação dos usuários;
- Agrupamento de Atividades – as atividades devem ser agrupadas de acordo com seu contexto, facilitando o encontro das mesmas;
- Escolha de Avatares – possibilitar aos usuários escolher o Avatar que mais lhe agrade, de modo a motivá-lo a fazer as atividades;

- Tela Limpa – o ambiente deve conter apenas informações relevantes para seu usuário de maneira a não poluir o visual do mesmo;
- Acompanhamento das Atividades – o ambiente deve conter recursos, onde os educadores possam acompanhar seus alunos individualmente;
- Cadastramentos Diversos – o ambiente deve prover ferramentas para o cadastro de diferentes perfis de usuários;
- Executor de Atividades – deve conter a inclusão e exclusão de comandos para a realização das atividades;
- Tamanho da Tela – a tela deve ter dimensões adaptáveis;
- Ajuda – o ambiente deve ter um “help” adequado à faixa etária que o utilizará;
- Ambientação – de maneira a facilitar o aprendizado do ambiente, ele deve prover uma ferramenta que permita às crianças interagirem com o mesmo e se ambientarem na utilização dele;
- Editor de Atividades – de maneira a tornar o ambiente adaptável e rico em atividades, ele deve conter um editor que permita aos educadores criarem novas atividades;
- Simulador – de maneira a testar atividades antes de cria-las, o ambiente deve prover recursos de simulação delas, permitindo testes pelos educadores.

## 5.3. Interface

Sendo o publico alvo constituído prioritariamente por crianças, a interface do ambiente é essencial precisa ter uma boa aceitação para permitir a obtenção dos melhores resultados.

> A interface do software, de modo geral, cuida da ligação do usuário com a máquina, e tem a incumbência de facilitar o processo de comunicação. No software educativo tem que se apresentar com muita qualidade e de modo a facilitar o processo ensino aprendizagem fazendo com que o aluno atinja o objetivo proposto com mais facilidade, rapidez, sem medo e ansiedade, conseguindo assim aumentar o seu desempenho (CRISTÓVÃO, 1997).

Na tabela 2 – Itens considerados para a criação da Interface do Ambiente proposto, há um paralelo entre alguns itens propostos por CRISTÓVÃO (1997) para avaliação

da interface de softwares educacionais e a interface do ambiente proposto nesta dissertação.

**Tabela 2 - Itens considerados para a criação da Interface do NewProg**

| Item de avaliação | Descrição (Cristóvão, 1997) | Ambiente proposto |
|---|---|---|
| **Facilidade de aprendizado** | O primeiro contato com a interface deve ser tranquilo e sem problemas. É recomendável que na função de ajuda do software apareça um tutorial de forma a permitir que o aluno experimente diferentes estratégias de solução associadas ao seu problema. | A ajuda do ambiente terá uma linguagem simples adequada à faixa etária que o utilizará. Seus ícones e botões terão aspectos visuais bastante atrativos do ponto de vista da criança. |
| **Facilidade de uso** | Após o aprendizado da interface, o nível de dificuldade exigido para manter o seu uso não deve ser alto. O software deve permitir a independência de uso e ação, sem assistência humana. O aluno deve conseguir lembrar com facilidade o manuseio da interface, após algum tempo (1 semana, por exemplo) sem ter tido contato com o ambiente. | O ambiente mantém o mesmo padrão da interface para todos os recursos de modo a não ser necessária uma nova ambientação. |

| | | |
|---|---|---|
| **Ortogonalidade de Conceitos** | O software deve oferecer uma linguagem sintaticamente homogênea e ter comportamento semelhante em situações semelhantes, solicitando do usuário ações similares para tarefas similares. Cada ação, comando ou objeto deve receber um nome que seja consistente durante todo o diálogo entre o usuário e o software. | A linguagem do ambiente será a linguagem de sala de aula o que facilitará na obtenção da consistência desejada. |
| **Tolerância a erros de operação do usuário** | O sistema em hipótese alguma pode se 'enrolar' caso o usuário cometa algum erro na operação da interface. Na repetição do erro, a situação deve ser reapresentada de maneira mais adequada, ou então uma ajuda pode aparecer a fim de que o aluno consiga completar ou reiniciar a tarefa interrompida pelo erro. | As mensagens serão sempre de incentivo ao repensar, com dicas que levem a descobrir e corrigir o erro por meio da reflexão e da análise cognitiva. |
| **Organização estrutural** | As categorias e regras de classificação devem ser usadas de maneira adequada, a fim de que o aluno possa perceber como estão sendo relacionadas. As sequências de telas devem ser organizadas de forma análoga a tarefas semelhantes já conhecidas | Ao manter o mesmo padrão da interface para todos os recursos, o aprendiz facilmente percebe a relação entre os conceitos e relações apresentados para ele nas telas do ambiente. |

| | | |
|---|---|---|
| **Feedback** | O feedback aqui tratado não é o de conteúdo, mas o reflexo do software perante a ação do usuário. O feedback associado à interface deve orientar o usuário com relação a sua participação na tarefa, e informar o efeito que sua ação teve sobre o sistema mostrando todas as consequências possíveis da mesma, e ainda exibindo o novo estado do sistema bem como a nova localização do usuário. | O ambiente trabalha com diversos *feedbacks*, permitindo ao usuário acompanhar cada passo de sua solução e identificar os possíveis erros. |
| **Layout da tela** | A tela não deve ser muito densa nem muito vazia, e também deve manter relações lógicas e funcionais entre todos os seus itens. As informações importantes devem ficar na tela e com facilidade de acesso, mas sem a tornar demasiadamente congestionada. As cores apresentadas, em geral devem aparecer em número reduzido, produzindo assim um efeito agradável. Se mal utilizadas podem até provocar sentimentos agressivos ou de apatia, e pode chamar atenção para algo não muito importante. | O visual do ambiente foi trabalhado junto com a pesquisa etnográfica, de maneira a atender da melhor forma seus usuários. No decorrer da pesquisa onde o ambiente foi posto em utilização, foram feitas varias correções visuais indicadas pelos próprios utilizadores. Mantendo um ambiente limpo e informativo, com cores alegres e motivantes. |

| | | |
|---|---|---|
| **Ícones** | Devem transmitir o que representam sem complicações. Eles devem ser grandes e fáceis de serem selecionados com o movimento do cursor. | Todos os ícones foram produzidos/selecionados de maneira a serem entendidos facilmente. |
| **Memorização** | As informações da interface que precisam ser memorizadas devem aparecer em número pequeno, para que o aluno possa desempenhar adequadamente as tarefas atuais e subsequentes que são objetivos de fato do software. | Foi reduzida ao máximo a necessidade de memorização, foi trabalhado junto com os ícones maneiras de ajudar neste quesito. |
| **Histórico** | O software deve disponibilizar para o professor, ou mesmo para o aluno, um histórico de todas as suas ações no decorrer da aula, bem como conseguir armazenar de forma permanente (em disco) o registro de utilização de um usuário juntamente com o seu histórico. Se for usado em rede, o professor, numa estação, deve poder verificar o andamento de todos os alunos através dos históricos pessoais. | O histórico é um dos principais recursos do ambiente. É com ele que o educador acompanha o desenvolvimento dos seus aprendizes, podendo com isto traçar estratégias para estimular o crescimento cognitivo de seus alunos. |

| | | |
|---|---|---|
| **Vocabulário e regionalidade** | Não deve apresentar erros gramaticais, mesmo que estes já sejam pertencentes à linguagem cotidiana, pois se o software é educativo deve usar a linguagem formal. O vocabulário usado no software deve ser adequado ao nível do aluno que irá operá-lo. As gírias, os termos e os costumes regionais apresentados no software devem ser adequados ao local onde a escola se situa. Se possível o software deve usar um vocabulário neutro em relação à regionalidade. | O vocabulário utilizado na concepção do ambiente foi neutro, usando uma linguagem adequada ao seu público alvo, sem uso de gírias ou termos e costumes regionais. |

# 6. ANÁLISE E PROJETO DO AMBIENTE

Este capítulo apresenta o projeto detalhado do ambiente, considerando desde seus objetivos até a modelagem do software proposto nesta dissertação.

## 6.1. Objetivos do Ambiente

O principal objetivo do ambiente é propiciar um espaço onde possam ser criadas e executadas atividades para o ensino de programação, oferecendo ao professor condições para acompanhar individualmente seus alunos por meio de um histórico das ações executadas.

## 6.2. Funções Importantes

As principais funções do ambiente serão subdividas em três partes, sendo funções orientadas aos aprendizes, funções pedagógicas orientadas a educadores e funções administrativas.

### 6.2.1 FUNÇÕES ORIENTADAS AOS APRENDIZES

As principais funções destinadas aos aprendizes podem ser vista na Figura 11 – Tela Execução de Atividades, sendo possível a programação das soluções das atividades.

- **Andar 1 vez** – ao clicar em uma seta pertencente a esta função, o Avatar andará uma vez de acordo com o sentido (seta) escolhido;
- **Andar 2 vezes** – ao clicar em uma seta pertencente a esta função, o Avatar andará duas vezes de acordo com o sentido (seta) escolhido;
- **Andar 3 vezes** – ao clicar em uma seta pertencente a esta função, o Avatar andará três vezes de acordo com o sentido (seta) escolhido;
- **Andar 4 vezes** – ao clicar em uma seta pertencente a esta função, o Avatar andará quatro vezes de acordo com o sentido (seta) escolhido;

- **Remover última ação** – permitindo apagar o último comando da sequência construída.
- **Informações** – ao clicar nessa função será exibida uma tela com informações sobre a atividade;
- **Exibir Bordas** – ao clicar nessa função, o tabuleiro da atividade será dividido em quadrados, permitindo ao aprendiz identificar melhor o caminho;
- **Menor caminho** – esta função esta desabilitada no início da atividade, após cinco tentativas a mesma é habilitada, permitindo ao aprendiz visualizar o menor caminho (melhor solução).


**Figura 11 – Tela Execução de Atividades**

### 6.2.2 FUNÇÕES PEDAGÓGICAS ORIENTADAS A EDUCADORES

As principais funções pedagógicas são listadas a seguir:

- **Cadastra alunos** – função destinada ao cadastro de alunos;
- **Visualizador de Atividades Realizadas** – função destinada ao acompanhamento de atividades realizadas pelos alunos, permitindo o acompanhamento por atividades ou por alunos;

- **Construtor de Atividades** – permite a construção de atividades de programação ou de atividades de programação com matemática;
- **Listar Alunos** – permite a visualização e exclusão de alunos cadastrados.

### 6.2.3 FUNÇÕES ADMINISTRATIVAS

As principais funções administrativas são listadas a seguir:

- **Cadastrar Usuários** – permite o cadastro de novos usuários com perfis Administrador ou Professor;
- **Listar Atividades** – permite a visualização e exclusão de atividades cadastradas;
- **Listar Usuários** - permite a visualização e exclusão de usuários cadastrados.

## 6.3. Modelagem

A especificação de modelos abstratos tem como principais objetivos a compreensão, através da simplificação, e a comunicação dos grupos ligados ao projeto, que podem “falar” a mesma língua, gerando um entendimento mútuo, com a possibilidade de uma cooperação mais eficaz (CONALLEN 2003).

Segundo CONALLEN (2002), o Web Application Extension (WAE) é um processo de desenvolvimento de software Web baseado na interatividade, em que as fases planejamento, requisitos, análise, projeto, implementação, teste e avaliação são repetidos até que cada incremento esteja ajustado às tarefas que desempenhará e seja implantado, conforme apresenta a Figura 12.

Desenvolver, sem uma técnica que especifique um modelo da aplicação web e suas particularidades, implica provavelmente em prejuízo a integridade e a rastreabilidade do programa. A utilização da WAE permite representar os elementos importantes do ponto de vista da modelagem. Segundo CONALLEN (2003), a WAE é representada por estereótipos que permitem que uma nova interpretação semântica possa ser adicionada a um elemento do modelo.

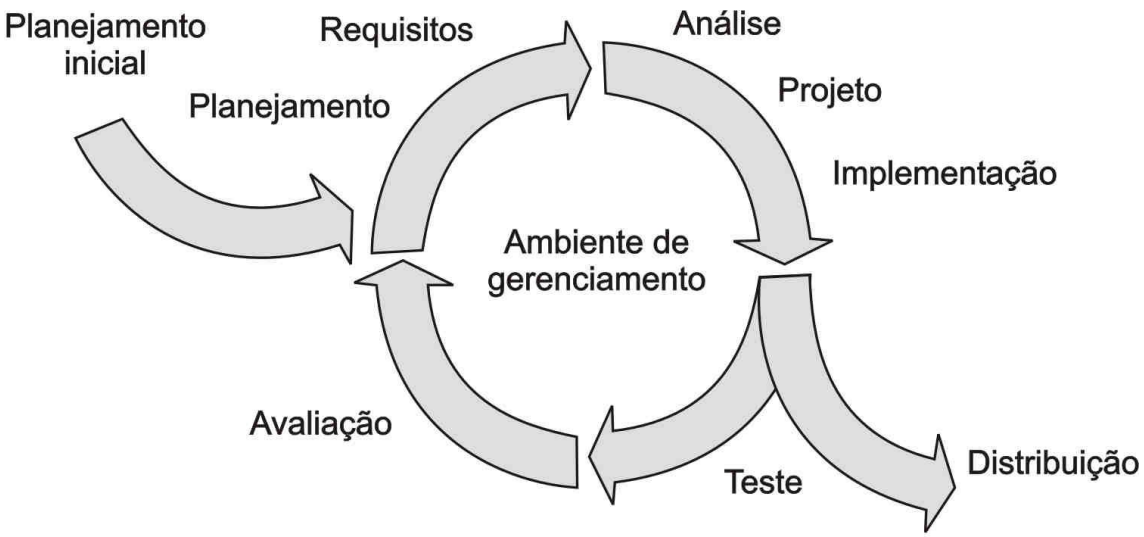


**Figura 12 - Processo iterativo (MARACCI, 2009)**

### 6.3.1 CASOS DE USO

As Figuras 13, 14 e 15 apresentam os casos de uso do administrador, do educador e do aprendiz, respectivamente.

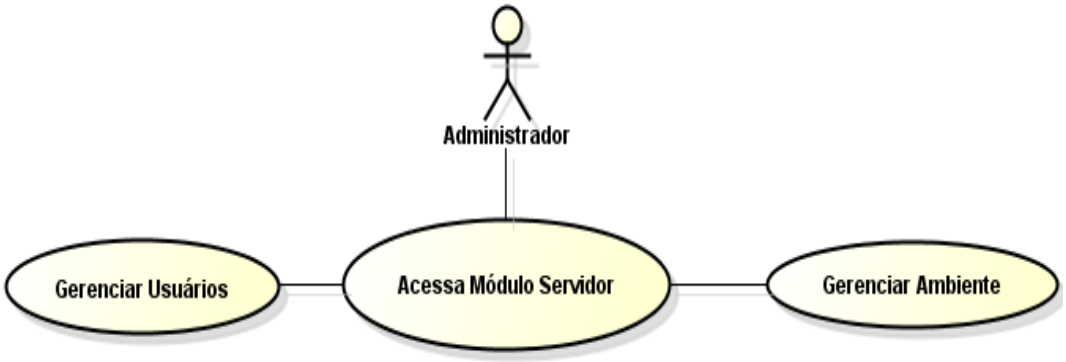


**Figura 13 – Caso de Uso Administrador**

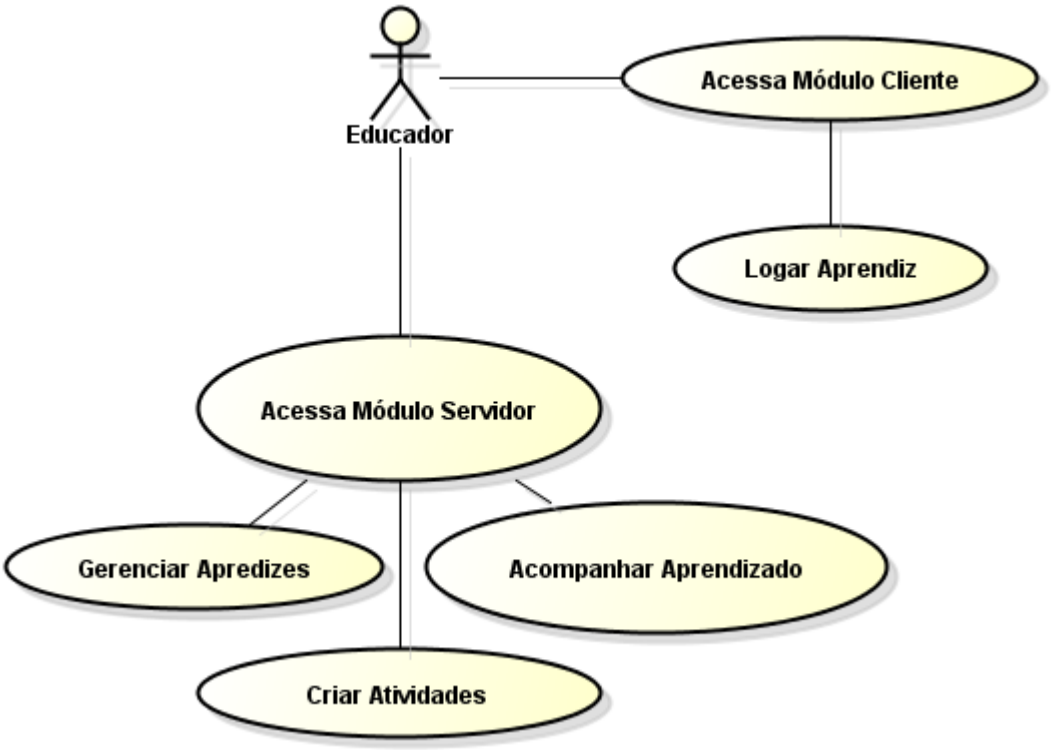


**Figura 14 – Caso de Uso Educador**


**Figura 15 – Caso de Uso Aprendiz**

### 6.3.2. DIAGRAMA DE CLASSES

O modelo conceitual (diagrama de classes) representa a estrutura dos dados do sistema, onde temos informações sobre Aprendizes, Turmas, Atividades, Soluções das Atividades, Educadores, Administradores, representados por meio das classes e seus relacionamentos. O ambiente proposto nessa dissertação necessita de sete classes e que se relacionarão segundo o diagrama de classes exibido na Figura 16.

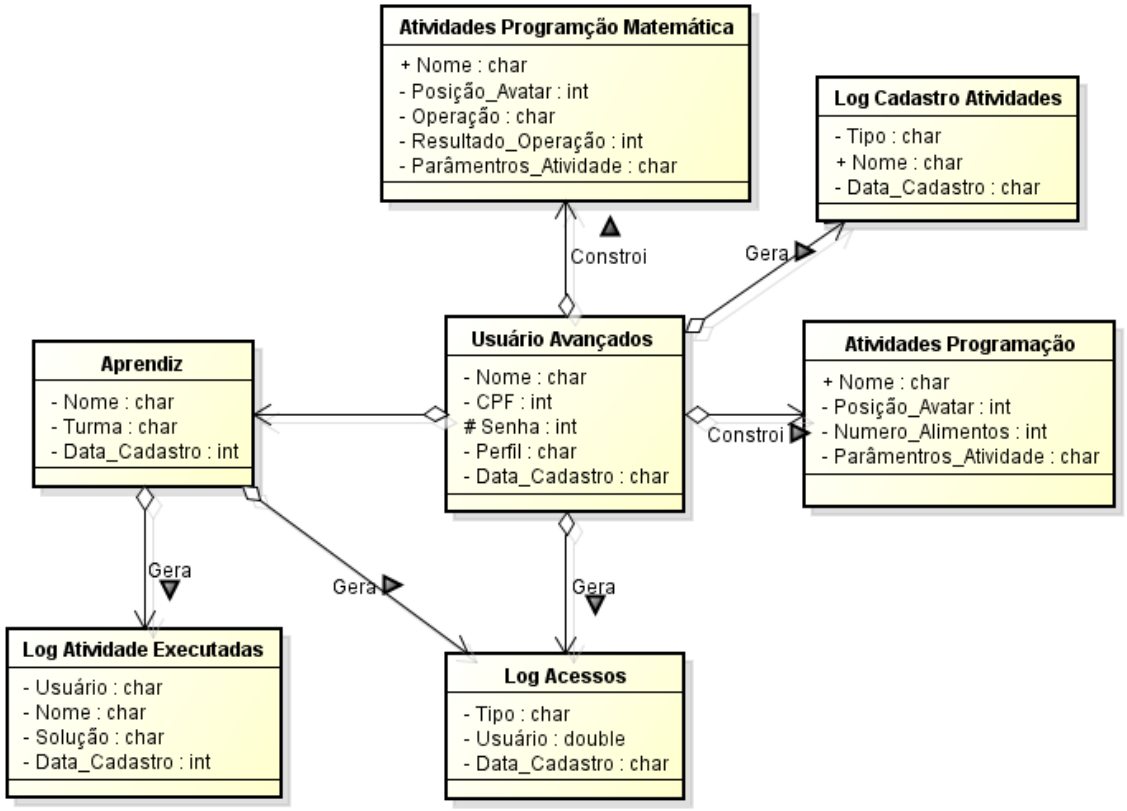


**Figura 16 – Diagrama de Classes**

### 6.3.4. MODELO NAVEGACIONAL

O modelo navegacional gerado a partir da notação WAE representa a estrutura de interação entre os módulos do ambiente, por meio das páginas da ambiente (aplicação) e seus relacionamentos. As Figuras 17, 18, 19 e 20, mostram a estrutura do ambiente e seus relacionamentos.

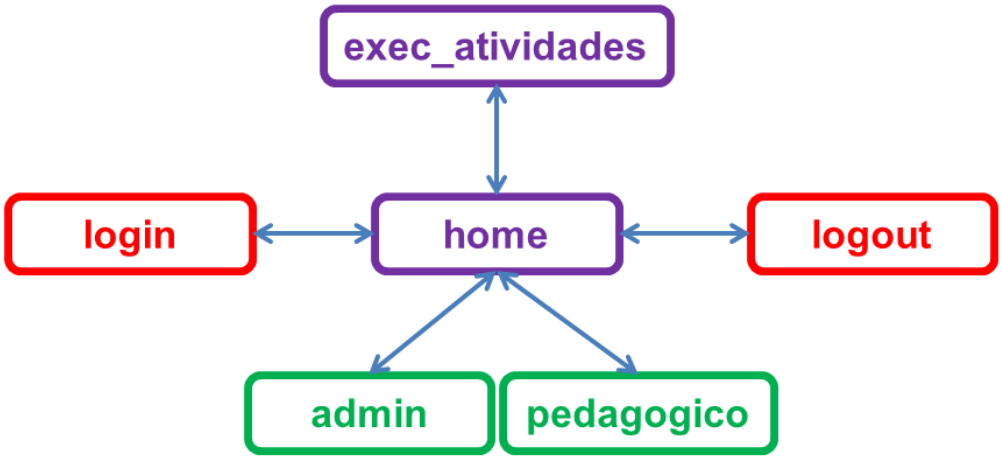


**Figura 17 – Modelo Navegacional 1**

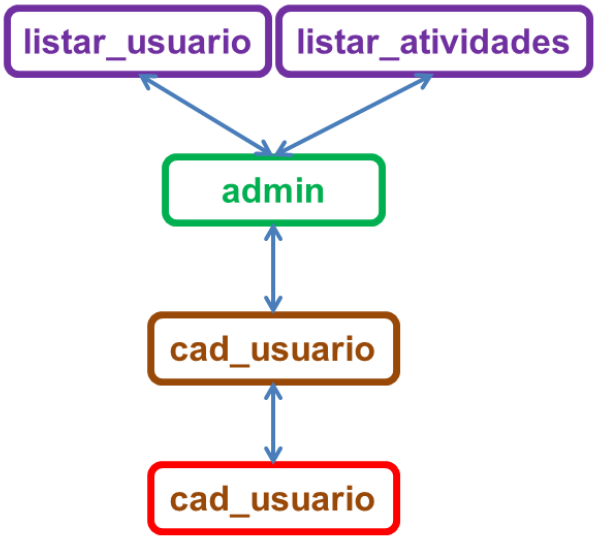


**Figura 18 – Modelo Navegacional 2**

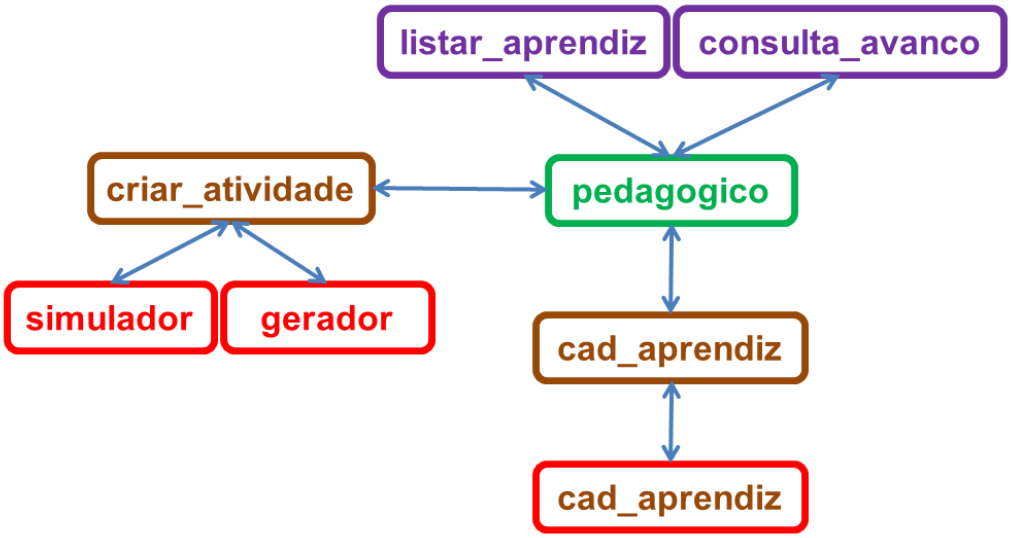


**Figura 19 – Modelo Navegacional 3**

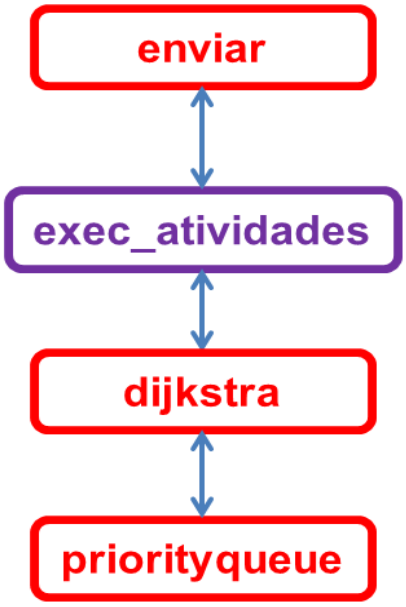


**Figura 20 – Modelo Navegacional 4**

Legenda Figuras 17, 18, 19 e 20.

- Cor **verde** são páginas Cliente (client page) representa componentes no cliente (browser).
- Cor **marrom** são páginas HTML Form, representa componentes no cliente, para envio de informações.
- Cor **vermelha** são páginas Servidoras (server page), representa componentes no servidor.
- Cor **roxa** são páginas Servidoras e Cliente ao mesmo tempo, representa componentes no servidor e no cliente.

A Figura 17 demonstra a interação da interface inicial do ambiente, onde, a página "home" interage com as páginas:

- Página login – Permite aos usuários fazerem a autenticação;
- Página logout – Permite a saída do ambiente;
- Página exec_atividades – Permite a execução das atividades do ambiente;
- Página admin – Permite a utilização de recursos administrativos;
- Página pedagogico – Permite a utilização de recursos pedagógicos.

A Figura 18 demostra as ligações da página "admin" com as páginas:

- Página listar_usuario – Responsável pela listagem e exclusão de usuários;
- Página listar_atividades – Responsável pela listagem e exclusão de atividades;
- Página cad_usuario – Responsável pelo cadastro de novos usuários.

A Figura 19 demonstra as ligações da página “pedagogico" com as páginas:

- Página listar_aprendiz – Responsável pela listagem e exclusão de aprendizes;
- Página consulta_avanco – Responsável pelas funções que permitem o acompanhamento dos aprendizes;
- Páginas criar_atividade, simulador e gerador – Responsáveis pela construção e simulação de atividades;
- Página cad_aprendiz – Responsável pelo cadastro de novos aprendizes.

A Figura 20 demonstra as ligações da página “exec_atividades” com as páginas:

- Página enviar – Responsável pelo recebimento dos dados a serem armazenados no banco de dados;
- Páginas dijkstra e priorityqueue – Responsáveis pela analise do menor caminho.

# 7. IMPLEMENTAÇÃO DO AMBIENTE E AVALIAÇÃO DOS RESULTADOS

O ambiente foi totalmente implementado, buscando-se criar um protótipo que tivesse todas as funções, de modo que pudesse ser avaliado por profissionais que trabalham com educação nas séries iniciais do ensino fundamental.

De modo a atender todos os objetivos do projeto, a implementação do ambiente necessitou de um conjunto de linguagens de programação.

## 7.1. Linguagens utilizadas na implementação

Esta seção apresenta as linguagens utilizadas para o desenvolvimento do ambiente NewProg e um breve descrição das mesmas.

### 7.1.1 LINGUAGEM PHP

O PHP (HyperText Processor) é uma linguagem desenvolvida a partir de um projeto pessoal de Rasmus Lerdorf, inicialmente como uma pequena linguagem de Script para adicionar alguma lógica ao processamento de formulários de seu site. De 1994 para cá, tem estado em constante evolução e conquistado um espaço significativo no mercado, com grande penetração em ambiente Unix (LERDORF *apud* CAMPANA, 2009).

Segundo CAMPANA (2009), é uma alternativa rápida, barata e confiável, disponível em várias plataformas, inclusive Microsoft/Intel, e pode ser uma solução interessante para aplicações de pequeno a médio porte, integrando-se a vários servidores WEB como o IIS (Microsoft) e Apache, acessando uma grande variedade de bancos de dados.

Dado o contexto atual de mercado e a vasta documentação, justifica-se sua escolha para trabalhar do lado do servidor, além das características mencionadas, as seguintes características abaixo influenciaram na escolha:

- Código Aberto: Código fonte do PHP está disponível;

- Custo Zero: É gratuito, necessitando apenas baixá-lo e instalá-lo;
- Multiplataforma: Compatível com uma gama de sistemas operacionais, alguns deles, Unix, Linux, Windows;
- Eficiência: Necessita de poucos recursos do servidor, sendo o PHP como módulo nativo do servidor WEB, evita chamadas externas, tornando-o mais eficiente;
- Acesso a Bancos de Dados: Podemos acessar diretamente os principais bancos de dados utilizados;
- Capacidade de ler informação do padrão XML, processamento de arquivos (leitura e gravação), manipulação de variáveis complexas, utilização de funções e classes, geração de códigos.

### 7.1.2 LINGUAGEM HTML

Segundo PACIEVITCH (2014), o HTML é uma linguagem de marcação utilizada para desenvolvimento de sites, sendo constituída de códigos que delimitam conteúdos específicos, segundo uma sintaxe própria, criado em 1991, por Tim Berners-Lee, no CERN (European Council for Nuclear Research) na suíça. Inicialmente o HTML foi projetado para interligar instituições de pesquisa próximas, e compartilhar documentos com facilidade. Em 1992, foi liberada a biblioteca de desenvolvimento WWW (World Wide Web), uma rede de alcance mundial, que junto com o HTML proporcionou o uso em escala mundial da WEB.

### 7.1.2 LINGUAGEM CSS

O Cascading Style Sheets (CSS) é uma "folha de estilo" composta por “camadas” e utilizada para definir a apresentação (aparência) em páginas da internet que adotam para o seu desenvolvimento linguagens de marcação (como XML, HTML e XHTML). O CSS define como serão exibidos os elementos contidos no código de uma página da internet e sua maior vantagem é efetuar a separação entre o formato e o conteúdo de um documento (PEREIRA, 2009).

Os principais benefícios de seu uso são:

- Controle do layout de vários documentos a partir de uma simples folha de estilos;
- Maior precisão no controle do layout;
- Aplicação de diferentes layouts para servir diferentes mídias;
- Emprego de variadas, sofisticadas e avançadas técnicas de desenvolvimento.

### 7.1.2 LINGUAGEM JAVASCRIPT

Segundo CAMPANA (2009), JavaScript é uma poderosa linguagem de criação de scripts baseada em objetos; os programas em JavaScript podem ser incorporados diretamente nas páginas Web que usam HTML. Quando combinado com o modelo de objeto de documento definido por um navegador Web, JavaScript permite criar conteúdo em Dynamic HTML (chamada DHTML que é um tipo de HTML dinâmico) e aplicativos interativos do lado cliente da Web. A sintaxe de JavaScript baseia-se nas linguagens de programação muito utilizadas, como C, C++ e Java, o que a torna familiar e fácil para programadores experientes. Além disso, o JavaScript é uma linguagem de criação de scripts interpretada, fornecendo um ambiente flexível de programação, de fácil aprendizado para novos programadores.

## 7.2. Facilitações adotadas

Para concluir um protótipo que possuísse todos os recursos funcionais para testes e que abrangesse os objetivos descritos anteriormente, alguns requisitos funcionais tiveram que ser implementados de modo diferente. Por exemplo, em alguns casos, o envio de dados foi feito com o método POST em outros casos, com o método GET.

Facilidades adotadas na construção:

- Padronização do layout do ambiente;
- Padronização do tabuleiro da atividade com 36 células;
- Construção de editores para dois modelos de atividades;
- Feedback padronizado com dois ou três modelos dependendo da atividade;
- Aproveitamento de códigos próprios e de outros desenvolvedores;

- Execução do Agente Analisador de Caminho no lado Servidor;
- Desativação do botão direito do *mouse*;
- Filtro de entrada de dados em alguns campos, evitando erros;
- Utilização de SESSIONS para autenticação dos usuários.

### 7.3. Dificuldades encontradas

Durante o desenvolvimento do protótipo foram encontradas diversas dificuldades referentes à programação que são descritas a seguir:

- Devido à utilização de mais de uma linguagem houve a necessidade de maior número de pesquisas e tempo para o aprendizado das mesmas, tomando mais de dois anos para o desenvolvimento das pesquisas e implementação.
- Falta de padronização na execução dos códigos pelos navegadores webs, o que exigiu a modelagem e remodelagem de vários recursos. Dessa forma, houve um grande consumo de tempo, de maneira a atender a um número maior de navegadores.
- Construção do Agente Analisador de Caminho, consumindo boa parte do desenvolvimento. Inicialmente foi tentado implementa-lo para execução do lado cliente e, após dezenas de tentativas, ele foi implementado do lado Servidor, enviando informações para o lado cliente.
- Desenvolvimento de um sistema de autenticação diferenciado, de modo que o usuário de perfil aprendiz (crianças pequenas) não precisarem recordar (memorizar) usuário e senha. A autenticação implementada permite a um usuário avançado (perfil administrador ou educador) autenticar os aprendizes.

### 7.4. Tela Inicial

A Tela Inicial do Ambiente NewProg é ajustável ao tipo de usuário, habilitando recursos (ferramentas) extras dependendo do perfil do mesmo. Ao abrir o ambiente sem que um usuário esteja autenticado (modo anônimo) é habilitada a **Janela de Login**. A sequência de Figuras 21, 22, 23, 24 e 25, exibe a Tela Inicia de acordo com o perfil de usuário.


**Figura 21 – Tela Inicial perfil Anônimo**


**Figura 22 – Tela Inicial perfil Aprendiz**

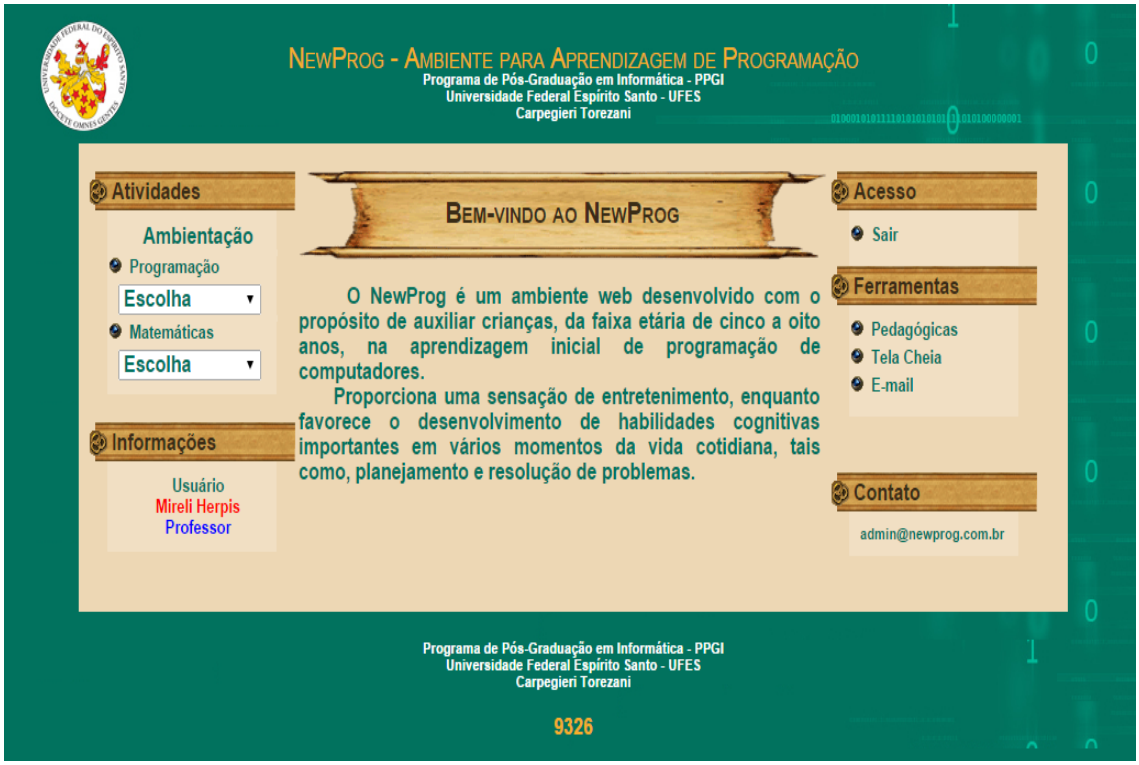


**Figura 23 – Tela Inicial perfil Professor**


**Figura 24 – Tela Inicial perfil Administrador**


**Figura 25 – Tela Inicial Descritiva**

## 7.5. Login do Sistema

De modo a habilitar ou restringir recursos, o ambiente necessita da autenticação do usuário, não obstante, o Login no Sistema permite o arquivamento (histórico identificado) de ações executadas pelos usuários.

### 7.5.1 LOGIN TERCEIRIZADO

De maneira a ter um histórico identificado de cada usuário é necessário seu Login com suas credenciais. Devido ao público alvo do ambiente ser constituído por crianças do ensino infantil e fundamental, houve a necessidade de um sistema de autenticação diferenciado, visto que o público alvo não recordaria de seu usuário e senha. A Figura 26 exibe a tela de autenticação com as devidas explicações, demonstrando com é feito o Login Terceirizado.

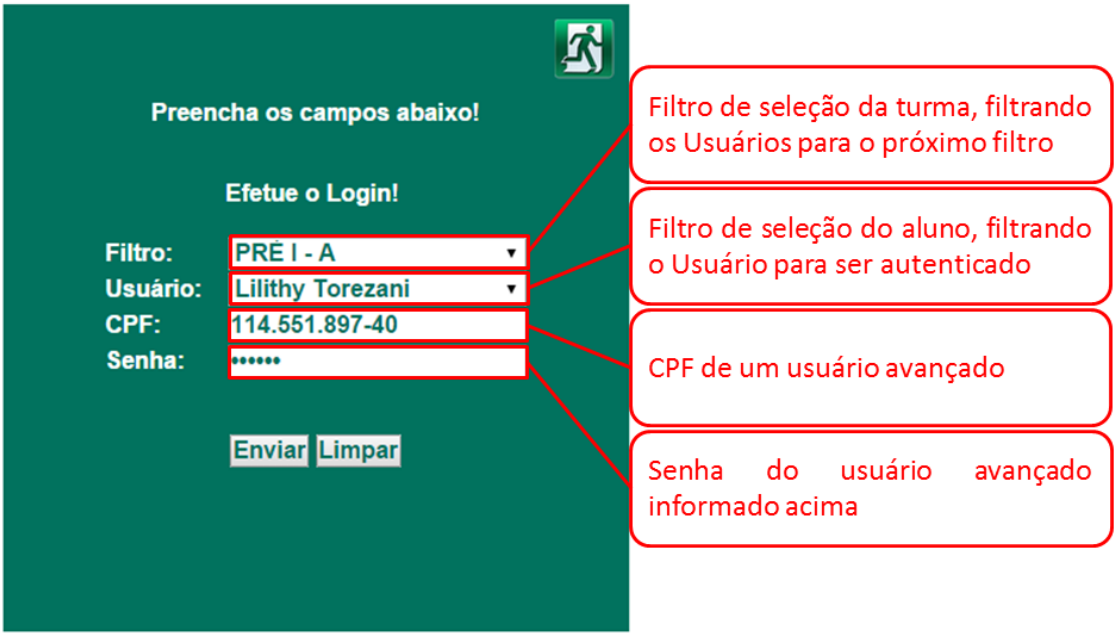


**Figura 26 – Tela Login**

## 7.6. Funções Administrativas

Nesta seção, serão apresentadas as funções administrativas implementadas no ambiente NewProg.

### 7.6.1. CADASTRO DE USUÁRIOS

A Figura 27 exibe a tela de Cadastro de Usuários, para acessá-la é necessário escolher a Função Cadastrar Usuários no Menu das Funções Administrativas. Para cadastra um novo usuário é necessário informar os seguintes campos:

- Tipo de Usuário (Perfil Administrador ou Professor);
- Nome do Usuário;
- CPF do Usuário;
- Senha do Usuário;
- Repetir Senha do Usuário.

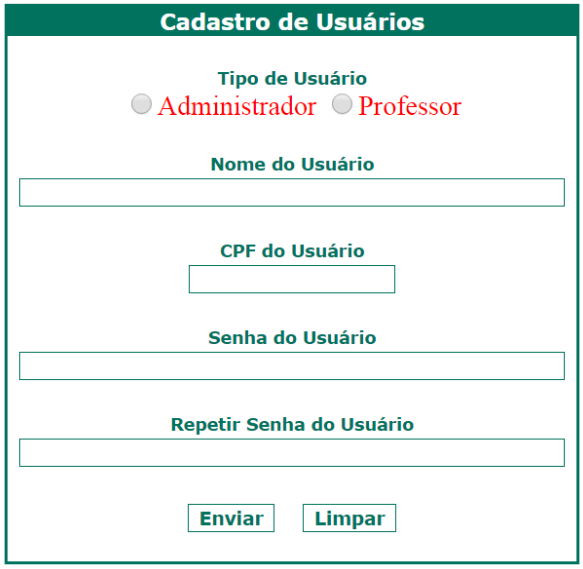


**Figura 27 – Tela Cadastro de Usuários**

### 7.6.2. LISTAR ATIVIDADES

A Figura 28 exibe a tela de Listar Atividades, para acessá-la é necessário escolher a Função Listar Atividades, no Menu das Funções Administrativas. Ao executar esta função, são listadas todas as atividades existentes no ambiente, sendo possível **Visualiza-las** ou **Excluí-las**.


**Figura 28 – Tela Listar Atividades**

### 7.6.3. LISTAR USUÁRIO

A Figura 29 exibe a tela de Listar Usuários, para acessá-la é necessário escolher a Função Listar Usuários no Menu das Funções Administrativas. Ao executar esta função será listados todos os usuários avançados existentes no ambiente, sendo possível **Excluí-los**.

Lista de Usuários Cadastrados

| Nome | CPF | Perfil | Data Cadastro | Excluir |
|---|---|---|---|---|
| Carpegieri Torezani | Ocultado | Admin | 2013-08-05 21:45:19 | Excluir |
| Crediné Silva de Menezes | Ocultado | Admin | 2013-09-17 23:37:31 | Excluir |
| Mireli Herpis | Ocultado | Prof | 2014-10-09 18:26:43 | Excluir |
| Orivaldo De Lira Tavares | Ocultado | Admin | 2013-09-05 10:52:48 | Excluir |

**Figura 29 – Tela Listar Usuários**

## 7.7. Funções Pedagógicas

Nesta seção, serão apresentadas as funções pedagógicas implementadas no ambiente NewProg.

### 7.7.1. CADASTRO DE ALUNOS

A Figura 30, exibe a tela de Cadastro de Alunos, para acessá-la é necessário escolher a Função Cadastrar Alunos no Menu das Funções Pedagógicas. Para cadastra um novo aluno é necessário informar os seguintes campos:

- Nome do Aluno;
- Turma (PRÉ I, PRÉ II, [1º, 2º, 3º, 4º, 5º, 6º, 7º, 8º, 9º Ano]);
- Módulo (A, B, C, D, E, F, G, H ou I).


**Figura 30 – Cadastro de Alunos**

## 7.7.2. VISUALIZADOR DE ATIVIDADES REALIZADAS

A Figura 31 exibe a tela de Visualizador de Atividades Realizadas. Para acessá-la é necessário escolher a Função Visualizador de Atividades Realizadas no Menu das Funções Pedagógicas. O visualizador de avanços permite os seguintes filtros:

- Filtra por Atividades;
- Filtra por Aluno.

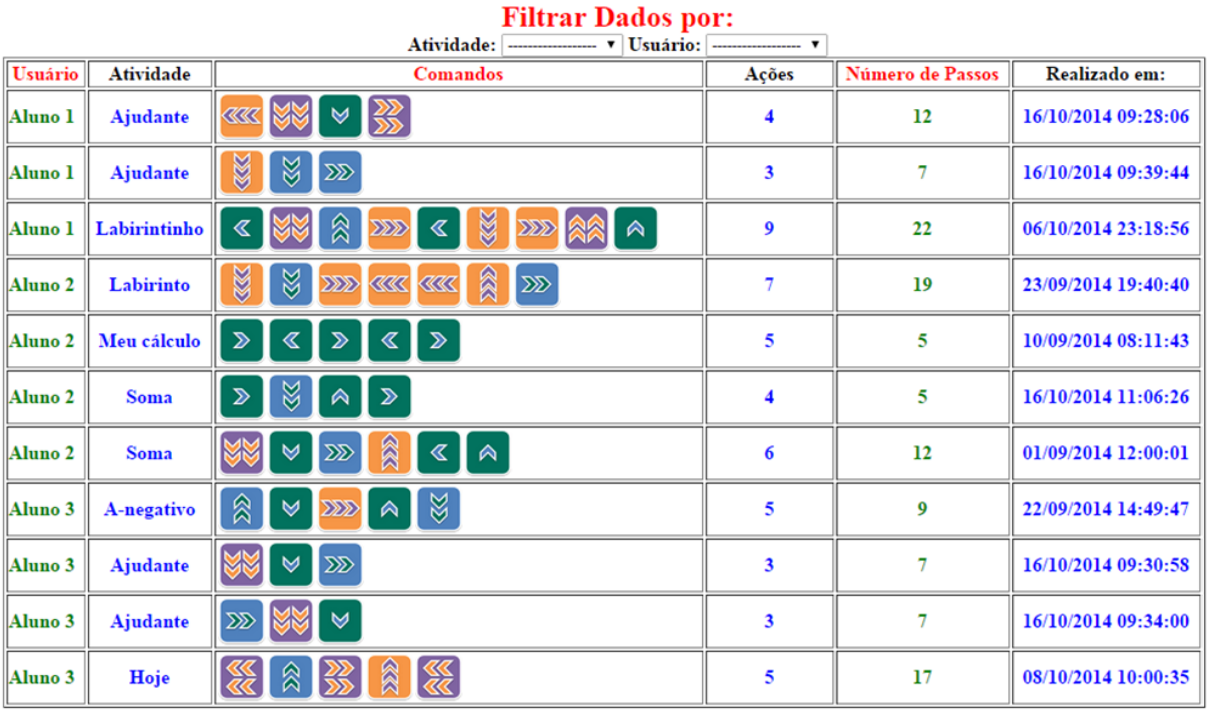


**Figura 31 – Visualizador de Atividades Realizadas**

### 7.7.3. LISTA DE ALUNOS

A Figura 32 exibe a tela Lista de Usuários Cadastrados. Para acessá-la é necessário escolher a Função Listar Alunos no Menu das Funções Pedagógicas. Ao executar esta função, serão listados todos os usuários aprendizes existentes no ambiente, sendo possível **Excluí-los**.

| Lista de Usuários Cadastrados | | | |
|---|---|---|---|
| Nome | Turma | Data Cadastro | Excluir |
| Antoni Amburgo | 1º Ano - A | 2013-08-06 10:49:41 | Excluir |
| Antônio Carlos | PRÉ II - A | 2013-08-06 10:48:51 | Excluir |
| João Lucas | 2º Ano - A | 2013-08-06 10:50:26 | Excluir |
| Julio Ferreira | 1º Ano - A | 2013-08-06 10:49:04 | Excluir |
| Lilithy Torezani | PRÉ I - A | 2014-10-09 18:22:31 | Excluir |
| Pedro Lucas | PRÉ II - A | 2013-08-06 10:48:32 | Excluir |

**Figura 32 – Lista de Alunos Cadastrados**

### 7.7.4. EDITORES DE ATIVIDADES - EANewProg

Nesta seção, serão apresentados os Editores de Atividades implementados no ambiente NewProg, denominados EANewProg, exibindo-os e detalhando a criação das atividades nos mesmos.

#### 7.7.4.1 Editor de Atividades de Programação

A Figura 33 exibe a tela do editor de atividade de programação. Por meio dele é possível à edição de atividades com alimentos e com barreiras, permitindo ao educador criar uma gama de atividades diferentes para a aprendizagem de programação.

O designer do editor foi moldado de acordo com a entrevista etnográfica, de modo a atender as exigências/necessidades dos educadores pesquisados.

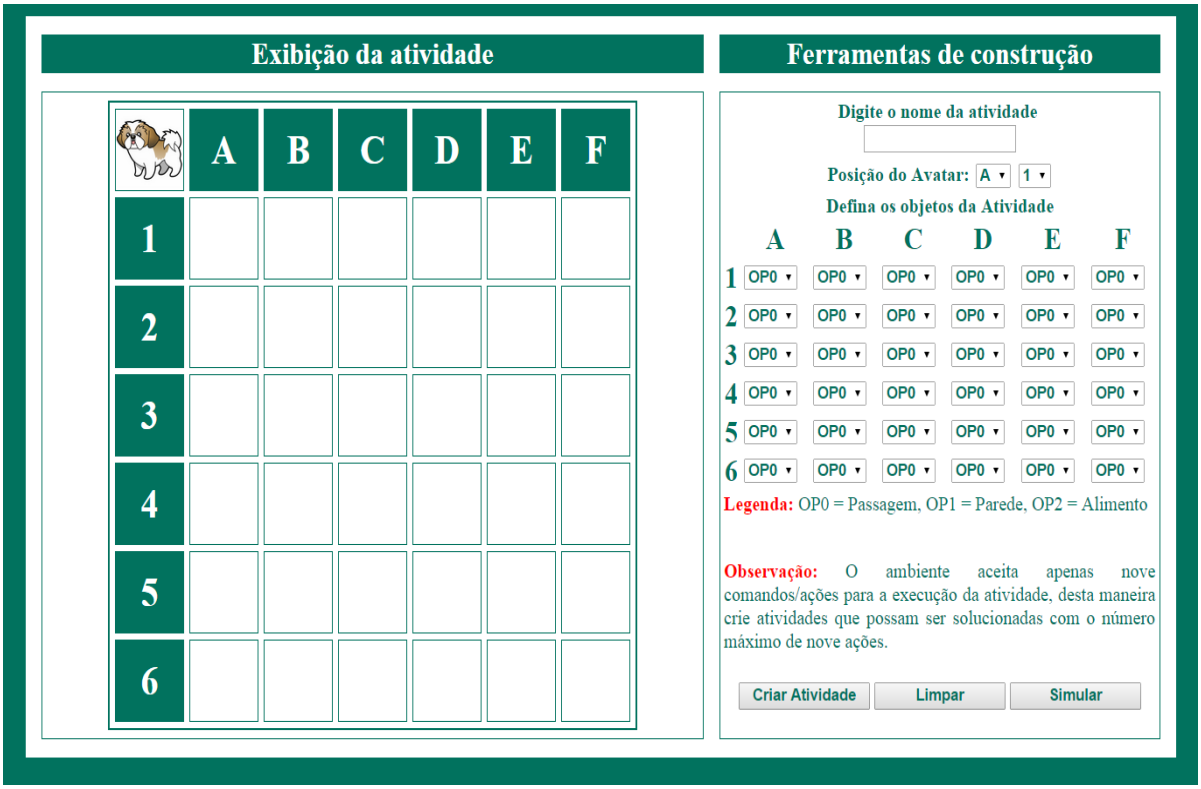


**Figura 33 – Editor de Atividades de Programação**

O editor de atividades é composto de duas janelas, sendo elas, **Janela Exibição da Atividade,** para permitir ao usuário a visualização de como será a atividade após sua construção, e a **Janela Ferramentas de Construção,** destinada às configurações da atividade que permite definir os critérios/informações da atividade.

Para a construção de uma atividade são necessários os seguintes requisitos:

- **Nome da atividade** – Caso seja informado um nome já existente, a atividade será criada com o nome informado seguido do dia e hora de sua criação;
- **Posição do Avatar** – Posição inicial do Avatar na atividade, informando o mesmo através de dois campos, definindo linha e coluna;
- **Alimento**(s) – Sendo obrigatório no mínimo um e no máximo três alimentos;
- **Barreira**(s) – As barreiras são opcionais, sendo criadas de acordo com a necessidade do educador.

#### 7.7.4.1 Editor de Atividades de Programação com Matemática

Durante a realização da pesquisa etnográfica, fomos indagados da possibilidade da construção de atividades multidisciplinares, dado a relação de proximidade entre informática e matemática, definimos a construção de um editor que possibilitasse a construção de atividades que abrangessem as duas áreas.

A Figura 34 exibe a tela do Editor de Atividades de Programação com Matemática, através do mesmo é possível à edição de atividades multidisciplinares, permitindo ao educador criar uma gama de atividades diferentes para o ensino aprendizado de programação e matemática.

Construído com os mesmos padrões do Editor de Atividades de Programação, este editor foi construído de maneira a simplificar ao máximo as interações com o mesmo, permitindo que atividades fossem criadas com poucos segundos.

Para a construção de uma atividade de programação com matemática são necessários os seguintes requisitos:

- **Nome da atividade** – Caso seja informado um nome já existente, a atividade será criada com o nome informado seguido do dia e hora de sua criação;
- **Posição do Avatar** – Posição inicial do Avatar na atividade, informando o mesmo através de dois campos, definindo linha e coluna;
- **Missão a ser realizada** – Operação matemática a ser respondida;
- **Números das Células** – Preenchimento de todas as trinta e seis células do tabuleiro (Caso alguma célula não seja preenchida com algum valor, será atribuída à mesma o valor zero).


**Figura 34 – Editor de Atividades de Programação com Matemática**

## 7.8. Atividades

O ambiente apresentado nesta dissertação permite a construção e execução de dois tipos de atividades, sendo elas **Atividades de Programação** e **Atividades de Programação com Matemática**. As atividades são acessadas pelos campos Seletores de Atividades informados na Figura 25.

### 7.8.1. SELEÇÃO AVATARES

Em um primeiro momento não foi levantada a ideia da construção de um ambiente que permitisse a utilização de avatares diferentes, sendo o mesmo construído apenas com um Avatar (Cachorrinho).

Em uma das avaliações do ambiente, um aprendiz que possuía um trauma com cachorros, afastou-se imediatamente do computador. Após uma conversa com o mesmo, ficou evidente a necessidade da criação do ambiente com a possibilidade de escolha de Avatares. A partir daí, foi implementada a possibilidade de escolher

um Avatar entre oito disponíveis. A Figura 35 exibe a janela de escolha dos Avatares, sendo necessária sua escolha em cada execução das atividades.

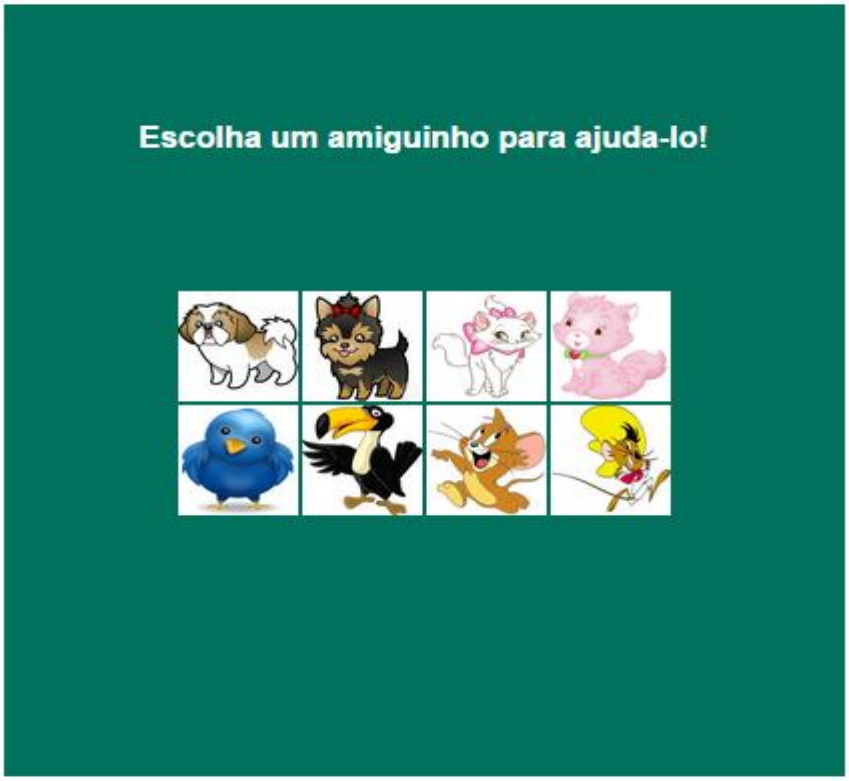


**Figura 35 – Janela escolha do Avatar**

### 7.8.2. MENU PRINCIPAL

Os dois modelos existentes de atividades no ambiente NewProg compartilham do mesmo modelo de interação, logo compartilham o mesmo Menu Principal, a Figura 36 exibe sua interface.

O Menu Principal é composto das seguintes funções:

- **Restaurar** – Inicia a atividade, desde a escolha do Avatar;
- **Início** – Permite voltar à tela inicial do ambiente;
- **Andar 1 vez** – Ao clicar em uma seta pertencente a esta função, o Avatar andará uma vez de acordo com o sentido(seta) escolhido;
- **Andar 2 vezes** – Ao clicar em uma seta pertencente a esta função, o Avatar andará duas vezes de acordo com o sentido(seta) escolhido;
- **Andar 3 vezes** – Ao clicar em uma seta pertencente a esta função, o Avatar andará três vezes de acordo com o sentido(seta) escolhido;

- **Andar 4 vezes** – Ao clicar em uma seta pertencente a esta função, o Avatar andará quatro vezes de acordo com o sentido(seta) escolhido;
- **Remover última ação** – Permitindo apagar o último comando da sequência de ícones (solução) editada pelo aprendiz.


**Figura 36 – Menu Principal das Atividades**

### 7.8.3. MENU SECUNDÁRIO

Devido algumas peculiaridades das atividades, o Menu Secundário foi construído com informações e recursos diferentes, ambos serão apresentados a seguir.

#### 7.8.3.1 Menu Secundário Atividades de Programação

Este Menu é destinado a orientação do aprendiz durante a execução das atividades de programação, a Figura 37 – Menu Secundário Atividades de Programação, exibe todas suas funções.


**Figura 37 – Menu Secundário Atividades de Programação**

O Menu Secundário Atividades de Programação é composto das seguintes funções:

- **Ajuda** – Composta com duas ou três funções dependendo da atividade, sendo elas:
    1. **Informações** – Ao clicar nessa função será exibida uma tela com informações sobre a atividade;
    2. **Exibir Bordas** – Ao clicar nessa função, o tabuleiro da atividade será dividido em quadrados, permitindo ao aprendiz identificar melhor o caminho;
    3. **Menor caminho** – Está função está desabilitada no início da atividade, após cinco tentativas a mesma é habilitada, permitindo ao aprendiz visualizar o menor caminho (melhor solução).
- **Número Ajudas** – Exibe número de ajudas solicitadas (esta função conta às vezes que o recurso Exibir Bordas foi utilizado);
- **Número de Erros** – Exibe o número de erros que aconteceram na execução do código programado (Os erros são movimentos impossíveis dentro do tabuleiro, sendo eles: bater na lateral do tabuleiro ou bater nas barreiras dentro do tabuleiro);

- **Número de Passos** – Exibe o número de passos que o Avatar andou dentro do tabuleiro;
- **Número de Ações** – Exibe o número de Ações programadas pelo Aprendiz;
- **Resultado** – Exibe o *feedback* personalizado apresentado ao aprendiz, de acordo com a solução programada por ele;
- **Informações** – Exibe os dados do usuário autenticado.

7.8.3.2 Menu Secundário Atividades de Programação com Matemática

Este Menu é destinado a orientação do aprendiz durante a execução das atividades de programação com matemática, a Figura 38 – Menu Secundário Atividades de Programação com matemática, exibe todas suas funções.


**Figura 38 – Menu Secundário Atividades de Programação com Matemática**

O Menu Secundário Atividades de Programação com Matemática é composto das seguintes funções:

- **Missão** – Composto com duas funções, sendo elas:
    1. **Informações** – ao clicar nessa função será exibida uma tela com informações sobre a atividade;

2. **Resolver a Equação** – Exibe a equação a ser solucionada.

- **Número de Passos** – Exibe o número de passos que o Avatar andou dentro do tabuleiro;
- **Número de Ações** – Exibe o número de Ações programadas pelo Aprendiz;
- **Resultado** – Exibe o resultado da soma dos valores do caminho percorrido pelo Avatar no tabuleiro;
- **Informações** – Exibe os dados do usuário autenticado no ambiente.

### 7.8.4. PROGRAMAÇÃO E EXECUÇÃO DAS ATIVIDADES

Dado o contexto do ambiente e seu público alvo, a programação dentro do ambiente é feita através de uma sequência de comandos visuais (setas), para resolver um problema (atividade). Um mesmo problema pode ser solucionado por uma gama de sequências (programas) diferentes, sendo possível a utilização ou não de repetição, ou caminhos diferentes para solucionar o problema.

De modo a exemplificar o grande número de possibilidades de soluções para uma mesma atividade/problema, escolhemos a atividade denominada Ajudante criada por um educador, a Figura 39 – Atividade Ajudante exibe a tela inicial da atividade (como é apresentada ao aprendiz), a Tabela 3 – Soluções para atividade Ajudante exibe algumas das possíveis soluções para essa atividade.

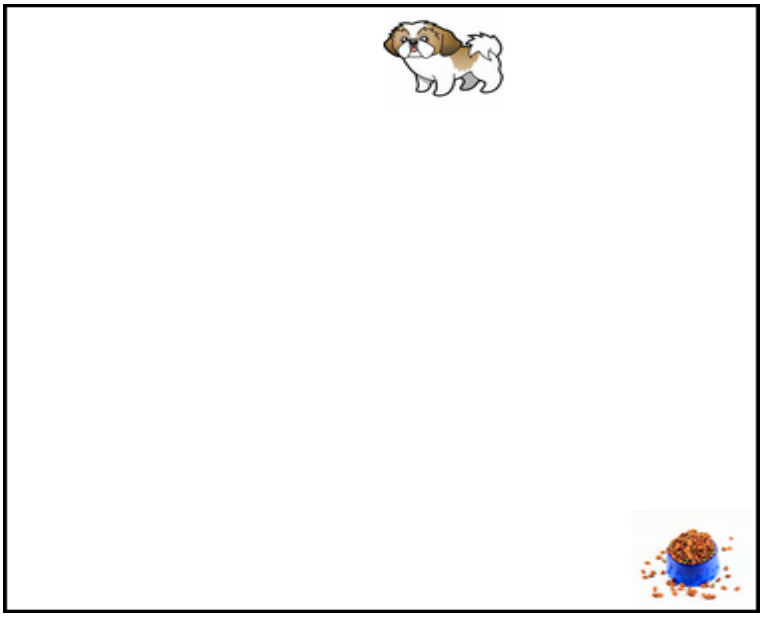

**Figura 39 – Atividade Ajudante**

**Tabela 3 – Soluções para atividade Ajudante**

| Atividade | Soluções | Observações |
|---|---|---|
| Atividade Ajudante, nesta atividade existe apenas um alimento, onde o aprendiz deve programar uma sequência de comandos de modo o Avatar chegar ao alimento. |  | Solução utilizando um caminho longo e repetição. |
| |  | Solução utilizando o menor caminho com repetição. |
| |  | Segunda Solução utilizando o menor caminho e repetição. |
| |  | Solução utilizando o menor caminho sem repetição. |
| |  | Solução utilizando o menor caminho indicado pelo Agente Analisador de Caminho com repetição. |
| |  | Segunda Solução utilizando o menor caminho indicado pelo Agente Analisador de Caminho com repetição. |
| |  | Solução utilizando o menor caminho indicado pelo Agente Analisador de Caminho sem repetição. |

As atividades devem ser resolvidas em uma sequência, de acordo com o modelo abaixo:

- Programá-las usando o Menu Principal;
- Conferir a solução usando a Função sequência de Ações;
- Executá-la usando a Função Executar;
- Acompanhar o movimento do Avatar no Tabuleiro Atividade;
- Visualizar o Feedback retornado no Menu Secundário.

Todas as descrições acima estão expostas na Figura 40 – Tela Atividade Descritiva.


**Figura 40 – Tela Atividade Descritiva**


**Figura 41 – Execução da Atividade**

Após o usuário clicar na Função Executar, o Avatar executará todas as ações programadas; o ambiente exibirá todos os *feedback* ao usuário, para uma autorregulação (o usuário, com ajuda do ambiente, vê imediatamente se a solução elaborada por ele está correta), permitindo a correção de possíveis erros. O *feedback* de retorno do ambiente pode ser visto na Figura 41 – Execução da Atividade.

### 7.8.5. EXEMPLOS DE APLICAÇÃO

É importante destacar que o uso do NewProg deve ser cuidadosamente planejado pelos educadores, de modo a utilizar ou criar atividades compatíveis com seus aprendizes, buscando ajudar da melhor maneira o processo de aprendizagem da criança.

> O software educacional necessita ser avaliado de alguma forma. A avaliação sob o ponto de vista de aprendizagem é muito complexa para ser realizada em curto prazo e implica na utilização de instrumentos que abordariam o aspecto qualitativo do sistema. O que pode ser feito dentro do escopo da dissertação, é avaliar o potencial pedagógico do protótipo via consultas à especialista (RIBEIRO, 2003).

Segundo Ribeiro (2003), a avaliação pedagógica do protótipo é a verificação do quanto o sistema pode auxiliar nas atividades de ensino e aprendizagem em situação de sala de aula real.

#### 7.8.6.1 Exemplo 1 – Melhor Caminho

O educador trabalha em sala de aula atividades envolvendo distância, exibindo as possíveis soluções, explicando as vantagens dos melhores caminhos. Após esse primeiro passo, utiliza atividades do ambiente que permitam ao aluno encontrar diversas soluções e identificar o melhor caminho.

A Figura 42 exibe uma atividade que permite várias soluções, dentre as soluções, existe uma solução, para o problema proposto, com um caminho menor.


**Figura 42 – Atividade Melhor Caminho**

A Figura 43 – Melhor Caminho exibe o melhor caminho identificado pelo Agente Analisador de Caminho.


**Figura 43 – Melhor Caminho**

### 7.8.6.2 Exemplo 2 – Soma

O educador trabalha os conteúdos envolvendo soma (adição) em sala de aula. Após isso, utiliza o ambiente para aplicar atividades que permitam a utilização dos conteúdos estudados. Essa é a proposta do professor de matemática que solicitou atividades de matemática no ambiente.

A Figura 44 – Atividade Soma exibe uma atividade com uma equação envolvendo a operação adição, nela é possível encontrar várias soluções para resolver o problema proposto. Esta atividade também trabalha com valores negativos, permitindo a operação subtração junto com adição.


**Figura 44 – Atividade Soma**

## 7.8.7. EXEMPLOS DE ATIVIDADES CRIADAS PELOS EDUCADORES

Esta seção é dedicada e exibição de algumas atividades criadas pelos usuários pedagógicos com a utilização dos editores de atividades. Sendo divididas em dois grupos, Atividades de Programação e Atividades de Programação com Matemática.

### 7.8.7.1 Atividade de Programação

Nesta seção serão exibidas as atividades de programação. Cabe ressaltar que o ambiente foi utilizado por dezenas de usuários diferentes, resultando em um enorme número de atividades. A escolha das atividades exibidas foi baseada na sua criatividade e no problema apresentado a ser resolvido.


**Figura 45 – Atividade Letra A**


**Figura 46 – Atividade Cantinho da Alegria**


**Figura 47 – Atividade Come Tudo**


**Figura 48 – Atividade Diamante**

### 7.8.7.2 Atividade de Programação com Matemática

Nesta seção serão exibidas as atividades de programação com matemática. Cabe ressaltar que o ambiente foi utilizado por dezenas de usuários diferentes, resultando num enorme número de atividades. A escolha das atividades exibidas foi baseada no problema apresentado a ser resolvido.

Para a solução desse tipo de atividades, os aprendizes podem fazer movimentos de ida e volta, permitindo a soma dos valores em cada passagem por uma célula, permitindo um maior número de soluções para cada atividade.


**Figura 49 – Atividade Matematiquinha**


**Figura 50 – Atividade Tabuada**


**Figura 51 – Atividade Tabuada de 3**


**Figura 52 – Atividade Soma Fácil**

## 7.9. Recurso Ambientação

Durante os primeiros testes do Ambiente NewProg verificou-se a necessidade de um recurso de Ambientação, visto que os aprendizes tinham dificuldades de assimilarem o correto funcionamento dos comandos a serem programados. Após uma analise inicial e uma entrevista com os educadores, modelamos o Recurso de Ambientação seguindo o mesmo padrão das atividades.

O recurso de ambientação conta com um tabuleiro do mesmo tamanho das atividades, onde os aprendizes são livres para movimentarem o Avatar, permitindo-os entender o correto funcionamento dos comandos. A Figura 53 – Recurso de Ambientação exibe a tela do Recurso de Ambientação. Para o melhor entendimento do funcionamento dos comandos, no recurso ambientação não existe a Função Executar, ao clicar no comando desejado o Avatar realiza o movimento do mesmo.


**Figura 53 – Recurso de Ambientação**

## 7.10. Avaliação dos resultados

A validação do ambiente foi dividida em duas etapas. A primeira etapa avaliou as atividades sendo executadas pelos aprendizes e a segunda etapa foi à avaliação dos editores sendo utilizados pelos educadores.

### 7.10.1. VALIDAÇÃO DAS ATIVIDADES

Para a validação das atividades foram realizados alguns experimentos, finalizando com uma pesquisa de campo em uma instituição de ensino fundamental com vinte crianças na faixa etária de cinco a oito anos e quatro professores.

A pesquisa foi dividida em duas atividades, iniciamos com uma atividade simples, onde o aluno precisava programar uma sequência de ações, de modo a fazer um Avatar se locomover na tela do computador e chegar ao alimento. Nesta atividade não existia barreiras.

Na segunda atividade, utilizamos uma atividade contendo obstáculos a serem contornados pelo Avatar escolhido, onde o aprendiz tinha a missão de programar o Avatar para percorrer um caminho, desviando dos obstáculos, e chegar ao alimento.

A pesquisa foi realizada durante o segundo bimestre de 2013, os dados obtidos são apresentados a seguir. A Tabela 4 – Média dos dados Primeira Atividade, exibe os dados da pesquisa referente à primeira atividade e a Tabela 5 – Média dos dados Segunda Atividade, exibe os dados referentes à segunda atividade.

**Tabela 4 – Média dos dados Primeira Atividade**

| | **Primeira amostragem** | **Segunda amostragem** | **Terceira amostragem** | **Quarta amostragem** |
|---|---|---|---|---|
| **Êxito** | 12% | 35% | 70% | 95% |
| **Erros** | 5 | 4 | 2 | 1 |
| **Motivação** | Baixa | Média | Boa | Alta |

**Tabela 5 – Média dos dados Segunda Atividade**

| | Primeira amostragem | Segunda amostragem | Terceira amostragem | Quarta amostragem |
|---|---|---|---|---|
| **Êxito** | 30% | 55% | 85% | 98% |
| **Erros** | 9 | 6 | 3 | 2 |
| **Motivação** | Média | Boa | Alta | Alta |

Destacamos que na segunda atividade, houve um aumento significativo no número de erros, pois, devido à complexidade maior da atividade, alguns alunos apresentaram dificuldades para a correta programação do caminho.

Ao longo da pesquisa, procurávamos evidências que comprovassem a aceitação do ambiente NewProg pelos alunos e professores. A cada etapa do processo, questionávamos os participantes sobre o uso do ambiente, indagando-os se concordavam com seu uso e se percebiam benefícios. A Figura 54 exibe a avaliação do ambiente por parte de alunos e professores.

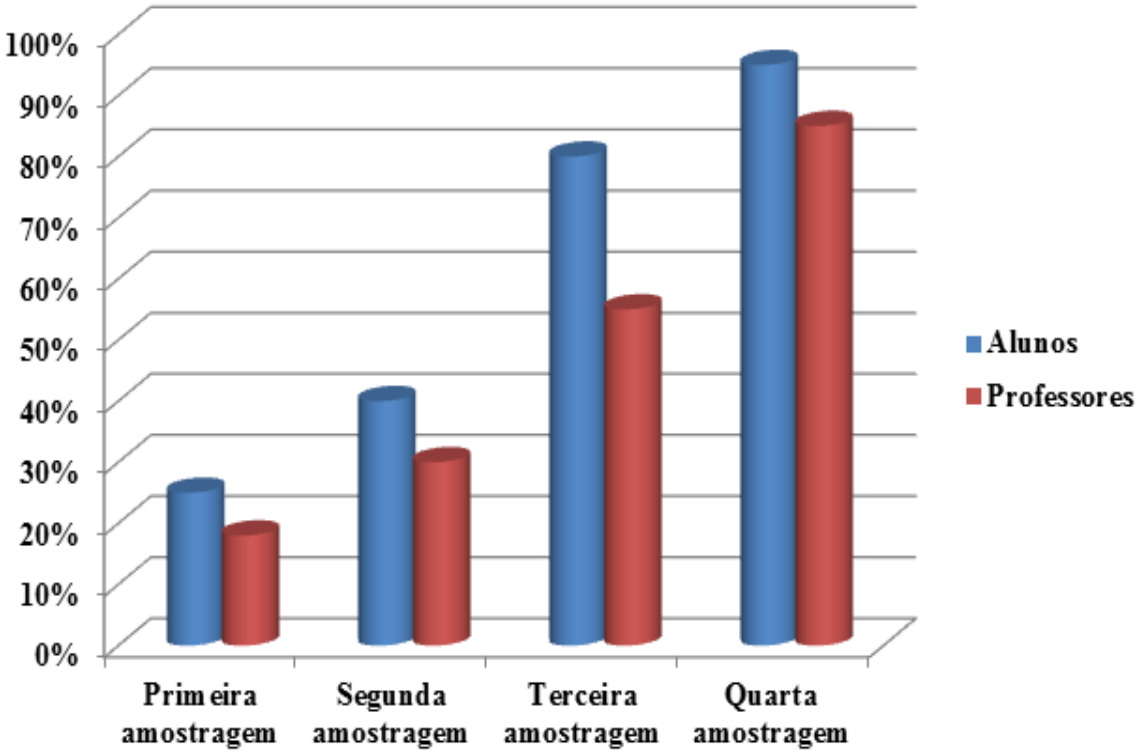


**Figura 54 – Avaliação do ambiente NewProg**

### 7.10.2. VALIDAÇÃO DOS EDITORES DE ATIVIDADES - EANewProg

A validação dos editores de atividades foi realizada durante o segundo bimestre de 2013 e o primeiro semestre de 2014, com dezesseis educadores de escolas da rede municipal, responsáveis por turmas da faixa etária de cinco a oito anos. Sendo dividida em duas etapas:

1ª etapa, os educadores desenvolveram suas atividades sem qualquer intervenção – eles ficaram livres para criá-las. Durante essa primeira etapa, observamos que a minoria utilizava o recurso de Simular, onde poderiam testar suas atividades, antes de criá-las definitivamente no ambiente NewProg.

2ª etapa, apresentamos situações onde as atividades construídas poderiam ser insolúveis, algumas por falhas na construção ou pelo limite de nove ações (36 passos) do ambiente NewProg. Retornamos ao desenvolvimento das atividades, agora, instruindo-os a criarem apenas atividades com uma ou mais soluções possíveis. Nesse momento, percebemos que a maioria começou a usar o recurso Simular, de modo a apenas criarem atividades que poderiam ser solucionadas.

Durante a pesquisa, ao finalizarmos uma etapa de criação de atividades, coletamos os dados e incentivamos os professores a construírem novas atividades mais difíceis de serem resolvidas. A Tabela 06 – Média dos dados das atividades criadas, exibe os resultados obtidos durante a pesquisa.

**Tabela 6– Média dos dados das atividade criadas**

| Atividades | Primeira amostragem | Segunda amostragem | Terceira amostragem | Quarta amostragem |
|---|---|---|---|---|
| **Insolúvel** | 45% | 10% | 5% | 0% |
| **Uma solução** | 0% | 5% | 15% | 65% |
| **N soluções** | 55% | 85% | 80% | 35% |

No decorrer da pesquisa questionamos os educadores sobre o uso do editor, indagando-os se concordavam com seu uso e se percebiam benefícios. A Figura 55 – Avaliação do EANewProg, exibe a avaliação feita pelos professores sobre os benefícios da utilização do editor para a criação de atividades para o NewProg.

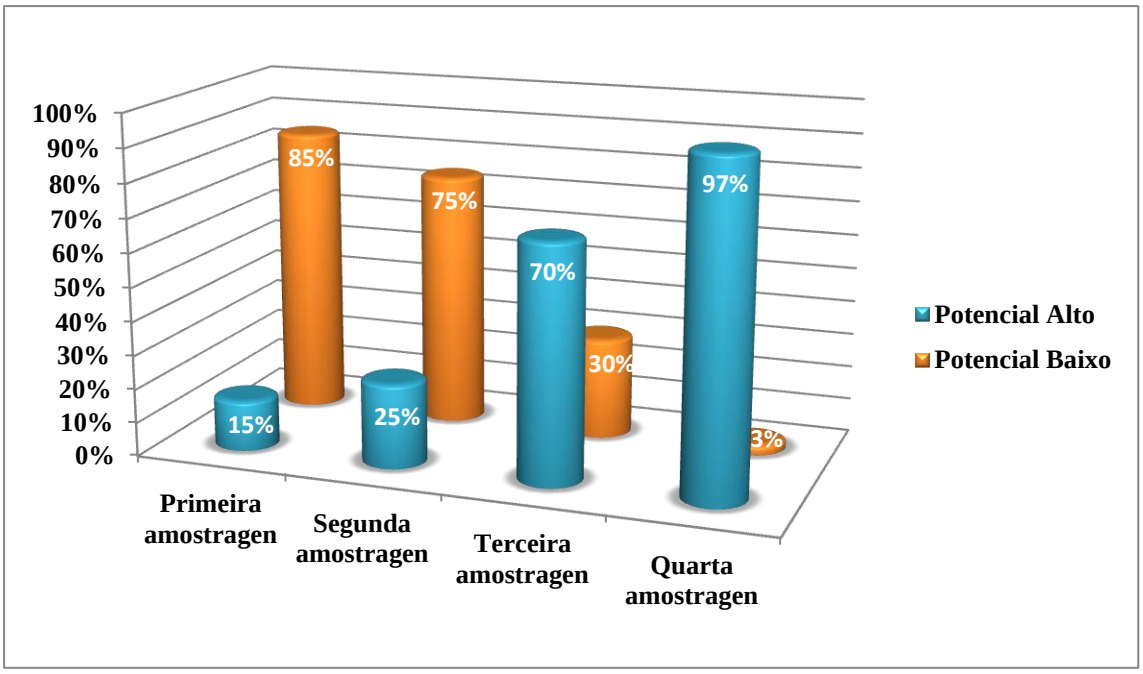


**Figura 55 – Avaliação do EANewProg**

### 7.10.3. COMPARAÇÃO ENTRE OS OBJETIVOS INICIAIS E OS RESULTADOS

Concluímos que o ambiente NewProg, juntamente com seu editor de atividades EANewProg, cumprem os objetivos planejados para esta pesquisa. Portanto, esse ambiente, com os componentes NewProg e EANewProg, é uma ferramenta útil para o processo de construção de conhecimentos e também para o acompanhamento da evolução dos aprendizes, permitindo educadores desenvolverem novas estratégias/metodologias que possibilitem aos aprendizes evoluírem de acordo com ritmo de cada um deles.

A avaliação final do ambiente o coloca em sintonia com as teorias de VYGOSTSKY (1988), onde o desenvolvimento de crianças e adolescente deve ser olhado de maneira prospectiva, isto é, com referência ao que está para acontecer na trajetória de cada um. De acordo com VYGOSTSKY, devemos procurar os “brotos”, “as flores” ou “ramos” do desenvolvimento e seus rumos, em vez de somente seus frutos.

Em relação à evolução dos aprendizes, acreditamos que o processo de reflexão possível de ser realizado com o uso do ambiente pode produzir diversos níveis de abstração, os quais, de acordo com PIAGET, provocarão alterações na estrutura mental do estudante.

> O nível de abstração mais simples é a abstração empírica, que permite ao aluno extrair informações do objeto ou das ações sobre o objeto, tais como a cor e a forma do objeto. A abstração pseudoempírica permite ao aprendiz deduzir algum conhecimento da sua ação ou do objeto. A abstração reflexiva permite a projeção daquilo que é extraído de um nível mais baixo para um nível cognitivo mais elevado ou a reorganização desse conhecimento em termos de conhecimento prévio (abstração sobre as próprias ideias do aluno) (Valente, 1993).

Desse modo, concretizamos e colocamos à disposição do educador uma ferramenta para ele criar atividades que possam desenvolver habilidades cognitivas nos seus alunos (crianças), de acordo com as necessidades cognitivas individualizadas. Com as atividades apropriadamente preparadas, o ambiente NewProg é uma poderosa ferramenta para potencializar a passagem da abstração empírica para a reflexiva, enquanto ajuda a concluir o ciclo da aprendizagem, permitindo à criança aplicar os conceitos que está aprendendo.

## 8. CONSIDERAÇÕES FINAIS

Este trabalho teve como motivação inicial a dificuldade de alunos universitários em aprender programação de computadores, visto que uma grande parcela chega à universidade sem o desenvolvimento de algumas habilidades cognitivas de ordem superior. Diversas pesquisas e experimentos são realizados de forma a ajuda-los a aprenderem programação, no entanto, essas pesquisas quase sempre são tardias sendo destinadas a alunos do ensino médio ou universitário.

A carência de ambientes que apoiem/permitam a aprendizagem de programação e o acompanhamento individualizado dos aprendizes motivou o desenvolvimento desta pesquisa. Ao realizar tal ação, tentou-se suprir essa necessidade e permitir o desenvolvimento de habilidades cognitivas de ordem superior logo nas séries iniciais.

Desenvolver um ambiente de aprendizagem requer cautela e a participação de profissionais de várias áreas, pois o mesmo deve contribuir no processo de ensino-aprendizagem. É importante que a sua concepção esteja vinculada a três elementos importantes: “o ‘saber fazer’, o ‘como fazer’ e ‘para que fazer’ da atividade intelectual” (MANTOAN et al, 1998).

O desenvolvimento do ambiente proposto foi embasado em pesquisas etnográficas. Durante sua utilização foram adicionadas diversas melhorias, a partir de sugestões de profissionais das áreas de educação infantil, psicologia, psicopedagogia, de informática educacional e, por fim, mas não menos importante, dos próprios aprendizes.

Esperamos que, com os resultados desta pesquisa, possamos cooperar para o avanço da educação nas séries iniciais do ensino fundamental e também da educação infantil, de modo a favorecer a aprendizagem das crianças e potencializar o desenvolvimento de habilidades cognitivas importantes.

Valendo-se dos resultados desta pesquisa, pode-se subsidiar também a formação inicial e continuada de professores que ensinam nessas séries, permitindo-os construir atividades individualizadas para seus aprendizes.

## 8.1. Trabalhos Futuros

Durante o desenvolvimento desta pesquisa todos os objetivos foram alcançados. No entanto, é interessante que outros recursos/atividades não implementados sejam contemplados para que o ambiente NewProg tenha uma aceitação máxima pelos educadores.

É importante a implementação de novos editores e executores de atividades multidisciplinares. Durante a pesquisa etnográfica foi levantada a possibilidade de criação de atividades que seguissem o padrão do ambiente NewProg e que contemplassem outras áreas de estudos.

Dentre das possíveis atividades que poderiam ser implementadas no ambiente destacam-se:

- Quiz;
- Montagem de Operações;
- Montagem de textos/historinhas infantis.

### 8.1.1. ATIVIDADE QUIZ

Neste modelo de atividade, o ambiente contaria com um Avatar e uma quantidade de portas definidas pelo educador. Cada porta conteria uma resposta e o aprendiz deveria responder uma pergunta programando o Avatar a chegar à porta correta.

Ao clicar em uma porta, durante a elaboração da solução, é exibida uma janela com uma resposta. Ao encontrar a resposta correta/desejada, o aprendiz constrói o código de maneira ao Avatar alcançar a porta. A Figura 56 – Atividade Quis, exibe um visual possível para essa atividade.

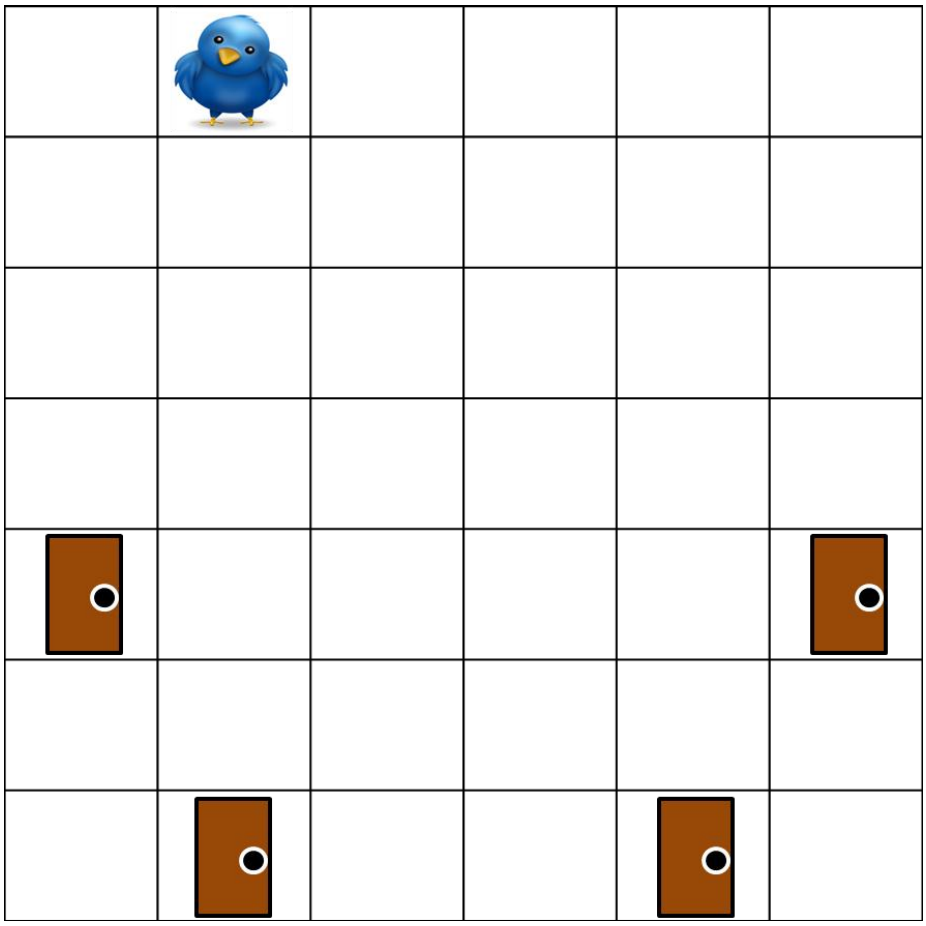

**Figura 56 – Atividade Quiz**

### 8.1.2. ATIVIDADE MONTAGEM DE OPERAÇÕES

Neste modelo de atividade, o ambiente contaria com um Avatar colocado na primeira fileira do tabuleiro. Na segunda fileira haveria números. A terceira fileira estaria vazia para os movimentos do Avatar. Na quarta fileira estariam operadores matemáticos; seguida pela quinta fileira, vazia para movimentos e, por fim, a sexta fileira haveria números. A Figura 57 apresenta esse cenário descrito.

Nesta atividade, o educador define um valor numérico, sendo este o resultado para a operação que o aprendiz precisa fazer o avatar traçar, ao se deslocar pelo cenário definido na atividade. A Figura 57 – Atividade Montagem de Operações, exibe o visual desta atividade.


**Figura 57 – Atividade Montagem de Operações**

Cabe ressaltar que esta atividade foi implementada para testes. No entanto, não foi desenvolvido um editor para este modelo de atividades, de modo que os educadores pudessem construir atividades individualizadas para seus aprendizes.

### 8.1.3. ATIVIDADE MONTAGEM DE TEXTOS/HISTÓRIA

Neste modelo de atividade o ambiente contaria com um Avatar e uma quantidade de portas definidas pelo educador. Cada porta conteria uma parte do texto/história. O aprendiz deveria montar o texto/história programando o Avatar a passar por todas as portas, montando o texto na sequência correta.

Ao clicar em uma porta é exibida uma janela com uma parte do texto/história. Ao encontrar a ordem correta do texto, o aprendiz constrói o código de maneira a fazer o Avatar passar pelas portas construindo o texto/história. A Figura 58 – Atividade Montagem de Texto/História, exibe o visual desta atividade.

| | | | | | |
|---|---|---|---|---|---|
| |  | | | | |
| | | | | | |
| | | | | |  |
| |  | | | | |
| | | |  | | |
| | | | | | |
| |  | | | | |

**Figura 58 – Atividade Montagem de Texto/História**

## 9. REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, F. J. de. Educação e informática: os computadores na escola. 5 ed. – São Paulo, Cortez, 2012.

ANDRADE, M.; SILVA, C.; OLIVEIRA T. Desenvolvendo games e aprendendo matemática utilizando o Scratch. XII SBGames – São Paulo – SP, 2013.

AZEVEDO, R. F. C. O uso de software educativos numa perspectiva ludica, para a construção do conhecimento. Mestrado em Educação, Universidade Estácio de Sá, RIO DE JANEIRO, 2007.

CAMPANA, V. F. Concepção de recursos digitais para apoiar o processo de ensino-aprendizagem – Uma abordagem etnográfica. Mestrado em Informática, CT/UFES. Vitória-ES, 2009.

CHARNIAK, Eugene; MCDERMOTT, Drew. A Bayesian Model of Plan Recognition. Massachusetts: Addison-Wesley, 1985.

CHAVES, E.; STEZER, W. O uso do computador em escola. São Paulo, Scipione, 1988.

CONALLEN, Jim. Desenvolvendo aplicações WEB com UML, Campus 2ª Ed, 2003.

COX, K. K. Informática na Educação Escolar. 2. Ed. Campinas – SP, Autores Associados, 2008.

CRISTÓVÃO, H. M. Ambientes Computacionais para Apoio à Aprendizagem: Um Experimento com Frações. Vitória - ES. Mestrado em Informática. Universidade Federal do Espírito Santo, 1997.

DIAS, P.; OSÓRIO, A. J., “Challenges 2009 : actas da Conferência Internacional de TIC na Educação, 6, Braga, Portugal, 2009”. Braga : Centro de Competência da Universidade do Minho, 2009.

EIVAZIAN, A. M. B. O ensino de ciências usando simulações. Acesso, São Paulo, v. 11, 1995.

FAGUNDES, LEA; NEVADO, Rosane; BASSO, Marcus; BITENCOURT, Juliano; MENEZES, C. S.; MONTEIRO, V. C. Projetos de Aprendizagem – Uma experiência mediada por ambientes Telemáticos. RBIE, 2006.

FESSAKIS, G.; GOULI, E.; MAVROUDI, E., Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. Computers & Education 63 (2013) 87–97.

GEE, J. P. What Video Games Have to Teach Us about Learning and Literacy, Palgrave Macmillan: New York, 2003.

GENNARI, M. C. Minidicionário de informática. São Paulo, Saraiva, 1999.

HAUGELAND, John. Artificial Intelligence: The Very Idea. Massachusetts: The MIT Press, 1985.

HAYES-ROTH, B. An Architecture for Adaptive Intelligent Systems, Artificial Intelligence: Special Issue on Agents and Interactivity, vol. 72, 1995.

KAPTELININ, V., NARDI, B. A. Activity Theory: Basic Concepts and Applications. CHI 97 Electronics Publications: Tutorials, march/1997.

KURZWEIL, Ray. The Age of Spiritual Machines. Massachusetts: The MIT Press, 1990.

LA TAILLE, Y. De. Ensaio sobre o lugar do computador na educação. São Paulo, Iglu, 1990.

MANTOAN, M. T. E.; MARTINS, M. C. e MISKULIN, R. G. S. Análise Microgenética de Processos Cognitivos em Contextos Múltiplos de Resolução de Problemas; In.: Revista Brasileira de Informática na Educação; No. 3; ISSN: 1414-5685; págs. 27-44; setembro, 1998.

MARACCI, F. V.; COLANZI, T. E.; JUBILEU, A. P.; ROSA, M. V. C.; BRANCO, E. F. S. WAEI/MSE: uma instância do processo WAE para micro e pequenas empresas de software. Acta Scientiarum Technology, Maringá, 2009.

MARQUES, C. P. C.; MATTOS, M. I. L. De; LA TAILLE, Y. De. Computador e Ensino – Uma aplicação à língua portuguesa. São Paulo, Editora Ática S.A., 1986.

MARTINS, L. E. G.; DALTRINI, B. M. Utilização dos Preceitos da Teoria da Atividade na Elicitação dos Requisitos do Software. Anais do XIII SBES – Simpósio Brasileiro de Engenharia de Software. Florianópolis - SC, 1999.

MATTOS, C. L. G. A abordagem etnográfica na investigação científica. Disponível em: http://people.ufpr.br/~marizalmeida/celem05/abord_etnogr_invest_cient.doc. Acesso em: 15 de setembro de 2014.

MORAES, R. de A. Informática na Educação. 2. ed. Rio de Janeiro: DP&A, 2000.

MORAN, J. M. Mudanças na Comunicação Pessoal. São Paulo: Paulinas, 1998.

NARDI, B. A. Context and Consciousness - Activity Theory and Human-Computer Interaction. MIT Press, 1996.

NEMHAUSER, G.L.; Rinnoy K. A.H.G. & Todd, M.J. Handbooks in Operations Research and Management Science. North-Holland, Vol 1, 1989.

Neto, F. A. A.; CASTRO, T. H. C.; JÚNIOR, A. N. de C., Utilizando o Método Clínico Piagetiano para Acompanhar a Aprendizagem de Programação. XVII Simpósio Brasileiro de Informática na Educação – SBIE – UNB/UCB – 2006.

OLIVEIRA, E. De; VAHLDICK, A. Um estudo de caso de utilização da WAE para UML em aplicações GWT. Anais V SULCOMP, Vol. 5, No 1, 2010.

PACIEVITCH Y. HTML. Disponível em: http://www.infoescola.com/informatica/html/. Acesso em: 09 de outubro de 2014.

PAPERT, S. Logo: Computadores e Ensino. São Paulo, Brasiliense, 1985.

_____. A máquina das crianças. Porto Alegre: Artes Médicas, 1994

PEREIRA, A. P. O que é CSS? Disponível em: http://www.tecmundo.com.br/programacao/2705-o-que-e-css-.htm. Acesso em: 09 de outubro de 2014.

PINTO, A. S. Scratch na aprendizagem da Matemática no 1.º Ciclo do Ensino Básico: estudo de caso na resolução de problemas. Mestrado em Estudos da Criança – Tecnologias de Informação e Comunicação, Instituto de Educação, Universidade do Minho, Portugal, 2010.

POOLE, D.; MACKWORTH, A. K.; GOEBEL, R. Computational Intelligence: A Logical Approach. Oxford: Oxford University, 1998.

RIBEIRO, J. C. de C. e S. Uma Experiência de Agentificação Aplicada a Software Educacional. Dissertação de Mestrado em Ciência da Computação, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre-RS, 2003.

RICH, E.; KNIGHT, K. Inteligência Artificial. 2 Ed., Editora McGraw-Hill, 1994.

RUSSEL, S.; NORVIG, P. Inteligência Artificial. 2. Ed., Rio de Janeiro: Campos, 2004.

SANCHES, D. R. Uma Arquitetura MultiAgente para Ambientes Virtuais de Aprendizagem. Mestrado em Informática, CT/UFES. Vitória-ES, 2006.

SICA, F. C.; BORTOLINI, N. das G. de S. Educação e Informática: um diálogo essencial. Ouro Preto: Universidade Federal de Ouro Preto, 2007.

SOARES, A. R. ProjetoLogo - Uma Apresentação. Disponível em: http://projetologo.webs.com/texto1.html. Acesso em: 02 de outubro de 2014.

TAVARES, O. L.; MENEZES, C.S.; NEVADO, R.A.: Pedagogical architectures to support the process of teaching and learning of computer programming: In FIE2012-Frontiers in education conference, 2012.

TENÓRIO, R. M. Cérebros e Computadores: a complementaridade analógica-digital na informática e na educação. 4. ed. São Paulo, Escrituras Editora, 2003.

TOREZANI, C ; CHAGAS, L. B. C.; TAVARES, O. L., NewProg - um ambiente online para crianças aprenderem programação de computadores. XIX Workshop de Informátca na Escola, 2013.

VALENTE, J. A.. Diferentes Usos do Computador na Educação. Computadores e Conhecimento: repensando a educação. Campinas, SP: Gráfica da UNICAMP, 1993.

_____. Computadores e conhecimento: repensando a educação. 2ª ed. Campinas - SP, UNICAMP/NIED, 1998.

_____. O computador na sociedade do conhecimento. Campinas, SP:UNICAMP/NIED, 1999.

Vygotsky, L. S. A formação social da mente. São Paulo-SP: Martins Fontes, 3ª ed. 1988.

WINSTON, P. H. Inteligência Artificial; Tradução de PAVEL. C. O. Rio de Janeiro, LTC-Livros Técnicos e Científicos, 1988.