

# **Takkou: Uma Ferramenta Proposta ao Ensino de Algoritmos**

**Leônidas S. Barbosa, Teresa C. B. Fernandes, André M. C. Campos**

Departamento de Informática e Matemática Aplicada – Universidade Federal do Rio  
Grande do Norte (UFRN)

Caixa Postal 1679 – 59.072-970 – Natal – RN – Brasil

barbosaleonidas@aol.com, {tfernandes.cc, amccampos}@gmail.com

**Abstract.** *This paper presents a tool, named Takkou, to be used as support on algorithms classes. The tool aims to provide a playful learning experience based on puzzle games. Takkou is a game where the player must solve a puzzle through the actions performed by his/her game character. However, the player does not directly control his/her character, but instead provide a set of algorithmic instructions to it. Takkou is based on theory of Meaningful Learning, and its main aim is to motivate students to exercise and improve their algorithms skills.*

**Resumo.** *Este artigo apresenta uma ferramenta, chamada Takkou, a ser utilizada como ferramenta de suporte em aulas de algoritmos. O objetivo da ferramenta é criar uma experiência lúdica através de jogos de raciocínio. Takkou é um jogo onde o jogador deve resolver um problema através de ações realizadas por seu personagem no jogo. Entretanto, o jogador não controla diretamente seu personagem. Ele o controla através de um conjunto de instruções algorítmicas. Takkou é baseado na Teoria de Aprendizagem Significativa e seu principal objetivo é motivar os estudantes a se exercitarem e aperfeiçoarem suas habilidades em algoritmos.*

## **1. Introdução**

A disciplina de algoritmos, em cursos de computação, representa a base na formação de estudantes em ciências da computação. Por meio de algoritmos são propostas soluções de problemas de ordem diversa, desde qual a melhor rota para um duto de petróleo até a definição de quantos passos um robô deve executar para trocar uma lâmpada. Por ser uma disciplina de base, aos alunos que ingressam no curso de computação, é exigido que se tenha boa habilidade algorítmica. Entretanto, durante os últimos anos o que se observa é que há um aproveitamento ineficiente dos alunos nesta disciplina. Em virtude do mal aproveitamento muitos alunos perdem o interesse pelo curso enxergando a disciplina como barreira para sua formação. Segundo Rocha (1993), os números de evasão e desistência na disciplina de algoritmos chegam a 60%. Ainda segundo o autor, é defendido que o mal aproveitamento na disciplina se deve a má metodologia abordada em sala de aula. Porém, em Jesus e Brito (2009), é afirmado que cabe aos alunos possuir certas habilidades requeridas no curso e a falta delas é a responsável pelo mal aproveitamento culminando em desistências e evasões. Dentre outras discussões o fator motivacional é apresentado como forte fator responsável pelo mal aproveitamento,

onde a falta de motivação faz com que os alunos sintam-se desencorajados em continuar no curso.

Como o objetivo de esclarecer quais os problemas que afetam o ensino de algoritmos, em Campos et al. (2011), foi realizado um levantamento sobre o aproveitamento na disciplina de algoritmos durante o período de setembro a novembro de 2010. Além dessa abordagem, foi realizado um comparativo com os dados de 2003 a novembro de 2010 advindos de artigos que abordavam o tema, aproveitamento em algoritmos. Junto aos alunos foi realizado uma solicitação de dados através de um questionário quantitativo quanto ao aproveitamento e dificuldades vivenciadas durante a disciplina de algoritmos, e realizado um levantamento estatístico junto a algumas universidades brasileiras. Ao final, os autores realizaram uma revisão bibliográfica que frente as observações obtidas confirmavam os apontamentos.

Com os dados levantados, percebeu-se que há na verdade dois problemas principais responsáveis pelo mal aproveitamento na disciplina de algoritmos: o problema do ensino, que é referente a abordagem pedagógica aplicada em sala de aula, bem como a experiência dos alunos e a interação professor-aluno, dentre outros fatores relatados no trabalho citado; e o problema do raciocínio lógico, representando habilidade extremamente requerida nas disciplinas dos cursos de computação, onde, por vezes, o ingressante não a possuindo bem desenvolvida sente dificuldades em transcrever a resolução de problemas em algoritmos. Como forma de confrontar o problema do raciocínio lógico, no trabalho citado, foi proposta uma ferramenta com características de jogo que, por meio de *puzzles*, abordam conceitos de algoritmos e auxiliam os estudantes no exercício desta habilidade. Através de uma metodologia de uso onde o aluno não só soluciona os *puzzles*, mas, também, constrói desafios, e, assim exercita seu aprendizado, foi proposta a ferramenta *Takkou* a ser disponibilizada via web para os usuários.

Nesse trabalho será apresentada a ferramenta *Takkou*, ferramenta proposta como um protótipo do trabalho abordado em Campos et al. (2011), que intenta auxiliar o aluno no exercício do raciocínio lógico através de um jogo suscitando maior motivação com base em aspectos pedagógicos inspirados na teoria de David Ausubel.

## **2. Fundamentação Teórica**

A teoria de David Ausubel, teoria do Aprendizado Significativo, aborda aspectos importantes para tornar a aprendizagem mais significativa, tais como a formulação do material potencialmente significativo, fazendo uso de signos que remetam a alguma situação real ou não, até a troca constante de informações entre professor-aluno e a disposição dos alunos em aprender conforme a teoria. Sendo uma teoria cognitivista, preconiza grande importância ao papel do aluno no modo como as informações são assimiladas por estes e como esta informação se adéqua e se reestrutura na sua estrutura cognitiva. Há, portanto, uma constante preocupação no aluno como fator do processo de aprendizagem, pois é desejável que o aluno aprenda de forma significativa, enriquecendo sua estrutura cognitiva, e não mecanicamente, onde os conceitos ficam dispersos nessa estrutura. Ao estruturar sua teoria, Ausubel afirma que quando um conceito menor se liga a um conceito maior, o qual identifica algum domínio de

conhecimento específico, esse conceito maior é chamado de **subsunçor**. O conceito subsunçor define uma das principais características da teoria de Ausubel, pois é através dele que o teórico justifica todo o processo empregado em sua teoria afirmando que, se os conceitos recém-adquiridos se ligam a um determinado subsunçor, há o aprendizado significativo, caso contrário o aprendizado é dito mecânico, pois não há nenhum conceito que esteja ligado ao conceito recém-adquirido e este ficará disperso na estrutura cognitiva do aluno sem nenhuma ligação, o que fará com que o esqueça com maior facilidade, Moreira e Mansini (2006).

Além do aspecto cognitivista da teoria, há o construtivista afirmando que os conhecimentos passados devem ser significativos, devem provocar no aluno alguma importância, Ausubel, em outras palavras, diz que cabe ao aluno em determinado momento tomar para si se um conteúdo é importante de acordo com a sua estrutura cognitiva e, a partir disso, construir o seu próprio conhecimento. Pelizzari (2002), afirma que há dois aspectos importantes que levam o aluno a aprender significativamente. O primeiro é o significado lógico, no qual para o aluno, a informação passada deve possuir algum sentido que se adeque ao conteúdo passado. O segundo é o significado psicológico, esse é particular ao aluno, é ele que, em dado momento, irá ligar na sua estrutura cognitiva algum conceito a um subsunçor próprio daquele conceito próprio ou temporário, onde subsunçores próprios são aqueles que ligam de forma direta um conceito menor a um conceito maior em um domínio específico. Diz-se que um subsunçor **A** é próprio de **B** se os conceitos aos quais **B** referencia é subconjunto dos conceitos referenciados por **A**.

Para a criação da ferramenta *Takkou* preocupou-se no modo como esta iria transmitir ao aluno signos do dia a dia, mas que com o auxílio da ferramenta referenciassem conceitos de algoritmos. A criação da ferramenta com base na proposta de Ausubel se justifica por um outro ponto abordado nessa teoria, o de que o material passado deve ser potencialmente significativo. Para que fosse possível criar uma ferramenta como material potencialmente significativo pensou-se em como os objetos e o visual desta estariam dispostos de forma a relacionar conceitos de algoritmos. Na *Takkou*, os elementos que compõem o cenário do jogo são simples e representam algum objeto do cotidiano, uma pedra, um lago, um arbusto, uma chave, porta, etc. Entretanto, quando o aluno tenta alcançar algum dos objetivos do jogo, são apresentados, de forma tangencial, conceitos de algoritmos em forma de instruções, tais como: estruturas de condicionais, laços de repetição, variável, etc. Dessa forma para que o objetivo do jogo seja alcançado o aluno deve criar um algoritmo que solucione o *puzzle*. Em contrapartida a resolução de problemas por meio dos alunos, há ainda a possibilidade do aluno criar o seus próprios desafios. Nessa abordagem, o aluno é colocado na condição de arquiteto e solucionador do problema, pois acredita-se que a medida que o aluno cria algum *puzzle* ele está, de certo modo, pensando duas vezes, a primeira em como criar e a segunda em como resolver a sua criação. Assim alcançasse, neste ponto, uma das características marcantes do construtivismo, fazendo com que o aluno construa o seu próprio conhecimento. Há ainda o fator de que, dessa forma, o aluno se esforça muito mais a aprender os conceitos de algoritmos, pois só entendendo como estes funcionam ele poderá criar *puzzles* suficientemente desafiadores aos seus colegas.

Com base no que foi apresentado, para a implementação da ferramenta em

questão, preocupou-se em atender aos seguintes aspectos voltados a teoria que a inspira.

- Apresentação de signos, símbolos do cotidiano: na ferramenta alguns símbolos e signos comuns do dia a dia dos alunos são transcritos visualmente deixando-os visualizar situações comuns já viciadas por eles;

- Possibilidade de desafiar os outros usuários (criação de *puzzles*): esta característica é de grande importância por tornar a abordagem pedagógica da ferramenta mais proveitosa, não o bastante, além desse aspecto pedagógico há o aspecto motivador, que surge ao permitir que os usuários possam desafiar-se;

- Temporizador de desafios: a questão tempo foi abordada na ferramenta de modo a tornar os desafios mais interessantes, não apenas isso, mas conforme se sabe, em cursos de algoritmos é comum que alunos resolvam problemas em um dado tempo. Com a ideia do temporizador em cada *puzzle* intenta-se atender os fatores motivacionais e forçar o aluno a pensar em tempo hábil tornando o jogo mais interessante, o que também irá indiretamente implicar em uma melhoria nas suas habilidades deixando-o mais produtivo na resolução de problemas;

- Usabilidade: a ferramenta apresenta características interessantes no tocante a usabilidade. Esta preocupação deveu-se ao fato de “jogadores” sentirem-se mais atraídos por jogos com boa usabilidade facilitando o seu uso e deixando-os à vontade no ambiente da ferramenta. Outro ponto importante é que, como citado no item anterior, é dado ao aluno um tempo determinado para solucionar um *puzzle*, logo, o uso da ferramenta, a sua facilidade de uso, deve se dar a contento possibilitando que este consiga usá-la de forma produtiva no tempo que lhe foi dado;

No referente aos aspectos técnicos da ferramenta tem-se:

- Ferramenta usável via web: a criação da ferramenta *Takkou* está voltada ao ambiente web, uma vez que, em qualquer lugar onde o usuário estiver poderá fazer uso dela, exercitar suas habilidades, servindo até mesmo aos propósitos de educação a distância conforme interesse;

- Criação de um ambiente de registro de *puzzles*: conforme já fora mencionado, na *Takkou* há a possibilidade de serem criados *puzzles* pelos usuários, para que estes possam ser mantidos foi sugerido a criação de uma base de dados acessível por meio de serviço web que recupera e registra os *puzzles*. Essa base poderá futuramente ser usada com diversos propósitos, como por exemplo uma avaliação via ferramenta.

### **3. Trabalhos Correlatos**

Há na literatura grande número de trabalhos voltados ao ensino de algoritmos. Alguns destes trabalhos abordam através de ferramentas formas novas e mais motivadores de se melhorar o ensino desta disciplina. Outras abordam o problema por meio de metodologias com base em teorias pedagógicas. Dentre os trabalhos que abordam a criação de ferramentas voltadas ao ensino de algoritmos, ou que podem ser utilizadas com tal finalidade, destaca-se, aqui, alguns deles de acordo com a seguinte categorização.

- 1.Ferramentas de linguagem: ferramentas voltadas a alguma linguagem de programação específica ou uma nova linguagem criada com o propósito de melhorar o

ensino de algoritmos. Enquadram-se nesta categoria as ferramentas: JavaTool e Tepequém. A primeira aborda o ensino algoritmos através de um subconjunto de instruções da linguagem Java por meio de uma ferramenta visual, Mota et al, (2008). A segunda aborda o ensino de algoritmos através de uma linguagem criada em português que possui características de acentuação e outros detalhes para o ensino de algoritmos e programação, Hinterholz Jr, (2009 );

2.Ferramentas de simulação: são ferramentas que, de algum modo, simulam o comportamento de algoritmos, seja via depuração de código ou através do auxílio de fluxograma. As ferramentas CIFluxProg e VisuAlg se enquadram nesta categoria. A primeira aborda o ensino de algoritmos, simulação, através de fluxogramas e depuração, Santiago (2004) . Já a segunda faz uma abordagem mais voltada a depuração do conteúdo das variáveis em dado momento da execução do algoritmo, Souza (2009) ;

3.Ferramenta de interação com algoritmos: não necessariamente são ferramentas voltadas ao ensino de algoritmos, mas fazem constante referencia aos seus conceitos. Dentre as ferramentas que se enquadram nesta categoria cita-se o Scratch e o Robomind. A primeira é uma ferramenta do MIT de propósito geral onde o usuário pode interagir e criar animações e vídeos utilizando de pseudo código por aglutinação de comandos, possuindo como uma das suas características ter sido desenvolvida sob base pedagógica construtivista, Maloney et al, (2008) . O robomind por outro lado é uma ferramenta que simula um robô em um ambiente com obstáculos e objetos a serem capturados. Esse ambiente pode ser estendido a um robô real por meio do Lego Mindstorm. Para mover o robô virtualmente ou real é possível executar instruções algorítmicas por meio de código em uma linguagem inspirada na linguagem LOGO.

4.Jogos aplicados ao ensino de algoritmos: são jogos que possuem geralmente, de forma indireta, conceitos de algoritmos incorporados na interação do jogador. Como exemplo de jogos que envolvem o usuário, muitas vezes de forma tangencial, com os conceitos de algoritmos têm-se: Robozzle e o Light-Bot, estes são jogos similares que abordam através de *puzzles* ideias que se referem a algoritmos. Em ambos é apresentado um puzzle que deve ser solucionado por meio de comandos através de caixas laterais que fazem referencia a conceitos da disciplina de algoritmos.

A pesar dos inúmeros trabalhos voltados e não voltados ao ensino de algoritmos, mas que podem ser utilizados com tal propósito, a ferramenta *Takkou* foi proposta por apresentar diferenciais frente as já existentes. Dentre as ferramentas citadas as que mais se assemelham a discutida neste artigo são os jogos Robozzle, Light-Bot e Scracht. Para diferenciar a ferramenta *Takkou* das citadas especifica-se as diferenças entre elas e onde a ferramenta proposta é melhor as demais.

- Takkou* x Robozzle: a ferramenta *Takkou* apresenta algumas similaridades com a Robozzle, entretanto, a primeira possui características pedagógicas fundamentadas em teoria, o que não é o caso da segunda, conforme o autor nos informou via e-mail. Além disso, ao contrário do robuzzle, a ferramenta *Takkou* apresenta um ambiente potencialmente significativo no que concerne aos símbolos do dia a dia do usuário;

- Takkou* x Light-Bot: por ser a Light-Bot similar a Robozzle, pode se reiterar o

que já foi afirmado no item anterior. Além das similaridades a ferramenta citada não possui um ambiente muito intuitivo, e se propõe a ser apenas um jogo, mesmo que abordando de forma direta conceitos de algoritmos;

•*Takkou* x Scratch: a grande diferença entre estas duas ferramentas se encontra nas teorias pedagógicas utilizadas em cada uma e no seu propósito. A ferramenta Scratch, como já fora citado, possui base pedagógica no construtivismo, entretanto, a ferramenta *takkou* vai além por abordar uma teoria diferencial, a do Aprendizado Significativo. Além disso, a ferramenta Scratch se enquadra na categoria de ferramentas de uso geral, por não está diretamente ligada a utilização no ensino de algoritmos e não se apresenta como um jogo, mas como uma ferramenta que possibilita a criação de animações, vídeos, etc. Já a ferramenta *Takkou*, se apresenta ligada a teoria de David Asubel e com características de um jogo de puzzle, ou antes, uma jogo onde se resolvem *puzzles* voltados diretamente para o ensino de algoritmos mesmo que de forma tangencial;

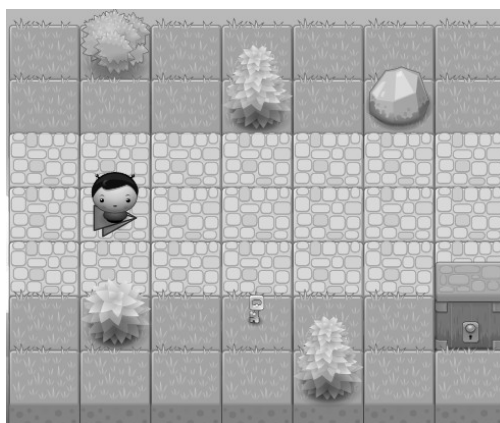
•*Takkou* x as demais ferramentas citadas: como pôde-se observar as ferramentas de linguagem e simulação apresentam-se como ferramentas técnicas. Já a ferramenta *Takkou* se enquadra no ramo das ferramentas com características de jogo, e, por sua vez, mais motivadora, partindo para uma abordagem diferencial, não técnica, mas lúdica.

Portanto, como fora explanado, percebeu-se que a ferramenta proposta neste artigo possui uma diferencial as já existentes, e, por isso, além de agregar um grande fator motivacional, apresenta-se como uma nova alternativa ao auxílio do ensino de algoritmos de forma lúdica.

#### **4. Takkou**

*Takkou* é uma ferramenta lúdica voltada ao ensino e prática de algoritmos e programação. Nela é preciso compor um bloco de instruções que, quando executado, seja possível cumprir uma determinada tarefa representada por um puzzle. O objetivo é fazer com o que o jogador cumpra com a tarefa em um menor tempo possível, refletindo uma situação em que ele terá de pensar rápido no desenvolvimento da lógica na resolução de um dado problema.

Cada puzzle contém um mapa representando um terreno e os obstáculos em que o jogador, representado por um boneco, deverá capturar itens que contribuem tanto para acúmulo de pontos quanto para o cumprimento da missão. Esse último tipo de item, representado por uma chave, deve ser capturado antes de o jogador ir em direção a um objeto de fim de jogo, representado por uma porta. Além da chave e da porta, outro fator que afeta a condição de fim de jogo é a presença de um temporizador, relógio, em que é definido um tempo para que o jogador complete a missão. Assim que o jogador começa a trabalhar na resolução do puzzle, ou seja, quando a missão é carregada, o relógio inicia uma contagem regressiva.



**Figura 1 : Ilustra o mapa do jogo com o boneco representando o jogador e a chave e a porta como itens de fim de jogo.**

A ferramenta foi projetada para possibilitar ao jogador uma interface simples e ao mesmo tempo eficiente. Para o que ela propõe, o jogador não deve ter dificuldades em compreender os elementos de interface assim como a sua manipulação, pois ele terá que focar apenas na missão do jogo.

Para fazer com que o *player*, boneco, se mova, deve-se montar uma estrutura formada por blocos de instruções. Uma vez inserido um bloco no painel, o mesmo pode ser movido de posição ou ser removido.



**Figura 2: As opções de comandos (do lado esquerdo) podem ser utilizadas para inserir blocos de instrução no painel (do lado direito).**

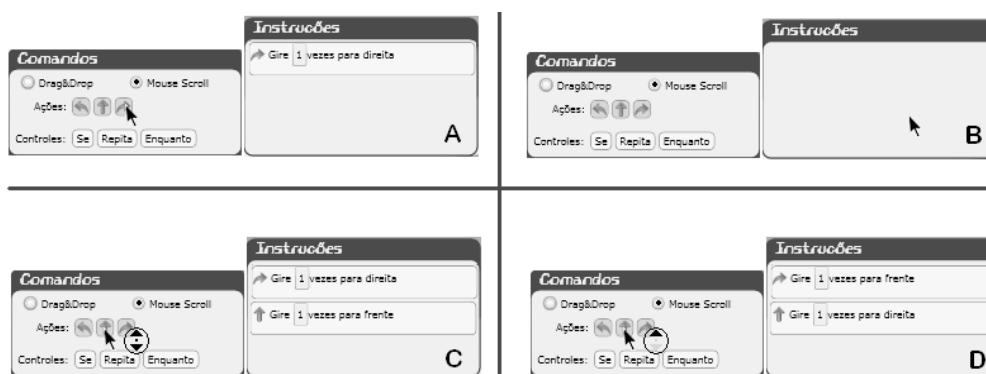
Os comandos são simples e fazem com que o *player* se mova em algumas direções como para frente, para o lado direito ou para o lado esquerdo. Além desses comandos há os comandos que permitem ao usuário checar uma dada condição de acordo com os elementos dispostos no mapa. É possível, por exemplo, verificar se há algum obstáculo no caminho em que se quer percorrer. E, por fim, é possível determinar quantas vezes determinado bloco de instrução deve ser executado através de um laço de repetição.

Durante a implementação dos comandos foram contrapostas duas abordagens em termos de eficiência. Levou-se em conta a seguinte questão: como fazer com que o jogador gaste pouco tempo manipulando os elementos de interface? Essa questão é bem colocada uma vez que a abordagem não deve afetar o tempo que se tem para cumprir a missão.

A primeira abordagem usa o artifício de *Drag and Drop*. Essa é a abordagem mais utilizada nos jogos de ensino de algoritmo. A sua maior vantagem é ser um mecanismo intuitivo. Em uma abordagem *Drag and drop* é dado ao usuário a

possibilidade de clicar em um botão, arrastá-lo e soltá-lo no painel de instruções. Muitas vezes, porém, o manuseio dos comandos torna-se um pouco demorado, o que acaba desmotivando o jogador e atrapalhando o objetivo da ferramenta, que no caso da *Takkou* é conseguir uma solução em um determinado tempo. Há, então, uma segunda abordagem que se baseia no uso do *scroll*, botão de rolagem do meio do mouse, para manipular os blocos de instrução. Apesar de ser menos intuitiva do que o uso do mecanismo de arrastar e soltar, ela se mostra mais adequada para a *Takkou*. Essa abordagem foi implementada disponibilizando um recurso de pré-visualização dos blocos no painel que é habilitado quando o mouse é passado por cima de algum comando. Para inserir um bloco no painel é necessário apenas um único movimento do *scroll*. A vantagem desta abordagem é que, quando não é movimentado o ponteiro do mouse, é possível deslocar o bloco recentemente inserido, mudando a sua ordem de execução. Assim, o jogador gastará menos tempo para deslocar um bloco. Na abordagem *Drag and drop* ele teria de mover o ponteiro do mouse até o bloco que se queira mudar de posição e arrastá-lo.

A Figura 3 mostra o funcionamento desta abordagem. Na situação **A** o mouse encontra-se apenas sobre o botão correspondente a instrução de “seguir em frente” e por isso o bloco é, temporariamente, inserido no painel até que o mouse saia da área do botão (situação **B**). Na situação **C** o *scroll* é movido uma vez e o bloco é inserido definitivamente no painel. Quando movido o *scroll* do mouse mais uma vez, a ordem do bloco em foco no painel é mudada de acordo com o sentido em que o *scroll* foi movido – para cima ou para baixo. Essa ação é representada na situação **D**.



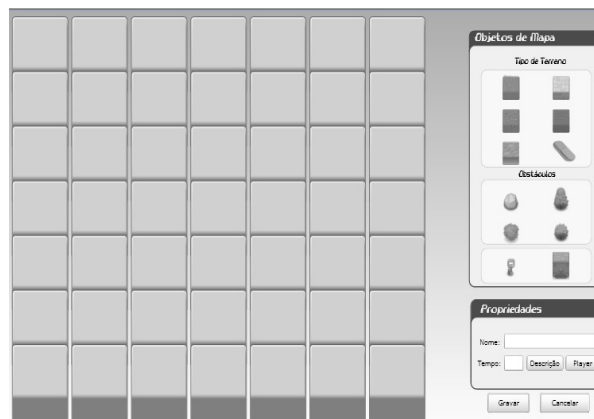
**Figura 3: Ilustra as quatro situações possíveis com o uso do mouse scroll.**

#### 4.1. Criação de mapas

A criação de mapas ou *puzzles* é outro diferencial na *Takkou* possibilitando a criação de uma missão. Ao criar um mapa o usuário pode definir todos os elementos que farão parte da missão. Desta forma, quem o cria pode definir qual nível de “programação” deseja-se explorar do jogador, podendo construir mapas mais complexos ou simples.

Para criar um mapa, primeiro, é escolhida a sua dimensão, após esta definição o usuário deve construir os mapas que se dividem em: terreno, obstáculos e itens. No mapa de terreno encontram-se disponíveis os tipos de terreno: água, terra, pedra ou grama; no mapa de obstáculo: pedra e arbustos; e, por fim, no de itens, que servem tanto para o jogador acumular pontos quanto para contribuir com as definições de fim de jogo, são definidos os itens: frutos, a chave e a porta.





**Figura 4: Página de construção de mapas de missão.**

Na figura 4 ilustra-se a página de construção de mapas. À direita há as opções de itens que são usados para preenchê-lo. Os itens de fim de jogo são considerados obrigatórios, uma vez que sem eles o jogo não teria um objetivo a ser atingido. À esquerda está o mapa inicialmente sem nenhuma definição de preenchimento. Ao clicar em alguma opção de item (à direita) e em seguida clicar numa área do mapa, o item é incluso. Após a construção do mapa, pode-se inserir uma descrição da missão, que é ilustrada assim que o jogador a carrega; um nome e um tempo limite de conclusão da missão e a posição inicial do jogador.

## 5. Avaliação Preliminar

Em uma avaliação preliminar a ferramenta *Takkou* mostrou-se satisfatória frente ao fator motivacional. A isso atribui-se o modo como a ferramenta aborda os conceitos de algoritmos, de forma tangencial. Por ser uma ferramenta protótipo ainda em desenvolvimento, não foi possível realizar uma avaliação profunda do seu uso frente aos usuários.

Em trabalhos futuros, intenta-se aprimorar a ferramenta no que se refere aos elementos que a compõe tornando o uso direcionado através da especificação de níveis mais detalhado. Outra proposta para o futuro é adequar a ferramenta para ser disponibilizada para rodar em celulares. Dessa forma, possibilitando aos usuários fazer uso da ferramenta em qualquer lugar e a qualquer hora, retirando, deste modo, a obrigatoriedade do usuário estar conectado em um computador para treinar suas habilidades.

## 6. Conclusões Finais

A ferramenta *Takkou* se apresenta como uma alternativa diferenciada das demais como forma de auxiliar os alunos de computação no exercício de suas habilidades algorítmicas concernentes ao raciocínio lógico. Como já fora citado em Campos et al. (2011), a ferramenta surgiu como uma abordagem frente ao problema enfrentado por inúmeros alunos de sentirem dificuldades em raciocinar logicamente na resolução de problemas. A forma como a ferramenta se apresenta foi proposta com o intuito de melhor motivar os seus usuários. Houve, também, a preocupação em torno da usabilidade por parte do usuário criando-se mecanismos de facilitação no manuseio da ferramenta como *drag and drop* e uso do botão central do mouse, *scroll*, para se adicionar comandos a serem

executados.

Além das características já citadas, a *Takkou* se distancia das outras ferramentas por se apresentar com uma abordagem diferenciada de ensino e auxílio ao aprimoramento das habilidades na disciplina algoritmos. Por se apresentar como uma ferramenta com uma série de jogos, *puzzles*, esta encoraja o usuário a aprender algoritmos sem tomar conhecimento que assim está fazendo, ao menos não de forma direta.

Por fim, acredita-se que no que se propõe esta ferramenta ela atende em sua totalidade os objetivos almejados: motivar os alunos a exercitar os conhecimentos de algoritmos; buscar conhecer mais sobre resolução de problemas; tornar a prática do exercício em algoritmos mais proveitosa e prazerosa.

## Referencias

- Campos, A. M., Barbosa, L. S. and Fernandes, T. C. B. (2011). A new approach for teaching algorithms . Submitted to The 2011 International Conference on Frontiers in Education: Computer Science and Computer Engineering.
- Hinterholz, JR, O. (2009). Tepequém: uma nova Ferramenta para o Ensino de Algoritmos nos Cursos Superiores em Computação. XVII WEI – Anais do Workshop sobre Educação em Informática. Bento Gonçalves – RS.
- Jesus, A. e Brito, G.S. (2009). Concepção de Ensino-aprendizagem de Algoritmos e Programação de Computadores: A Prática Docente. I ENINED - Encontro Nacional de Informática e Educação. Universidade Estadual do Oeste do Paraná (UNIOESTE). Cascavel, Paraná.
- Maloney, J., Peppeler, K. and Kafai, Y. Resnick, M. Rusk, N. (2009). [Programming by Choice: Urban Youth Learning Programming with Scratch. In ACM SIGCSE Bulletin archive, Vol 40, 367-371, ISSN:0097-8418, 2008.](#)
- Moreira, M.A. e Masini, E.F.S. (2006). Aprendizagem Significativa - A Teoria de David Ausubel. São Paulo: Editora Centauro.
- Mota, M.P., Pereira, L.W.K. e Favero, E.L. (2008) . JavaTool: Uma Ferramenta para o Ensino de Programação. In: [Congresso da Sociedade Brasileira de Computação. Belém. XXVIII Congresso da Sociedade Brasileira de Computação. p. 127-136.](#)
- Pelizzari, A. et al. (2002) Teoria da aprendizagem significativa segundo Ausubel. Rev.PEC, Curitiba, v.2, n.1, p.37-42.
- Rocha, H. (1993). “Representações Computacionais Auxiliares ao Entendimento de Conceitos de Programação”, In: [“Computadores e Conhecimento: Repensando a Educação”](#). Livro organizado por Valente, J. A. Editora Unicamp.
- Santiago, R. (2004). Ferramenta de apoio ao ensino de algoritmos. XII Seminário de Computação. Blumenau/SC.
- Souza, C.M. (2009). [VisuAlg – Ferramenta de Apoio ao Ensino de Programação. Revista TECCEN – Vol. 2, no.2; ISSN 1984-0993.](#)