

# MineScratch: Integração Minetest-Scratch para Apoiar o Ensino de Programação

Jhonata da Rocha<sup>1</sup>, Jean C. R. Hauck<sup>1</sup>

<sup>1</sup> Departamento de Informática e Estatística (INE)  
Universidade Federal de Santa Catarina (UFSC)  
Caixa Postal 476 – CEP 88040-900  
Florianópolis – SC – Brasil

{jhonata11, jean.hauck}@ufsc.br

**Abstract.** *Despite the growing demand for IT professionals, most students in this area do not complete their undergraduate courses. Several initiatives have applied different tools and educational games, since elementary school, as support for computer education, in an attempt to change this scenario. Scratch has been one of the most used tools for teaching programming and Minetest is an open-world building block game, a very popular format today. Thus, this article presents the development of a library for integration between these two softwares to support programming teaching. The library is evaluated through the development and use of an educational game that uses it. The initial evaluation raises initial indications that there are improvements to be implemented, but the library is efficient, robust and functional.*

**Resumo.** *Apesar da crescente demanda por profissionais de TI, grande parte dos estudantes dessa área não conclui os cursos de graduação. Diversas iniciativas vêm buscando aplicar diferentes ferramentas e jogos educacionais desde o ensino fundamental, como apoio ao ensino de computação, no intuito de tentar mudar esse cenário. O Scratch tem sido uma das ferramentas mais utilizadas para ensino de programação e o Minetest é um jogo de construção em blocos de mundo aberto, num formato muito popular atualmente. Assim, este artigo apresenta o desenvolvimento de uma biblioteca de integração entre esses dois softwares para apoiar o ensino de programação. A biblioteca é avaliada por meio do desenvolvimento e uso de um jogo educacional que a utiliza. A avaliação inicial do trabalho levanta indícios iniciais de que existem melhorias a serem implementadas, mas a biblioteca mostra ser eficiente, robusta e funcional.*

## 1. Introdução

O Brasil possui o 5º maior mercado interno de tecnologia da informação segundo um estudo feito pela BRASSCOM (2013), tendo uma participação de 5,2% no PIB nacional. Entretanto, apesar de números positivos, um assunto é objeto de

discussões: a carência de profissionais qualificados. Essa carência de profissionais qualificados tem relação com a alta taxa de evasão dos cursos de tecnologia.

Segundo o observatório SOFTEX (2012), anualmente 21% dos estudantes matriculados abandonam seus cursos da área de TI. Isso pode estar aliado ao fato de que, no Brasil, as crianças e jovens não possuem contato com conceitos de computação em suas escolas, podendo reduzir o interesse em áreas afins.

Nesse sentido, um esforço crescente tem sido realizado para aumentar o ensino de conceitos de computação para crianças [Computação na Escola 2016]. Diversas ferramentas têm sido desenvolvidas para apoiar esse ensino, tais como o Scratch (<http://scratch.mit.edu>) e o AppInventor (<http://ai2.appinventor.mit.edu>), ambas permitindo a programação utilizando blocos coloridos, facilitando o aprendizado.

Jogos educacionais também têm sido utilizados com sucesso como apoio ao ensino das diversas áreas da computação [Battistella & Wangenheim 2016]. Um dos jogos mais populares nos últimos anos entre as crianças é o jogo Minecraft (<https://minecraft.net>), um jogo proprietário, baseado na construção de blocos, em mundo aberto que estimula a criatividade. Uma alternativa gratuita e de código ao Minecraft é o Minetest (<http://www.minetest.net/>), que possui funcionalidades semelhantes, apesar de restritas.

Assim, inspirado na popularidade de jogos de construção de blocos em mundo aberto, como o Minetest, e na facilidade de aprendizado do Scratch, este artigo apresenta a modelagem, implementação e avaliação de uma biblioteca que une estes dois softwares para apoio à construção de jogos educacionais para ensino de programação.

Este artigo apresenta na seção 2 os trabalhos correlatos, seguida da seção 3 onde Minetest e Scratch são brevemente apresentados. Na seção 4 a modelagem e desenvolvimento da biblioteca são apresentados e na seção 5 a avaliação da biblioteca é também apresentada. Por fim, a seção 6 traz as conclusões e trabalhos futuros.

## **2. Trabalhos Correlatos**

Para identificar os trabalhos correlatos, foi realizada uma Revisão Sistemática de Literatura (RSL) seguindo a abordagem proposta por Kitchenham (2009). A seguir uma visão geral do planejamento, execução e resultados da RSL é apresentada.

A principal pergunta de pesquisa definida para a RSL foi: “Como são as bibliotecas de integração com Minetest e/ou com Scratch para apoio ao ensino de programação?”. A partir da definição desta pergunta de pesquisa foram derivados os termos de busca, definida a base de pesquisa como o Google Scholar (<http://scholar.google.com>) e elaborada a String de busca.

A String de busca, foi definida como: (*"scratch" OR "programação visual" OR "visual programming"*) OR (*"minetest" OR "minecraft"*) AND (*"integração" OR "integração de software" OR "integração de sistemas" OR "software integration" OR "system integration"*) AND (*"biblioteca" OR "biblioteca de software" OR "módulo" OR "módulo de software" OR "biblioteca estática" OR "library" OR "software library" OR "module" OR "software module" OR "static library"*) AND (*"ensino de programação" OR "ensino de lógica de programação" OR "ensino de algoritmos" OR "ensino de*

*construção de algoritmos" OR "Programming education" OR "programming logic education" OR "algorithm education" OR "algorithm building education") PERÍODO (2010 a 2016).*

Como critérios de inclusão e exclusão buscando selecionar os trabalhos relacionados a ferramentas que façam utilização do Scratch (ou de qualquer forma de programação visual) ou do Minetest (ou Minecraft) por meio de uma integração. A busca foi então realizada em três iterações, aplicando os critérios de inclusão e exclusão e, dos 800 trabalhos encontrados, foram considerados relevantes somente 3 trabalhos.

A maioria dos trabalhos não considerados relevantes, foram descartados por não se tratarem do desenvolvimento de uma integração entre as ferramentas, mas somente relatando experiências de utilização das ferramentas, principalmente o Scratch, de forma pedagógica no ensino de programação. Alguns exemplos interessantes são: o trabalho de Lomba (2014), relatando a utilização do Scratch como estratégia de ensino e aprendizagem da programação, como uma experiência no âmbito do ensino profissional; um relato de experiência da utilização do Scratch nas aulas de Lógica de Programação do Proeja, de Alencar (2014), e; dominando para não ser dominado: Autonomia tecnológica com o Projeto Jovem Hacker, apresentado em Amiel et al. (2015).

Dentre os três trabalhos selecionados como relevantes na RSL, pode-se destacar um Mod (extensão ou modificação do Minetest) que permite programar as ações do personagem no Minetest: *Visual Programming for Minetest* [Pruss 2015]. Este Mod utiliza a ferramenta de programação visual Blockly (<https://developers.google.com/blockly/>) como forma de programação visual, e é desenvolvido nas linguagens de programação Python e Lua.

Outro trabalho interessante é apresentado por Yorinks (2016), integrando Scratch com Arduino (<https://www.arduino.cc/>) por meio da extensão *s2a\_fm*, que consiste em uma ferramenta para o controle de microprocessadores Arduino utilizando programação visual. O projeto é desenvolvido na linguagem de programação Python. Já para fazer a comunicação com o Scratch, é utilizada a extensão HTTP, sendo que o servidor é criado na linguagem Python e se comunica com o Scratch por meio de requisições HTTP.

Um protótipo de aplicação, que utiliza a programação visual como forma de controle de mecanismos e sensores no Raspberry Pi (<https://www.raspberrypi.org/>) é apresentado por Heinen et al. (2014). O projeto faz o uso da ferramenta para programação visual Blockly para permitir a programação dos sensores. O trabalho foi avaliado por meio da utilização da aplicação em um ambiente de testes e os resultados foram positivos.

Nesta RSL não foi possível encontrar nenhum trabalho relatando uma integração entre Scratch e Minetest para apoio ao ensino de programação.

### **3. Minetest e Scratch**

Minetest (<http://www.minetest.net/>) é um jogo de código livre criado em 2010, composto por um ambiente virtual em 3D. Em sua versão 0.4.9 ele é disponível nas plataformas Windows, GNU/Linux e OSX. Funciona por meio de blocos cúbicos que podem se conectar uns aos outros, formando construções maiores e mais complexas, de

forma similar ao conhecido jogo Minecraft (<https://minecraft.net/>), como se pode ver na Figura 1.

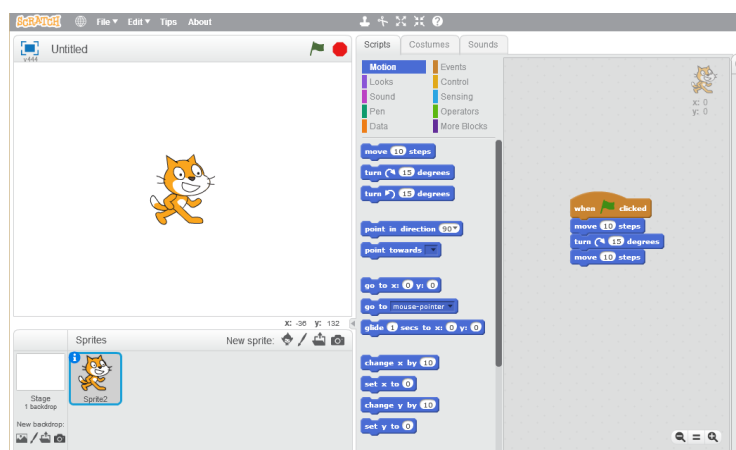
O Minetest possui uma API para a construção de Mods, que são qualquer alteração no funcionamento do jogo sem modificação do código fonte, desenvolvidas em Lua (<https://www.lua.org/portugues.html>) na forma de extensões plugáveis.



**Figura 1 – Exemplo de mapa no Minetest (MINETEST, 2016)**

O Minetest pode ser jogado por um jogador ou em *multiplayer*. Ele possui ainda uma API de comunicação, o que permite que seja integrado a outros softwares. Sua comunicação se dá por meio do protocolo UDP. Essa interface de comunicação é utilizada para a integração apresentada neste trabalho.

O Scratch (<https://scratch.mit.edu/>) é uma plataforma de desenvolvimento visual de programas criada pelo Instituto de Tecnologia de Massachusetts. Nele é possível a construção de algoritmos por meio de blocos de instruções visuais, permitindo programar graficamente, o que tem se mostrado atrativo para crianças. A Figura 2 apresenta a tela inicial do Scratch.



**Figura 2 - Tela inicial do Scratch [Scratch 2016]**

O Scratch possui versões online e off-line. No editor off-line, existe a funcionalidade de importar uma extensão, que é uma forma do Scratch permitir a conexão com outros softwares por meio de requisições HTTP. Esta funcionalidade é utilizada na construção da biblioteca proposta por este trabalho.

## 4. Desenvolvimento da biblioteca e do jogo

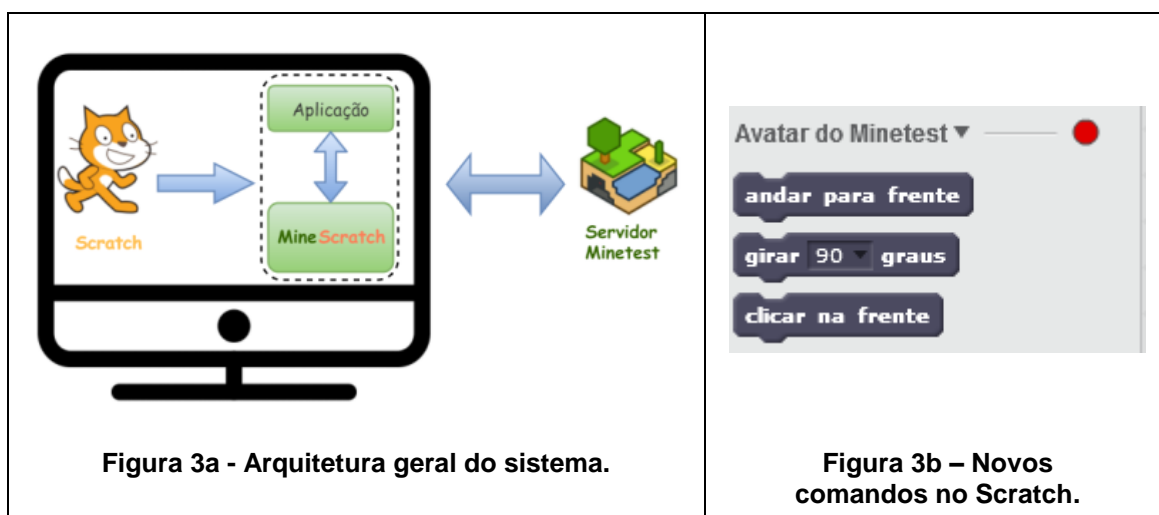
O objetivo da biblioteca apresentada neste trabalho é possibilitar uma integração entre os softwares Minetest e Scratch. Assim, ela foi nomeada como: MineScratch. Nesta seção o desenvolvimento da biblioteca é brevemente apresentado, bem como o desenvolvimento de um jogo educacional para permitir a avaliação da biblioteca.

### 4.1 Desenvolvimento da biblioteca

Para permitir a integração entre as duas ferramentas como apoio ao ensino de programação, suas principais funcionalidades foram definidas como:

- Criação de conexão com o Scratch.
- Autenticação em um servidor Minetest.
- Criação de instruções que podem ser recebidas no Scratch.
- Criação de comandos para serem enviados ao Minetest.
- Envio e recebimento de comandos do Minetest.
- Recebimento de instruções vindas do Scratch.
- Registro das informações do personagem durante a conexão.

A partir dessas funcionalidades identificadas, foi documentado um conjunto de requisitos, que deram origem à arquitetura apresentada na Figura 3a. A biblioteca MineScratch foi dividida em 3 módulos, sendo 2 deles os principais: módulo Scratch e módulo Minetest. O módulo Scratch faz todo o gerenciamento das requisições HTTP provenientes da extensão importada no Scratch (Figura 3b). Já o módulo Minetest gerencia a conexão com o Minetest, além de manter informações sobre o personagem.



A biblioteca foi então desenvolvida na linguagem de programação Java, gerando, ao final um arquivo com a extensão *jar*. Dessa forma, para que uma aplicação faça uso dessas funções, basta estender as classes necessárias. O código-fonte do MineScratch pode ser encontrado em <https://github.com/jhonata11/MineScratch>.

## 4.2 Desenvolvimento do jogo

O jogo educacional que utiliza o MineScratch é desenvolvido utilizando a abordagem para desenvolvimento de jogos educacionais ENgAGED (Battistella & Wangenheim 2015). O jogo é desenvolvido em forma de labirinto, com regras simples, onde o jogador/aluno deve programar as ações e movimentos do personagem até alcançar determinado objetivo, afim de receber uma pontuação. Para cada jogador é instanciado um personagem no Minetest, e o jogador controla suas ações por meio da programação no Scratch.

O jogo educacional implementado para testar a biblioteca trata-se de um sistema modular, uma vez que possui diversos subsistemas autossuficientes que se comunicam entre si de forma autônoma. Um dos tipos de módulos existentes são as bibliotecas de aplicações [Meyer 1988]. Elas são um conjunto de funções implementadas em alguma linguagem de programação. Elas podem ser categorizadas em [Yacoub et al. 1999]: Bibliotecas estáticas: são aquelas que possuem seus códigos invocados durante a construção do programa. Bibliotecas dinâmicas: possuem seus códigos invocados após a construção do programa, seja no início da execução ou no meio.

Assim, os principais componentes do jogo desenvolvido são:

- Minetest:
  - Um mapa criado especialmente para o jogo (vide Figura 5).
  - Um script Lua para o gerenciamento da pontuação.
- Scratch:
  - Uma extensão HTTP para a criação de novas instruções que podem ser utilizadas pelo aluno.
- Uma aplicação-ponte que faz uso da biblioteca MinteScratch.



Figura 4 – Mapa do jogo no Minetest

## 5. Avaliação da Biblioteca

Após os testes unitários e de integração da biblioteca e do jogo desenvolvidos, a biblioteca foi avaliada em duas etapas: primeiramente em ambiente de laboratório e depois por usuários utilizando o jogo desenvolvido. Utilizando-se a abordagem GQM – *Goal/Question/Metric* [Basili et al. 1994], foram definidos três objetivos de avaliação, contemplando os seguintes critérios: Eficiência, Robustez e Funcionalidade, a partir das características de qualidade esperadas de boas bibliotecas [Sarkar et al. 2007]. Para cada objetivo de medição, seguindo a abordagem GQM, foram definidas perguntas, medidas e um questionário como instrumento de coleta de dados.

### 5.1 Primeira aplicação

A primeira aplicação, em ambiente controlado, teve o objetivo de obter os resultados no quesito Eficiência. Foram realizadas oito execuções completas e obtidas suas medidas, principalmente com relação ao tempo de execução dos comandos recebidos do Scratch, coletados por meio do registro de logs da biblioteca.

A Figura 5 apresenta os resultados da primeira aplicação. Em azul, observa-se o tempo desde o momento em que uma instrução é acionada no Scratch até a biblioteca receber sua requisição. Já em laranja é possível observar o tempo desde o envio da biblioteca de um comando até sua execução no Minetest. É possível observar que o tempo de execução dos comandos no Minetest são relativamente altos, o que se deve, em grande parte à comunicação entre o gerenciamento das requisições HTTP para o Scratch e a comunicação com o Minetest, que utiliza comunicação UDP de baixo nível. Essa comunicação precisa ser melhorada em versões futuras.

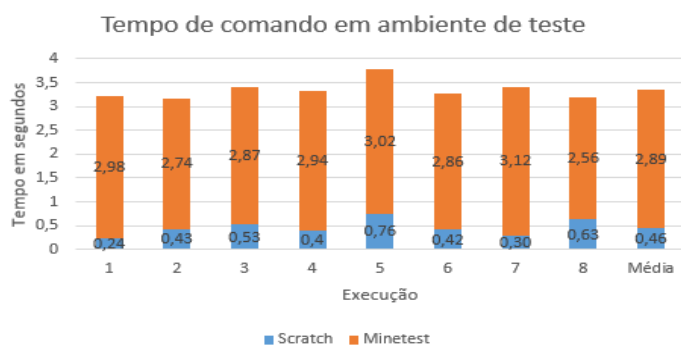
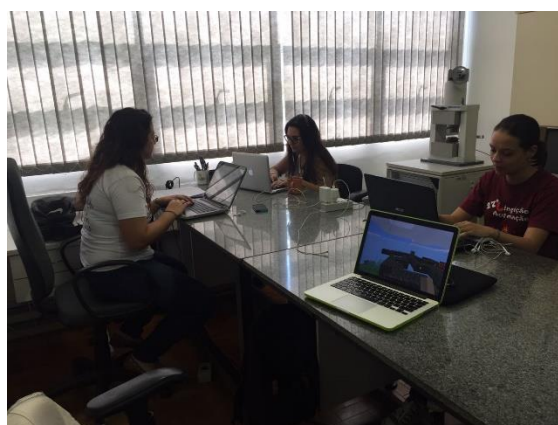


Figura 5 - Resultados da primeira aplicação.

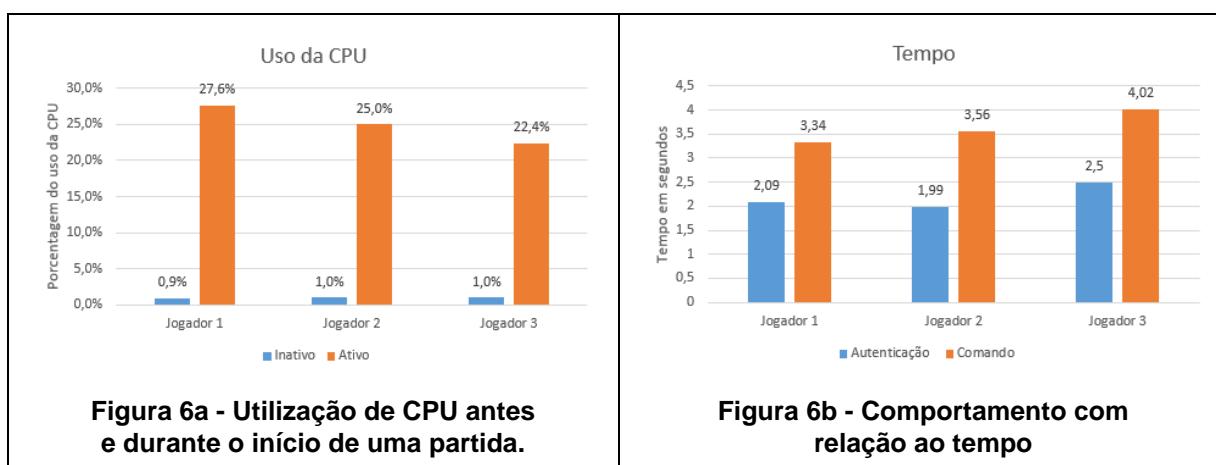
### 5.2 Segunda aplicação

A segunda aplicação foi realizada em um laboratório de informática, contando com 3 estudantes da Universidade Federal de Santa Catarina, sem experiência prévia com programação. Cada usuário possuía 1 computador contendo a biblioteca, a aplicação e as aplicações Scratch e Minetest. Os usuários participaram de três rodadas do jogo, e ao fim responderam o questionário (Figura 6). Os resultados coletados são apresentados a seguir, relativos a cada objetivo de medição.



**Figura 6 - Aplicação do jogo que utiliza o MineScratch**

Com relação à **Eficiência** na utilização de recursos, a biblioteca se mostrou satisfatória, já que não houve sobrecarga das máquinas durante sua utilização. Já com relação ao comportamento em relação ao tempo, assim como na primeira aplicação, foi possível notar que o tempo de comunicação também foi alto, principalmente na comunicação com o Minetest. Nas Figuras 6a e 6b são apresentados os resultados de forma gráfica.



Com relação à **Robustez**, durante a aplicação houve uma queda de rede. Nenhum usuário se mostrou insatisfeito com a forma como a biblioteca se comportou, avisando sobre a queda de energia. Contudo, uma sugestão apontada foi tentativa de reconexão.

Sobre a **Funcionalidade**, foi apontada nas respostas do questionário a dificuldade de trabalhar com diversos sistemas ao mesmo tempo, pois vários programas funcionavam em paralelo, e o usuário deve trabalhar com todos. Contudo, este não é um problema especificamente da biblioteca, mas sim do jogo em si, que necessita que o jogador utilize o Scratch para programar e a tela do Minetest para verificar o resultado do conjunto de comandos enviados.



### 5.3 Ameaças à validade da avaliação

A principal ameaça à validade da avaliação é o fato da biblioteca ser avaliada indiretamente, por meio de seu uso na construção de um jogo educacional. De fato, uma forma de melhor avaliá-la seria aplicá-la na construção de aplicações por outros desenvolvedores que não o autor, porém, pela complexidade de desenvolvimento de um software, se tornaria inviável no tempo desenvolver várias aplicações diferentes.

Outra ameaça se refere ao tamanho da amostra, de apenas três estudantes. Esse número foi limitado à disponibilidade dos participantes para a avaliação. Novas avaliações com maior número participantes estão previstas em trabalhos futuros.

Em relação ao ambiente de aplicação por não ser exatamente o que seria encontrado em uma aplicação em sala de aula ou laboratório de informática. Mas, para simular uma rede instável, esta foi manualmente desconectada durante os testes.

## 6. Conclusão e Trabalhos Futuros

Este artigo trata do desenvolvimento de uma biblioteca de integração entre Scratch e Minetest para apoiar o ensino de programação. Após o levantamento do estado da arte por meio de uma Revisão Sistemática de Literatura, a biblioteca é modelada e desenvolvida. Um jogo educacional simples utilizando a biblioteca também é desenvolvido, como prova de conceito, para permitir a avaliação da biblioteca.

A biblioteca desenvolvida foi avaliada em duas etapas, primeiramente com testes em ambiente controlado e, posteriormente, sendo utilizado o jogo desenvolvido com base na biblioteca. Nessas avaliações a biblioteca se mostrou eficiente, robusta e funcional. Porém nos testes foi percebido que poderia ser otimizada a comunicação com o Minetest.

Levando em consideração as informações obtidas durante o desenvolvimento deste trabalho, seriam possíveis trabalhos futuros a melhorar da eficiência na comunicação com o Minetest, por meio da otimização nos algoritmos de rede e do gerenciamento de múltiplas *threads*; a possibilidade de alteração da utilização do Scratch para a biblioteca Blockly, diretamente integrada no Minetest, e; a aplicação do MineScratch com uma amostra significativa, tornando possíveis análises estatísticas dos dados referentes ao seu uso.

## Referências

- Amiel, T., de Souza, F. G., Arantes, F. L., Aguado, A. G. Dominando para não ser dominado: Autonomia tecnológica com o Projeto Jovem Hacker. Workshop de Software Livre, 2015.
- Alencar, G A.; Freitas, A. K.; Danielle, J. S. Utilizando o SCRATCH nas aulas de Lógica de Programação do Proeja: Um relato de experiência. Nuevas Ideas en Informática Educativa - TISE, 2014.
- Basili, V. R., G.; Caldiera, H. D.; Rombach. Goal/Question/Metric Approach. In J. Marciniak (ed.), Encyclopedia of Software Engineering, volume 1. John Wiley & Sons, 1994.

- Battistella, P. E.; von Wangenheim, C. G. Games for Teaching Computing in Higher Education? A Systematic Review. *IEEE Multidisciplinary Engineering Education Magazine*, v. 9, p. 8-30, 2016.
- Battistella, P. von Wangenheim, C. G. ENgAGED: Processo de Desenvolvimento de Jogos para Ensino em Computação. Relatório técnico. Instituto Nacional para Convergência Digital, Universidade Federal de Santa Catarina, Florianópolis, 2015.
- BRASSCOM, 2013. Mercado Brasileiro de TI <<http://www.brasscom.org.br/brasscom/Portugues/detInstituodCategoria=21>>. Acesso em: 21 jul. 2015.
- Computação na Escola. < <http://www.computacaonaescola.ufsc.br/>>. Acesso em: 23 dez. 2016.
- Heinen, E.; Lauer, A. G.; Lealdino Filho, P. Protótipo de Aplicação de Programação Visual para Controle de Mecanismos e Sensores com Raspberry-Pi e Node.js. III Semana Acadêmica de Tecnologia em Sistemas para Internet. Guarapuava, 2014.
- Kitchenham, B. A. Systematic literature reviews in software engineering. *Information and Software Technology*, vol. 51, no. 1, p. 7-15, 2009.
- Lomba, C. A. A utilização do Scratch como estratégia de ensino e aprendizagem da programação: uma experiência no âmbito do ensino profissional. 2014
- Meyer, B. Object-oriented software construction. New York: Prentice hall, 1988.
- Pruss, A. Raspberryjammod-minetest, disponível em: <<https://github.com/arpruss/raspberryjammod-minetest/>>. Acessado em 14 out. 2016.
- Sarkar, S.; Rama, G. M.; Kak, A. C. API-based and information-theoretic metrics for measuring the quality of software modularization. *IEEE Transactions on Software Engineering*, v. 33, n. 1, 2007.
- Scratch. Disponível em <<http://scratch.mit.edu/>>. Acessado em: 21 jul. 2016.
- SOFTEX. Software e Serviços de TI: A indústria brasileira em perspectiva. 2012. Disponível em: <<http://www.softex.br/wp-content/uploads/2013/07/2012-Observatorio-Softex-Industria-Brasileira-Software-Servicos-TI-em-perspectiva-Versao-Completa-Portugues.pdf>>. Acessado em 21 jul. 2015.
- Yacoub, S.; Ammar, H.; Mili, A. Characterizing a software component. In: *International Workshop on Component-Based Software Engineering*. 1999.
- Yorinks, A. S2A\_FM. Disponível em: <[https://github.com/MrYsLab/s2a\\_fm](https://github.com/MrYsLab/s2a_fm)>. Acessado em 14 out. 2016.