iVProg: Programação Interativa Visual e Textual na Internet

Igor Moreira Félix[1], Lucas Mendonça de Souza[1], Leônidas de Oliveira Brandão[1], Bernardo Martins Ferreira[1], Anarosa Alves Franco Brandão[2]

[1]Instituto de Matemática e Estatística – Universidade de São Paulo (USP)
São Paulo – SP – Brasil

[2]Escola Politécnica – Universidade de São Paulo (USP)
São Paulo – SP – Brasil

igormf@ime.usp.br, lucasmens@ime.usp.br, leo@ime.usp.br,
bernardomartinsf@ime.usp.br, anarosa.brandao@poli.usp.br

Resumo. *Ensinar e aprender programação não é uma tarefa simples. Uma prova dessa dificuldade são os altos índices de reprovação em disciplinas que introduzem os conceitos de algoritmo ou de uma linguagem específica. Neste artigo, apresentamos o iVProg, uma ferramenta educacional com um ambiente visual e interativo, focado no aprendizado de algoritmos e programação. Seus principais recursos incluem avaliador automático, integração com o Moodle e uma série de elementos gráficos que dispensam a codificação textual para a criação de algoritmos, deixando os estudantes livres para praticarem seu raciocínio computacional.*

1. Cenário de Uso

O aprendizado de programação não é uma tarefa fácil [Boulay 1986], pois exige do iniciante uma série de habilidades e conhecimentos, dos quais ele não está habituado [de-la Fuente-Valentín et al. 2013]. Estes itens incluem, desde a compreensão do funcionamento do computador, raciocínio lógico e do domínio da sintaxe e semântica da linguagem de programação que se está aprendendo. As dificuldades encontradas passam ainda pelas questões associadas à pragmática da programação, que abrangem as habilidades de abstração de problemas, codificação, testes e depuração utilizando-se as ferramentas empregadas [Boulay 1986].

Estas dificuldades enfrentadas pelos estudantes se refletem nos índices de reprovação e desistência em disciplinas nos diferentes níveis educacionais. Por exemplo, [Bosse e Gerosa 2015] analisaram os resultados dos estudantes de graduação da disciplina MAC115 de introdução a programação da Universidade de São Paulo, entre os anos 2010 e 2014, encontrando um dado alarmante, de que mais de 50% dos estudantes não obtiveram êxito no curso. Além disso, [Souza et al. 2016] observa que a dificuldade na disciplina de introdução a computação pode inclusive ser o motivo de desistência no curso no qual o estudante está matriculado.

Para tentar mitigar este problema, esforços têm sido voltados para o desenvolvimento de iniciativas que fujam do modelo tradicional de ensino e possam estimular tanto os estudantes quanto os professores, como a utilização de jogos educacionais [Arawjo et al. 2017], a computação desplugada [Ferreira et al. 2015] e a programação visual [Broll et al. 2017].

Esta última, a programação visual, tem ganhado destaque por estimular o pensamento computacional [Broll et al. 2017] e facilitar o aprendizado de conceitos de programação [Sáez-López et al. 2016]. Muitos ambientes de programação visual se utilizam de um espaço lúdico e interativo, permitindo que os usuários elaborem algoritmos pela simples manipulação de elementos gráficos, onde exige-se pouca ou nenhuma codificação textual, por exemplo, o iVProg [Brandao et al. 2018], que é apresentado neste artigo.

DOI: 10.5753/cbie.wcbie.2019.1164

Ao adotar a utilização de um ambiente de programação visual, o aluno pode focar-se na elaboração do raciocínio algorítmico, diminuindo a sua carga mental relacionada à sintaxe das linguagens de programação tradicionais.

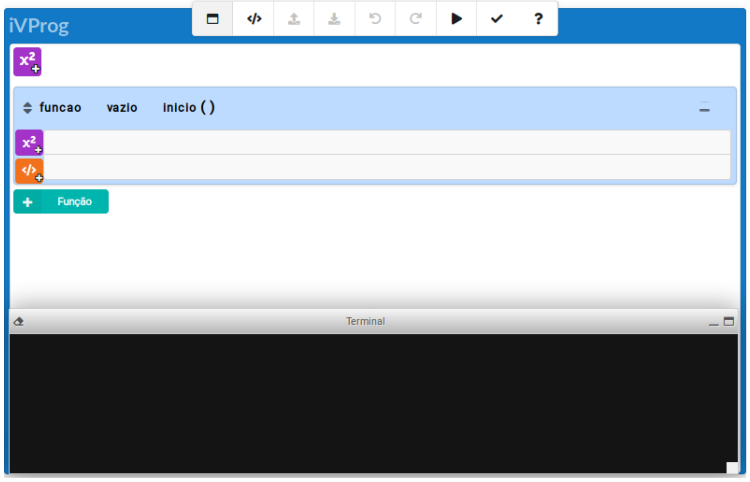

Figura 1. Tela principal do iVProg

Desta forma, a ferramenta educacional iVProg (figura 1) foi concebida, buscando oferecer tanto para o professor, quanto para o aluno, um ambiente visual e interativo focado no aprendizado de algoritmos e programação. Entre seus recursos, destacam-se o avaliador automático de exercícios e a total integração com o ambiente gerenciador de cursos Moodle. Estes e os demais recursos são apresentados no tópico 3.

O iVProg é desenvolvido especificamente para ser executado em navegadores *Web*, utilizando-se para isso as tecnologias HTML5, JavaScript e CSS. Além disso, o iVProg faz parte da comunidade de software livre, sendo seu código aberto e distribuído gratuitamente.

2. Desenvolvimento

A versão do iVProg (Programação Visual interativa na Internet) apresentada neste artigo teve seu desenvolvimento iniciado em agosto de 2018, sendo produto de código aberto e gratuito elaborado pelo Laboratório de Informática na Educação do Instituto de Matemática e Estatística da Universidade de São Paulo, disponível em: *https://www.usp.br/line/ivprog/*.

Para melhor gerenciar o desenvolvimento do iVProg, o processo foi dividido em três partes: interface gráfica de usuário, núcleo de interpretação de código e integração com o Moodle. As tecnologias empregadas em comum para as três partes do iVProg incluem o tripé da *Web* 2.0: HTML5 (Hypertext Markup Language, versão 5), JavaScript e CSS (Cascading Style Sheets). As divisões do desenvolvimento são detalhadas a seguir.

2.1. Interface gráfica de usuário

A fim de garantir que o aprendiz explore a ferramenta e seus recursos, a interface gráfica do iVProg foi elaborada pensando na interatividade do usuário com o algoritmo que está sendo elaborado. Além disso, o iVProg minimiza possíveis erros de lógica e programação, corroborando para os princípios do paradigma de programação visual [Sykes 2007].

A interatividade da ferramenta é garantida pela disposição dos elementos gráficos na tela e a possibilidade de todos os componentes do algoritmo serem manipulados. Na imagem 2, é apresentado um algoritmo que recebe dois números inteiros e em seguida imprime a soma entre eles. Na ilustração, o usuário está alterando a ordem de leitura das variáveis. Todas as operações são realizadas na parte visual do iVProg, não sendo necessária intervenções de código textual por parte do aprendiz.

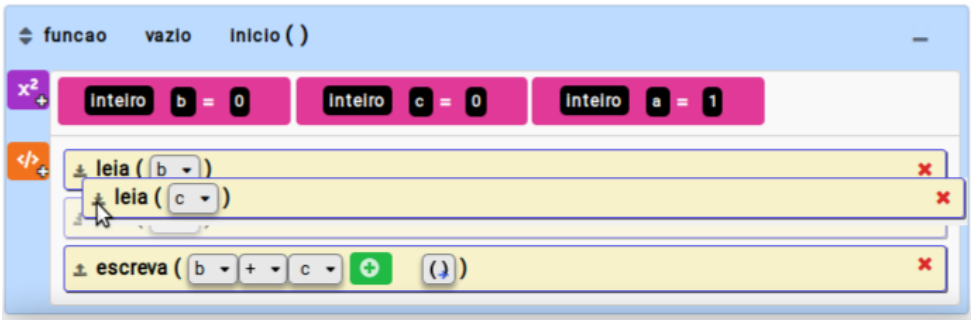

Figura 2. Interação com os comandos no iVProg

Para a parte gráfica do iVProg, foram utilizados as seguintes bibliotecas:

- jQuery versão 3.3.1: destinado à manipulação de forma simplificada dos elementos HTML presentes no iVProg.
- jQuery UI versão v1.12.1: dentro do iVProg é utilizado para a criação e gerenciamento de alguns elementos gráficos.
- SortableJS: garante a movimentação dos elementos visuais, desde os blocos das funções, como também as variáveis e os comandos que compõem os algoritmos.
- Semantic UI versão 2.4.2: principal biblioteca responsável pelo visual do iVProg, sendo empregada em ícones, blocos, gerenciamento de eventos e outros.

2.2. Núcleo de interpretação de código

O núcleo de interpretação é composto por diversos módulos e é responsável pela validação e interpretação do código gerado pela interface visual. Na parte de validação, um avaliador sintático e semântico é executado sobre o texto recebido, de acordo ao idioma selecionado, para garantir que o código recebido é válido e pode ser executado. Todavia, a verificação semântica feita nessa fase ainda é rasa, pois alguns aspectos só podem ser avaliados em tempo de execução. O processo de validação tem como resultado uma árvore abstrata sintática (AST) que representa o código e permite a sua execução.

Durante o processo de execução, interfaces dentro do núcleo abstraem os processos de entrada e saída. Essas interfaces permitem que as entradas fornecidas pelo usuário sejam recebidas pelo núcleo de processamento. Além disso, essas mesmas interfaces permitem que as saídas geradas, além dos alertas e erros, também possam ser apresentadas ao usuário da mesma forma.

Dentro do núcleo também reside o módulo de avaliação automática. Esse módulo utiliza os dados da atividade, quando fornecido, para verificar se o programa gera as saídas esperadas a partir das entradas fornecidas. O resultado da verificação é transmitido ao estudante e possui uma interface que permite ao professor armazenar essa informação.

Por ser multi-idioma, o núcleo permite que traduções sejam fornecidas não só para a linguagem como também para as funções definidas nas bibliotecas das mesmas. Um módulo para gerenciar textos internacionalizáveis foi implementado exatamente para prover essa funcionalidade.

O núcleo do iVProg utiliza as seguintes bibliotecas:

- antlr4 versão 4.7.2: é responsável pelo avaliador léxico da linguagem utilizada iVProg.
- decimal.js versão 10.1.1: esta biblioteca é utilizada para gerenciar todos os valores numéricos dentro do iVProg.

Além dos recursos citados acima, aqueles outros utilizados dentro do núcleo de interpretação de código do iVProg foram desenvolvidos pela própria equipe.

2.3. Integração com o Moodle

Para a integração com o gerenciador de cursos Moodle, o iVProg utiliza o módulo de atividade interativa chamado iTarefa, desenvolvido e mantido pelo mesmo grupo de pesquisa responsável pelo iVProg. O módulo encontra-se disponível no repositório online de plugins do próprio ambiente Moodle, sendo que sua principal funcionalidade é permitir a implementação de novos módulos de atividade interativa para o Moodle [Brandao et al. 2018], de forma simplificada, utilizando-se apenas a comunicação por meio de chamadas JavaScript, não exigindo do desenvolvedor que conheça toda a estrutura do Moodle, banco de dados e PHP.

Para garantir a integração com o iTarefa e consequentemente com o Moodle, algumas funções em JavaScript foram implementadas, para contemplar os aspectos de avaliação e de tarefa.

3. Apresentação do Software

Como mencionado anteriormente, o iVProg tem por objetivo oferecer aos estudantes um ambiente visual e interativo para a elaboração de algoritmos de forma simples, sem que seja necessária a codificação textual. Dessa forma, facilitando o aprendizado dos estudantes, estimulando-os a elaborar suas soluções computacionais, sem exigir a memorização de uma linguagem de programação específica e todas as suas particularidades léxicas e sintáticas.

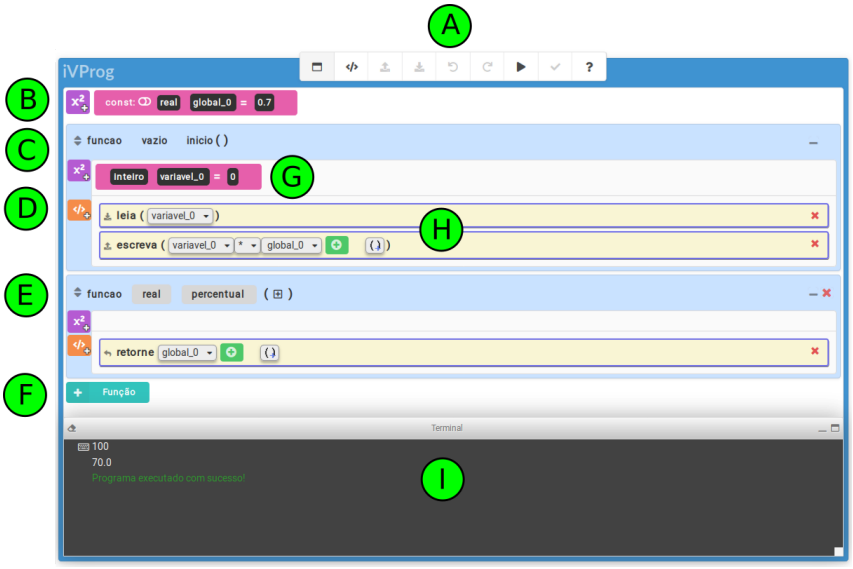

Figura 3. Elementos da área de trabalho do iVProg

A figura 3 apresenta a área de desenvolvimento de algoritmos do iVProg, onde o usuário pode elaborar sua solução de forma visual, intuitiva e interativa. As partes que compõem a área de trabalho são descritas a seguir:

A Menu principal
A partir das opções disponíveis neste menu, o usuário pode, por exemplo, alternar entre programação visual e textual. Sendo que a programação visual é apresentada na própria figura 3. Já a programação textual, o aprendiz poderá implementar seu algoritmo em uma linguagem inspirada no Portugol Studio, vide figura 4, onde é apresentado o mesmo algoritmo da figura 3. Além dessas funções, no menu, o usuário tem acesso ainda às opções de executar seu programa, avaliá-lo por meio do avaliador automático ou acessar o repositório online de ajuda, mantido pelo grupo responsável pelo iVProg.

B Gerenciador de variáveis globais

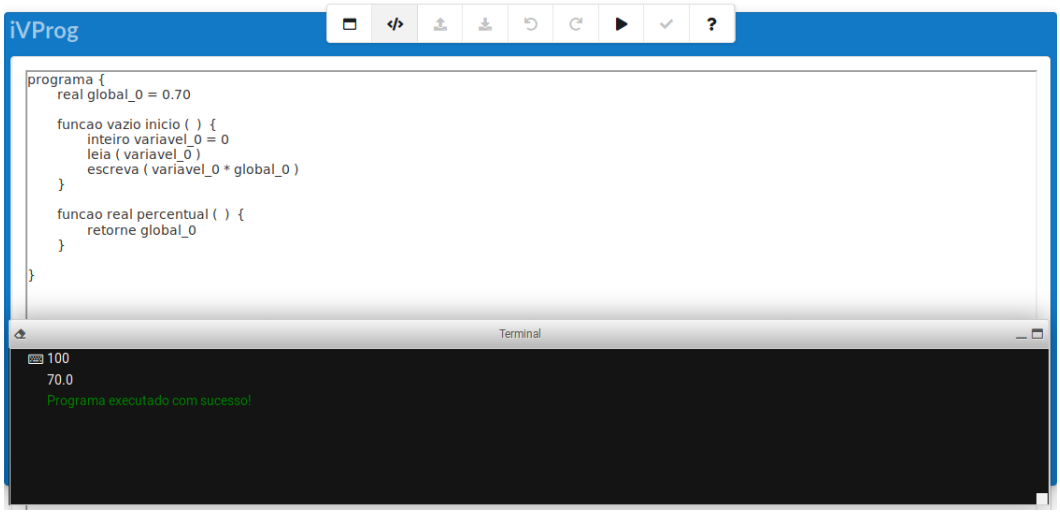

Figura 4. Codificação textual no iVProg

Nesta parte da interface visual, podem ser criadas novas variáveis globais, modificadas aquelas existentes ou mesmo excluídas do algoritmo. Estas variáveis ficarão acessíveis por todas as funções presentes no programa do usuário. O iVProg possibilita ainda que essas variáveis globais sejam definidas como constantes, ou seja, seu valor não poderá ser alterado em qualquer lugar do algoritmo.

C Função "inicio"

Esta é a função principal do algoritmo, como a "main" das linguagens Java e C. É o ponto de partida onde o iVProg iniciará a execução do programa do aluno. Toda vez que o estudante iniciar um novo algoritmo, esta função estará disponível, não sendo possível que esta seja excluída, nem mesmo tenha parâmetros ou retorno diferente de "vazio".

D Botão de comandos

A partir deste botão, o usuário tem à disposição todos os comandos pré-definidos no iVProg, que utilizados de forma sequencial e lógica comporão o algoritmo do estudante. A figura 5 apresenta o menu de comandos aberto após o acionamento do botão. Note que cada comando utilizado representará um bloco dentro do algoritmo.

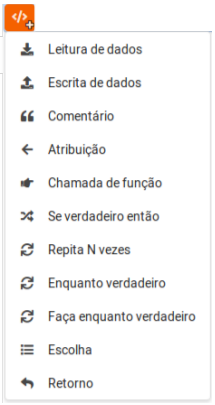

Figura 5. Menu de comandos do iVProg

No menu apresentado na figura 5, estão disponíveis os seguintes comandos:

- Leitura de dados: Cria uma instrução para a entrada de dados a partir do teclado dentro do terminal. É uma função genérica a todos os tipos de dados utilizados: inteiro, real, lógico e texto.
- Escrita de dados: Realiza a impressão de algum conteúdo, seja literal ou variável diretamente no terminal.

- Comentário: Gera um bloco para o usuário inserir um comentário em seu programa.
- Atribuição: Cria uma instrução de atribuição de valor a uma variável, podendo esta atribuição ser composta por expressões matemáticas, lógicas ou textuais.
- Chamada de função: Realiza a chamada de uma função, na qual o usuário não espera um retorno.
- Se verdadeiro então: Estrutura condicional do tipo “se/senão”.
- Repita N vezes: Laço de repetição controlada por contador, do tipo “for”.
- Enquanto verdadeiro: Laço de repetição do tipo “while”.
- Faça enquanto verdadeiro: Laço de repetição do tipo “do/while”.
- Escolha: Estrutura condicional de múltiplas escolhas do tipo “switch”.
- Retorno: Retorna o conteúdo para a chamada de função e a encerra.

E Função criada pelo usuário
Ao criar uma nova função, o usuário tem à disposição o gerenciador de variáveis locais e de instruções. Para as funções criadas pelo próprio usuário, os campos referentes ao tipo de retorno, identificador e também dos parâmetros ficam disponíveis para modificações.

F Botão para criar novas funções
A partir do acionamento deste botão, são acrescentadas novas funções ao algoritmo. Sendo que para removê-las, o usuário pode acessar o botão representado por um “X” vermelho dentro da mesma.

G Gerenciador de variáveis locais
Cada função presente no algoritmo possui uma área destinada às variáveis locais, onde o usuário pode adicionar, alterar os campos referentes ao tipo, ao identificador ou ao valor de inicialização.

H Área de trabalho para os comandos
Neste espaço, as instruções criadas a partir do menu de comandos são organizadas de forma sequencial pelo usuário. Esta sequência compõe o comportamento da função.

I Terminal
O terminal é a interface de interação entre o usuário e o programa em execução, sendo responsável por receber as entradas e imprimir as saídas produzidas pelo algoritmo.

Vale ressaltar que tanto em variáveis globais quanto aquelas locais, o iVProg trabalha com variáveis simples e também compostas (vetores e matrizes). Isso permite que o professor utilize o iVProg desde os conceitos iniciais até assuntos mais avançados em um curso introdutório de programação.

Do lado do professor, o software conta com uma área de autoria voltada especificamente à elaboração de atividades (figura 6). Para cada atividade, podem ser definidas as configurações dos recursos que estarão acessíveis ao estudante e também os casos de teste que serão utilizados pelo avaliador automático do iVProg.

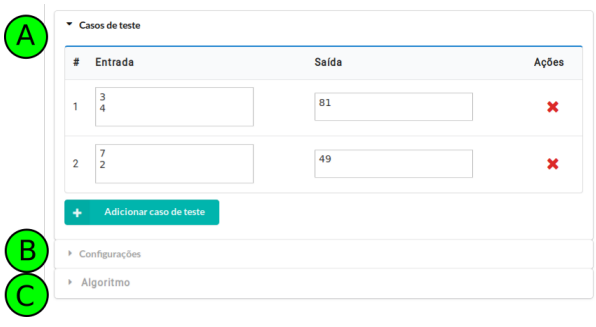

Figura 6. Área de autoria de atividades do iVProg

A área de autoria de atividades do professor é apresentado na figura 6, onde os principais componentes estão rotulados, descritos a seguir:

A Casos de teste
Nesta área, o iVProg disponibiliza ao professor uma tabela, onde na coluna nomeadamente "Entrada", o instrutor informa os valores que deverão ser recebidos pelo algoritmo do aluno. Enquanto na coluna "Saída", são definidas as saídas esperadas para o conjunto de entradas.
O avaliador automático é um componente importante do iVProg pois verifica se a solução elaborada pelo estudante antede às expectativas do docente: para cada conjunto de entrada e saída definidos, o programa do aluno é executado e os dados de entrada são inseridos. Enquanto que as saídas geradas pela execução são coletadas e estas devem ser equivalentes às saídas definidas pelo professor.

B Configurações
Nesta aba, o professor pode limitar os recursos que estarão disponíveis para o aluno solucionar o problema proposto. Por exemplo, poderá bloquear a criação de comandos de repetição, obrigando o aluno a criar um algoritmo com a utilização de recursividade.

C Algoritmo
Na aba de algoritmo, o professor tem à disposição a interface completa do iVProg (figura 3), podendo auxiliá-lo na elaboração dos casos de teste. Além disso, o professor pode partes de um algoritmo nesta área para que o aluno complete-o no exercício.

4. Considerações Finais

Este trabalho apresentou o iVProg, um software de programação visual, focado no ensino e aprendizagem dos conceitos de introdução a programação, bem como a prática do pensamento computacional. Seus principais recursos incluem uma área voltada à autoria de exercícios, avaliador automático das soluções e a integração com o ambiente gerenciador de cursos Moodle.

As possibilidades de utilização do iVProg são variadas, incluindo por exemplo, em cursos iniciais de programação, tanto no ensino presencial quanto naquele a distância. Podendo ser aplicado a diferentes níveis de ensino, desde o fundamental até o superior. Sua possibilidade de utilização passa ainda por disciplinas de matemática, para estimular o estudante a elaborar soluções algorítmicas para problemas como Fibonacci, fatorial, somatórios e muitos outros.

Referências

Arawjo, I., Wang, C.-Y., Myers, A. C., Andersen, E., e Guimbretière, F. (2017). Teaching programming with gamified semantics. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 4911–4923. ACM.

Bosse, Y. e Gerosa, M. A. (2015). Reprovações e Trancamentos nas Disciplinas de Introdução à Programação da Universidade de São Paulo: Um Estudo Preliminar. In *WEI-Workshop sobre Educação em Computação.(2015)*, pages 1–10.

Boulay, B. D. (1986). Some difficulties of learning to program. *Journal of Educational Computing Research*, 2(1):57–73.

Brandao, L. d. O., Felix, I. M., Pereira, P. A., e Brandão, A. A. F. (2018). Evolving technology to better support teaching introductory programming inside moodle. In *Conferência Latino-Americana de Informática - CLEI*. SBC.

Broll, B., Lédeczi, A., Volgyesi, P., Sallai, J., Maroti, M., Carrillo, A., Weeden-Wright, S. L., Vanags, C., Swartz, J. D., e Lu, M. (2017). A visual programming environment for learning distributed programming. In *ACM SIGCSE*, pages 81–86.

de-la Fuente-Valentín, L., Pardo, A., e Kloos, C. D. (2013). Addressing drop-out and sustained effort issues with large practical groups using an automated delivery and assessment system. *Computers & Education*, 61:33 – 42.

Ferreira, A. C., Melhor, A., Barreto, J., de Paiva, L. F., e Matos, E. (2015). Experiência prática interdisciplinar do raciocínio computacional em atividades de computação desplugada na educação básica. In *Anais do Workshop de Informática na Escola*.

Souza, D., Batista, M., e Barbosa, E. (2016). Problemas e dificuldades no ensino e na aprendizagem de programação: Um mapeamento sistemático. *Revista Brasileira de Informática na Educação*, 24(1):39.

Sykes, E. R. (2007). Determining the effectiveness of the 3d alice programming environment at the computer science i level. *Journal of Educational Computing Research*, 36(2):223–244.

Sáez-López, J.-M., Román-González, M., e Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "scratch" in five schools. *Computers & Education*, 97:129 – 141.