

Universidade de São Paulo  
Instituto de Matemática e Estatística  
Bachalerado em Ciência da Computação

Gabriel Eiji Nomura

**iVProg - Programação Visual e  
Interativa na Internet**

São Paulo  
Janeiro de 2017

# **iVProg - Programação Visual e Interativa na Internet**

Monografia final da disciplina  
MAC0499 – Trabalho de Formatura Supervisionado.

Supervisor: Prof. Dr. Leônidas Brandão de Oliveira

São Paulo  
Janeiro de 2017

# Resumo

Os recursos audiovisuais proporcionados por programas de computadores podem ajudar na motivação de estudantes para aprenderem. Em particular isso também é possível no aprendizado de algoritmos e de programação com o emprego do paradigma de programação visual (PV) [Kamiya, R. R., Brandão, 2009]. Mais ainda, dependendo da implementação, a PV pode ser utilizado até com estudantes do ensino fundamental, o que pode ampliar o interesse em carreiras ligadas à computação. Além disso o aprendizado de algoritmos pode auxiliar no aprendizado dos fundamentos da computação, pois demanda a compreensão básica de lógica booleana e o desenvolvimento da organização do raciocínio de modo sistemático. Neste contexto foi iniciado o desenvolvimento do projeto iVProg em 2008, utilizando a linguagem Java e em 2014 iniciou-se o desenvolvimento de uma versão do iVProg em HTML5 (iVProgH5). O modelo de PV implementado pelo iVProg foi pensado para estudantes a partir do último ano do ensino fundamental II. Ele apresenta os conceitos básicos de programação como variáveis, seleção e laço de repetição, mas por utilizar componentes gráficos é mais fácil para iniciantes que a programação tradicional, que demanda digitação de códigos na forma de texto. Este projeto estendeu o protótipo inicial do iVProgH5, que agora dispões de vetores e funções. Desse modo a usabilidade didática do iVProgH5 foi bastante incrementada, permitindo que todos os conceitos típicos da programação possam ser explorados. Neste texto está descrito o processo empregado para realizar a extensão do iVProgH5 e algumas lições aprendidas.

**Palavras-chave:** programação interativa, programação visual, iVProg.



# Abstract

The audiovisual resources provided by computer programs can help motivate students to learn. In particular, this is also possible in the learning of algorithms and programming using the visual programming paradigm (VP) [Kamiya, R. R., Brandão, 2009]. Moreover, depending on the implementation, VP can be used even with elementary students, which can broaden the interest in careers related to computing. In addition, learning algorithms can help in learning the basics of computing, because it requires the basic understanding of Boolean logic and the development of the organization of reasoning in a systematic way. In this context the development of the iVProg project was started in 2008, using the Java language and in 2014 the development of a version of iVProg in HTML5 (iVProgH5) was started. The VP model implemented by iVProg was designed for students from the last year of elementary school II. It presents the basics of programming such as variables, selection, and loop repetition, but by using graphics components it is easier for beginners than traditional programming, which requires typing codes in the form of text. This project extended the initial prototype of iVProgH5, which now has vectors and functions. In this way the didactic usability of iVProgH5 has been greatly increased, allowing all the typical programming concepts to be explored. This paper describes the process used to perform the extension of iVProgH5 and some lessons learned.

**Keywords:** Interactive programming, visual programming, iVProg.



# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Contexto do projeto</b>	<b>3</b>
<b>3</b>	<b>Sobre o iVProg</b>	<b>5</b>
3.1	Como programar utilizando os componentes visuais do iVProg . . . . .	5
3.2	Leitura de dados . . . . .	6
3.3	Escrita de dados . . . . .	6
3.4	Criação de fluxo condicional . . . . .	6
3.5	Criação de repetição . . . . .	6
3.6	Criação de repetição (enquanto) . . . . .	7
<b>4</b>	<b>Desenvolvimento do projeto</b>	<b>9</b>
<b>5</b>	<b>Tecnologias empregadas no desenvolvimento do iVProgH5</b>	<b>11</b>
5.1	JavaScript . . . . .	11
5.1.1	NodeJS . . . . .	11
5.1.2	NPM JS . . . . .	11
5.1.3	AngularJS . . . . .	12
5.1.4	Gulp JS . . . . .	12
5.2	CSS . . . . .	12
5.2.1	Bootstrap . . . . .	12
5.3	HTML5 . . . . .	12
<b>6</b>	<b>Implementações Realizadas</b>	<b>15</b>
<b>7</b>	<b>Dificuldades encontradas</b>	<b>17</b>
<b>8</b>	<b>Trabalhos futuros</b>	<b>19</b>
<b>9</b>	<b>Conclusões</b>	<b>21</b>





# Capítulo 1

## Introdução

Alguns dos benefícios da utilização de software no processo educacional são a melhora da experiência de aprendizagem do estudante, bem como facilitar o processo da didática do ponto de vista do professor. Esse é o foco central do Laboratório de Informática na Educação (LInE) que procura desenvolver e aplicar novos recursos computacionais na educação. Em particular o LInE tem desenvolvido programas na forma de Módulos de Aprendizagem Interativa (iMA) [Brandão, Rodrigues 2013]. Esses iMA podem ser integrados a Sistemas de Gerenciamento de Cursos de modo a propiciar um uso integrado do módulo com os ambientes Web para cursos. A criação de programas para apoiar a educação é uma atividade multidisciplinar, dependendo de conhecimento técnico de computação e de conceitos educacionais. Além disso, para o desenvolvimento deve-se trabalhar em conjunto com os professores que utilizarão tais ferramentas, pois quanto mais simples para eles e quanto mais adequado aos modelos didáticos que adotam, maior poderá ser a adesão ao uso das ferramentas. O foco deste projeto é o ensino e aprendizagem de algoritmos e dos fundamentos da programação, a partir de um particular módulo, o iVProg. Ele é iMA para programação interativa na Internet, desenvolvido dentro do LInE, cujo alvo é facilitar a aprendizagem de algoritmos e estrutura do código. Para isso adota um modelo de programação visual, com o qual o aprendiz tem facilitada a tarefa de memorizar a sintaxe de comando ou a declaração de variáveis. Este projeto iniciou-se em uma versão Java do iVProg codificado em Java em 2009 (projeto de mestrado do Reginaldo Rideaki Kamiya), depois foi refeito completamente a partir de 2014 em Java no projeto de mestrado de Romenig da Silva Ribeiro e em HTML5 por estagiários do LInE. A versão HTML5 foi iniciada em função dos responsáveis pelas máquinas virtuais Java apresentarem dificuldades para o usuário final rodar em seu navegador um `?applet?`. Este projeto é uma extensão do último citado. Aqui focamos estudos de interface, além ampliarmos as funcionalidades oferecidas, particularmente conseguimos implementar o uso de vetores e de funções. Devido à facilidade de uso do iVProg, esse pode ser utilizado como ferramenta para introdução à programação, antes de mudar para uma linguagem real de programação.



## Capítulo 2

### Contexto do projeto

A interatividade é uma importante característica que programas educacionais devem geralmente ter para aumentar a motivação e o interesse dos alunos. Outra característica interessante para esses programas é estarem integrados ao sistema gerenciador de cursos (SGC). Nesse sentido, o iVProgH5 apresenta essas características, a interatividade advém principalmente de seu paradigma de programação visual e a integração a um particular SGC é provida por outro pacote livre do LinE, o iTarefa [Moura e Brandão, 2005, Rodrigues, Brandão, Brandão, 2010].

De modo geral, os sistemas desenvolvidos no LinE buscam essas características de interatividade e integração com SGC, além de do reuso de código, pensando também do lado do docente procurando trazer mais conteúdo com mais facilidade de uso. Existem alguns problemas encontrados no desenvolvimento desse tipo de sistema como o elevado custo de desenvolvimento, a chance de encontrar um problema na aplicação que não seja de fácil correção. Caso o desenvolvimento seja feito por estudantes existe o problema da baixa maturidade da codificação, de um desenvolvimento ser interrompido pela conclusão do curso e consequente abandono do projeto, pela possível defasagem da tecnologia quando o projeto estiver num estado de pronto, entre outros problemas [Dalmon, Brandão, Brandão, 2012]. Este último item aplica-se ao presente projeto. Assim, foi necessário um período adicional de trabalho, de mais de um semestre (este projeto foi iniciado em 2015), para a familiarização com o projeto e com seu código legado.



# Capítulo 3

## Sobre o iVProg

O iVProg é um sistema para programação de forma visual e foi construído inicialmente utilizando a tecnologia Java, rodando como aplicativo e como applet (integrável à Web). Para utilizar o sistema era necessário ter instalado uma máquina virtual (JVM), que era frequentemente atualizado e em alguns casos causava incompatibilidade com a versão anterior. Assim, para executar o iVProg era necessário manter determinadas configurações no computador e caso o JVM fosse atualizado para uma versão mais recente poderia resultar no não funcionamento do sistema. Dependendo de como era feito o suporte das máquinas em que eram utilizadas o sistema iVProg, poderia ser muito trabalhoso ficar voltando para versões antigas da JVM somente para manter o programa executando ou tendo que atualizar a JVM em alguns casos. Na versão mais atual do iVProg não ocorre problema de dependência de instalação de um programa externo, pois para utilizá-lo é necessário apenas um navegador que suporte as tecnologias de HTML5 e javascript, que o restante dos componentes necessários serão baixados através do navegador. Desde a sua concepção o iVProg foi pensado de forma a permitir que o aluno possa criar um algoritmo sem se preocupar com as sintaxes de programação ou estruturas das linguagens. A programação é feita de forma visual, apenas indicando os valores quando necessário. Dessa maneira o sistema auxilia o aluno a montar a lógica, selecionando as variáveis e valores para comparação, criação de laços, e demais instruções, o que facilita o ensino dos algoritmos.

### 3.1 Como programar utilizando os componentes visuais do iVProg

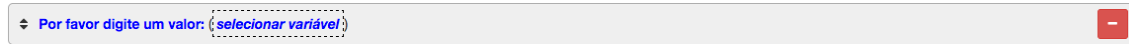
Ao clicar no botão “+ Variáveis” será criado uma nova variável na área destinada à criação de variáveis, que poderão ser utilizadas nas instruções criadas na área inferior a área da declaração de variáveis.



**Figura 3.1:** Botão para criar variável.

## 3.2 Leitura de dados

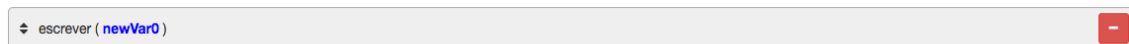
Ao inserir a instrução “Leitura de dados” será criada uma linha como na figura 3.2, permitindo que o aluno possa selecionar uma variável que receberá um conteúdo a ser digitado quando o programa estiver executando.



**Figura 3.2:** Instrução para leitura de valor e atribuir à variável.

## 3.3 Escrita de dados

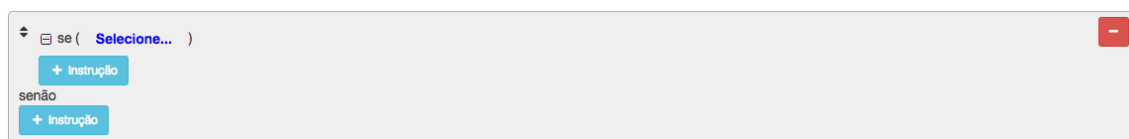
Ao inserir a instrução “Escrita de dados” será criada uma linha como na figura 3.3. Esta instrução vai fazer com que o valor da variável selecionada seja impressa na área de console.



**Figura 3.3:** Instrução para escrita do valor de uma variável no console.

## 3.4 Criação de fluxo condicional

Com a instrução “Se verdadeiro então” aparecerá uma região para entrada da validação de uma instrução lógica que será avaliada e executará uma instrução a ser definida caso a comparação seja verdadeira ou será executada outra instrução a ser definida na região abaixo do texto “senão” caso a comparação retorne falso.



**Figura 3.4:** Instrução condicional “Se”.

## 3.5 Criação de repetição

Quando utilizar a instrução “Repita N vezes” é possível determinar a quantidade de vezes que uma instrução deverá ser executada 3.5. Essa instrução também pode ser configurada para incrementar uma determinada variável enquanto realiza as repetições 3.6. Ou ainda é possível incrementar uma variável de um valor, que pode ser editado, e determinar o número de repetições com base no valor dessa variável 3.7.

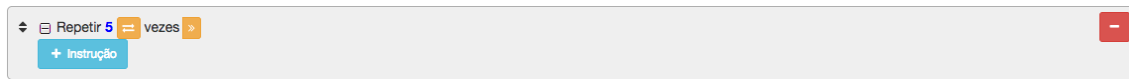


Figura 3.5: Instrução de repetição.



Figura 3.6: Instrução de repetição 2.

## 3.6 Criação de repetição (enquanto)

A instrução “Enquanto verdadeiro” utiliza uma função lógica na entrada para verificar se a instrução determinada abaixo continuará sendo executada ou não. Quando usar essa instrução o aluno deve verificar se há possibilidade de o programa sair do laço, pois a execução pode entrar em um modo de laço infinito e o programa não chegará num estado final.

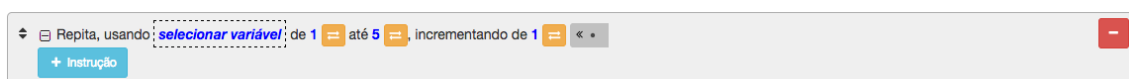
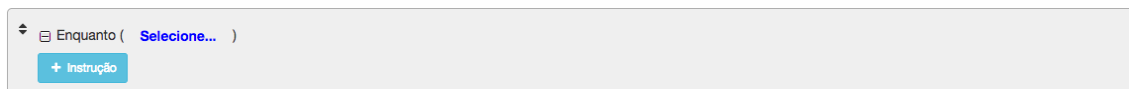


Figura 3.7: Instrução de repetição 3.



**Figura 3.8:** *Instrução de repetição “Enquanto”.*



## Capítulo 4

# Desenvolvimento do projeto

Como o sistema iVProg já tinha uma versão em um estado funcional, para dar continuidade ao desenvolvimento do sistema foi necessário conhecer as tecnologias utilizadas, entender a arquitetura que foi criada e conhecer as ferramentas que iriam auxiliar a dar prosseguimento da atividade. As tecnologias presentes no sistema iVProg são bastante utilizadas em sistemas *web* e por isso é fácil encontrar referências e dúvidas a respeito da utilização das funcionalidades. Nem sempre uma solução utilizada em um sistema irá resolver em outro, o que tomou um tempo significativo quando encontrado alguns entraves no desenvolvimento.

Apesar de ter sido despendido um tempo no início do projeto para conhecimento das tecnologias, foi necessário dedicar mais tempo ao estudo e pesquisa para encontrar soluções que resolvessem determinados problemas. No capítulo a seguir está uma descrição breve das tecnologias utilizadas no sistema.

Para testar o iVProg numa máquina local basta seguir o passo a passo descrito no arquivo *LeiaMe* que será possível executar e realizar testes. Para o desenvolvimento das novas partes do sistema foram feitos testes utilizando um navegador e acompanhamento com log e um recurso disponível no navegador onde é possível acompanhar passo a passo a montagem da tela, chamada de funções, passagem de parâmetros e retornos ou erros apresentados num console disponível.

Como o projeto já tinha uma parte pronta, era necessário seguir a estrutura utilizada e fazer as novas implementações mantendo coerência com o código existente, o que em alguns momentos resultou em algumas horas de estudo da arquitetura criada devido a falta de documentação do sistema.



# Capítulo 5

## Tecnologias empregadas no desenvolvimento do iVProgH5

O iVProg HTML5 está codificado utilizando o HTML versão 5 que dispõe de área para desenho, associado com folha de estilo (CSS) e com JavaScript para possibilitar controles e desenhos. Além disso, foram empregados alguns arcabouços ou bibliotecas de códigos JavaScript para agilizar o desenvolvimento da ferramenta, como o NodeJS, NPM JS, AngularJS, Gulp JS, Bootstrap, que são rapidamente explicadas nas subseções seguintes.

### 5.1 JavaScript

JavaScript é uma linguagem de programação interpretada orientada a objetos que foi originalmente criada para tratar do comportamento do lado do cliente de páginas *Web*. Com o passar dos anos o JavaScript acabou sendo utilizado em outras aplicações, como veremos nos tópicos a seguir. Com ela é possível criar variáveis, definir funções, atribuir a variáveis alguma função, que será tratada como objeto e pode ser utilizada como parâmetro para outra função, definir de forma dinâmica o tipo da variável, entre outras funcionalidades que podem facilitar a vida do programador caso tenha ciência da estrutura que esteja utilizando.

#### 5.1.1 NodeJS

NodeJS é uma plataforma para desenvolvimento de aplicações do lado do servidor. A linguagem utilizada é o JavaScript e ela é orientada a evento, como criação de conexão, entrada de dados, etc. O Node.js foi criado para suprir a necessidade de servidores que necessitam de muitas conexões, porém o conteúdo de cada uma dessas conexões não são muito altos. Para cada nova conexão o Node.js não fica criando threads ou alocando espaço para manter essa conexão, assim otimiza o espaço utilizado no servidor.

#### 5.1.2 NPM JS

O NPM (Node Package Manager) é um gerenciador de pacotes em JavaScript para ambientes de execução do NodeJS. Com ele é possível disponibilizar um pacote que resolva um determinado problema para que outras pessoas possam utilizar, além de ser possível fazer um controle de versão e de atualização desses módulos ou pacotes utilizados. Um desenvolvedor pode, por exemplo, definir em um arquivo as dependências e os pacotes que serão necessários instalar para que a sua aplicação tenha as configurações mínimas para poder ser executada e

referenciar esses pacotes no NPM, assim quando outro desenvolvedor quiser executar a aplicação, será necessário somente executar um comando que irá baixar os pacotes diretamente do repositório do NPM.

### 5.1.3 AngularJS

A tecnologia AngularJS é um arcabouço que foi criado pela empresa Google. Suas principais características são a reutilização de estruturas, criação através de MVC (Model - View - Controller) e traz para o lado do cliente (navegador) alguns trabalhos que antes eram feitos do lado do servidor, como os controladores para os componentes visuais. A versão do arcabouço utilizada foi a AngularJS v1.0.6, que atualmente está na versão v2.4.0. Devido as alterações realizadas da versão 1 para a versão 2, não existe compatibilidade para muitos elementos implementados, dificultando a atualização para a versão mais atual.

### 5.1.4 Gulp JS

Gulp JS é um conjunto de ferramentas úteis para gerar aplicações JavaScript como a redução de tamanho de arquivos JS. Como um pacote JS roda no navegador cliente, ao solicitar uma página todo seu código deve ser descarregado, assim, uma de suas ferramentas é o redutor de arquivos JS. Outra ferramenta útil é o recurso para recarregar um arquivo assim que ele for alterado. O Gulp JS também utiliza a linguagem JavaScript e roda em cima do Node.JS.

## 5.2 CSS

O nome CSS vem de *Cascading Style Sheets*, que poderíamos traduzir como Folha de Estilos em Cascata. É uma linguagem utilizada para criar formatações e criar padrões a serem utilizados em páginas *Web*. Enquanto o HTML fica responsável pela criação e estruturação do conteúdo, o CSS facilita a padronização dos estilos de cada objeto apresentado na tela. Com a utilização do CSS é possível determinar a apresentação de apontadores de forma diferenciada, posicionar elementos em posições fixas na tela ou de forma flutuante, padronizar bordas, alturas, larguras, margens, fontes, agrupar elementos, dentre outras opções.

### 5.2.1 Bootstrap

Bootstrap é um arcabouço que surgiu para facilitar o desenvolvimento de páginas *Web* auto-ajustáveis (ou que respondem, vindo do termo em Inglês *responsive*). Os sítios *Web* auto-ajustáveis são aqueles que adequam a disposição dos seus objetos conforme o tamanho da tela de quem esteja acessando a página. Isso poderia ser feito recebendo uma informação do tamanho da tela do cliente e lhe enviando um formato adequado, mas usualmente não existe a informação do tamanho da tela do cliente, assim o Bootstrap usa os fundamentos de CSS que hoje conseguem, no lado cliente, detectar o tamanho da tela e fazer uma seleção de qual configuração utilizar.

## 5.3 HTML5

O HTML é uma linguagem de marcação que surgiu com o início da *WorldWideWeb* em 1989 para estruturar o conteúdo a ser apresentado em páginas disponibilizadas na *web*. É

uma linguagem de A versão mais atual é o HTML 5 e tem como principais modificações a capacidade de utilizar marcações personalizadas, maior facilidade de identificação de vídeos, áudios e outras funcionalidades como padronizações das estruturas.



## Capítulo 6

# Implementações Realizadas

O projeto possui muitas funcionalidades que podem ser implementadas de modo que sua experiência de usabilidade torne cada vez mais atrativa para os estudantes. Foram feitas algumas tentativas de implementação de funcionalidades, mas por entender que o desenvolvimento dessas funcionalidades levariam mais tempo do que o esperado, optou-se por uma implementação de uma funcionalidade que levaria tempo necessário e suficiente para a construção e possível teste. O projeto já possuía algumas funcionalidades como criação de variáveis (inteiro, real, booleana e texto) e algumas funções aplicadas a essas variáveis como atribuição de valor à variável, leitura da entrada pelo usuário, associando esse valor a alguma variável, escrita do valor de uma variável na tela, implementação de laço de repetição e fluxo de condicional. Somente com essa estrutura implementada é possível criar muitos algoritmos utilizando esses tipos de dados e estruturas disponíveis. Foi realizado o desenvolvimento da estrutura de listas para cada tipo de variável e as regras de utilização nas instruções disponíveis. As validações para usar esses tipos de variáveis e a validação automática também foram funções que foi implementado.

Ao iniciar o projeto a versão corrente do iVProgH5 dispunha dos seguintes recursos:

- definição de variáveis simples (não vetores) dos tipos booleano, inteiro, real e cadeia de caracteres (string);
- comando de seleção (if-else);
- comandos de entrada e de saída;
- comandos de repetição com condição de entrada (while) e com passos fixos (for).

Deste modo, para a construção de algoritmos importantes em cursos introdutórios de programação, faltavam algumas funcionalidades, como funções e vetores. Como o tempo de aprendizado da tecnologia necessário, notadamente HTML5, JavaScript, AngularJS e NodeJS, além de precisar conhecer bem o código legado tomou muito tempo (cerca de 10 meses - concorrente com as várias disciplinas cursadas). Foi necessário optar por implementar apenas um das duas funcionalidades acima citadas. Após analisar alguns cursos de programação básica, decidiu-se implementar o conceito de vetores.





## Capítulo 7

### Dificuldades encontradas

A aplicação exige certo conhecimento prévio de algumas tecnologias para dar continuidade no desenvolvimento da aplicação. Mesmo aprendendo os conceitos de cada tecnologia, analisando as estruturas e sabendo do estado final esperado do projeto, algumas informações poderiam estar melhores documentadas, o que auxiliaria no desenvolvimento. Em alguns pontos o desenvolvimento pode ser impactado pelo não conhecimento das ferramentas disponibilizadas pelos programas utilizados, como a opção de ferramenta do desenvolvedor disponível no navegador Chrome, onde é possível acompanhar passo a passo a montagem de uma tela, execução de roteiros, acompanhar os parâmetros passado entre eles. Mas para conseguir tirar proveito desses recursos úteis no desenvolvimento é necessário conhecer os pontos certos em que a ferramenta permite que seja acompanhada a construção das telas.



# Capítulo 8

## Trabalhos futuros

O sistema iVProg preenche uma lacuna existente no início dos estudos da programação, pois seu foco está no desenvolvimento de algoritmos ou sequência de instruções de modo a solucionar determinado problema sem ter que se preocupar com sintaxes existente nas linguagens de programação.

Mesmo tendo sido implementadas funções básicas para solucionar problemas, o iVProg ainda possui funcionalidades que deverão ser implementadas num futuro próximo. Ainda que todas as funcionalidades programadas estejam implementadas, uma atualização das tecnologias será fundamental visto que essas tecnologias continuam em constante evolução.

Algumas novas implementações já estão em vista, como:

- Inclusão do bootstrap
- Inclusão dos botões “desfazer” e “refazer”
- Inclusão de um dicionário para facilitar a portabilidade para outros idiomas
- Alteração do item “Instrução” escolhida
- Melhoria na documentação
- Alteração visual, de modo a reduzir os tamanhos dos elementos apresentados, mas sem deixar de ser intuitivo
- Medir o impacto das novas implementações acompanhando a utilização por parte dos estudantes e validando as experiências de uso da ferramenta.



# Capítulo 9

## Conclusões

A utilização do sistema iVProg como ferramenta de auxílio para o ensino de programação básica pode ajudar na diminuição de evasão, bem como motivar os estudantes a se interessarem mais por cursos relacionados a essa área. O modo interativo criado por esse sistema atrai mais o estudante a seguir adiante nessas matérias e a buscar mais conhecimento na área de tecnologia e engenharia.

Pelo ponto de vista do professor, o sistema auxiliará no ganho de tempo na hora da correção das soluções apresentadas e também nas criações dos problemas, visto que será necessário dedicar pouco tempo em como utilizar o sistema devido a sua facilidade.

O conhecimento necessário para ajudar na manutenção do sistema, criação de novas soluções, corrigir erros e melhorar o desempenho do sistema iVProg podem se tornar cada vez mais complexos para alunos iniciantes na área de programação. Mas mantendo uma boa modularização e documentação das funcionalidades será possível reduzir o tempo necessário de aprendizado e gasto no desenvolvimento.

