**RELATÓRIO GERAL**

**DE PSI**

**(Estado da Arte)**

**Researched by:**

João Bandarra

Nº 17806

Índice

[Índice de imagens 3](#_Toc451854962)

[Flowgorithm 4](#_Toc451854963)

[O que é? 4](#_Toc451854964)

[Linguagens suportadas pelo conversor 7](#_Toc451854965)

[Características da linguagem 8](#_Toc451854966)

[Línguas de suporte 8](#_Toc451854967)

[Outras informações 9](#_Toc451854968)

[SICAS 10](#_Toc451854969)

[O que é? 10](#_Toc451854970)

[Características do programa 10](#_Toc451854971)

[Para estudantes: 10](#_Toc451854972)

[Para professores: 10](#_Toc451854973)

[Para ambos: 11](#_Toc451854974)

[Conclusão 15](#_Toc451854975)

[RAPTOR 16](#_Toc451854976)

[Introdução 16](#_Toc451854977)

[Características do programa 16](#_Toc451854978)

[Estrutura do programa 17](#_Toc451854979)

[Outras informações 21](#_Toc451854980)

[Portugol Studio 22](#_Toc451854981)

[Introdução 22](#_Toc451854982)

[Características do programa 22](#_Toc451854983)

[Estrutura do programa 23](#_Toc451854984)

[Outras informações 28](#_Toc451854985)

[Scratch 2 29](#_Toc451854986)

[Introdução 29](#_Toc451854987)

[Características do programa 30](#_Toc451854988)

[Outras informações 34](#_Toc451854989)

[Outras referências 35](#_Toc451854990)

[LARP 35](#_Toc451854991)

[Logic scheme compiler 39](#_Toc451854992)

[Alice 40](#_Toc451854993)

[Visual Logic 42](#_Toc451854994)

[BlueJ 43](#_Toc451854995)

[NetBeans BlueJ Edition 44](#_Toc451854996)

[Code2Flow 45](#_Toc451854997)

[Netgrafia 46](#_Toc451854998)

[Flowgorithm 46](#_Toc451854999)

[Sicas 46](#_Toc451855000)

[Raptor 46](#_Toc451855001)

[Portugol Studio 46](#_Toc451855002)

[Scratch 46](#_Toc451855003)

[LARP 46](#_Toc451855004)

[Logic Scheme Compiler 47](#_Toc451855005)

[Alice 47](#_Toc451855006)

[Visual Logic 47](#_Toc451855007)

[BlueJ 47](#_Toc451855008)

[NetBeans BlueJ Edition 47](#_Toc451855009)

[Code2Flow 47](#_Toc451855010)

# Índice de imagens

Figura 1 - Exemplo do funcionamento da consola de Output 5

Figura 2 - Janela de Variáveis 6

Figura 3- Exemplo do fluxograma 6

Figura 4 - Exemplo da conversão em código Java 7

Figura 5- Especificação de uma repetição 12

Figura 6- Exemplo do desenho do algoritmo 13

Figura 7- Simulação do algoritmo 14

Figura 8- Interface do RAPTOR 16

Figura 9- Exemplo de um fluxograma em RAPTOR 18

Figura 10- Página inicial do Portugol Studio 23

Figura 11- Estrutura do Separador Programar 24

Figura 12- Exemplo da estrutura do separador Programar com código 25

Figura 13- Exemplo da estrutura do separador Biblioteca 26

Figura 14- Código do jogo Moon Lander que se encontra nos exemplos 27

Figura 15- Aplicação Moon Lander 28

Figura 16 - Página inicial do Scratch2 Online 29

Figura 17- Logótipo do Scratch2 29

Figura 18 - Interface de programação do Scratch2 30

Figura 19- Exemplo de um cenário com um ator e as instruções indicadas ao ator 31

Figura 20- Exemplo do tutorial do lado direito " crie um jogo de pong " 31

Figura 21- Exemplo de realização de um tutorial 32

Figura 22- Cores para um ator 32

Figura 23- Lista de Cenários 33

Figura 24 - Lista de Atores 33

Figura 25- Exemplo de um projeto feito por um utilizador 34

Figura 26- Escolha do tipo de desenvolvimento do algoritmo 36

Figura 27- Aspeto do LARP no desenvolvimento de Fluxogramas 36

Figura 28- Consola do LARP 37

Figura 29- Menu help do LARP 37

Figura 30 - Aspeto do desenvolvimento de pseudocódigo no LARP 38

Figura 31 - Aspeto geral da aplicação Logic Scheme Compiler 39

Figura 32- Aspeto geral da aplicação Alice 40

Figura 33 - Criação de uma classe em Alice 41

Figura 34 - Aspeto geral da aplicação Visual Logic 42

Figura 35 - Aspeto geral do BlueJ 43

Figura 36 - Aspeto geral do Code2Flow 45

# Flowgorithm

## O que é?

O flowgorithm oferece-nos um **output** completamente diferente do normal, a forma de utilização da consola é disponibilizada por um output parecido com o Messenger, isto é, a utilização de bolhas de chat diferenciando a informação dada pelo computador e pelo utilizador por cores fazendo assim parecer que o utilizador está a falar com o computador.

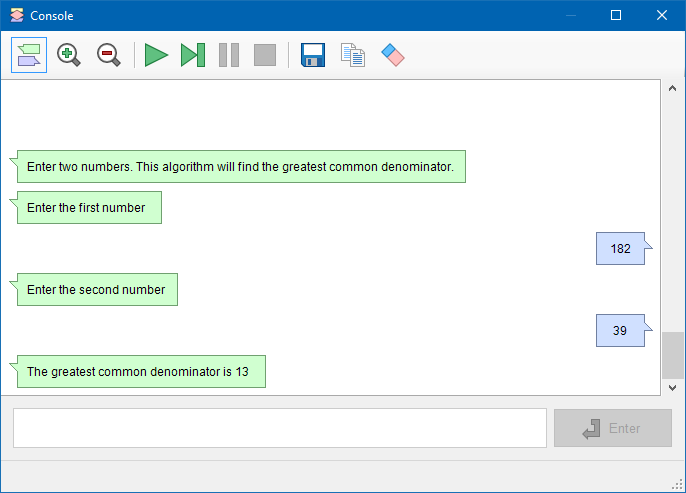


Figura 1 - Exemplo do funcionamento da consola de Output

Esta IDE oferece-nos também uma opção bastante eficiente de verificar o que está a acontecer com as nossas variáveis, esta opção é uma **janela de variáveis** onde podemos seguir os estados das variáveis enquanto o programa está em execução. Cada uma destas variáveis tem uma cor própria de modo a simplificar a identificação de cada variável.

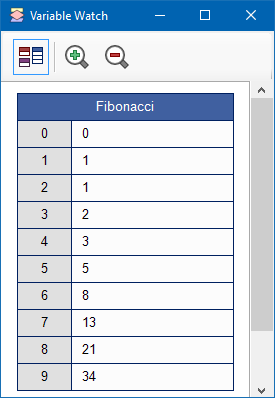
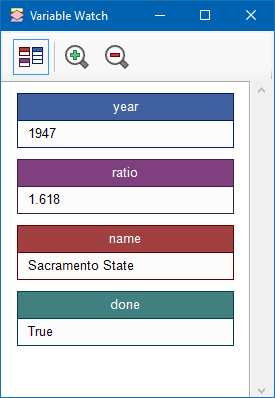


Figura 2 - Janela de Variáveis

O flowgorithm também nos oferece um conversor de fluxogramas para linguagem de programação. Ao inserir os fluxogramas temos a opção de poder converter em várias linguagens sem ter qualquer trabalho. Ao mesmo tempo o código terá a mesma cor que coincidir nos fluxogramas, isto é, cada forma do fluxograma terá uma cor diferente no código manterá essas cores para associar cada parte do código à forma do fluxograma.

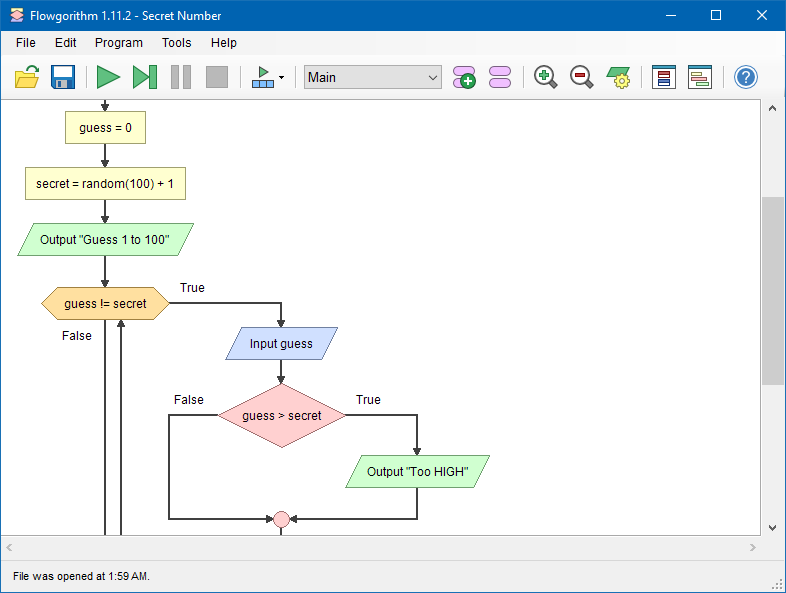


Figura 3- Exemplo do fluxograma

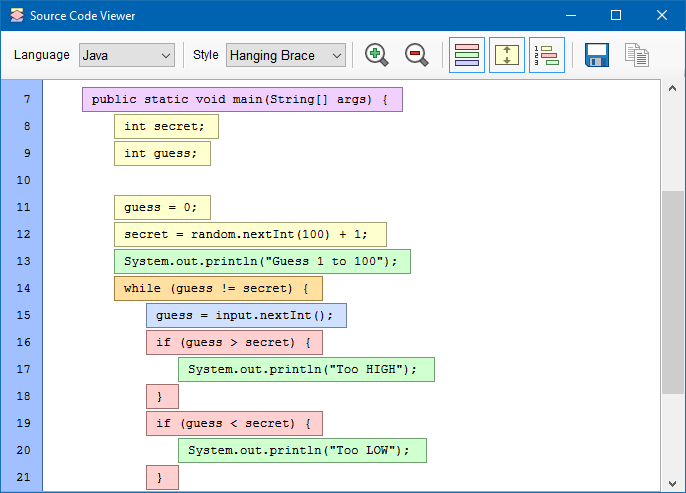


Figura 4 - Exemplo da conversão em código Java

## Linguagens suportadas pelo conversor

* C++
* C#
* Delphi / Pascal
* Java
* JavaScript
* Lua
* Perl
* Python
* QBasic
* Ruby
* Swift 2
* Visual Basic .NET
* VBA (Visual Basic for Applications)
* Gaddis Pseudocode - used in the book "Programming Logic and Design" by Tony Gaddis.

## Características da linguagem

* Arrays de dimensão única
* Testes pré e pós ciclo
* Para os ciclos

1. Declaração de variáveis explícitas. Os fluxogramas geralmente usam declaração implícita, porém praticamente todas a linguagens usam declarações explícitas.
2. Recursividade segura. Os fluxogramas contéem uma pilha interna, se um estudante acidentalmente cria uma recursividade (runaway), o programa não vai crachar, ao invés disso, vai é aparecer uma mensagem de erro.
3. Os operadores da linguagem C e BASIC são suportados simultaneamente.
4. Suporta funções com retorno e sem retorno.
5. Mais de 20 funções intrínsecas

## Línguas de suporte

* Chinês (simplificado)
* Checo
* Inglês Americano
* Inglês da Grã-Bretanha
* Francês
* Galego
* Alemão
* Húngaro
* Italiano
* Japonês
* Português (De Portugal e do Brasil)
* Espanhol
* Espanhol do México
* Castelhano

## Outras informações

* Vários estilos de visualização de fluxogramas suportados. Inclui estilo clássico, IBM, SDL e outros.
* Os fluxogramas podem ser impressos a cores ou a preto e branco
* Os fluxogramas podem ser gravados com a extensão PNG.
* O Editor tem uma característica chamada “layout windows”. Tem o objectivo de mostrar todos os conteúdos no ecrã sem ter de estar a redimensionar manualmente cada janela.
* Os programas são gravados com a extensão XML

O flowgorithm tem ainda um site oficial, onde é disponibilizado a aplicação de forma gratuita, contém tutoriais sobre a aplicação. Contém ainda uma janela onde diz quais foram as ultimas atualizações feitas e disponibilizadas.

# SICAS

## O que é?

Aplicação baseada em fluxogramas desenvolvido por professores da Universidade de Coimbra com o intuito de utilizar na própria universidade, de modo a simplificar a aprendizagem aos alunos.

## Características do programa

### Para estudantes:

* Inclui suporte de dois tipos de atividades:

1. Design/edição de soluções para os problemas propostos.
   * + O estudante constrói algoritmos usando representações visuais, os fluxogramas.
2. Execução/simulação dessas soluções.
   * + O estudante pode simular e animar a execução dos algoritmos que desenvolveu, analisando com detalhe e com o ritmo desejado dos vários passos e as entidades que participam.

Os estudantes podem selecionar um algoritmo feito por eles ou algoritmos do professor que tenham sido criados para esse problema.

### Para professores:

Para além dos dois tipos de atividades mencionados acima, o Sicas inclui ainda um modo professor que permite a criação e edição de uma biblioteca de problemas sugeridos. Conterá também uma solução definida pelo professor.

Pode abrir um algoritmo criado por um estudante.

Permite ao professor verificar se a solução do aluno apresentada resolve o problema.

A ideia da existência de vários algoritmos para o mesmo problema é permitir diferentes tipos de resolução e dar suporte a várias formas de compreensão, permitindo ao aluno comparar a sua resolução e estratégia com a solução mais adequada ao problema.

### Para ambos:

O utilizador pode consultar ou editar um dos seus algoritmos

O design do algoritmo é suportado por ícones onde o utilizador pode construir um fluxograma que representa as suas ideias ao invés de construir um algoritmo com pseudo-código, o que facilita a sua aprendizagem.

Pode-se tirar a conclusão de que:

* Demora menos tempo para compreender
* Produz-se menos erros por falta de compreensão
* Dá mais confiança aos alunos na compreensão do algoritmo
* Reduz o tempo de resposta relacionado com o algoritmo
* Reduz o número de vezes que os alunos precisam para analisar o algoritmo

As opções que os alunos podem usar quando criam fluxogramas no SICAS são:

* Atribuição – Define um valor de uma variável com o resultado de uma expressão
* Input/Output – Torna possível a leitura de valores do utilizador que escreve para a consola
* Repetição – Permite a repetição de execução de algumas ações. Temos como exemplo a figura 1 que mostra uma caixa de diálogo onde o estudante pode especificar as características da repetição.
* Seleção – Permite a escolha entre duas ações que poderão ser executadas

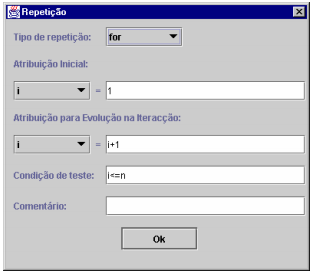


Figura 5- Especificação de uma repetição

As caixas de diálogos usadas foram projetadas para incluir o mínimo de informação necessária para executar a estrutura do programa correspondente. Isto evita erros comuns de sintaxe nos programadores mais inexperientes e faz com que eles pensem mais cuidadosamente sobre os diferentes componentes de cada estrutura.

Os alunos podem facilmente eliminar, modificar ou copiar qualquer componente em qualquer altura. Pode ser feito através de menus, ícones correspondentes na barra de ferramentas ou ativar/selecionar menus com o botão direito do rato.

A aplicação SICAS suporta apenas o uso de quatro tipos de dados: “números”, “strings”, “arrays de strings” e “arrays de números”. O objetivo principal do SICAS é dar suporte á aprendizagem de programação básica. Considera-se que a diversidade de tipos de dados, apesar de ser um aspeto importante nas linguagens de programação, constitui, várias vezes, uma fonte de distração de dificuldade acrescentada nos alunos para o design do algoritmo.

A construção de expressões no SICAS usa uma sintaxe similar á linguagem C e Java. Essa decisão foi tomada em consideração aos alunos, que mais tarde irão programar numa dessas linguagens. No entanto tentaram minimizar os detalhes sintáticos, para que o aluno possa concentrar completamente na tarefa principal (criação do algoritmo que resolve o problema).

Outra característica importante é o suporte a funções definidas por alunos. Essa característica tenta introduzir os alunos aos conceitos de modularização. A modularização tem várias vantagens:

* Torna o algoritmo mais legível e fácil de compreender
* Ajuda a gerir e resolver problemas mais complexos e grandes
* Permite a alteração do módulo ou função sem alterar o resto do algoritmo
* Evita a redundância, tornando possível a redução do número de componentes

A construção de funções é feita usando as estruturas mencionadas acima de acordo com as regras e opções previamente descritas para qualquer resolução. O SICAS também tem funções pré definidas que podem ser utilizadas para a construção de expressões sem necessitar da sua declaração.

Existem funções para a manipulação de números e strings.

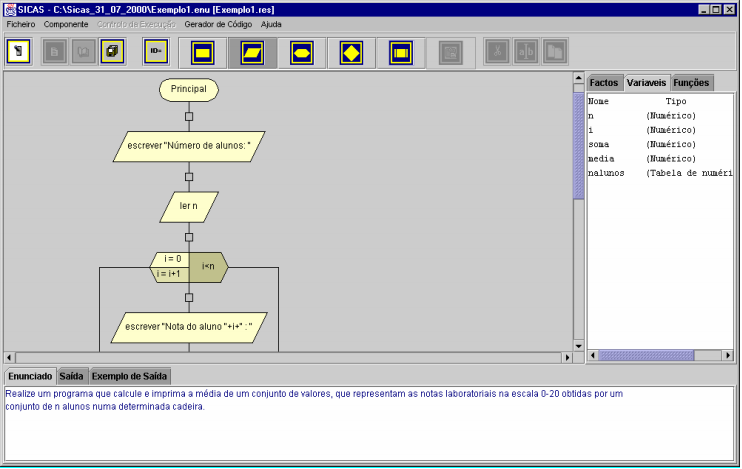


Figura 6- Exemplo do desenho do algoritmo

Qualquer algoritmo criado com SICAS pode ser automaticamente traduzido para pseudo-código, C ou JAVA. Estas alternativas não estão relacionadas com a sua importância, mas são a maneira de mostrar aos estudantes que um algoritmo bem desenhado pode ser facilmente traduzido em várias linguagens. Logo o algoritmo bem desenhado é o mais importante e não a linguagem que irá ser implementada para corresponder aos detalhes da sintaxe.

Depois de construído o algoritmo para solucionar um problema, o estudante/aluno pode ver uma simulação animada. O aluno pode ter algum controlo na simulação sendo:

* Controlo do ritmo e do progresso da simulação, pode avançar a instrução passo por passo, que faz com que a função continue mas de uma forma mais lenta ou então pode correr a função sem interrupções.
* Pode pausar a simulação, permitindo uma análise mais profunda dos dados (valor de uma variável) e/ou um debate com o professor ou com outros alunos.
* Permite retroceder passos da execução, para que o aluno possa repetir essas instruções para ajudar nas partes menos compreendidas pelos alunos nas execuções.

Durante a simulação, existe um componente que vai mudar de cor, para facilitar ao aluno o que se está a passar no programa. A Figura 3 mostra os aspetos do SICAS durante a simulação do algoritmo.

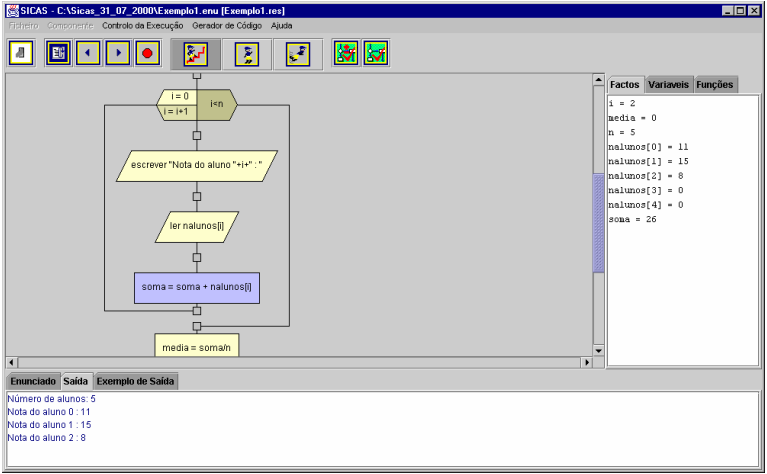


Figura 7- Simulação do algoritmo

Se o professor tem dado um conjunto de dados de teste (entradas e correspondente resultado esperado do algoritmo) para o problema ser resolvido, o aluno também pode testar o seu próprio algoritmo com os mesmos dados. É útil se o professor especificar casos especiais dos dados de entrada, uma vez que muitos alunos tendem a ficar satisfeitos se as suas soluções funcionarem para o que é pedido sem se importar com outros casos.

## Conclusão

O SICAS é uma aplicação educacional que tem intenção para ajudar o aluno a compreender e a tornar-se “profissional” na utilização de estruturas de programação básicas. O aluno pode desenhar/criar algoritmos que solucionam os problemas colocados pelo professor. O sistema usa uma interface gráfica interativa, onde as estruturas básicas de programação são representadas através de símbolos pré definidos.

Este ambiente pode ser usado pelos alunos em grupo ou em singular, nas aulas ou em casa. Permite aos estudantes com mais dificuldades (em acompanhar o ritmo nas aulas) de poder acompanhar as aulas praticando através da aplicação, permitindo assim uma melhoria da sua lógica e experiência.

# RAPTOR

## Introdução

O RAPTOR é uma aplicação de programação baseado num ambiente de fluxogramas, desenhado especificamente para ajudar os alunos a visualizar os seus algoritmos. Com o RAPTOR, a programação é criada de forma visual e executada visualmente traçando a execução por fluxogramas. Requer o mínimo de sintaxe. Os alunos preferem usar fluxogramas para expressar os seus algoritmos que, de certa forma, traz mais sucesso para os alunos do que as tradicionais linguagens de programação ou desenhar fluxogramas sem o RAPTOR.

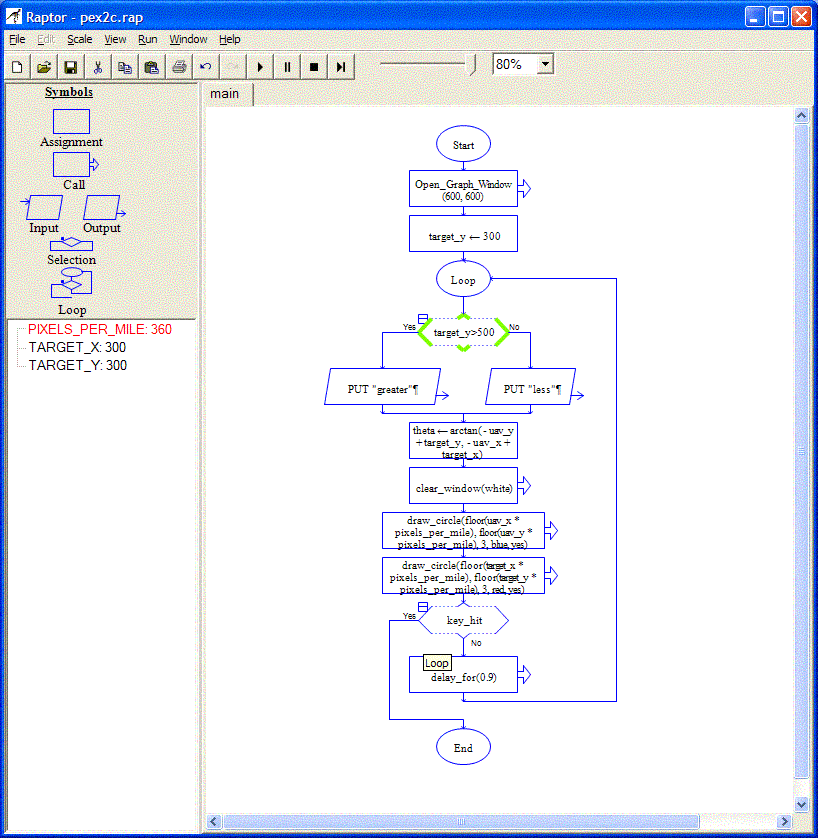


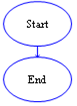
Figura 8- Interface do RAPTOR

## Características do programa

A linguagem RAPTOR tem o mínimo de sintaxe quando comparado com outras linguagens de programação. Isto faz com que seja mais fácil escrever algoritmos com RAPTOR, tornando mais eficiente e reduzindo o numero de linguagens que é necessário para aprender para se tornar produtivo.

Em segundo lugar, a linguagem RAPTOR é visual, ou seja, a programação é feita por diagramas (fluxogramas), como resultado, facilita a compreensão da programação.

## Estrutura do programa



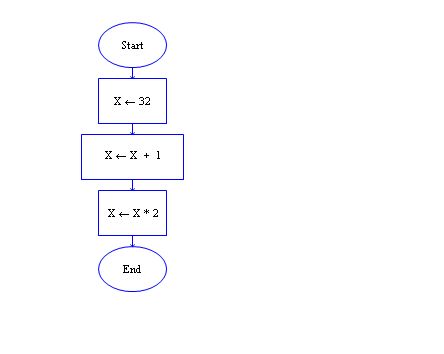
A aplicação RAPTOR é executada diretamente num grafo com início = start ----🡪 fim = end. Quando se quer executar um programa, começa por carregar no botão START (botão com símbolo parecido ao ”play”) que está no topo, na barra de ferramentas.

Certamente que se executar o programa só com as instruções inicio/fim, não fará nada.

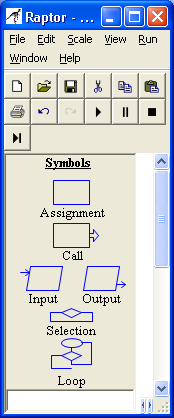
Arrastando instruções entre o início e o fim do grafo, irá criar fluxogramas mais complexos e mais significativos.

As variáveis são memórias locais que armazenam um valor. Em qualquer altura uma variável pode armazenar um valor específico de um tipo de dado, porém durante a execução do algoritmo, essa variável poderá sofrer alterações. Todas as variáveis terão de conter um nome distinto dado pelo programador, para facilitar a sua compreensão.

Os tipos de dados do RAPTOR para armazenar variáveis são:

* Inteiros
* String
* Boolean
* Float
* Arrays

Ao contrário da maioria das outras aplicações, as variáveis do RAPTOR são declaradas logo com um valor.



O RAPTOR tem 6 instruções básicas, Entrada (INPUT), Saída (OUTPUT), Declaração (ASSIGNMENT), Chamar função (CALL), Seleção (SELECTION) e Ciclo (LOOP). Qualquer um destas instruções é designado por um símbolo diferente.

As instruções de Seleção de Ciclo são também conhecidas como instruções de estrutura de controlo. São essas instruções que controlam como o programa executa.

Para todas as instruções, dentro da janela de cada instrução, contém uma pequena informação a explicar o que se pode fazer nessa instrução, normalmente localizada acima das áreas de texto.

****

Figura 9- Exemplo de um fluxograma em RAPTOR

Quando executados os operadores ou funções, esta irá devolver um resultado. O RAPTOR utiliza a seguinte construção de operadores e funções:

**Matemática básica:** +, -, \*, /, ^, \*\*, rem, mod, sqrt, log, abs, ceiling, floor

**Trigonometria:** sin, cos, tan, cot, arcsin, arcos, arctan, arccot

**Relacional:** =, !=, /=, <, >, >=, >=

**Logica:** and, or, not

**Outros:** random, Length\_of

Os operadores básicos de matemática incluem os operadores familiares (+, -, \*, /) assim como uns menos familiares.

\*\* e ^são exponenciais, exemplo 2\*\*4=16, 3^2=9.

rem (remainder = restante) e mod (module = módulo) retornam o resto (o que restou do resultado) quando o operando direito divide o operando esquerdo, exemplo: 10 rem 3 = 1, 10 mod 3 é também 1.

Sqrt retorna uma raíz quadrada, ex: sqrt(4) é 2

Log retorna um logaritmo natural, ex: log(e) é 1

Abs retorna um valor absoluto, ex: abs(-9) é 9

Ceiling arredonda para cima para um valor inteiro, ex: ceiling(3.14159) é 4

Floor arredonda para baixo para um valor inteiro, ex: floor(10/3) é 3

O + também trabalha como operação de concatenação para juntar duas strings ou um string e um número.

Length\_of retorna o numero de caracteres numa variável de strings, ex: teste é 5.

Os operadores de lógica e de relação só podem ser usados em instruções de seleção ou de ciclo.

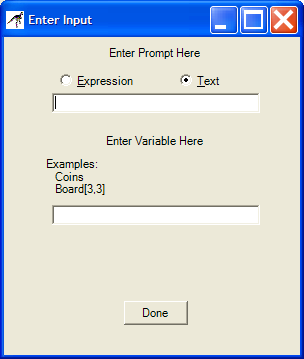
As constantes são variáveis pré-definidas cujos valores não podem ser alterados. O RAPTOR contem as seguintes variáveis:  
 **Constantes:** pi, e , true, false, yes, no

O PI tem o valor de 3.14159274101257.

O numero de nepper (e) tem o valor de 2.71828174591064

True e yes são definidos a 1

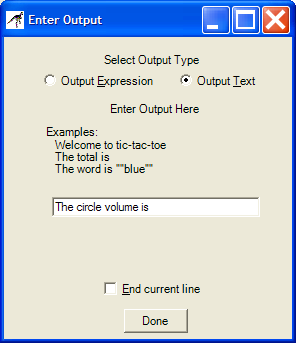
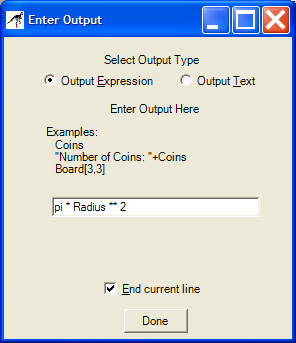
False e no são definidos a 0



Toda a linguagem de programação tem instruções que permite ao programa obter informação do utilizador do programa através do teclado, enviando essa informação para a consola.

Em RAPTOR, uma instrução de input mostra uma consola para o utilizador (para inserir um valor), e armazenar esse valor numa variável.

No RAPTOR, uma instrução de output é um tipo especial de procedimento que imprime o valor na consola. Uma caixa de diálogo pergunta para identificar se o utilizador está a escrever uma expressão ou texto.



Quando se imprime dados na consola, pode-se meter espaços depois do texto como o texto que está na figura acima do lado direito.

Como qualquer outra linguagem de programação, o RAPTOR permite adicionar comentários para o nosso algoritmo. O propósito de deixar um comentário no nosso algoritmo é explicar o algoritmo em si tornando-se mais fácil a sua compreensão.

## Outras informações

Esta aplicação foi desenvolvida de duas formas, isto é, o desenvolvedor fornece a execução de instalador e uma versão portátil que pode executar a partir de uma pen, não necessitando de fazer a instalação.

Para mais informações, pode consultar a página <http://raptor.martincarlisle.com/>

Tem toda a documentação disponível (tutorial), links youtube e ainda várias versões da aplicação juntamente com as suas melhorias.

# Portugol Studio

## Introdução

O Portugol Studio é uma aplicação IDE para aprender a programar, tem como publico alvo pessoas que sejam iniciantes em programação e que falem português.

Possui uma sintaxe fácil, vários exercícios resolvidos e materiais de apoio para uma melhor aprendizagem. Possibilidade de criar jogos e aplicações.

Foi criado com o intuito de os alunos (no Brasil) poderem desenvolver a sua lógica sem ter que perceber bem o Inglês.

## Características do programa

Ao correr a aplicação aparece um menu com várias opções e abre uma janela com um pequeno tutorial sobre o primeiro funcionamento do menu.

A aplicação necessita sempre de conexão à internet, pois, está dependente pelas videoaulas, a biblioteca e lista.

As opções disponibilizadas no menu são: programar, ajuda, videoaulas, biblioteca e ainda uma lista de exemplos simples.

Ainda no menu, ao fundo da página contém ainda:

* + - Dicas de Interface
    - Atalhos do teclado
    - Relatar um Bug
    - Ajudar no desenvolvimento
    - Sobre….
    - Atualizações



Figura 10- Página inicial do Portugol Studio

## Estrutura do programa

Depois de acedido ao modo de programação, a interface parece-se um pouco como um browser, onde é possível clicar no botão Portugol Studio para ir para o menu, continuando a janela programação ativa, num separador.

Dito isto, é possível ter vários projetos abertos ou até mesmo outras páginas abertas de exemplos, bibliotecas, etc;

É no separador programar que os utilizadores vão desenvolver o seu raciocínio e desenvolver para código.

A aplicação como características no separador programação tem:

* Correr o código instrução a instrução ou pode correr automaticamente
* Pode interromper a execução da aplicação, neste caso, instrução a instrução
* Permite guardar o projeto podendo escolher a diretoria
* Permite personalização da fonte (tamanho, tipo de letra)
* Numa aba lateral direita, contem ainda, em estrutura de árvore e de uma forma ordenada, as funções

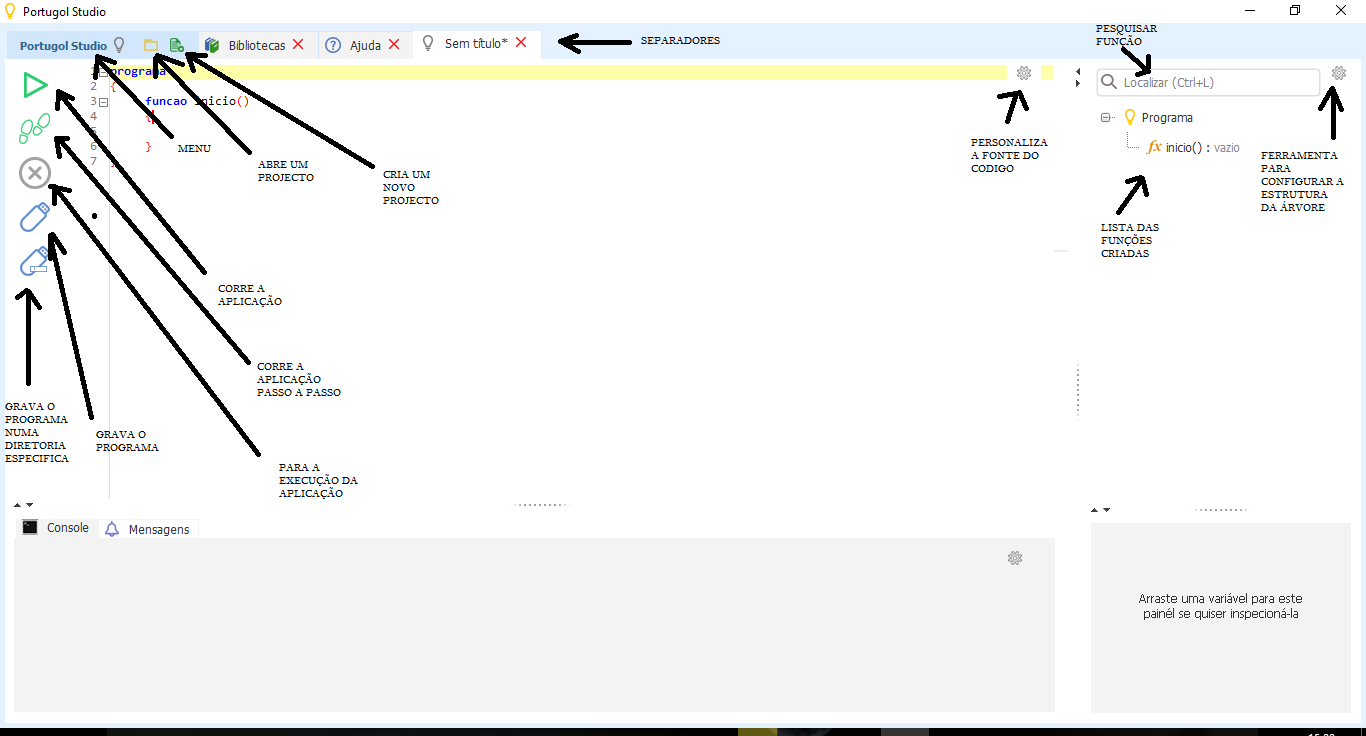


Figura 11- Estrutura do Separador Programar

As instruções utilizadas encontram-se em Português, logo é sempre aconselhável dar uma vista de olhos nos exemplos já resolvidos ou ir ao separador biblioteca para saber os nomes das instruções.

Temos como exemplo certas instruções típicas de programação (linguagem C neste caso) e a instrução respetiva em Português:

* var 🡪 inteiro(int), real(float), cadeia(array), logico(boolean)
* PrintF 🡪 escreva
* ScanF 🡪 leia
* If…else 🡪 se…senão
* Do…while 🡪 faça…enquanto
* While 🡪 enquanto
* For 🡪 para
* Function 🡪 função

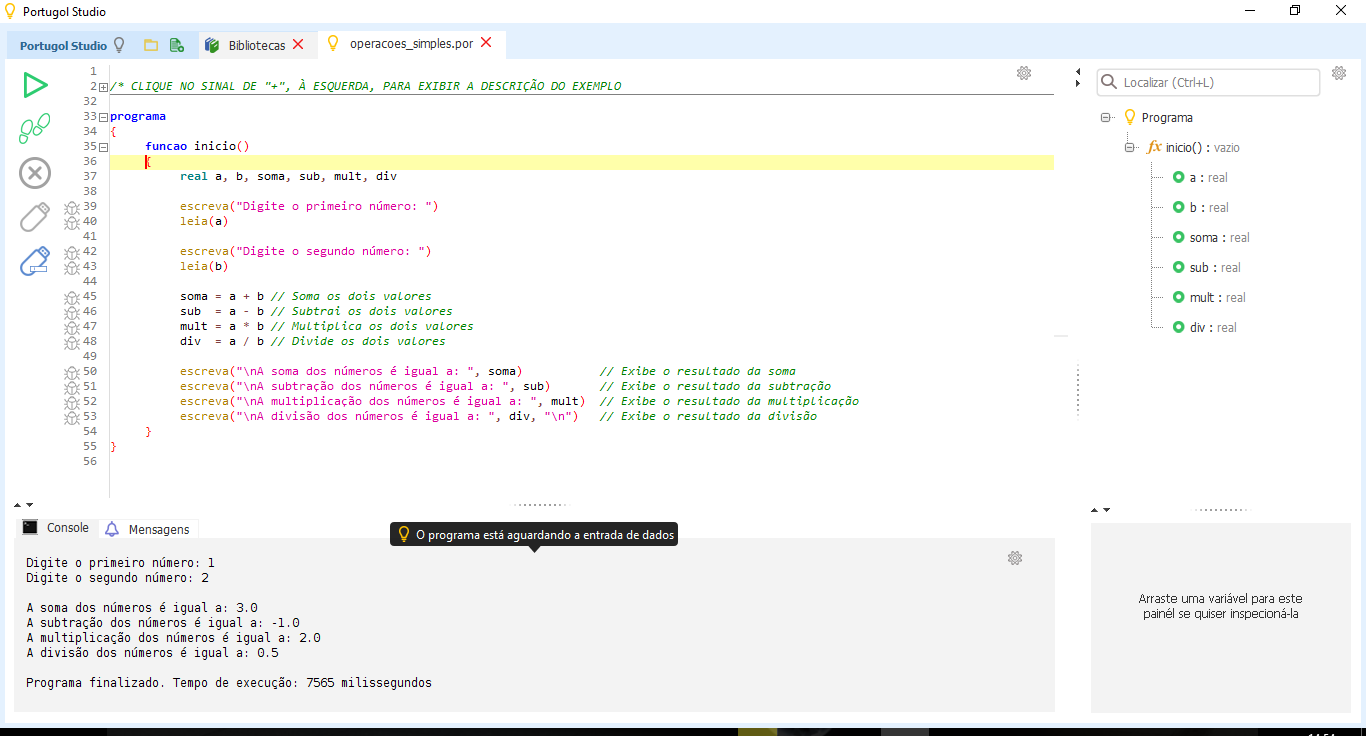


Figura 12- Exemplo da estrutura do separador Programar com código

O Portugol Studio contém bibliotecas, contém também uma biblioteca de jogos.

Essas Bibliotecas estão divididas em secções:

* + - Arquivos – Permite ler e escrever arquivos
    - Gráficos – Permite iniciar e utilizar um ambiente gráfico com suporte ao desenho de primitivas gráficas e de imagens carregadas do sistemas de arquivos.
    - Matemática – Contém uma série de funções matemáticas mais comuns.
    - Mouse (“rato”) – Contém um conjunto de funções para manipular a entrada de dados através do rato do computador. De mencionar que a biblioteca não funciona na consola de entrada e saída de dados, ela só funciona com a biblioteca gráficos, se o modo gráfico estiver ativado.
    - Sons – Contém funções que permitem reproduzir sons dentro dum programa. Suporta apenas a extensão MP3.
    - Teclado – Contém um conjunto de funções para manipular a entrada de dados através do teclado do computador. No entanto, esta biblioteca não funciona na consola de entrada e saída de dados. Só funciona com a biblioteca gráficos, se o modo gráfico estiver ativado.
    - Texto – Esta biblioteca contém funções para manipulação de texto.
    - Tipos – Contém funções que permitem converter os tipos de dados.
    - Util – Contém diversas funções utilitárias

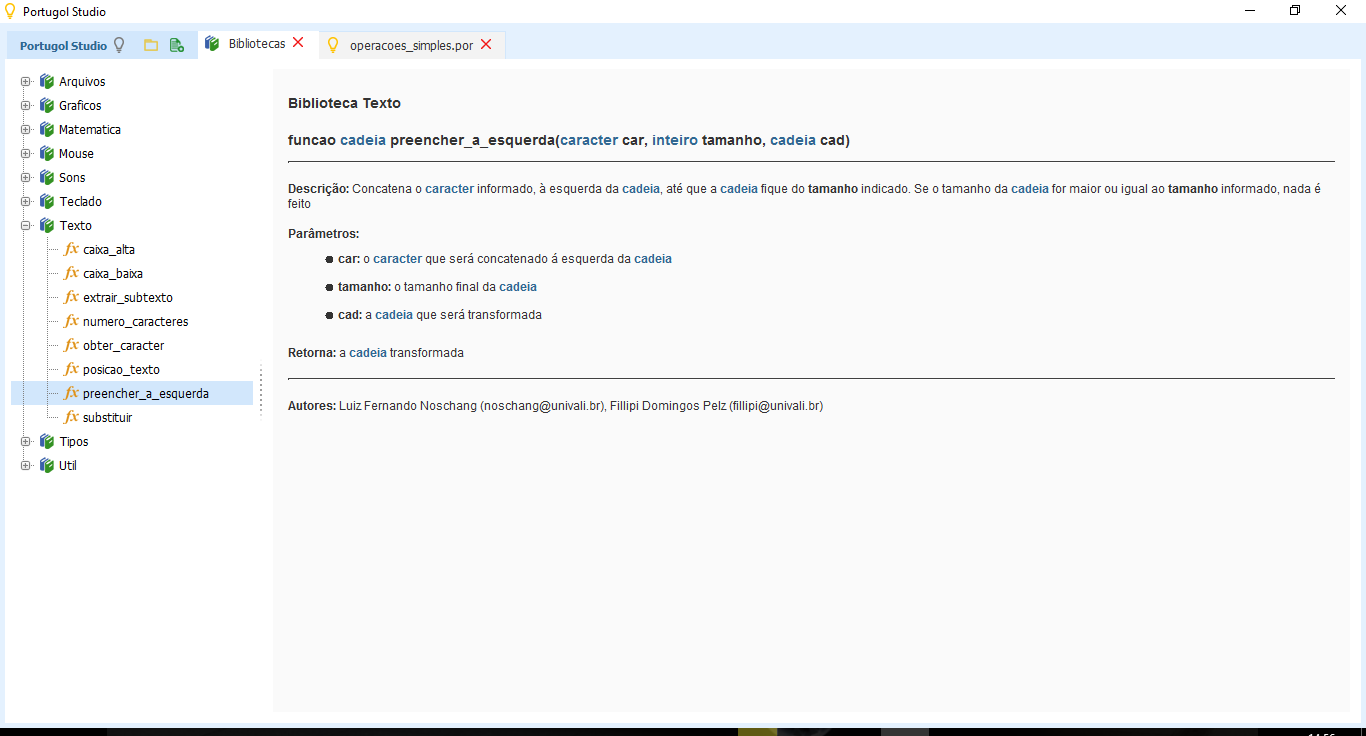


Figura 13- Exemplo da estrutura do separador Biblioteca

No separador Ajuda, encontra-se a informação que desejamos, em alguns dos casos, a sua resolução em fluxograma e ainda com um exercício resolvido como teste para que o utilizador possa estudar de forma a compreender a resposta á duvida que tinha inicialmente.

Pode ser dividido em secções, sendo elas:

* Bibliotecas
* Declarações
* Entrada e saída
* Estruturas de controlo
* Expressões
* Tipos

No menu principal do Portugol Studio tem vários exemplos de resolução que nos permite também perceber melhor a estrutura do código. Os tipos de exemplos disponibilizados são:

* Entrada e saída
* Operações Aritméticas
* Algoritmos sequenciais
* Desvios condicionais
* Laços de repetição
* Vetores e Matrizes
* Subrotinas
* Bibliotecas
* Música
* Jogos

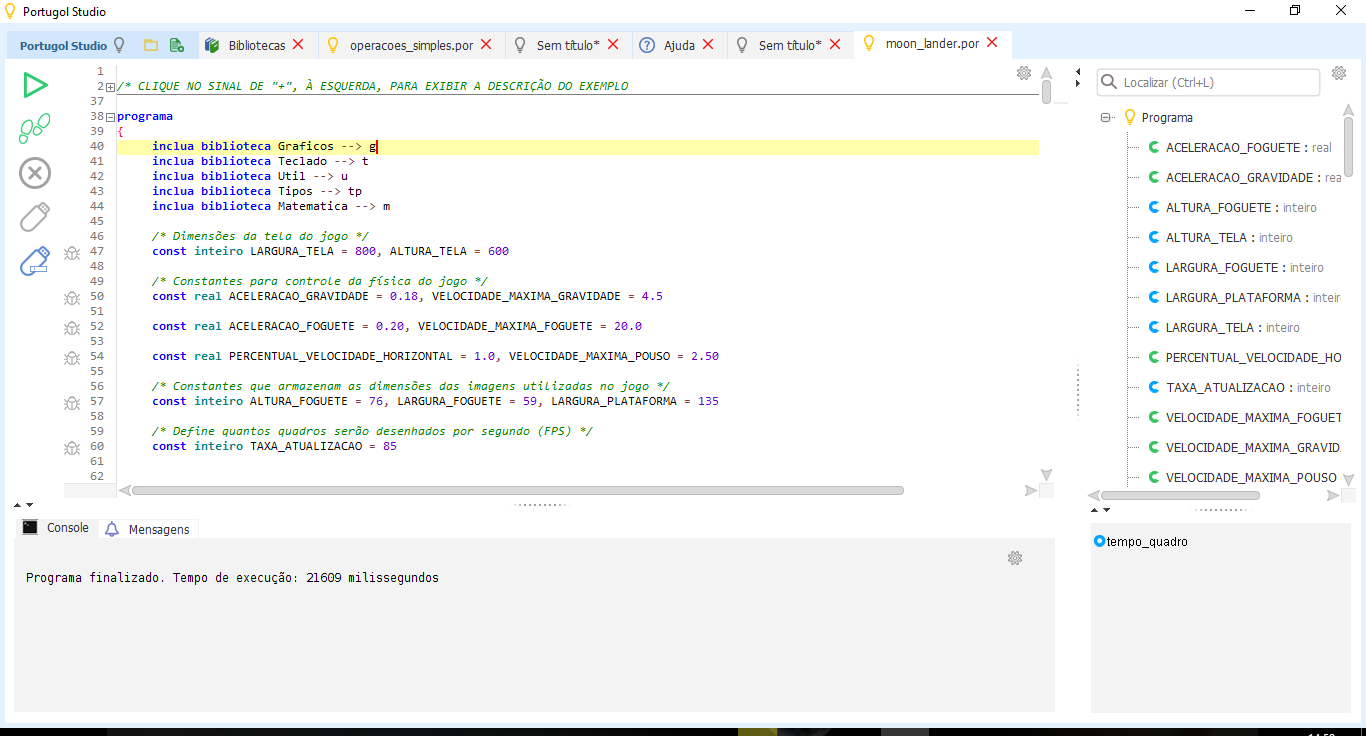


Figura 14- Código do jogo Moon Lander que se encontra nos exemplos



Figura 15- Aplicação Moon Lander

## Outras informações

A aplicação contém ainda Dicas de interface que funciona como um pequeno tutorial para navegar pela aplicação.

Contém outro botão “Atalhos do teclado” que abre outra janela com algumas funcionalidades com o teclado.

Pode sempre relatar problemas aos criadores de modo a resolverem esses problemas nas próximas atualizações, ou até mesmo ajudar no desenvolvimento (melhoria) da aplicação.

# Scratch 2

## Introdução

Desenvolvido pela Media Lab do MIT em 2003 e mais tarde, em 2013, está disponível online ou através de uma aplicação offline.

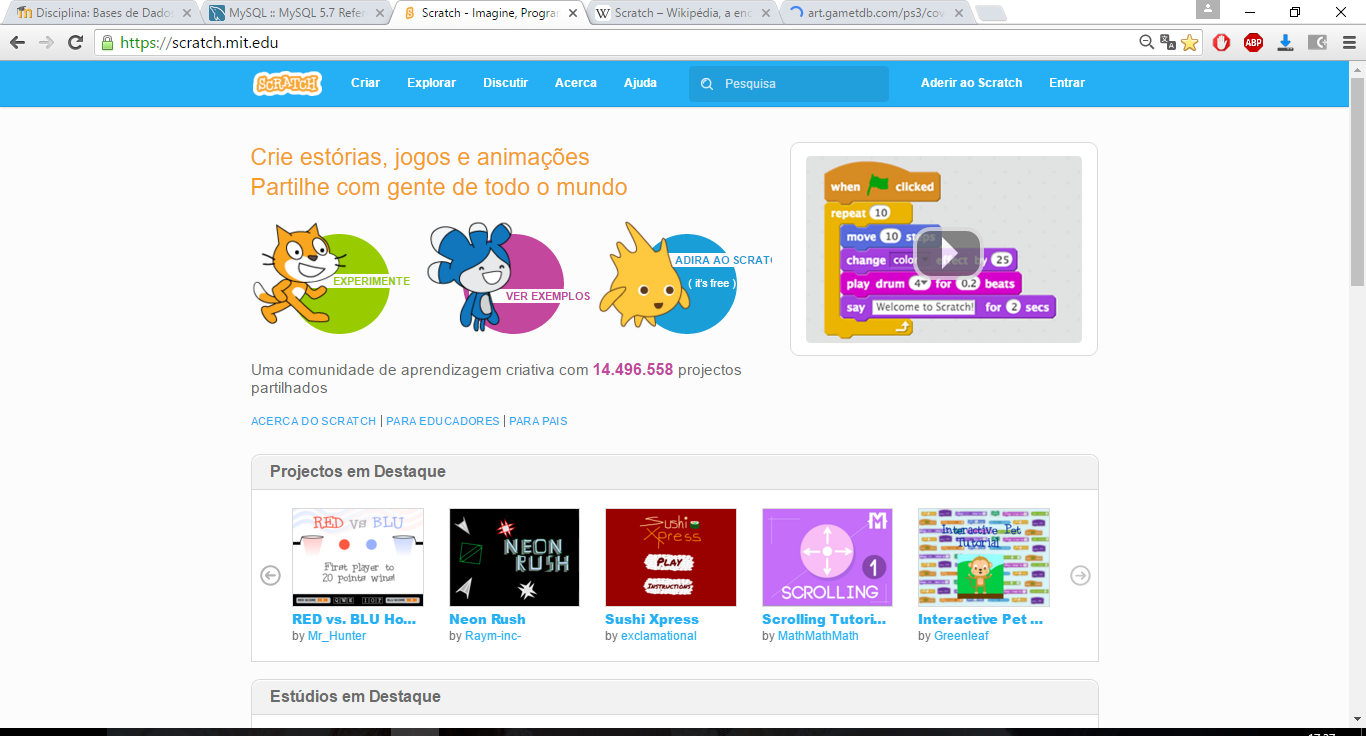


Figura 16 - Página inicial do Scratch2 Online

Foi pensado para as crianças poderem utilizar, desenvolvendo a lógica brincando como se tratasse de lego, pois o processo de programação é através de blocos.



Figura 17- Logótipo do Scratch2

## Características do programa

O programa permite ás crianças aprender conceitos de matemática computacional. É possível criar Histórias animadas, Jogos e outras aplicações.

É mais acessível que as outras linguagens de programação por conter uma interface gráfica.

As instruções estão divididas em várias categorias sendo elas:

* Movimento
* Aparência
* Som
* Caneta
* Dados
* Eventos
* Controlo
* Sensores
* Operadores
* Mais Blocos

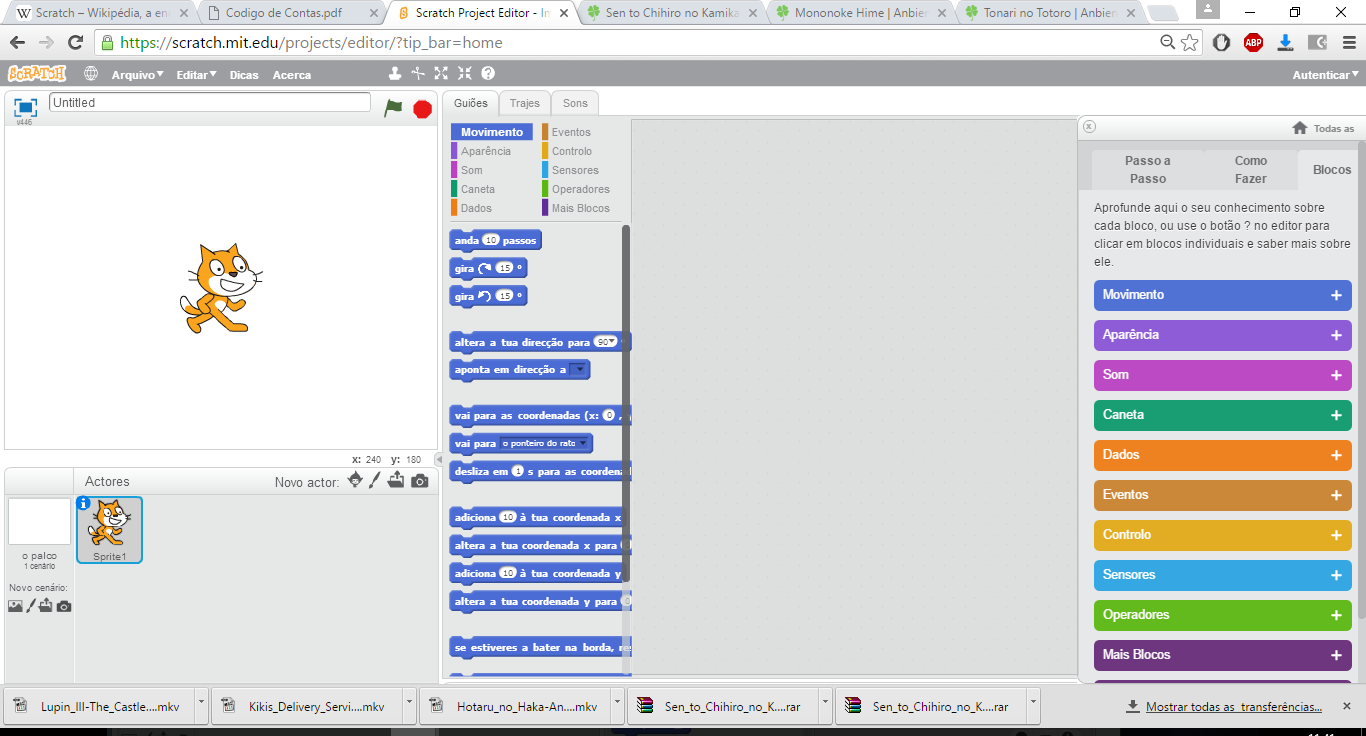


Figura 18 - Interface de programação do Scratch2

Pode criar atores. Esses atores, conforme o código que vai sendo criado para eles, vão “interagir” com o código. Por exemplo podemos inserir um excerto de código para fazer com que o ator diga alguma coisa, ou podemos clicar nele, se for essa a ação desejada, e o ator dança ao som de uma música.

Junto com o ator, pode adicionar um cenário para parecer mais animado. Neste caso optou-se por pôr um palco.



Figura 19- Exemplo de um cenário com um ator e as instruções indicadas ao ator

Como se pode ver no canto direito, a aplicação dispõe de tutoriais que permitirá ao utilizador aprender manusear as ferramentas. Tem como exemplo animações, sons, contas, etc.

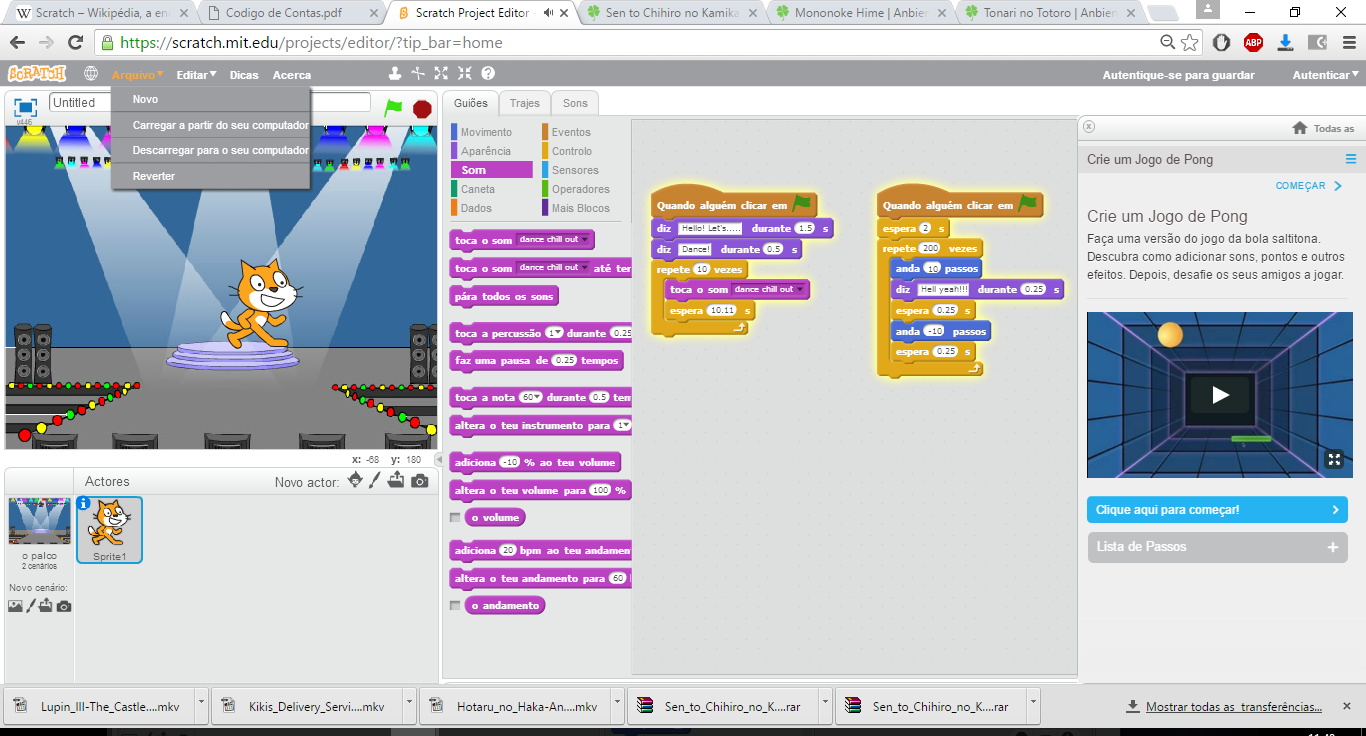


Figura 20- Exemplo do tutorial do lado direito " crie um jogo de pong "

Quando selecionado um tutorial, este irá ensinar os procedimentos passo a passo, permitindo também adicionar mais funções ao gosto do utilizador.

Quando o utilizador estiver pronto para “correr” com as suas instruções, carrega no botão com um desenho de bandeira verde, se por algum motivo o utilizador se esqueceu de algo ou não fizer o que é pedido pode sempre interromper a operação através de um botão que tem como símbolo um stop, localizado ao lado do botão bandeira verde.

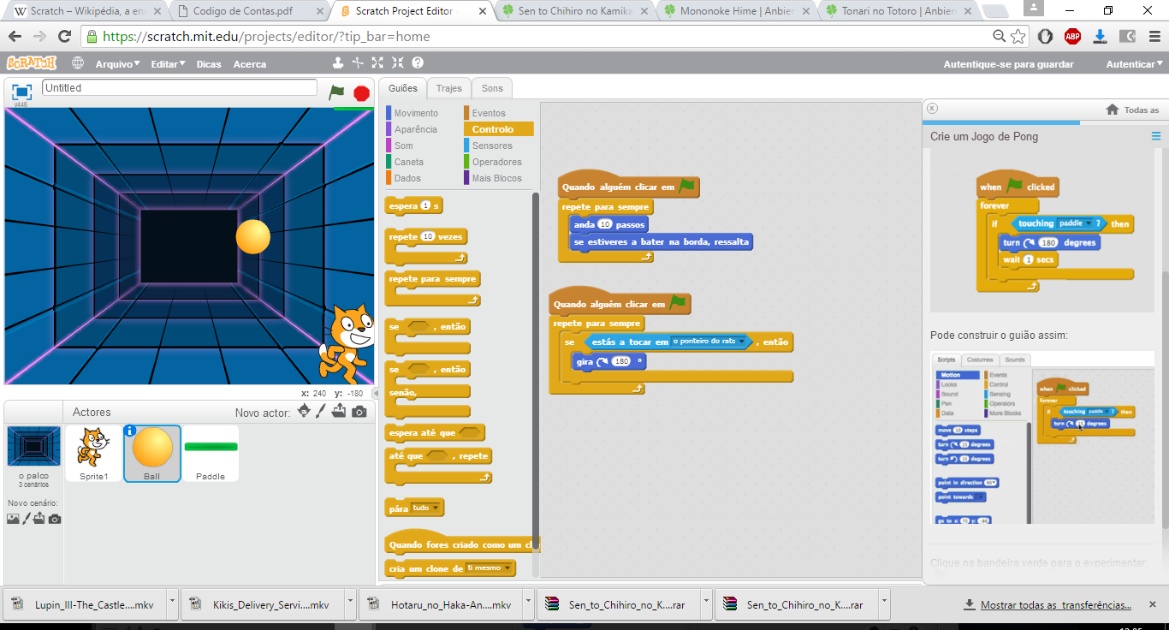


Figura 21- Exemplo de realização de um tutorial

A aplicação já tem cores predefinidas para cada ator, no entanto, o utilizador pode personalizar ao seu gosto. Existe também certos atores que contém mais do que um Sprite para facilitar a realização de animações.

Scratch 2 contém várias bibliotecas, sendo eles de atores, cenários e musicas.

A biblioteca atores contém inúmeros atores para adicionar á janela principal, o mesmo se pode dizer dos cenários.

Na biblioteca músicas, existem dezenas de excertos de música para adicionar ao cenário.

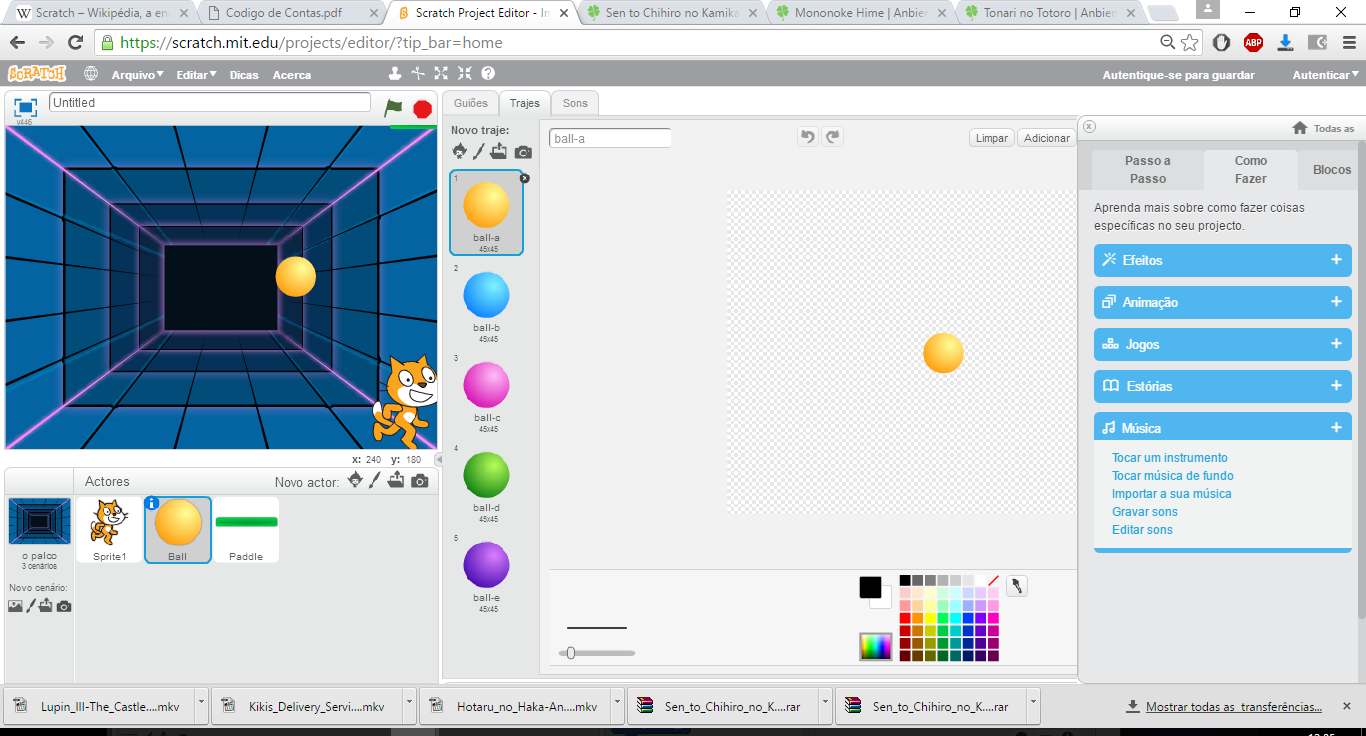


Figura 22- Cores para um ator

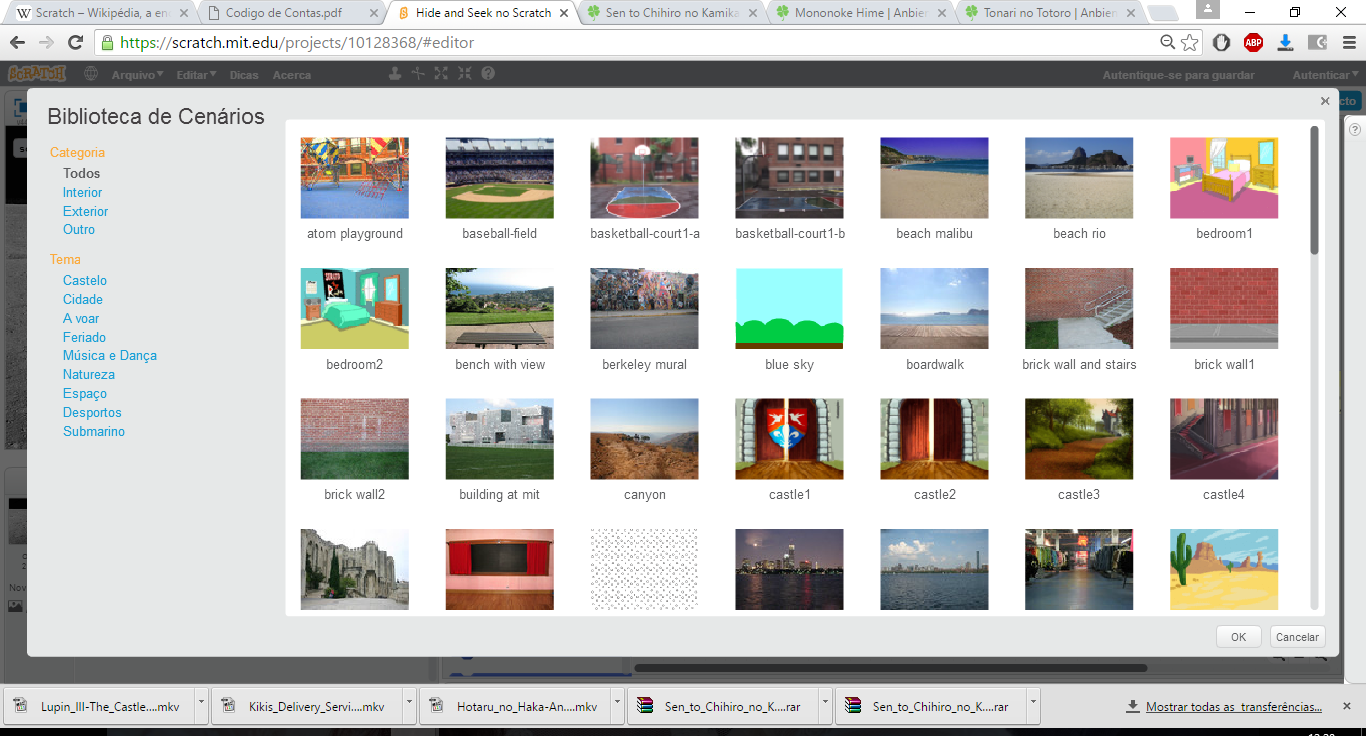


Figura 23- Lista de Cenários

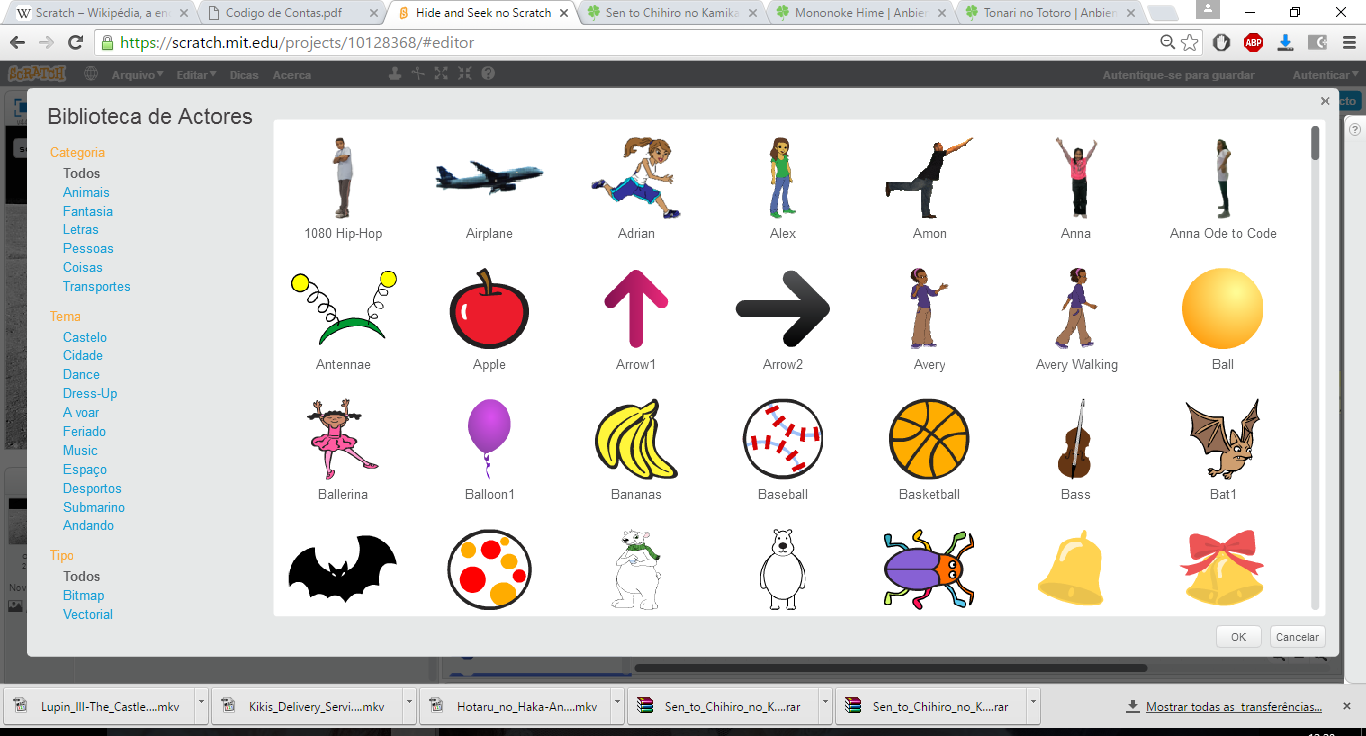


Figura 24 - Lista de Atores

## Outras informações

Na página oficial do Scratch contém ainda projetos realizados por outros utilizadores permitindo ao utilizador abrir e ver/testar o projeto. Pode aceder ao código todo desse projeto e alterá-lo para uso próprio.

Poderá inscrever-se na página oficial, podendo submeter os projetos feitos pelo utilizador ou para colocar perguntas no fórum ou até mesmo responder.

Contém ainda alguma informação acerca do Scratch, quem usa o Scratch, o porquê de aprender a programar, investigações, citações, o Scratch nas escolas, apoios e financiamentos, etc.

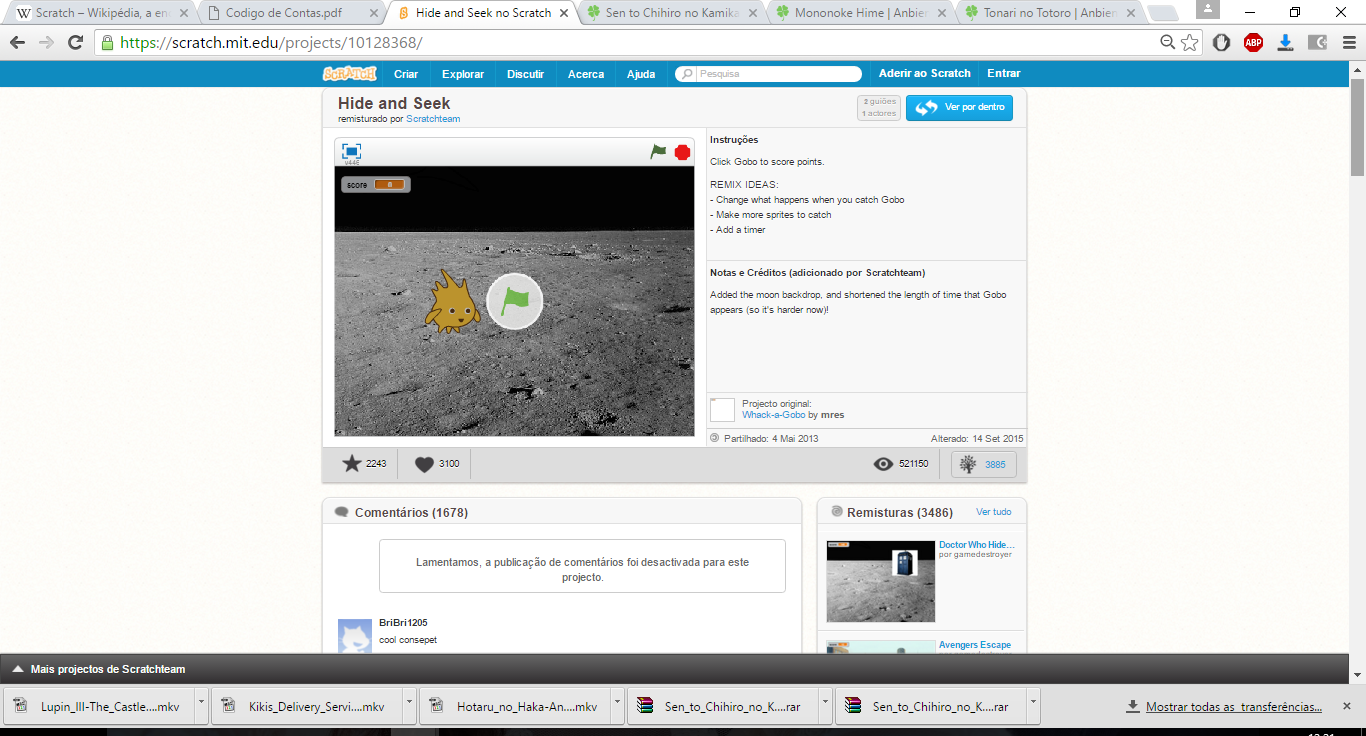


Figura 25- Exemplo de um projeto feito por um utilizador

# Outras referências

## LARP

LARP é uma aplicação para o desenvolvimento de algoritmos através de pseudocódigo ou fluxogramas. Foi criado por Marco Lavoie em 2004, atualmente vai na versão 4.0.6 lançado em 2008.

A sintaxe permite tirar partido dos conceitos de programação estruturada para construir pseudo-código e / ou algoritmos baseados em fluxogramas.

Os conceitos abordados incluem variáveis ​​e contentores, estruturas condicionais e iterativos, modularidade e armazenamento de dados.

O ambiente de desenvolvimento de LARP inclui funcionalidades de programação abrangentes, como a edição, compilação e prevenção de plágio, o que torna o software particularmente adequado para aprender a programar.

Usando LARP, os programadores iniciantes podem concentrar-se em desenvolver algoritmos ao invés de dominar a sintaxe difícil num ambiente de desenvolvimento dura, isto é, NetBeans, Eclipse, etc.

A facilidade do uso do pseudo-código de LARP e as sintaxes dos fluxogramas e do convívio do seu ambiente de desenvolvimento torna o software particularmente adequado para o ensino e desenvolvimento de programação estruturada.

O LARP foi concebido por um instrutor de programação e usado em cursos de nível universitário para a introdução ao ensino de desenvolvimento de algoritmos.

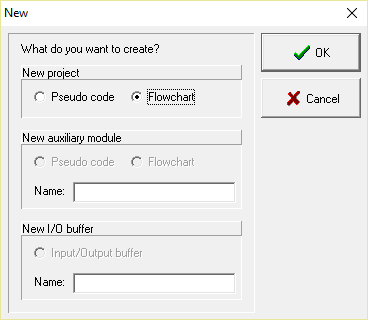
A facilitar ainda mais o uso do software, o LARP inclui ajuda sensível ao contexto, on-line e impressa, que descreve em detalhes a sintaxe do pseudo-codigo e fluxogramas, bem como todas as funcionalidades do ambiente de desenvolvimento. Esta documentação é apresentada de uma forma instrutiva.

Inclui muitos exemplos e oferece explicações detalhadas de conceitos relacionados, tais como código binário e recursão. Ambos os alunos e seus professores irão beneficiar usando a documentação para estudar algorítmica baseada resolução de problemas.

LARP é um software educacional essencial para o ensino e aprendizagem de programação estruturada e desenvolvimento de algoritmos. A facilidade de uso e funcionalidades tornam a ferramenta ideal para ensinar os alunos de todas as idades para a programação.

Acedendo á aplicação LARP:

É-nos dada a opção de como queremos desenvolver o nosso algoritmo quando criamos um novo projeto.



Neste caso foi optado o desenvolvimento do algoritmo por fluxogramas.

No modo de desenvolvimento do algoritmo em fluxogramas, no lado esquerdo da aplicação encontra-se instruções de fluxogramas. Para por uma dessas instruções no fluxograma basta fazer drag and drop para um dos pontos que existem ao longo do fluxograma.

Figura 26- Escolha do tipo de desenvolvimento do algoritmo

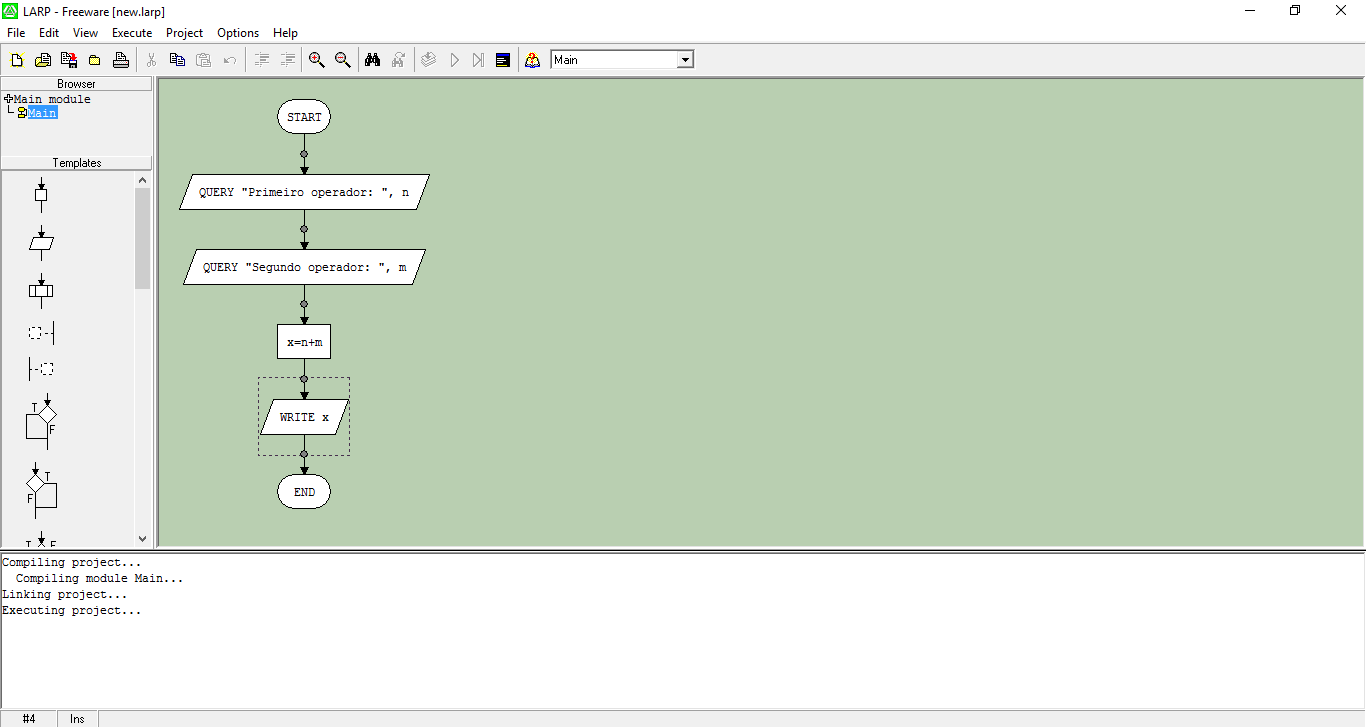


Figura 27- Aspeto do LARP no desenvolvimento de Fluxogramas

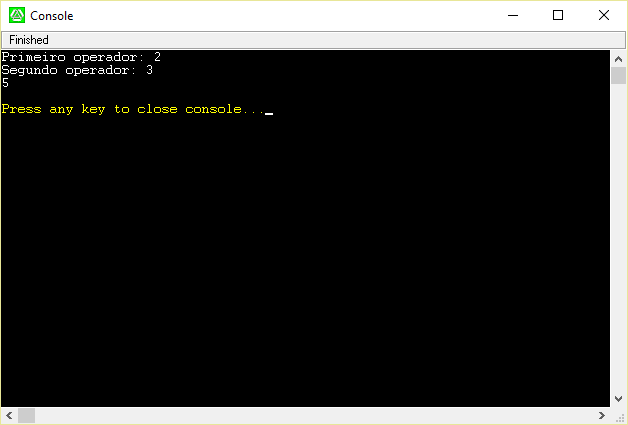


Figura 28- Consola do LARP

O LARP oferece várias características tais como a execução de passo a passo, compilação, execução de passo animado, meter um breakpoint, pausar a execução, continuar a execução. Permite ter várias funções, aceita recursividade.

Contém opções de configuração do tipo de letra, tamanho, cor, definições de cores para variáveis, funções, operações, cor de fundo, etc.

Na barra de ferramentas, o botão Help contém mais informação sobre a aplicação, nomeadamente, como trabalhar com a aplicação, funcionalidades, tipos de variáveis com que se pode utilizar, etc.



Figura 29- Menu help do LARP

No modo de desenvolvimento do algoritmo em pseudocódigo contém no lado esquerdo a aplicação uma lista de templates drag and drop que permitem a simplicidade de não cometer erros de sintaxe.

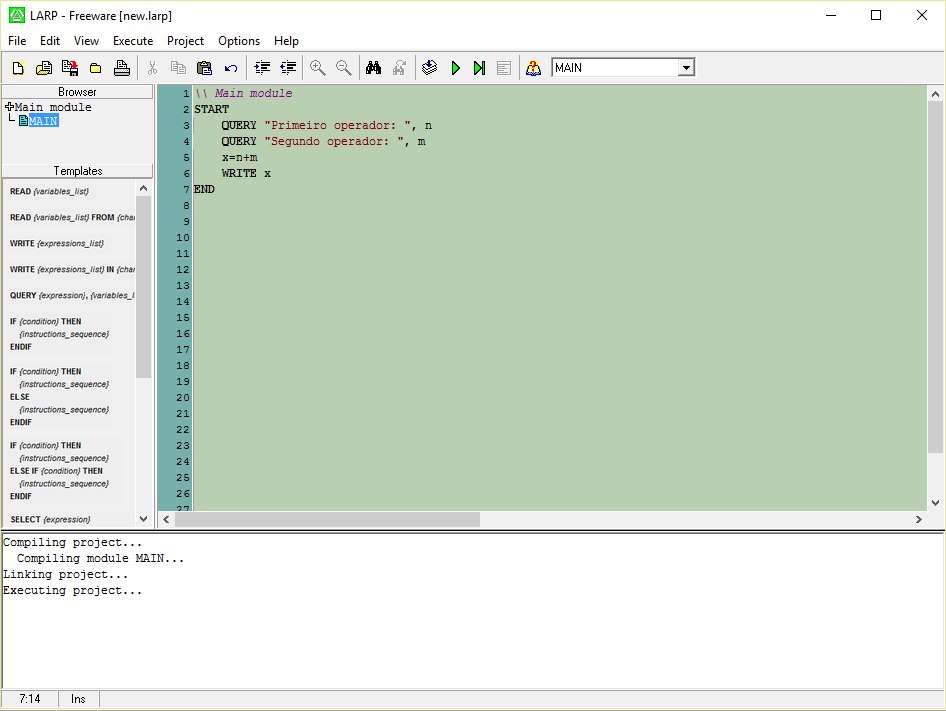


Figura 30 - Aspeto do desenvolvimento de pseudocódigo no LARP

Depois de finalizado o algoritmo por fluxograma pode converter para pseudo-código, e vice-versa, isto é, depois de finalizado o pseudo-código, pode convertê-lo para fluxograma.

Ao contrário das outras aplicações, o LARP não tem website oficial, no Wikipédia menciona o link, porém, esse já não se encontra activo.

É possível ao utilizador, para o caso de o LARP conter bugs, relatar problemas para melhorar a aplicação.

## Logic scheme compiler

Outra aplicação para desenvolvimento de algoritmos por fluxogramas, desenvolvido por Ursuleanu Andrei. Pouco se sabe sobre a aplicação. O link onde disponibiliza a aplicação para download é <http://logic-scheme-compiler.soft112.com/>.

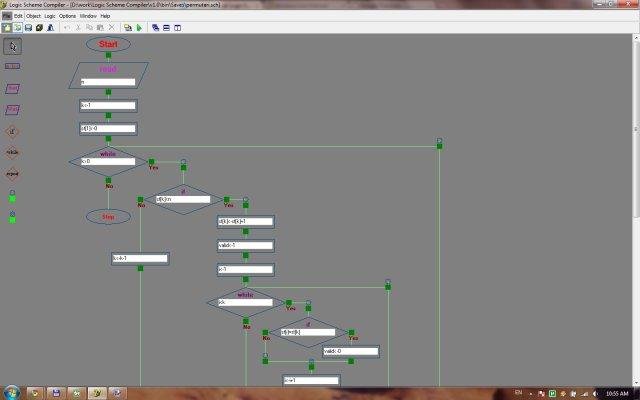


Figura 31 - Aspeto geral da aplicação Logic Scheme Compiler

## Alice

É uma aplicação inovadora em 3D que facilita a criação de histórias, criar um jogo interativo ou um vídeo para partilhar na internet.

Alice é uma aplicação gratuita de aprendizagem para aprender o fundamental sobre a programação orientada a objetos. Permite aos estudantes aprenderem conceitos tais como a criação de animações e videojogos simples. Os objetos 3-D em Alice (Pessoas, animais, veículos, etc) ocupam um mundo virtual. Cabe aos alunos criar um programa para animar os objetos.



Figura 32- Aspeto geral da aplicação Alice

O interface interativo de Alice permite aos utilizadores utilizar o sistema de drag and drop de várias instruções para criar um programa, que correspondem às instruções standard de uma programação orientada a objetos como o Java, C++ e C#.

Alice permite aos utilizadores verem imediatamente como a animação decorre á medida que percorre as instruções, facilitando a sua compreensão a relação das instruções com a animação.

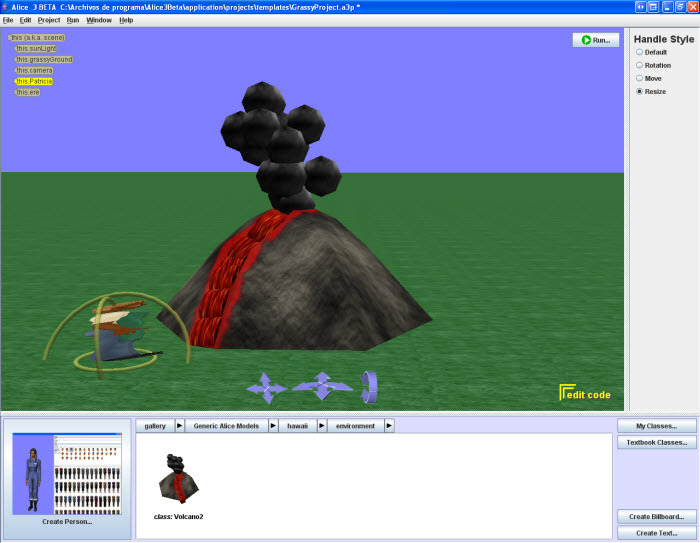


Figura 33 - Criação de uma classe em Alice

Ao manipular os objetos no mundo virtual, os utilizadores ganham experiência com toda a construção do programa tipicamente utilizada numa cadeira introdutória á programação orientada a objetos.

Alice contem um website oficial <http://www.alice.org/index.php> onde disponibilizam mais informação sobre a aplicação, downloads de várias versões, desde alice 2 a alice 3.

Contém um separador teaching onde se pode encontrar toda a informação necessária para a introdução da aplicação, um tutorial, um guia, matérias dadas em aulas.  
 Contém uma comunidade onde se pode obter os contactos de professores dedicados a esta aplicação, a página de facebook de Alice, um blog e fórums.

No separador publicações encontra-se papeis e pesquisa, é aí que são disponibilizados alguns artigos de pesquisa em pdf.

Por fim, mas não menos importante, o site contém um separador de suporte onde se pode obter ajudas da aplicação e os contactos dos criadores da Alice

## Visual Logic

Aplicação de realização de algoritmos através de fluxogramas, providencia o mínimo de sintaxe para a programação dos conceitos essenciais. Inclui variáveis, entrada, atribuição, saída, condições, ciclos, procedimentos, arrays e ficheiros.

Baseado na pesquisa que mostra que os alunos têm mais sucesso quando começam a focar mais nos conceitos que a sintaxe. O Visual Logic permite aos instrutores a mostrarem aos alunos o conteúdo de uma maneira limpa e ilustrada.

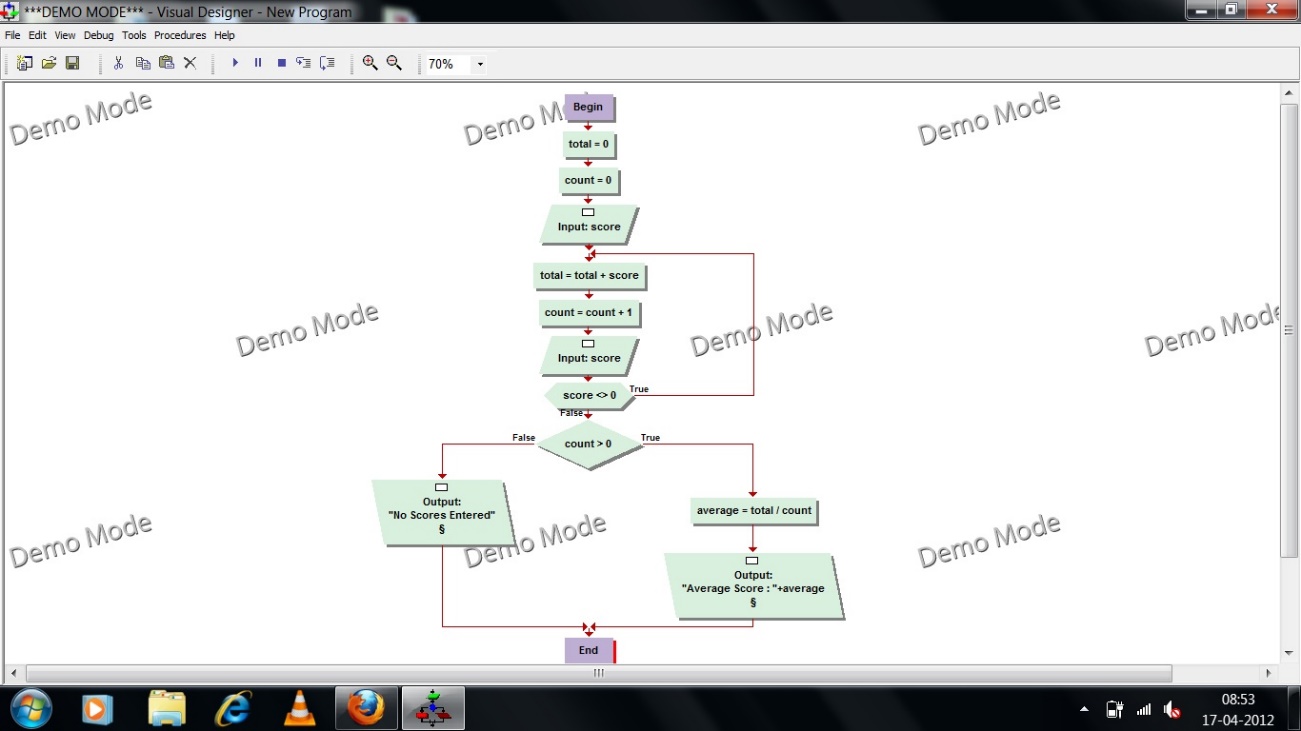


Figura 34 - Aspeto geral da aplicação Visual Logic

Características da aplicação:

* Fluxogramas executáveis – Providencia aos alunos feedback imediato.
* Ciclos – Pré-teste, pós-teste e dentro de um ciclo.
* Condições – condições do tipo nested e compound.
* Procedimentos – Parâmetros passados por valor, referência. Recursão.
* Debugging – Janela de variáveis e debugging passo a passo.
* Entrada/Saída melhorada – Caixa de diálogo, consola e ficheiro de texto.
* Interface intuitivo – cortar, colar, copiar, drag and drop.
* Características adicionais – Arrays, suporte multimédia e gráficos.

Site oficial - <http://www.visuallogic.org/>

Onde: pode encontrar a aplicação disponível para download de forma gratuita, contem ainda um tutorial, faq e contactos.

## BlueJ

Aplicação IDE para programação de linguagem em Java, desenvolvido para propósitos educacionais mas também serve para ser utilizados para projetos de pequena escala. Precisa do JDK instalado na máquina para poder correr com o BlueJ.

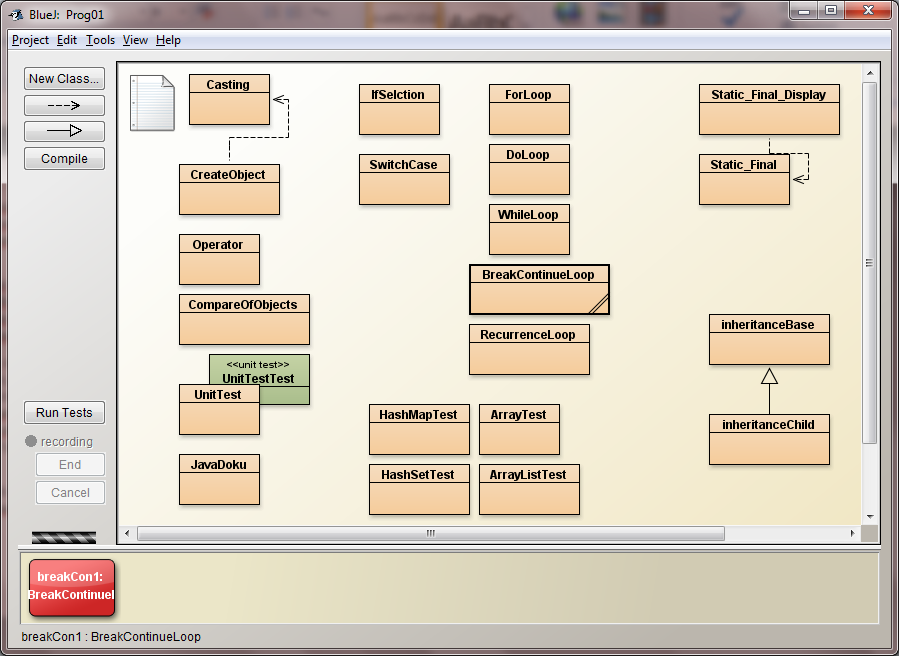


Figura 35 - Aspeto geral do BlueJ

BlueJ foi desenvolvido para dar suporte de aprendizagem e ensino de programação orientada a objetos e como resultado o design é diferente de todos os outros ambientes de desenvolvimento. A janela principal mostra graficamente a estrutura de uma classe de uma aplicação.

Os objetos podem ser interactivamente criados e testados. Esta facilidade de interação, combinado com um simples interface permite fácil utilização dos objetos em desenvolvimento.

Os conceitos de orientação a objetos são representados visualmente e o seu design de interação no interface.

## NetBeans BlueJ Edition

O NetBeans BlueJ Edition é uma aplicação IDE que serve como introdução aos alunos para depois utilizarem o Blue J, transitarem para a aplicação mais profissional, NetBeans. Esse plugin, instalado no NetBeans oferecia várias funções tais como a vista do projeto e uma barra de objetos. No entanto o plugin da aplicação deixou de ter suporte quando o NetBeans ia na versão 6.5.

As diferenças entre o NetBeans BlueJ Edition e o BlueJ são:

|  |  |  |
| --- | --- | --- |
|  | **BlueJ** | **NetBeans BlueJ Edition** |
| **Banco de Objetos** | Integrado no BlueJ | Não tem |
| **Interatividade com classes ADHOC** | Inerente no banco de objetos | Não tem |
| **Programas de Compilação** | Usa versão selecionada do Javac | Usa versão selecionada do Javac, mas com parâmetros as usar no Java 1.4 quando usado um projeto BlueJ |
| **Método principal, Classe principal** | Não afeta a aplicação | Requer correr a aplicação |
| **Vista BlueJ** | Janela principal. Permite o movimento de classes. Mostra as relações entre classes. | Uma vista opcional, só usa imagens para listas simplificadas. Não mostra os movimentos nem as relações. |
| **Debugging** | Tem uma consola de debugging. Tem bugs para projetos de grande escala. | Tem todas as vistas tradicionais de debugging. |

## Code2Flow

O code2flow é um website, mais concretamente um script experimental que corre através do código fonte do projeto, procurando definições de função, depois corre outra vez, procura onde as funções são chamadas.

O script conecta os pontos e apresenta como um fluxograma estimado pela estrutura funcional do programa criado pelo utilizador. É especialmente útil para desembaraçar código mal otimizado (“confuso”).

O code2flow pretende fornecer uma visão geral da estrutura de projetos simples. O código fonte que é um pouco esotérico vai contornar isso. Mesmo com código normal, existem muitas limitações conhecidas e muitos erros.

Depois de escrever o código desejado, pode guardar o fluxograma em PDF, PNG ou SVG.

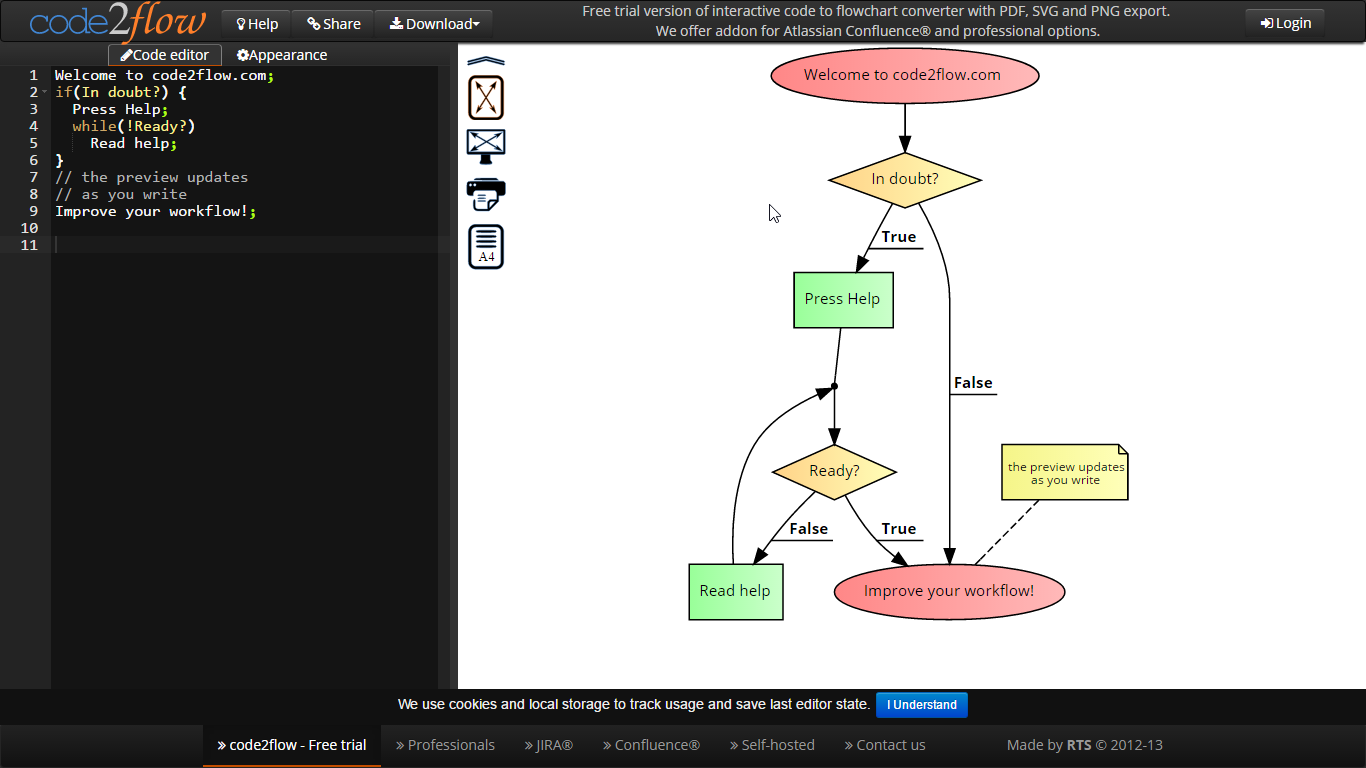


Figura 36 - Aspeto geral do Code2Flow

# Netgrafia

## Flowgorithm

* Site oficial - <http://www.flowgorithm.org/>
* Aplicação oficial - <http://www.flowgorithm.org/download/index.htm>
* Mais informação sobre Flowgorithm - <https://en.wikipedia.org/wiki/Flowgorithm>

## Sicas

* Artigo publicado - <https://www.researchgate.net/publication/228328489_SICAS_Interactive_system_for_algorithm_development_and_simulation>

## Raptor

* Página Oficial - <http://raptor.martincarlisle.com/>

## Portugol Studio

* Download da Aplicação - <https://sourceforge.net/projects/portugolstudio/>

## Scratch

* Página oficial - <https://scratch.mit.edu/>
* Download da aplicação - <https://scratch.mit.edu/scratch2download/>
* Outras informações - <https://en.wikipedia.org/wiki/Scratch_(programming_language)>

## LARP

* Download da aplicação e informação sobre a aplicação - <http://larp.software.informer.com/>

## Logic Scheme Compiler

* Download da aplicação - <https://sourceforge.net/projects/logicschemecomp/>

## Alice

* Página oficial - <http://www.alice.org/index.php>
* Download da aplicação Alice 3 - <http://www.alice.org/index.php?page=downloads/download_alice3.1>
* Outras informações para o Alice 3 - <http://www.alice.org/3.1/index.html>

## Visual Logic

* Página oficial - <http://www.visuallogic.org/> (contém também o link para download)
* Alguma informação - <http://www.visuallogic.org/Overview.html>

## BlueJ

* Página oficial - <http://www.bluej.org/>
* Alguma informação - <http://www.bluej.org/about.html>

## NetBeans BlueJ Edition

* Página oficial (já sem suporte) - <http://bluej.org/netbeans/>

## Code2Flow

* Página oficial (aplicação free trial mode) - <http://code2flow.com/>