

PROGRAMA DE INTRODUÇÃO A PROGRAMAÇÃO

10.º ANO DE ESCOLARIDADE

COMPONENTE DE FORMAÇÃO ESPECÍFICA [Opção a)]
ÁREA: Ciência e Tecnologia

ENSINO SECUNDÁRIO
(Versão Experimental)

Ficha Técnica

Título

Programa de Introdução à Programação – 10.º Ano de escolaridade – Componente de Formação Específica Opção a) – Área de Ciência e Tecnologia

Editores/Autores

Ministério da Educação

Concetores:

Celestino I. de Barros

Érico Pinheiro Fortes

Benjamim Silva

Validador:

Elizabete Andrade

Coordenação

Direção Nacional de Educação / Serviço de Desenvolvimento Curricular

Elaboração

Universidade de Cabo Verde (Uni-CV)

Propriedade

Ministério da Educação

Palácio do Governo

C.P. 111

Tel.: +238 262 11 72 / 11 76

Cidade da Praia – Santiago

Data: setembro 2022

Índice

1. INTRODUÇÃO	3
2. APRESENTAÇÃO, FINALIDADES E ORIENTAÇÕES GERAIS DA DISCIPLINA	4
2.1. Propósito da Disciplina no Ensino Secundário	4
2.2. Finalidades.....	4
2.3. Competências a desenvolver	4
2.4. Visão Geral dos Temas /Conteúdos	5
2.5. Indicações Metodológicas gerais	7
2.6. Indicações gerais para a Avaliação das Aprendizagens.....	8
3. ROTEIROS DE APRENDIZAGEM.....	0
3.1. Natureza e Roteiros de Aprendizagens do 10.º, 11.º e 12.º anos	0
3.2. Roteiro de Aprendizagem e Indicadores de Avaliação do Programa do 10.º ano	0
4. BIBLIOGRAFIA CONSULTADA.....	9
5. RECURSOS EDUCATIVOS RECOMENDADO	9

VERSÃO EXPERIMENTAL

1. INTRODUÇÃO

A disciplina *Introdução à Programação* é uma disciplina do 10º e 11º ano de escolaridade do Ensino Secundário, tem uma carga horária de três (3) horas divididas por aulas de cinquenta (50) minutos ao longo das trinta e oito (38) semanas de cada ano letivo.

Esta disciplina pretende conduzir os estudantes à resolução de problemas de uma forma estruturada e sistematizada, promover a capacidade de análise crítica através de análises de todos os elementos intervenientes e permitir oportunidades de desenvolvimento profissional que permitam uma literacia digital generalizada. Pelo que, a aquisição de competências em *Introdução à Programação*, a partir do 10º ano, afigura-se fundamental para o cumprimento do objetivo estipulado no programa de Governo: *“O Governo promoverá, em linha com o dividendo digital progressivamente gerado pela sociedade cabo-verdiana, a inclusão e a integração digital de toda a comunidade educativa, através do fomento da literacia digital, associado às acessibilidades e ao fomento da cultura de usabilidade comunitária das tecnologias de base digital em todos os níveis do sistema de ensino, mediante reestruturação e reforço da capacidade tecnológica da Rádio e Televisão Educativas, desenvolvendo um Sistema Global de Comunicação e de Multimédia Educativas que envolva a adoção de um plano de desenvolvimento infraestrutural da rádio e televisão educativas e seu agregador digital e de multimédia que transformarão as escolas em Polos de Integração Digital da Comunidade Educativa para a Disseminação de conteúdos digitais e de multimédia (PID & RDDM) nas comunidades, ligando todas as escolas numa única rede, de modo a combater a exclusão digital das famílias* (Programa de Governo e Moção de Confiança do VIII Governo Constitucional da República de Cabo Verde, 2021-2026, p. 26).

No 10º ano, através do estudo de fluxograma, algoritmia e programação estruturada utilizando a ferramenta *Flowgorithm*, pretende-se criar as bases necessárias para a evolução do aluno na área das linguagens de programação.

No 11º ano, utilizando o *Python*, linguagem de grande divulgação, o aluno irá adquirir não só as bases da programação, como deverá ser capaz de elaborar pequenas aplicações informáticas utilizando uma ferramenta visual atual.

Dada a natureza da disciplina, é aconselhável que não existam mais de dois alunos por computadores, podendo a turma ser dividida em turnos, ou não, conforme o número de alunos e o número de computadores existentes na sala de aula.

2. APRESENTAÇÃO, FINALIDADES E ORIENTAÇÕES GERAIS DA DISCIPLINA

2.1. Propósito da Disciplina no Ensino Secundário

Nesta disciplina, pretende-se estimular os estudantes a atualizarem e aprofundarem os seus conhecimentos no que tange ao raciocínio lógico e a aquisição de competências básicas sobre a lógica de programação, técnicas de programação e ferramentas indispensáveis para entender melhor os problemas, segmentá-los em instruções e procedimentos, de modo a facilitar a sua compreensão e o desenvolvimento de soluções através de algoritmos. Almeja-se ainda que, os estudantes tenham uma noção clara sobre como se dá a entrada e saída de dados e a interação homem-máquina.

2.2. Finalidades

Pretende-se na disciplina de Introdução à Programação do 10º ano, alcançar os seguintes fins:

- Capacitar os estudantes no que tange a compreensão de fundamentos lógicos de programação, algoritmos, tipos de dados, estruturas e funções de controle elementares utilizados dentro da estrutura de paradigmas de programação imperativa;
- Fomentar o desenvolvimento do raciocínio lógico dos estudantes;
- Capacitar os estudantes de técnicas de análise de problemas, formalização, codificação e resolução de problemas tipificados.
- Apresentar e familiarizar com linguagens e técnicas de representação e codificação de algoritmos elementares;
- Apresentar de conceitos básicos de linguagens de programação e elementos estruturais para a construção de algoritmos;

2.3. Competências a desenvolver

No âmbito desta disciplina, pretende-se que os estudantes desenvolvam as seguintes capacidades:

- Os estudantes serão capazes de aplicar as suas habilidades de programação de forma assertiva e em conformidade com diferentes cenários e requisitos inerentes aos problemas de programação que se pretende resolver.

- Criatividade dos estudantes através dos seus projetos, permitindo-lhes expressar as suas ideias e apresentar soluções inovadoras.
- Aumento gradativo da capacidade de análise de problemas, pensamento lógico, compreensão de operações, condicionais e procedimentos essenciais para o desenvolvimento de soluções através de algoritmos.

2.4. Visão Geral dos Temas /Conteúdos

1. Introdução à lógica de Programação

1.1. Lógica

1.2. Etapas na solução de problemas

1.3. Sistematização de procedimentos

1.3.1. Formas de Representação (Fluxograma)

1.3.1.1. Simbologia

2. Introdução à Programação

2.1. Conceitos básicos

2.2. Tipo de dados

2.3. Constantes

2.4. Variáveis

2.4.1. Introdução

2.4.2. Tipos

2.4.3. Identificação

2.4.4. Declaração

2.4.4.1. Variáveis locais

2.4.4.2. Variáveis globais

2.4.5. Atribuição

2.4.6. Operadores

2.4.6.1. Introdução

2.4.6.2. Tipos

2.4.6.2.1. Aritméticos

2.4.6.2.2. Relacionais

2.4.6.2.3. Lógicos

2.4.6.3. Procedência de operadores

2.4.7. Estruturas de Controle / Tomada de decisão

2.4.7.1. Introdução

2.4.7.2. Valores lógicos

2.4.7.3. *If- Else*

2.4.7.4. Bloco de instruções

2.4.8. Estruturas de Seleção

2.4.8.1. Simples

2.4.8.2. Composta

2.4.8.3. Encadeada

2.4.8.4. Múltipla escolha

2.4.9. Boas práticas de Organização estrutural de um Programa

2.4.9.1. Indentação

2.4.9.2. Comentar o código

2.4.9.3. Atribuir o nome de variáveis de forma intuitiva

2.4.9.4. Atribuir o nome de funções de forma intuitiva

2.4.9.5. Padronização na atribuição do nome das constantes

2.4.10. Estruturas de Repetição (Ciclos)

2.4.10.1. Do

2.4.10.2. While

2.4.10.3. Do...While

2.4.10.4. Operadores de incrementação e decrementação

2.4.10.5. Ciclos infinitos

2.4.11. Funções e procedimentos

2.4.11.1. Introdução

2.4.11.2. Características de uma função

2.4.11.3. Atribuição do nome a uma função

2.4.11.4. Funcionamento

2.4.11.5. Parâmetros

2.4.11.6. Retorno de valores através de uma função

2.5. Indicações Metodológicas gerais

Atendendo que geralmente os estudantes, ao iniciarem a disciplina Introdução à Programação, apresentam níveis de conhecimento, diferenciados, é importante que o(a) professor(a) realize uma avaliação diagnóstica, com o propósito de poder orientar as suas planificações de modo a permitir aos alunos que mostrem dominar as competências essenciais delineadas para o respetivo ano de escolaridade o desenvolvimento de competências mais avançadas.

A disciplina tem um carácter predominantemente prático e experimental. Torna-se, por isso, necessário implementar metodologias ativas, tais como a experimentação, a pesquisa e a resolução de problemas. Neste sentido, as aulas deverão privilegiar a participação dos(as) alunos(as) em projetos, na resolução de problemas ou na abordagem de temas de outras áreas disciplinares.

Neste contexto, a articulação de saberes das várias disciplinas deverá ser posta em prática através da realização de projetos que permitam ao estudante encarar a utilização de aplicações não como um fim em si, mas como uma ferramenta para facilitar a comunicação e a resolução de problemas.

Sugere-se também a realização de projetos colaborativos com estudantes de outras escolas cabo-verdianas ou mesmo de outros países, otimizando assim as potencialidades de comunicação via Internet.

Na abordagem de conteúdos mais teóricos, aconselha-se dinâmicas mais ativas, uma vez que o tempo de concentração dos estudantes é limitado, nomeadamente com propostas de atividades lúdicas- pedagógicas, questionários, entre outros.

O(a) professor(a) deve utilizar as potencialidades da Internet na dinamização das atividades letivas, nomeadamente com recurso a:

- **GoAnimate** (www.goanimate.com);
- **KEduca** (<http://keduca-project.soft112.com/>);
- **HotPotatoes** (<https://ferramentaseducativas.com/index.php/aplicacoes/offline/60-hotpotatoes>), de entre outros.

Dinamização das Aulas

Nas aulas de introdução a um novo software, para exemplificação e/ou demonstração de aspetos práticos, o(a) professor(a) deverá recorrer a projeção, tornando-os visíveis a toda a turma.

Assim, numa primeira fase, sugere-se a metodologia da aprendizagem por execução de tarefas, através de exercícios sob a forma de fichas de trabalho, onde estejam listadas e identificadas as tarefas a executar pelos(as) alunos(as).

Numa segunda fase, dá-se maior ênfase à descoberta dos conteúdos por parte dos estudantes, pelo que se sugere a metodologia da descoberta guiada, mediante fichas de trabalho, com o exemplo do resultado a obter e em que são indicados alguns passos para a sua obtenção. Os alunos(as) experimentam e descobrem os procedimentos que estão em falta executando-os, a fim de conseguirem o resultado pretendido.

Assim, recomenda-se que o(a) professor(a):

- Realize um breve enquadramento teórico de cada tema e proceda à demonstração do funcionamento global do *software*;
- Exemplifique com a ajuda do computador;
- Privilegie as aulas práticas para que os(as) alunos(as) utilizem o computador;
- Estimule o trabalho de grupo;
- Proponha aos(as) alunos(as) atividades de carácter experimental e de pesquisa;
- Proponha aos(as) alunos(as) a realização de trabalhos práticos (que possam eventualmente ser postos ao serviço da comunidade e/ou outras disciplinas), nos quais tenham de aplicar os conhecimentos adquiridos;
- Incite os(as) alunos(as) a procurar, a manusear e a utilizar *software livres* e aplicações equivalente ao utilizado nas aulas;

[2.6. Indicações gerais para a Avaliação das Aprendizagens](#)

Os procedimentos de avaliação dos estudantes resultam da natureza eminentemente prática e experimental da disciplina, privilegiando-se a vertente

diagnóstica, formativa e sumativa que são indispensáveis à orientação dos processos de ensino e de aprendizagem.

Avaliação Diagnóstica

É fundamental que no início do ano letivo, seja realizada uma avaliação diagnóstica que permita identificar grupos diferenciados e estabelecer um plano de ação para cada grupo de estudantes, não perdendo de vista o desenvolvimento, para todos eles, das competências essenciais que se encontram definidas neste programa.

Avaliação Formativa

A avaliação formativa, enquanto autorreguladora das aprendizagens, acarreta uma postura bastante ativa do(a) professor(a), que deve procurar acompanhar o desenvolvimento de todas as atividades, identificando as principais dificuldades, que deverão ser superadas, no momento, na aula seguinte, ou no âmbito das atividades de complemento educativo, garantindo que o estudante se sinta permanentemente motivado e acompanhado.

A avaliação deverá também ter um carácter contínuo, devendo valorizar o processo /desenvolvimento de capacidades dos estudantes para alcançarem os objetivos. Assim, importa recorrer à observação direta dos trabalhos desenvolvidos pelo estudante durante as aulas, utilizando para isso grelhas de avaliação diversificados que permitam registar o seu desempenho nas situações que lhe são proporcionadas e a progressão na aprendizagem ao longo do ano letivo, nomeadamente, quanto ao interesse e à participação no trabalho, à capacidade de desenvolver trabalho em grupo, à capacidade de explorar, investigar e mobilizar conceitos em diferentes situações, bem como, relativamente à qualidade do trabalho realizado e à forma como estudante gere, organiza e autoavalia. Neste processo, o(a) professor(a), deve motivar, elogiar e instigar o estudante para que esta se sinta cada vez mais integrado(a) no seu processo educativo e que vá desenvolvendo o gosto pela disciplina.

Avaliação Sumativa

A disciplina tem um carácter predominantemente prático e experimental. Torna-se, por isso, necessário a realização de avaliação através de atividades que incidam sobre a aplicação prática e contextualizada dos conteúdos: a

experimentação, a pesquisa e a resolução de problemas. Neste sentido, a avaliação deverá privilegiar a participação dos estudantes em projetos práticos que permitam avaliar a consolidação dos conhecimentos adquiridos e das competências desenvolvidas ao longo do processo de ensino/aprendizagem. Outra fonte de informação que pode dar um contributo importante para a avaliação reside na conceção, realização, apresentação e discussão em turma de um ou vários projetos interdisciplinares, que permitem a mobilização dos saberes adquiridos na disciplina em função de problemas ou temas de pesquisa que poderão estar ligados a outras áreas do conhecimento.

Deverão ser previstos momentos de avaliação sumativa, procedendo-se à aplicação de provas de carácter prático ou teórico-prático de acordo com os conteúdos de cada área temática que permitam avaliar os conhecimentos e competências adquiridas.

VERSÃO EXPERIMENTAL

3. ROTEIROS DE APRENDIZAGEM

3.1. Natureza e Roteiros de Aprendizagens do 10.º, 11.º e 12.º anos

3.2. Roteiro de Aprendizagem e Indicadores de Avaliação do Programa do 10.º ano

No final o/a aluno/a deverá ser capaz de:

Áreas temáticas	Conteúdos e Conceitos	Objetivos de Aprendizagem (Conhecimentos, Procedimentos, atitudes)	Sugestões Metodológicas	Indicadores de Avaliação das Aprendizagens
3. Introdução à lógica de Programação	3.1. Lógica 3.2. Etapas na solução de problemas 3.3. Sistematização de procedimentos 3.3.1. Formas de Representação (Fluxograma) 3.3.1.1. Simbologia	- Caracterizar o que é lógica de programação - Explicar o conceito de algoritmo - Explicar o conceito de linguagem - Caracterizar as formas de representação de algoritmos	- Apresentação pelo professor de exemplos concretos dos conceitos introdutórios a desenvolver nas unidades seguintes, bem como nos anos subsequentes. - Discussão dos alunos em grupo sobre os	Avaliação deve ser essencialmente formativa, centrada no desenvolvimento do processo e do progresso da aprendizagem, pois é contínua e sistemática e tem função diagnóstica, permitindo ao professor, ao aluno, ao encarregado de

		<ul style="list-style-type: none"> - Explicar o que é um programa - Conceituar Linguagem de programação 	<p>diferentes conteúdos da unidade.</p> <ul style="list-style-type: none"> - Realização de tarefas práticas de carácter essencialmente lúdico que permitam aos alunos evidenciar competências relacionadas com a programação, nomeadamente ao nível da algoritmia (p. ex. reorganizar uma história, pela sua sequência lógica). 	<p>educação e a outras pessoas ou entidades legalmente autorizadas obter informação sobre o desenvolvimento das aprendizagens, com vista ao ajustamento de processos e estratégias.</p> <p>Avaliação sumativa</p> <p>A avaliação sumativa que consiste num juízo globalizante que conduz à tomada de decisão, no âmbito da classificação e da aprovação da disciplina, pois é o somatório de todos os itens avaliativos pré-definidos pelo professor.</p>
--	--	---	--	--

<p>4. Introdução à Programação</p>	<p>4.1. Conceitos básicos</p> <p>4.2. Tipo de dados</p> <p>4.3. Constantes</p> <p>4.4. Variáveis</p> <p>4.4.1. Introdução</p> <p>4.4.2. Tipos</p> <p>4.4.3. Identificação</p> <p>4.4.4. Declaração</p> <p>4.4.4.1. Variáveis locais</p> <p>4.4.4.2. Variáveis globais</p> <p>4.4.5. Atribuição</p> <p>4.4.6. Operadores</p> <p>4.4.6.1. Introdução</p> <p>4.4.6.2. Tipos</p> <p>4.4.6.2.1. Aritméticos</p> <p>4.4.6.2.2. Relacionais</p> <p>4.4.6.2.3. Lógicos</p> <p>4.4.6.3. Procedência de operadores</p>	<p>- Caracterizar os tipos primitivos</p> <p>- Explicar o conceito de variável e utilizá-las.</p> <p>- Conceituar e utilizar constantes em algoritmos</p> <p>- Explicar e utilizar expressões matemáticas</p> <p>- Explicar e utilizar expressões lógicas</p> <p>- Conceituar atribuição e utilizá-la em algoritmos</p> <p>- Realizar entradas e saídas de Dados</p> <p>- Criar blocos de instruções</p>	<p>- O professor deverá introduzir a unidade através de exemplos do dia-a-dia, simples de compreender, em que os alunos poderão reconhecer a utilização do raciocínio lógico formal, passando, em seguida, à sistematização através da álgebra, construindo expressões e aplicando os diferentes operadores existentes.</p> <p>- O professor deverá introduzir o conceito de algoritmo utilizando problemas de âmbito geral, decompondo uma ação complexa em</p>	<p>Deverá ser dada ênfase especial à avaliação do desempenho e da manipulação dos programas, assim como a toda a documentação que acompanha os programas, isto é, relatórios de análise e programação e manuais de utilização. Sugere-se o uso sistemático de grelhas de observação que permitirão avaliar a destreza do aluno com as novas aplicações, o equipamento, a sua capacidade de organização e concentração, a qualidade do trabalho realizado e o relacionamento do aluno em grupo de trabalho. A avaliação será, portanto, contínua, registrando a evolução do aluno aula a aula, permitindo ultrapassar de imediato qualquer dificuldade.</p>
---	---	--	--	--

	<p>4.4.7. Estruturas de Controlo / Tomada de decisão</p> <p>4.4.7.1. Introdução</p> <p>4.4.7.2. Valores lógicos</p> <p>4.4.7.3. <i>If- Else</i></p> <p>4.4.7.4. Bloco de instruções</p> <p>4.4.8. Estruturas de Seleção</p> <p>4.4.8.1. Simples</p> <p>4.4.8.2. Composta</p> <p>4.4.8.3. Encadeada</p> <p>4.4.8.4. Múltipla escolha</p>	<p>- Caracterizar as estruturas algorítmicas de seleção</p> <p>- Construir algoritmos utilizando a estrutura de seleção apropriada para a resolução de um determinado problema</p> <p>- Caracterizar as estruturas algorítmicas de repetição</p> <p>- Construir estruturas de repetição combinadas com estruturas de condição</p> <p>- Desenvolver algoritmos utilizando estruturas de repetição apropriadas para a resolução de um determinado problema</p>	<p>outras mais simples, como sejam, por exemplo, programar uma viagem de estudo, mudar um pneu num automóvel ou uma lâmpada num candeeiro, etc.</p> <p>O aluno deverá inserir tipos de dados e operações na construção de algoritmos, utilizando exemplos simples, como seja efetuar um cálculo ou analisar o conteúdo de uma frase. O professor deverá realizar, no quadro, algoritmos simples que clarifiquem a</p>	<p>No entanto, existirão momentos em que será necessário proceder, no final de cada unidade, a provas práticas individuais que permitam avaliar os conhecimentos e as competências adquiridos ao longo do processo de ensino/aprendizagem. Por outro lado, a consolidação das aprendizagens e das competências também poderá ser feita através da avaliação do desempenho e de Trabalho de Projeto ou do desenvolvimento de trabalhos de grupo e ou individuais.</p>
--	---	--	---	--

		<ul style="list-style-type: none">- Caracterizar as estruturas de decisão- Construir algoritmos utilizando a estrutura de decisão apropriada para a resolução de um determinado problema	<p>necessidade e potencialidade das estruturas de decisão. O professor deve insistir na diversidade de exemplos, acompanhando de perto a evolução dos alunos. O grau de dificuldade deverá crescer progressivamente.</p> <ul style="list-style-type: none">- O professor deve deixar espaço para que os alunos debatam entre si os problemas e as suas diferentes vias de resolução.- O professor não deverá descurar a estruturação e a boa	
--	--	---	---	--

VERSÃO EXPERIMENTAL

			<p>- O professor poderá ter os algoritmos implementados no computador em uma linguagem, a fim de os poder executar e mostrar os resultados através do data-show</p>	
Módulo 3 -	<p>4.4.9. Boas práticas de Organização estrutural de um Programa</p> <p>4.4.9.1. Indentação</p> <p>4.4.9.2. Comentar o código</p> <p>4.4.9.3. Atribuir o nome de variáveis de forma intuitiva</p> <p>4.4.9.4. Atribuir o nome de funções de forma intuitiva</p>	<p>- Caracterizar as estruturas de repetição</p> <p>- Construir estruturas de repetição combinadas com estruturas de condição</p> <p>- Desenvolver algoritmos utilizando estruturas de repetição apropriadas para a resolução de um determinado problema</p>	<p>- A demonstração da necessidade do uso de vetores pode ser conseguida através de exemplos como o cálculo de médias, de máximos e mínimos, etc., dentro de um conjunto de valores, que poderão ser fornecidos um a um, ou através de uma cadeia</p>	

	<p>4.4.9.5. Padronização na atribuição do nome das constantes</p> <p>4.4.10. Estruturas de Repetição (Ciclos)</p> <p>4.4.10.1. Do</p> <p>4.4.10.2. While</p> <p>4.4.10.3. Do...While</p> <p>4.4.10.4. Operadores de incrementação e decrementação</p> <p>4.4.10.5. Ciclos infinitos</p> <p>4.4.11. Funções e procedimentos</p> <p>4.4.11.1. Introdução</p> <p>4.4.11.2. Características de uma função</p> <p>4.4.11.3. Atribuição do nome a uma função</p> <p>4.4.11.4. Funcionamento</p>	<p>- Explicar o conceito de funções e procedimentos</p> <p>- Definir retorno de valores</p> <p>- Definir escopo de variáveis</p> <p>- Definir e caracterizar os tipos de passagem de parâmetros por valor e por referência</p>	<p>de caracteres, que deverá ser validada.</p> <p>- O professor poderá demonstrar que, para armazenar dados mais complexos, existem outros tipos de estruturas de dados, como os registos. Deverá ainda utilizar exercícios que permitam distinguir entre a armazenagem de dados em memória e em disco, introduzindo a noção de ficheiro. Os conceitos de registo e ficheiro deverão ser abordados nesta unidade de uma forma simples, sendo</p>	
--	---	--	--	--

	4.4.11.5. Parâmetros 4.4.11.6. Retorno de valores através de uma função		posteriormente mais aprofundados nos anos seguintes.	
--	---	--	--	--

VERSÃO EXPERIMENTAL

4. BIBLIOGRAFIA CONSULTADA

5. RECURSOS EDUCATIVOS RECOMENDADO

No culminar da apresentação do conteúdo programático desta disciplina, sugere-se que sejam adotadas atividades e recursos tecnológicos que fomentem a participação, a colaboração, comunicação e desenvolvimento de soluções criativas e interativas.

No quadro que se segue, são sugeridas algumas aplicações que podem servir par ajudar numa melhor facilitação e dinamização do processo ensino-aprendizagem.

Aplicações /Plataformas	Link
Flowgorithm	http://www.flowgorithm.org/download/index.html
Portugol Studio	http://lite.acad.univali.br/portugol/
Pictoblox	https://thetempedia.com/product/pictoblox/
Scratch	https://scratch.mit.edu/



Cântico da Liberdade

Canta, irmão
Canta, meu irmão
Que a liberdade é hino
E o homem a certeza.

Com dignidade, enterra a semente
No pó da ilha nua;
No despenhadeiro da vida
A esperança é do tamanho do mar
Que nos abraça,
Sentinela de mares e ventos
Perseverante
Entre estrelas e o Atlântico
Entoa o cântico da liberdade.

Canta, irmão
Canta, meu irmão
Que a liberdade é hino
E o homem a certeza!