

INICIAÇÃO A PROGRAMAÇÃO



SÉRIE
EMPREGABILIDADE



A iniciativa da criação deste módulo complementar surgiu da necessidade do mercado de trabalho de preparar pessoas para a área da programação. A formação de programadores é uma grande demanda da nossa sociedade, mas os candidatos enfrentam dificuldades em conceitos básicos, principalmente no que diz respeito à lógica de programação.

Buscando auxiliar na preparação inicial dos futuros programadores, a Pensamento Digital desenvolveu este módulo para auxiliar na aprendizagem dos conceitos básicos necessários para o trabalho com programação, buscando propiciar momentos de criação, criatividade e autoria, utilizando diferentes *softwares* e recursos disponíveis na internet.

O *software* utilizado para trabalhar com a lógica de programação é o Scratch, desenvolvido pelo Media Lab do MIT (Instituto de Tecnologia de Massachusetts) nos Estados Unidos a partir da linguagem de programação Squeak. Este *software* é gratuito, tem uma *interface* amigável e permite criações por pessoas com poucos conhecimentos em programação, pois os comandos podem ser selecionados a partir de pequenos blocos e agrupados. Maiores informações e *download* no site: <<http://scratch.mit.edu/>>.

O objetivo deste material é auxiliar na organização e preparação deste módulo. As idéias são norteadoras, sendo que o planejamento das aulas deve ser desenvolvido para o público a ser atendido, ou seja, não será apresentado aqui um planejamento de aulas, apenas dicas e objetivos para o curso.

■ FORMAÇÃO INICIAL

É importante lembrar que este módulo oferece apenas uma formação inicial, sendo que, após este curso, se os aprendizes tiverem interesse profissional pela área da programação, deverão buscar cursos avançados e profissionalizantes.

■ OBJETIVO GERAL

O curso tem por objetivo preparar o aprendiz para criar páginas e objetos animados, utilizando linguagem HTML e conceitos de programação, de lógica e de matemática, levando em conta o desenvolvimento do raciocínio lógico, aprimorando a criatividade e autonomia pela busca de informações.

■ OBJETIVOS ESPECÍFICOS

■ Oferecer uma preparação básica para que os aprendizes possam desenvolver produções em HTML e Scratch.

■ Apresentar *softwares* e recursos disponíveis e atuais para o trabalho com programação HTML.

■ Mostrar o Scratch como um recurso interativo, lúdico e fácil para o desenvolvimento de objetos animados, histórias, jogos e outras animações.

■ Possibilitar atividades práticas para o desenvolvimento de sites e histórias animadas, buscando utilizar sites e ambientes para a disponibilização dessas produções, permitindo interações e trocas de experiências com outros estudantes e profissionais de programação.

■ PÚBLICO-ALVO

Este módulo foi desenvolvido para oferecer uma preparação inicial às pessoas que desejam se aprofundar no conhecimento da área de programação. Assim, a sugestão é oferecer este curso para jovens acima de 14 anos e para adultos que buscam novas possibilidades profissionais.

■ PRÉ-REQUISITOS

Ter noções básicas de informática (sistemas operacionais, organização de pastas e diretórios de arquivos, editores de texto e internet – pesquisa, navegação, e-mail).

Para melhor aproveitamento do curso, sugere-se que os aprendizes estejam pelo menos finalizando o Ensino Fundamental. Por questões de limitação de vagas e pelas exigências do mercado de trabalho, o grau de escolaridade solicitado pode ser maior.

■ ESTRUTURA E FORMATO

O curso é presencial, sendo que para a certificação é necessário atingir os objetivos e ter frequência de 75%.

A carga horária é de cem (100) horas. A instituição deve organizar as aulas da forma mais adequada a sua realidade, contudo apresentamos duas sugestões de distribuição da carga horária:

■ Duas aulas por semana com carga horária de 3 h/aula, tendo 32 aulas e mais 4h de atividades complementares: passeios, palestras, vídeos, atividades à distância.

■ Duas aulas por semana com carga horária de 4 h/aula, tendo 25 aulas. Durante essa carga horária, atividades complementares podem ser planejadas.

CONTEÚDO DO MÓDULO

O conteúdo está sendo apresentado como uma lista de conceitos e assuntos a serem estudados. Isso não determina uma ordem rígida de abordagem e não impede que outros assuntos, além desses, sejam trabalhados. Esses conteúdos são apenas uma sugestão, pois eles foram trabalhados na turma piloto desenvolvida pela Pensamento Digital e acreditamos que são adequados para a preparação inicial em programação.

No caso da experiência com a turma piloto, notou-se um grande interesse dos cursistas pelo Scratch. Por este motivo, foi feita a opção de trabalhar inicialmente com Scratch e suas possibilidades e depois com o HTML.

■ HTML (PÁGINAS, SITES)

Informações básicas

- Padrão W3C
- Tags HTML
- Conhecimento do *software* a ser usado (Aptana, Nvu, Mozilla Composer, etc)
- Código básico

Função de cada Tag

- Formatação de texto e títulos
- Formatação de parágrafos
- Linhas horizontais

Código RGB

- Significado
- Determinação e código das cores
- Forma de utilização no HTML (cores de fundo e de fonte)

Imagens

- Tags de inserção e formatação de imagens
- Imagens de fundo

Listas

Tags para criação de listas

- Listas ordenadas
- Listas não ordenadas
- Listas ordenadas com caracteres especiais

Links

- Tags para criação e formatação de links
- Links para arquivos
- Links para sites externos

Tabelas

- Tags para linha e coluna
- Formatação de células
- Inserção de imagens

■ SCRATCH

- Compreensão sobre a criação de movimento
- Eixo cartesiano
- Ângulo e direção
- Como utilizar os comandos

Lógica

- Operadores matemáticos
- Condições (if, else) e outras
- Controles
- Como utilizar os comandos

Variáveis

- O que são e como funcionam
- Quais estão pré-definidas
- Como criar variáveis

■ PROJETO

- Criar página para internet (site), utilizando recursos HTML e usando criações do Scratch

■ PUBLICAÇÃO

- Utilização de *upload*
- Pesquisa por serviços de hospedagem de site

■ INFORMAÇÕES ADICIONAIS

A Pensamento Digital desenvolveu materiais a partir da experiência da turma piloto do módulo de Iniciação a Programação. São materiais que fornecem informações para os educadores que ministraram as aulas do curso. Os temas abordados são:

- Páginas web e código HTML
- Animações
- Desafios com Scratch, para incentivar e ajudar nas primeiras criações usando este *software*

Todos estes materiais estão disponíveis no endereço: <http://oficinas.pensamentodigital.org.br> ou acessando o site <http://www.pensamentodigital.org.br>

Para conhecer melhor o Scratch, visite o site [<http://scratch.mit.edu/>](http://scratch.mit.edu/) que congrega a comunidade de usuários de diversas partes do mundo. Ali são compartilhados projetos e tutoriais. Também é possível interagir em fóruns temáticos.

Uma sugestão para as aulas do curso é a utilização de vídeos que apresentem informações sobre mercado de trabalho, exigências da formação profissional, entrevista de emprego e outros assuntos relacionados. Muitos desses vídeos podem ser encontrados no YouTube (<http://br.youtube.com/>).

LINGUAGEM BÁSICA

Antes de começar a trabalhar HTML, é necessário que se conheça os termos básicos que envolvem esta linguagem.

■ **INTERNET** é um conjunto de redes de computadores. Podemos dizer que a internet é como uma cidade eletrônica com bibliotecas virtuais, lojas virtuais, escritórios virtuais, galerias de arte virtuais, etc. A internet oferece um grupo de serviços para usuários, como correio eletrônico, a World Wide Web, FTP, Gopher, IRC, grupos de notícias usenet, telnet e outros.

Normalmente quando abrimos uma página na internet usamos o **www**. Esta é a sigla de World Wide Web que significa Rede Mundial (a grande rede de computadores interligados no mundo todo). Web é o diminutivo para World Wide Web.

■ **PROTOCOLO** é um conjunto de regras estabelecidas com o objetivo de permitir a comunicação entre computadores. É um método de acesso a um documento ou serviço através da internet. São os chamados TCP/IP (Transmission Control Protocol ou Internet Protocol). Em português: Protocolo de Controle de Transmissão ou Protocolo Internet. O TCP/IP está disponível para qualquer tipo de CPU e sistema operacional. Existem dois tipos principais:

■ ■ **Protocolo HTTP** (HyperText Transfer Protocol ou, em português, Protocolo de Transferência de Hipertexto) é um protocolo da internet utilizado pelos computadores ligados à web para comunicar-se entre si. Ele tem como função ativar os navegadores da web para recuperarem informações de servidores da web.

■ ■ **Protocolo FTP** (File Transfer Protocol ou, em português, Protocolo de Transferência de Arquivo) é um protocolo que possibilita a transferência de arquivos de um local para outro pela Internet.

■ **HIPERTEXTO** é qualquer informação de texto em um computador que contenha saltos para outras informações, o que usualmente chamamos de *link*. Os documentos visualizados através dos *browsers* são escritos em hipertextos, utilizando-se uma linguagem especial chamada HTML (HyperText Markup Language).

■ **HOME PAGE** é um conjunto de páginas, documentos disponíveis na web, interligados entre si (através de *links*).

■ **SITE** é uma palavra em inglês que significa local, lugar. Na internet, designa um conjunto de páginas que representa uma pessoa, instituição ou empresa na rede.

■ **BROWSERS** são programas que lêem e interpretam arquivos HTML enviados pela web, também formata-os em páginas da web e os exibe ao usuário. Navegadores da web podem executar som ou arquivos de vídeo incorporados em documentos da web, se o usuário dispuser do *hardware* necessário. Existem vários tipos de *browsers*, os mais usados são o Internet Explorer, o Firefox e o Opera, entre muitos outros.

■ **URL** (Uniform Resource Locator ou, em português, Localizador de Recursos Uniforme) é uma seqüência de caracteres que fornece o endereço de um site da web ou um recurso da World Wide Web, juntamente com o protocolo (como FTP ou HTTP) através do qual o site ou o recurso é acessado.

■ **ENDEREÇO** é o caminho até um objeto, documento, arquivo, página ou outro destino.

Para entender o URL e o endereço colocamos a explicação a partir do site <http://www.pensamentodigital.org.br/>.

■ ■ **http://** = Protocolo, utilizado pelos computadores ligados à web para comunicar-se entre si.

■ ■ **www.pensamentodigital.org.br** = Nome do domínio. Domínio são as categorias de endereços da internet que representam países ou tipos de organização.

■ ■ **www** = Sigla de World Wide Web que significa rede mundial.

■ ■ **pensamentodigital** = Nome específico, que pode conter uma ou mais palavras, separadas ou não por hífens (ex.: pensamento-digital).

■ ■ **org** = Tipo de domínio, que indica a natureza do site. No caso, como trata-se de um site de uma fundação usa-se *.org*, organizações sem fins lucrativos. Outros exemplos: *.gov* (governo) e *.com* (endereços comerciais).

■ ■ **br** = Sigla do país, composta de duas letras, significa que a página está situada em um computador no Brasil. Páginas que não possuem terminação indicando o país de origem estão situadas nos Estados Unidos. Outros exemplos: *.pt* (Portugal) e *.jp* (Japão).

INTRODUÇÃO AO HTML

HTML significa HyperText Markup Language ou Linguagem de Formatação de Hipertexto. É a linguagem usada para criar páginas na web, que estabelece como um determinado elemento deve ser visualizado. A linguagem de formatação de exibição de textos funciona através de comandos conhecidos como tags.

O documento HTML pode ser escrito em qualquer editor de texto, como texto puro, sem formatação ou caracteres de controle. Pode-se usar, portanto, o Notepad do Windows ou o Kedit do Linux. Hoje existem vários programas que são próprios para a programação em HTML, como Mozilla Composer, Front Page, Aptana, Notepad++, entre outros.

■ TAGS

Tags são os comandos que são sempre em inglês e ficam entre <>. O padrão correto é escrever os comandos com letras minúsculas.

Podemos dizer que existem 3 tipos de tags:

- Tag de abertura de comando: usada quando começamos um comando. Exemplo: `<html>`
- Tag de fechamento de comando: usada quando finalizamos o comando. Exemplo: `</html>`
- Tag solteira: estes são comandos específicos. Exemplo: `
`

Muitas tags possuem o que chamamos de atributos. Os atributos são complementos dos comandos, seriam funções a mais que os comandos possuem.

■ ESTRUTURA BÁSICA

A estrutura básica de um documento HTML é a seguinte:

- `<html>` => Início do documento
- `<head>` => Início do cabeçalho do documento
- `<title>` => Início do título do documento
- Aqui entra o título do documento
- `</title>` => Fim do título do documento
- `</head>` => Fim do cabeçalho do documento
- `<body>` => Início do conteúdo do documento
- Aqui entra todo o conteúdo que será exposto pelo *browser*
- `</body>` => Fim do conteúdo do documento
- `</html>` => Fim do documento

Se observarmos a estrutura acima, podemos dizer que a página divide-se entre cabeçalho, que fica entre as tags `<head>` e `</head>` e corpo, que fica entre `<body>` e `</body>`. Tudo que vai aparecer na página deve estar em corpo.

■ REGRAS DO HTML

Quando se fala em regras no HTML, os iniciantes acreditam que o tema abordado é a seqüência lógica. Claro que isto também, mas além desta seqüência, as regras tratam das estruturas das páginas.

Para a fiscalização do sistema foi fundada uma instituição norteadora, a W3C, que é um consórcio de empresas que criaram o padrão HTML. No site <http://www.w3.org> é possível verificar se um site está dentro do padrão. Este site está em inglês, mas existe uma versão em português em <http://www.w3c.br/>

■ PARTES BÁSICAS DO HTML

■ CABEÇALHO DA PÁGINA (`<head>...</head>`)

Contém o cabeçalho do documento, ou seja as informações que não fazem parte do corpo da página e, portanto, não serão exibidas diretamente no site, mas, por exemplo, será o título exibido na barra de títulos do navegador.

Exemplo: `<head>...informações importantes sobre o documento...</head>`

■ TÍTULO (`<title>...</title>`)

Contém o título do documento. Exibe o título como nome da janela em que a página é visualizada. O título é obrigatório. Não confunda o título da página com o nome do arquivo gravado em disco.

Exemplo: `<title>Escola de Fábrica</title>`

■ SCRIPT (`<script>...</script>`)

É utilizado para inserção de código script, como Java Script, VB Script, etc, dentro do documento HTML.

Exemplo: `<script>...código em linguagem script conhecida pelo browser...</script>`

■ META (`<meta>`)

Define valores especiais. Os valores são definidos como pares name/value, ou seja nome/valor.

Atributos de Meta

Name: Especifica um nome ao qual um determinado valor será associado. O navegador precisa entender esse nome para que a tag tenha utilidade.

Exemplo: `<meta name="keywords" content="INFORMÁTICA, EDUCAÇÃO, APRENDIZADO">`

Content: Especifica o valor associado a um Name.

Exemplo: `<meta name="keywords" content="INFORMÁTICA, EDUCAÇÃO, APRENDIZADO">`

PARTES BÁSICAS DO HTML

É obrigatória a presença do atributo *Name* ou do atributo *HTTP-EQUIV*. `<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">`

O tag meta é utilizado, também, para especificar palavras-chaves que serão catalogadas por Ferramentas de Busca.

■ ■ CORPO DO TEXTO (`<body>...</body>`)

Este é o corpo da página, a própria página. Todo o conteúdo do site estará entre `<body>` e `</body>`.

Atributos de Body

Bgcolor: Define a cor de fundo da página

Exemplo: `<body bgcolor="white">`

Background: Define uma imagem que será utilizada como fundo da página. Caso a imagem tenha um tamanho inferior ao da página, ela será repetida diversas vezes de forma a cobrir o fundo do documento

Exemplo: `<body background="images/fundo.png">...diversas tags, textos etc...</body>`

■ ■ TÍTULOS E SUBTÍTULOS (`<hX>...</hX>`)

Indicam um título ou subtítulo que chamamos de *header* (cabeçalho). Os cabeçalhos têm seis níveis de importância, sendo o "1" o mais importante e o "6", o menos. Os níveis de cabeçalho são equivalentes aos tamanhos da fonte – h1 = 24 pt / h2 = 18 pt / h3 = 14 pt / h4 = 12 pt / h5 = 10 pt / h6 = 8 pt.

Visualização:

`<h1>Título 1</h1>`

`<h2>Título 2</h2>`

`<h3> Título 3</h3>`

`<h4> Título 4</h4>`

`<h5>Título 5</h5>`

`<h6> Título 6</h6>`

Atributos de hX

Align: Indica o alinhado a esquerda (padrão), centralizado ou à direita, usando-se as palavras *left*, *center* e *right*.

Exemplo e visualização:

`<h5 align="left">Pensamento Digital</h5>`
Pensamento Digital

`<h5 align="center">Pensamento Digital</h5>`
Pensamento Digital

`<h5 align="right">Pensamento Digital</h5>`
Pensamento Digital

■ ■ NEGRITO, ITÁLICO E SUBLINHADO

■ ■ NEGRITO (Bold `...`) OU STRONG (`...`)

Indica que o texto deve ser apresentado em negrito.

Exemplo: `Pensamento Digital`

Visualização: **Pensamento Digital**

■ ■ ITÁLICO (`<i>...</i>`) OU EM (`...`)

Indica que o texto deve ser apresentado em itálico.

Exemplo: `<i>Pensamento Digital</i>`

Visualização: *Pensamento Digital*

■ ■ SUBLINHADO (`<u>...</u>`)

Indica que o texto deve ser apresentado sublinhado.

Exemplo: `<u>Pensamento Digital</u>`

Visualização: Pensamento Digital

■ ■ PARÁGRAFOS (`<p>...</p>`)

Indica que o texto que esta entre as tags é um parágrafo e possui um alinhamento próprio. Os navegadores deixam uma quebra de linha antes de cada parágrafo.

Exemplo: `<p>O projeto Escola de Fábrica oferece a oportunidade de formação profissional inicial e continuada a jovens de baixa renda.</p>`

Atributos de P

Align: Indica o alinhamento do parágrafo a esquerda (padrão), centralizado, à direita ou justificado, usando-se respectivamente as palavras *left*, *center*, *right* e *justify*.

Exemplo: `<p align="center">Este é o site da Pensamento Digital.</p>`

Visualização:

Este é o site da Pensamento Digital.

■ ■ QUEBRA DE LINHA (`
`)

Passa para a próxima linha

Exemplo: `<p>A Pensamento Digital trabalha com inclusão digital.
A Pensamento Digital atua no Rio Grande do Sul e em São Paulo.</p>`

Visualização:

A Pensamento Digital trabalha com inclusão digital.

A Pensamento Digital atua no Rio Grande do Sul e em São Paulo.

PARTES BÁSICAS DO HTML

■ ■ FONTES (...)

Existem 7 tamanhos de fonte em HTML, numeradas de 1 a 7. O tamanho padrão é 3 que é equivalente a 12 pt.

Exemplo: Texto

Atributos de Fonte

Size: Indica qual o tamanho de fonte que deve ser usado dentro de sua área de influência.

Exemplo e visualização: Pensamento Digital

Pensamento Digital

Color: Indica qual a cor da fonte dentro de sua área de influência.

Exemplo e visualização:

Pensamento Digital

Pensamento Digital

Pensamento Digital

Face: Determina a fonte (tipo de letra) a ser utilizada. Deve ter um nome entre aspas ou uma lista de nomes de fontes separados por vírgula e entre aspas. O navegador procurará estas fontes em ordem.

Exemplo e visualização:

Pensamento Digital

Pensamento Digital

■ ■ LINK (<a>...)

Indica a região a ser tratada como hyperlink

Exemplo: Fundação Pensamento Digital

Visualização: Fundação Pensamento Digital

Atributos de A

HREF: Define que sua área de influência é um *link*.

Exemplo e visualização: <p> Clique aqui para acessar a página da Fundação Pensamento Digital.</p>

Clique aqui para acessar a página da Fundação Pensamento Digital.

Veja algumas formas de especificar um *link*:

Para outras páginas de um mesmo site

O local precisa ter um nome. Este nome é definido da seguinte forma: Link

Para outros sites

Deve-se usar a URL completa da página destino.

Exemplo:

Link para site externo

Visualização:

Link para site externo

■ ■ LINK ESPECIAL: "MAILTO"

Existe um tipo de *link* que chamamos de "mailto:". Se o protocolo utilizado for "mailto:", ao invés de "http://", o *link* abrirá uma janela especial para que se possa enviar um e-mail (correio eletrônico) para o endereço específico.

Exemplo:

Mande um e-mail para a Pensamento Digital.

Visualização:

Mande um e-mail para a Pensamento Digital.

■ ■ IMAGENS ()

Esta tag insere uma imagem na página.

Atributos de IMG

SRC: Indica a URL da imagem a ser exibida. Podem ser usados dois tipos:

■ URL absoluto (http://www.pensamentodigital.org.br/images/log.png)

■ URL relativo (/images/logo.png). Esta tag é obrigatória.

Exemplo:

ALT: Indica um texto associado à imagem. Quando a imagem não for exibida, o texto será exibido em seu lugar. O texto também será exibido quando o cursor ficar parado sobre a imagem.

Exemplo:

Align: Determina o alinhamento da imagem em relação ao texto existente na mesma linha. Os valores válidos são "TOP", "MIDDLE", "BOTTOM", "LEFT" e "RIGHT".

Exemplo:

Width: Determina a largura, em pixels, da imagem.

Exemplo:

Height: Determina a altura da borda da imagem.

Exemplo:

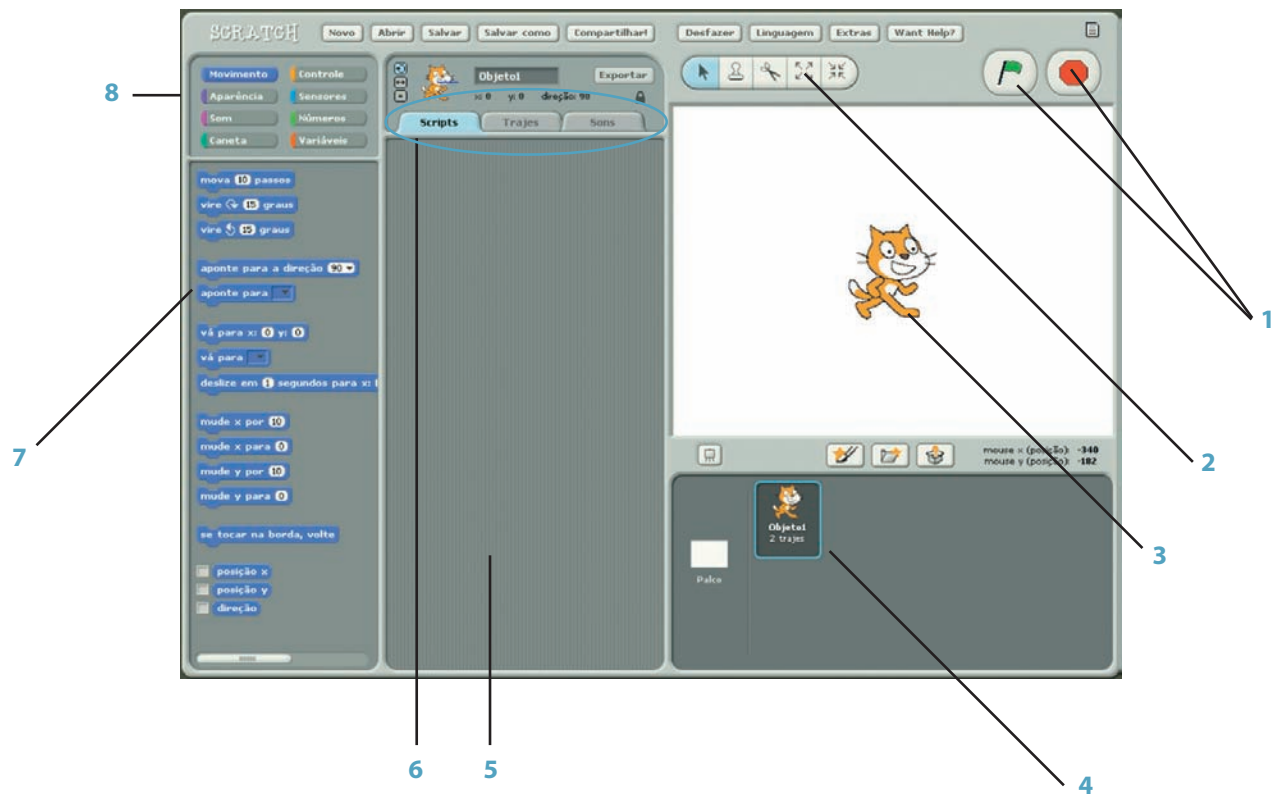
Border: Determina a largura da borda da imagem.

Exemplo:

APRESENTAÇÃO DO SCRATCH

Scratch é uma nova linguagem de programação que permite a criação de histórias, animações, jogos e outras produções. Tudo pode ser feito a partir de comandos prontos que devem ser agrupados.

Para fazer *download* do Scratch, entre no site <http://scratch.mit.edu/download> e após preencher um formulário será possível escolher a versão para *download*. Ele é gratuito. Abaixo veja a tela principal do Scratch traduzida para o português:



- 1 – Botões de iniciar e parar o script
- 2 – Botões para editar o objeto selecionado no palco
- 3 – Palco onde os objetos são colocados e onde é possível ver o resultado da programação criada. O objeto inicial que aparece no palco é o gato
- 4 – Área dos objetos usados na animação. O objeto em edição fica selecionado
- 5 – Área de edição e conexão de scripts
- 6 – Abas com opções para a área de script, para traje e para sons
- 7 – Blocos de comandos
- 8 – Categorias de comandos

MOVIMENTO

Veja agora como fazer um objeto se movimentar. Clique na categoria *Movimento* dos blocos de scripts.



Selecione o bloco *Mova* e arraste para a área de edição de scripts.



O clique duplo sobre o bloco faz o movimento acontecer.

ENTENDENDO O SCRATCH

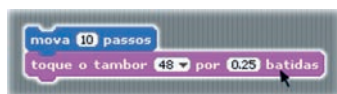
SOM

Para colocar som no script, você pode usar o comando *Toque o tambor*. Ele fica disponível na categoria som. Você pode usar este bloco de comando sozinho ou agrupado com outros comandos.

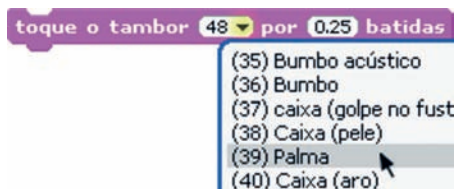


Clique e arraste o bloco para a área de edição de scripts. Se for o caso, encaixe este bloco com os já existentes no script.

Para ver o funcionamento, dê um duplo clique sobre o grupo de blocos.



Para escolher o som desejado, clique na seta destacada e escolha entre as opções do menu.



Se você desejar importar um arquivo de música (MP3 ou WAV) do seu computador ou desejar gravar um som, clique na aba *Som* e escolha entre *Gravar* e *Importar*.



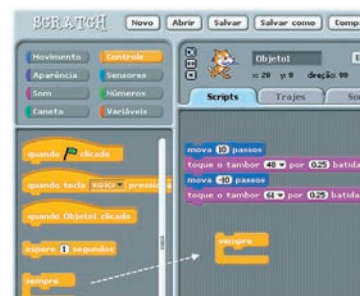
Para usar o som escolhido na sua programação, escolha o bloco *Toque o som* e encaixe no seu script.



Lembre-se: se o som não funcionar, verifique se este recurso funciona no seu computador (se o som está ligado e se existem caixas de som funcionando).

USO DO CONTROLE SEMPRE

É possível programar no Scratch que uma ou mais ações continuem acontecendo e se repetindo por tempo indeterminado. Para isso se usa o comando *Sempre*, disponível na categoria *Controle*.



Clique e arraste o bloco *Sempre* para a área de edição de scripts. Encaixe os comandos dentro do bloco *Sempre*.



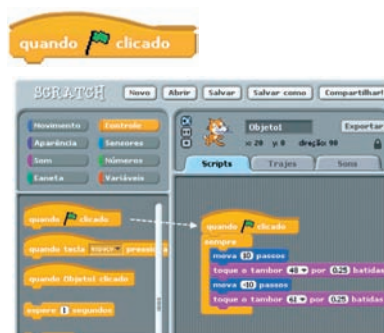
Para arrastar um conjunto de blocos, clique sobre o primeiro bloco (no topo do conjunto) e arraste tudo.

Para parar a programação, após usar o comando *Sempre*, clique no botão vermelho que significa *Parar Tudo*.



BANDEIRA VERDE – INICIAR O SCRIPT

O Scratch também possui controles para o início da execução dos scripts. Um exemplo é a bandeira verde que fica sobre a tela de visualização das programações. Ela pode ser usada para iniciar o funcionamento de um script. Para isso é necessário que seja colocado no script o bloco de controle que indica:



Clique no bloco e arraste para a área de edição de scripts. Encaixe o bloco sobre o conjunto já existente, se for o caso. Este controle deve ser o primeiro em um grupo de blocos, pois é ele que determina o início desta execução.



Para testar, clique sobre a bandeira verde que significa *Iniciar Scripts*.

ENTENDENDO O SCRATCH

■ USANDO TECLAS

Para iniciar um script, além de usar a bandeira verde, é possível determinar uma tecla do teclado que funcione como disparadora do script. Desta forma, quando a tecla for pressionada, o script inicia sua execução.

Para determinar que o início da execução será determinado por uma tecla, você precisa colocá-la no início de um script o controle.



Arraste o bloco para a área de edição de script e encaixe no início de um conjunto de blocos. Aperte a tecla determinada para fazer o teste.

Para determinar qual tecla será usada para iniciar o script, clique na seta destacada e escolha a opção desejada.



Você pode usar um controle inicial de script diferente para cada conjunto de blocos. É assim que se faz para determinar movimentos diferentes de um objeto de acordo com o clique nas setas de direção do teclado.

■ OBJETO NOVO

Quando o Scratch é aberto, no palco já está aparecendo o gato. Mas nem sempre se deseja usá-lo e então é possível inserir ou criar um novo objeto. Da mesma forma, é possível ter vários objetos em uma programação.

Veja abaixo como aparece um novo objeto no palco:



Para criar ou inserir um novo objeto você deve clicar em uma das seguintes opções:



Desenhar um objeto (abre um editor que permite pintar e desenhar um objeto).



Inserir objeto do arquivo (permite inserir um arquivo de imagem do computador).



Inserir objeto surpresa (clcando neste botão, surge um objeto surpresa no palco, ou seja, a pessoa não determina o objeto que surgirá).

■ FALAR ALGO

No Scratch é possível fazer um objeto falar. Para isto, basta usar o bloco de comando *Diga*. Nele você pode determinar o que será dito e o tempo que essa mensagem ficará aparecendo.



Coloque esse bloco no script do objeto que deverá falar.



■ FAZER ANIMAÇÃO

Para fazer uma animação no Scratch é bastante simples. O efeito é o mesmo de uma imagem gif, onde aparecem diferentes posições de um personagem e a troca das imagens das posições produz a idéia de animação.

Escolha o objeto que será animado e clique em *Trajes*. Você pode criar as diferentes posições do objeto desenhando o novo a partir do inicial (fazer uma cópia do original e editar) ou importar as posições.



Depois faça o script do objeto que será animado. Use o bloco *Sempre* e dentro dele o bloco *próximo traje*. Este bloco faz o objeto alternar entre seus trajes já criados.



É importante colocar um tempo após a troca de traje para que seja possível visualizar a troca, ou isso acontecerá muito rápido.

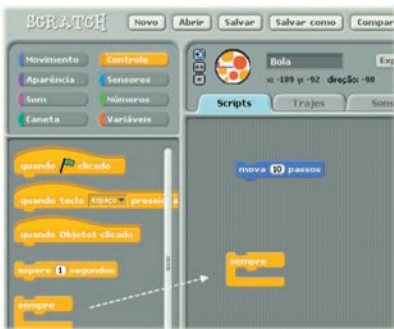
ENTENDENDO O SCRATCH

■ TOCAR NA BORDA E VOLTAR

Quando você faz algumas programações no Scratch, é importante que o objeto ao tocar na borda do palco volte. Um exemplo disso pode ser uma bola que rola, bate na borda e volta. Primeiro, puxe o bloco *Mova* para a área de edição de scripts.



Pegue o bloco *Sempre* e coloque na área de edição de scripts.

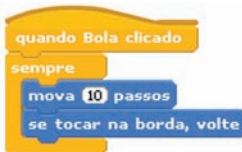


Encaixe o *Mova* dentro do *Sempre*.



Pegue o bloco *se tocar na borda, volte* na categoria *Movimento* e coloque dentro do *Sempre*.

Se você quiser que a bola comece a andar quando for pressionada pelo mouse (clificada), use o controle abaixo:



■ USO DE TESTES: SE (IF)

Para muitas programações, jogos e histórias é importante usar testes. Podemos fazer uma bola bater em um objeto e quando ela bater, voltar. Mas como ela vai saber que bateu? Como determinar o que acontece quando ela bate. Veja:



Você também pode determinar que o script inicie quando a bandeira verde for pressionada.

A bola cai, bate na cabeça do gato e volta para cima. Quando bate na borda superior ela volta e bate novamente no gato. Inicialmente mude a direção do objeto "bola" para 180 (para ela ir para baixo).



O script da bola é iniciado quando a bola é clicada com o mouse. Ela sempre irá se mover e, se tocar na borda (qualquer borda do palco), ela volta na direção contrária. Puxe o teste *se* na categoria *Controle* e coloque na área de edição de scripts.



Dentro do *se* coloque o sensor *tocando em* que fica na categoria *Sensores*.

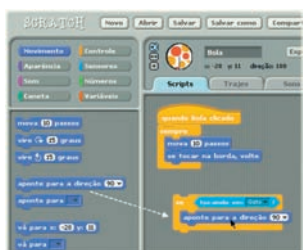


Escolha no menu o nome do objeto que será tocado (no caso, o gato).

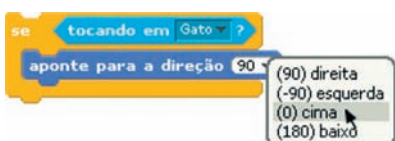


ENTENDENDO O SCRATCH

Dentro do bloco *se*, coloque o que acontece quando a bola tocar o gato, ou seja, acrescente o bloco *aponte para a direção*. Isto significa que quando a bola tocar no gato, ela irá mudar sua direção para aquela determinada no script.



Escolha a direção (0) cima pois após bater no gato a bola deve subir.



Acrescente o bloco *Sempre* para que o teste seja feito o tempo todo, e coloque o teste dentro do *Sempre*.



Acrescente o controle para o início da execução do script. Neste caso foi usado o *quando bandeira for clicada*.

Veja que foram feitos dois scripts separados: um para o movimento da bola e outro para o teste. Também é possível fazer tudo junto, usando apenas um controle de início do script e apenas um bloco *Sempre*. Experimente modificar este script e gerar novas versões.

■ USO DE TESTES: SE, SENÃO (IF, ELSE)

Agora vamos usar o teste completo: *se, senão*. O desafio é fazer uma bola ir na direção do gol e, se bater nele, dizer "Gol!" Inicialmente faça o desenho da goleira e escolha ou desenhe a bola para ficar mais ou menos assim:



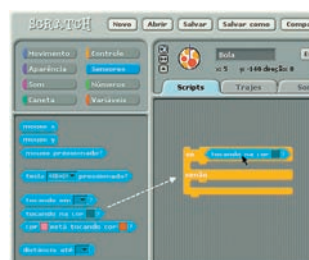
Inicialmente, mude a direção da bola para 0, para ela ir na direção da goleira.



Acrescente na área de edição de scripts o bloco *se, senão* que fica na categoria *Controle*.



Coloque no espaço do *se* o sensor *tocando na cor*.



Clique no quadrado da cor e escolha a cor da goleira.



Agora a cor no teste está correta.



Você pode colocar dentro do *senão* o comando *Mova* para que a bola ande caso não toque no gol. Mas se deixar o *senão* vazio, apenas não acontecerá nada quando a bola não tocar no gol.



Dentro do bloco *se* coloque a ação que deve ocorrer quando a bola tocar o gol, ou seja, coloque o bloco "diga Gol! por 2 segundos"



Você pode editar o texto do bloco *Diga* clicando e apagando o texto original.



ENTENDENDO O SCRATCH

Coloque o teste (*se, senão*) dentro de um bloco *Sempre* para que este teste seja feito o tempo todo. Depois coloque um controle que determine o início do script. No caso foi pedido que a bola ande quando for clicada com o mouse, mas outros controles podem ser usados.



Experimente fazer um jogo em que a bola ande e vire conforme algumas teclas do teclado são pressionadas.

■ COMPARTILHAR

No Scratch, após fazer seu projeto, é possível compartilhá-lo com outros publicando-o no site do Scratch. Para isso, basta clicar no botão *Compartilhar* na parte superior da tela do Scratch.

Mas atenção: para compartilhar seu projeto no site do Scratch você precisa ter feito seu cadastro. Se ainda não fez, entre em <http://scratch.mit.edu> e faça seu cadastro. É gratuito e fácil de fazer.

Compartilhar!

FONTES

■ <http://oficinas.pensamentodigital.org.br> ■ <http://pt.wikipedia.org>

REFERÊNCIAS PARA HTML

LOUREIRO, Gustavo. Curso Superior de Formação Específica em Gestão de Ambientes Internet Webmaster/Webdesigner [manual]. [Capturado em 2008 jun 20]. Disponível em: <http://www.apostilando.com/download.php?cod=2586&categoria=HTML>

PRODUÇÃO: FUNDAÇÃO PENSAMENTO DIGITAL

Av. Ipiranga, 6681, Prédio 94, Sala 13 – Tecnopuc – Bairro Partenon – Porto Alegre – RS – CEP 90619-900

Fone: 51 3433.5151 / 3433.5150 – www.pensamentodigital.org.br

Presidente: **Léa Fagundes** (lea@pensamentodigital.org.br)

Superintendente executiva: **Marta Voelcker** (marta@pensamentodigital.org.br)

O USO DAS TICs PARA PROMOVER A EMPREGABILIDADE

Com o objetivo de desenvolver Módulos Complementares (ao Módulo Tecnologias para a Vida – voltado para a inclusão digital e para o desenvolvimento da identidade) direcionados para a inserção de jovens no mercado de trabalho, a Pensamento Digital inscreveu um projeto no Rede de Parceria Social, uma iniciativa conjunta da Secretaria da Justiça e do Desenvolvimento Social, organizações sociais e empresas, com objetivo de realizar projetos sociais, abrangendo diversas áreas da assistência social. Foram trabalhados, durante 10 meses, quatro áreas (quatro Módulos): Iniciação à Programação, Tecnologias para o Trabalho, Edição de Imagens, Manutenção e Configuração de Computadores. Depois de desenvolvidos, os módulos complementares serão oferecidos a todas as organizações que adotam a proposta pedagógica da Pensamento Digital, através de cursos para os educadores sociais.

RESPONSÁVEIS PELO MÓDULO INICIAÇÃO À PROGRAMAÇÃO

Coordenadora pedagógica: **Susana Seidel** (susana@pensamentodigital.org.br)

Orientador pedagógico: **Cesar Felipe Ferreira** (cesar@pensamentodigital.org.br)

Elaboração do conteúdo: **Cesar Ferreira, Cláudio Gilberto César e Susana Seidel**

