

UNIVERSIDADE FEDERAL DO TRIÂNGULO MINEIRO - UFTM



MESTRADO PROFISSIONAL EM MATEMÁTICA EM REDE NACIONAL - PROFMAT



PROFMAT

DISSERTAÇÃO DE MESTRADO

TÓPICOS ABORDADOS NA DISCIPLINA DE RECURSOS
COMPUTACIONAIS

MICHELLE CRISTINA DE SOUSA BALTAZAR
MELISSA DA SILVA RODRIGUES

UBERABA - MINAS GERAIS

DEZEMBRO DE 2016

TÓPICOS ABORDADOS NA DISCIPLINA DE RECURSOS COMPUTACIONAIS

MICHELLE CRISTINA DE SOUSA BALTAZAR

MELISSA DA SILVA RODRIGUES

Dissertação de Mestrado apresentada à Comissão Acadêmica Institucional do PROFMAT-UFTM como requisito parcial para obtenção do título de Mestre em Matemática.

Orientador: Prof. Dr. Leandro Cruvinel Lemes.

Uberaba - Minas Gerais

Dezembro de 2016

*“A viagem tem sido longa e o custo tem sido
alto... mas nunca uma coisa grande foi
alcançada com facilidade.
Uma longa história, como uma Torre alta,
tem de ser construída pedra por pedra”
— Stephen King, A Torre Negra*

Sumário

| | |
|-----------------------------------|-----------|
| INTRODUÇÃO | 1 |
| 1 HTML | 2 |
| 1.1 CSS | 6 |
| 2 PYTHON | 10 |
| 2.1 MATPLOTLIB | 13 |
| 2.2 MATH | 16 |
| 3 LATEX | 17 |
| 4 OCTAVE | 21 |
| 5 RAPIDMINER | 23 |
| 6 GIMP | 26 |
| 7 INKSCAPE | 29 |
| 8 SCRATCH | 31 |
| REFERÊNCIAS BIBLIOGRÁFICAS | 34 |

INTRODUÇÃO

Iremos abordar as ferramentas estudadas no curso de Recursos Computacionais e será apresentado um pequeno resumo de cada ferramenta, seu funcionamento, léxico e sintaxe e algumas possíveis utilizações para cada uma delas.

A importância dessas ferramentas para o aluno de mestrado é ter nas mãos ferramentas poderosas e profissionais para trabalhar em vários seguimentos da educação. É de suma importância que o mestrando tenha não apenas conhecimento de sua existência mas também alguma experiência.

É exatamente destas experiências e trocas de conhecimento que surgem melhorias, inovações e novas formas de implementação e utilização. Todas as ferramentas são gratuitas e trabalham sob a licença GPL do GNU, permitindo sugestões, correções e melhorias com rapidez, eficiência e segurança.

O estudo atual não pretende abranger a totalidade das ferramentas nem tampouco esgotar os assuntos abordados. Porém temos aqui um norte a ser seguido para embasar um estudo mais aprofundado ou mesmo para ser apresentado a alguma ferramenta mais desconhecida.

Agradecemos desde já pela atenção e desejamos uma leitura agradável e instrutiva!

1 HTML

HTML é uma abreviação para *Hyper Text Markup Language* ou *Linguagem de Marcação de Hipertexto*. É uma linguagem utilizada pra criação de páginas para a internet que serão "interpretadas" no navegador do usuário.

As páginas de internet funcionam graças ao protocolo **HTTP** que, em português, seria *Protocolo de Transferência de Hipertexto*.

É um protocolo de comunicação na camada de aplicação que permite a comunicação de dados na **Wide World Web**.

Se observarmos a *barra de endereços* onde aparece o endereço do site atual, veremos que o texto *http://* ou *https://* precedem o endereço, informando ao navegador que este protocolo é o responsável pelo funcionamento do site digitado, permitindo a correta interpretação do endereço e do conteúdo da página.

Hipertexto é o texto estruturado que utiliza ligações lógicas (*hiperlinks*) e o *HTTP* é o protocolo que permite que estas ligações funcionem. Ou seja, a partir de uma determinada página eu cliço em um "*link*" (*hiperlink*) que irá me redirecionar para outra página no mesmo site ou em outro endereço.

Também podemos acessar uma página digitando diretamente o endereço da página correspondente ou simplesmente copiando o "*link*" e colando na *barra de endereços* do navegador.

O **HTTP** funciona em um modelo computacional *cliente-servidor* onde o navegador que



está abrindo a página é o cliente e a aplicação que está hospedada em um site da web é o servidor. O *HTTP* foi projetado para agilizar ao máximo a comunicação entre clientes e servidores de forma que sites de alto tráfego geralmente se utilizam de recursos como servidores de *cache* assim como do próprio *cache* local do computador cliente.

O **cache** é simplesmente uma cópia local do site armazenado online, de forma que ao acessarmos o site, a cópia local é aberta primeiro e depois as atualizações são baixadas do servidor. Desta forma, para o usuário a página abriu de forma quase imediata, mas na prática ele estava apenas vendo uma cópia do que foi acessado na sua última visita.

Muitos erros na navegação acontecem devido a problemas com o *cache* e mesmo no sistema operacional *Windows*. O que ocorre é que às vezes ao acessarmos uma determinada página ele carrega apenas do cache e não atualiza com a versão do servidor, de forma que o que vemos é apenas uma cópia não totalmente funcional (nem mesmo atual) da página previamente visitada e portanto pode não funcionar ou apresentar erros. Na maioria dos casos, teclar um simples "*F5*" pode resolver o problema atualizando a página e forçando o navegador a carregar a versão original do servidor. Em outros casos pode ser necessário efetuar uma limpeza no cache local a fim de carregar novamente todas as páginas diretamente dos servidores.

O *Windows* pode também apresentar travamentos e problemas de acesso exatamente por causa do cache, pois este vai se acumulando indefinidamente a cada página visualizada e pode chegar a consumir diversos *gigabytes* de espaço em disco. Por isso é sempre recomendável utilizar regularmente programas de limpeza tais como o *cCleaner* ou realizar o processo manualmente para eliminar o cache de cada navegador utilizado.

Durante muito tempo foi utilizado *html* simples para a criação de páginas, mas como a própria linguagem foi evoluindo, as ferramentas e técnicas utilizadas também foram se aprimorando e se tornando mais complexas e com visual mais arrojado e eficiente. Atualmente podemos utilizar sistemas complexos de banco de dados, criptografia de última geração e diversos tipos de aplicativos embutidos. Desta forma temos bancos online, animações, jogos, filmes, etc.

Esta evolução também foi necessária para atender as diversas plataformas utilizadas para a exibição das páginas *html*. Passamos a ter diversos navegadores concorrentes no mercado (*Mozilla Firefox*, *Google Chrome*, *Internet Explorer*, *Opera*, etc.), diversos sistemas operacionais tais como *Linux*, *Windows*, *OS2*, além de plataformas diferentes como celu-

lares, tablets, notebooks, computadores, smartTVs, etc.

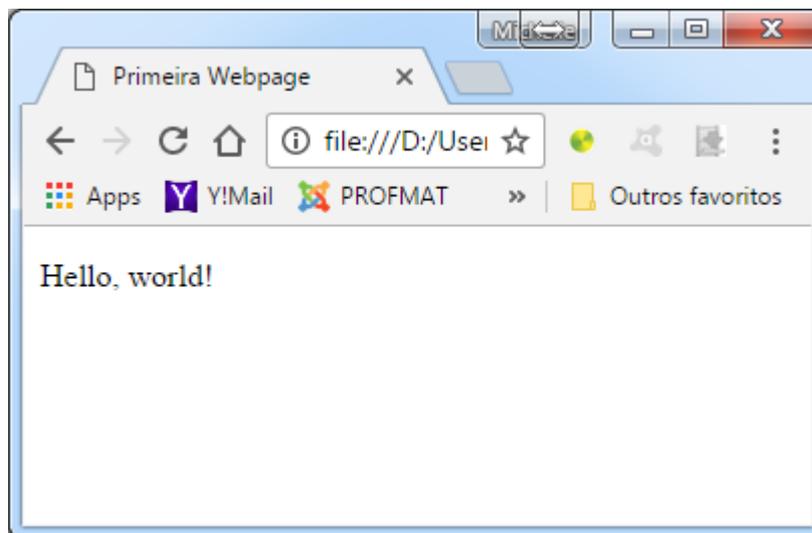
A estrutura funciona com o formato:

```
<abre>conteúdo</fecha>
```

A estrutura básica mínima para criação de uma página é a seguinte:

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>Primeira Webpage</title>  
</head>  
<body>  
  <p>Hello, world!</p>  
</body>  
</html>
```

O resultado será mostrado na figura a seguir:



HTML é basicamente formado por **TAGs** e cada declaração aberta deverá ter sua contrapartida fechando e o efeito desejado será aplicado sobre o conteúdo que estiver entre a declaração. Além desta estrutura básica, podemos inserir diversos tipos de formatação para adequar a página à estética desejada ou ao formato necessário para exibição.

Boas Práticas:**- Uso de Indentação ou Recuos:**

A indentação no HTML não é obrigatória, porém é sempre recomendável a fim de melhorar a organização e visualização do código fonte.

- Inserção de comentários:

A utilização de comentários é útil para deixar o código fonte mais facilmente localizável e identificável. Também podemos utilizá-los de forma instrutiva, explicando as ações tomadas e a forma de utilização.

```
<!-- este é um comentário -->
```

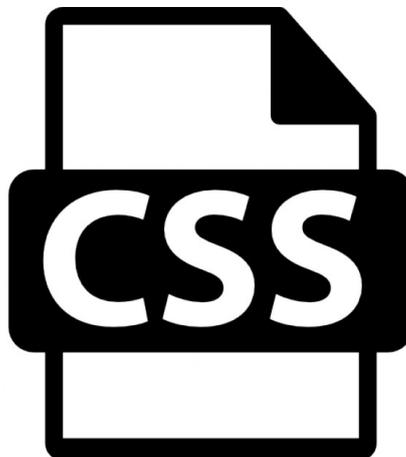
```
<!------->
```

```
<!--Tabela inicia aqui-->
```

Diversas outras opções podem ser inseridas em páginas **HTML**, inclusive comandos de programação e bancos de dados e animação, entre outros.

1.1 CSS

CSS ou *Cascading Style Sheets* é uma linguagem de folhas de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação como *HTML* ou *XML*. O seu principal benefício é a separação entre o formato e o conteúdo de um documento.



Da mesma forma que o *HTML*, também será interpretado no navegador do cliente ao ser aberta a página que possuir a sua referência.

Em vez de colocar a formatação dentro do documento, o desenvolvedor cria um *link* para uma página que contém os estilos, procedendo de forma idêntica para todas as páginas de um portal. Quando quiser alterar a aparência do portal basta portanto modificar apenas um arquivo.

Com a variação de atualizações dos navegadores como *Internet Explorer* que ficou sem nova versão de 2001 a 2006, o suporte ao *CSS* pode variar. O *Internet Explorer 6*, por exemplo, tem suporte total a *CSS1* e praticamente nulo a *CSS2*. Navegadores mais modernos como *Google Chrome* e *Mozilla Firefox* tem suporte maior, inclusive até a *CSS3*, ainda em desenvolvimento.

CSS tem uma sintaxe simples e utiliza uma série de palavras em inglês para especificar os nomes de diferentes estilos de propriedade de uma página.

Sintaxe

A linguagem de folhas de estilo precisa de uma sintaxe para assim ser expressa de uma forma legível para máquina. Por exemplo, aqui está uma folha de estilo simples escrita na sintaxe *CSS*:

```
h1 { font-size: 30px }
```

Neste exemplo, definimos que todo título "h1" na página destino terá o tamanho da fonte igual a 30 pixels.

Seletores

Os seletores especificam quais elementos devem ser influenciados pela regra de estilo. Como tal, seletores ligam a estrutura do documento às regras estilísticas nas folhas de estilo. No exemplo acima, o selector "h1" seleciona todos os elementos "h1" de um documento. Os seletores mais complexos podem selecionar elementos com base, por exemplo, em seu contexto, atributos e conteúdos.

Propriedades

Todas as linguagens de folhas de estilo têm algum conceito de propriedade que dá valores para alterar um aspecto de apresentação de um elemento. A propriedade *font-size* do *CSS* é usada no exemplo acima. As linguagens de folhas de estilo comuns costumam ter cerca de 50 propriedades para descrever a apresentação de documentos.

Valores e unidades

As propriedades mudam a apresentação de um elemento ao lhe atribuir um determinado valor. O valor pode ser uma sequência, uma palavra-chave, um número, ou um número com um identificador da unidade. Além disso, os valores podem ser listas ou expressões envolvendo vários dos valores acima mencionados. As linguagens de folhas de estilo comuns têm cerca de dez unidades diferentes.

Mecanismo de propagação de valor

Para evitar ter que especificar explicitamente todos os valores para todas as propriedades de todos os elementos, as linguagens de folhas de estilo têm mecanismos para propagar os valores automaticamente. O principal benefício da propagação de valor é tornar as folhas de estilo menos extensa. No exemplo acima, apenas o tamanho da fonte foi especificada; outros valores podem ser encontrados por meio de mecanismos de propagação de valores. A herança de valores iniciais e a escrita em cascata são exemplos de mecanismos de propagação de valores.

Modelo de formatação

Todas as linguagens de folhas de estilo suportam algum tipo de modelo de formatação. A maioria dessas linguagens têm um modelo de formatação visual que descreve, com algum detalhe, como o texto e outros conteúdos serão mostrados.

Exemplo de um arquivo de CSS básico:

```
body { background-color: lightblue;
}
h1 { color: white;
      text-align: center;
}
p { font-family: verdana;
      font-size: 20px;
}
```

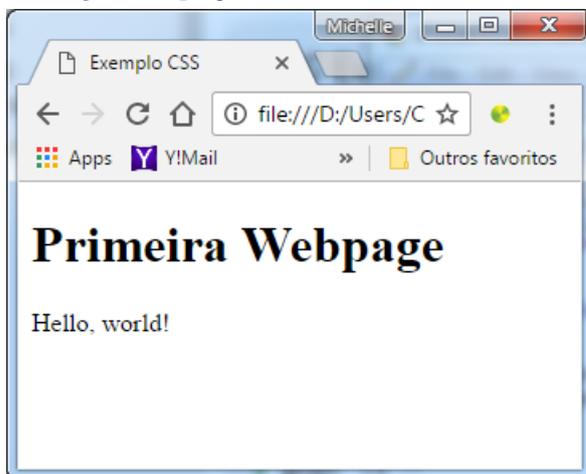
Página HTML sem CSS:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemplo CSS</title>
  </head>
  <body>
    <h1>Primeira Webpage</h1>
    <p>Hello, world!</p>
  </body>
</html>
```

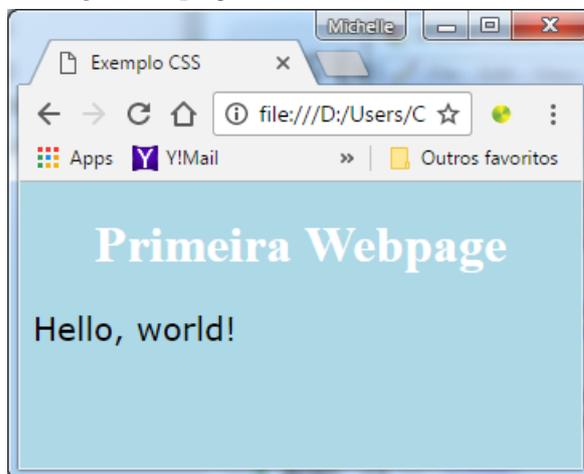
Página HTML com CSS:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemplo CSS</title>
    <link rel="stylesheet" href="exemplo_css.css">
  </head>
  <body>
    <h1>Primeira Webpage</h1>
    <p>Hello, world!</p>
  </body>
</html>
```

Exibição da página sem CSS:



Exibição da página com CSS:



Lembrando que, semelhantemente ao HTML, o CSS não necessita de indentação para seu funcionamento. A aplicação de tabulações e espaços em branco são recomendados apenas para melhor compreensão e visualização do código. A organização facilita encontrar erros e identificar estruturas com rapidez e eficiência.

Desta forma, os dois códigos a seguir geram o mesmo resultado:

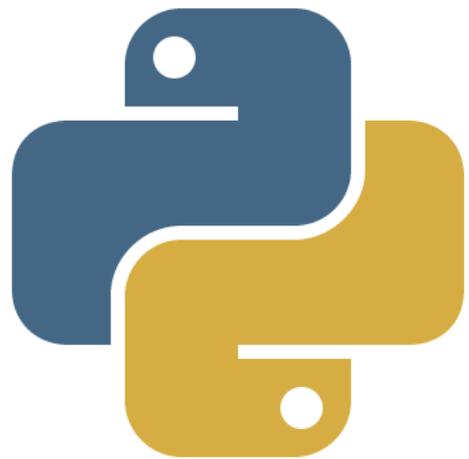
```
body{overflow:hidden;background:#000000;}
```

```
body {  
    overflow: hidden;  
    background: #000000;  
}
```

2 PYTHON

Python é uma linguagem de programação de *alto nível, interpretada, de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte.*

A linguagem foi projetada com a filosofia de enfatizar a importância do esforço do programador sobre o esforço computacional. Prioriza a legibilidade do código sobre a velocidade ou expressividade. Combina uma sintaxe concisa e clara com os recursos poderosos de sua biblioteca padrão.



Abaixo iremos pormenorizar cada um de seus atributos:

- **Linguagem Interpretada** – é uma linguagem de programação em que o código fonte nessa linguagem é executado por um programa de computador chamado interpretador e em seguida é executado pelo sistema operacional ou processador. Ou seja, de uma forma ela é semelhante à linguagem HTML que é interpretada no navegador do usuário no ato do acesso.
- **Linguagem de Script** – é uma linguagem que é executada no interior de programas ou de outras linguagem de programação, não se restringindo a estes ambientes. Ou seja, podem estender as funcionalidades de um programa sem contudo alterá-lo e/ou controla-lo.
- **Linguagem Imperativa** – que descreve o programa como um conjunto de ações, enunciados ou comandos que mudam o estado de um programa dizendo a cada passo o que deve ser executado ou interpretado. Ou seja, está relacionado ao tempo verbal imperativo, onde o programador diz ao computador “faça isso, depois isso, depois aquilo”.

- **Orientada a Objetos** – é um modelo de análise, projeto e programação baseado na composição e interação entre diversas unidades chamadas de objetos. Os objetos são partes de código independentes e funcionais que podem ser aproveitadas conforme o desejo e necessidade do programador. Depois de definidos, implementa-se um conjunto de classes que definem os objetos presentes no sistema e cada classe determina o comportamento e estados possíveis de seus objetos assim como o relacionamento com outros objetos.
- **Programação Funcional** – é uma abordagem de programação que enfatiza a aplicação de funções. A ideia básica é utilizar-se de funções predefinidas que evitam estados ou dados mutáveis. No caso de Python, as inúmeras funções preexistentes auxiliam desde a simples formatação de texto até a resolução de cálculos, concatenação de valores e até operações com arquivos. Este tipo de abordagem contrasta em parte com a programação imperativa que, além das funções, já possui construções mais complexas. Desta forma, Python não é considerada estritamente funcional, pois além de funções utilizadas no ato da interpretação, também se utiliza dessas construções imperativas.
- **Tipo forte** – Linguagem implementadas com tipificação forte exigem que o tipo de dado de um valor seja do mesmo tipo da variável ao qual este valor será atribuído. Ou seja, se declararmos uma variável como inteira, somente será aceita a entrada de valores inteiros para esta variável.
- **Tipo Dinâmico** – A verificação de um dado é feita de forma dinâmica, ou seja, em tempo de execução – característica de uma linguagem interpretada.

Construções em Python incluem:

- Estrutura de seleção (*If, else elif*)
- Estrutura de repetição (*for, while*)
- Construção de classes (*class*)
- Construção de sub-rotinas (*def*)
- Construção de escopo (*with*)

Tipos de dado

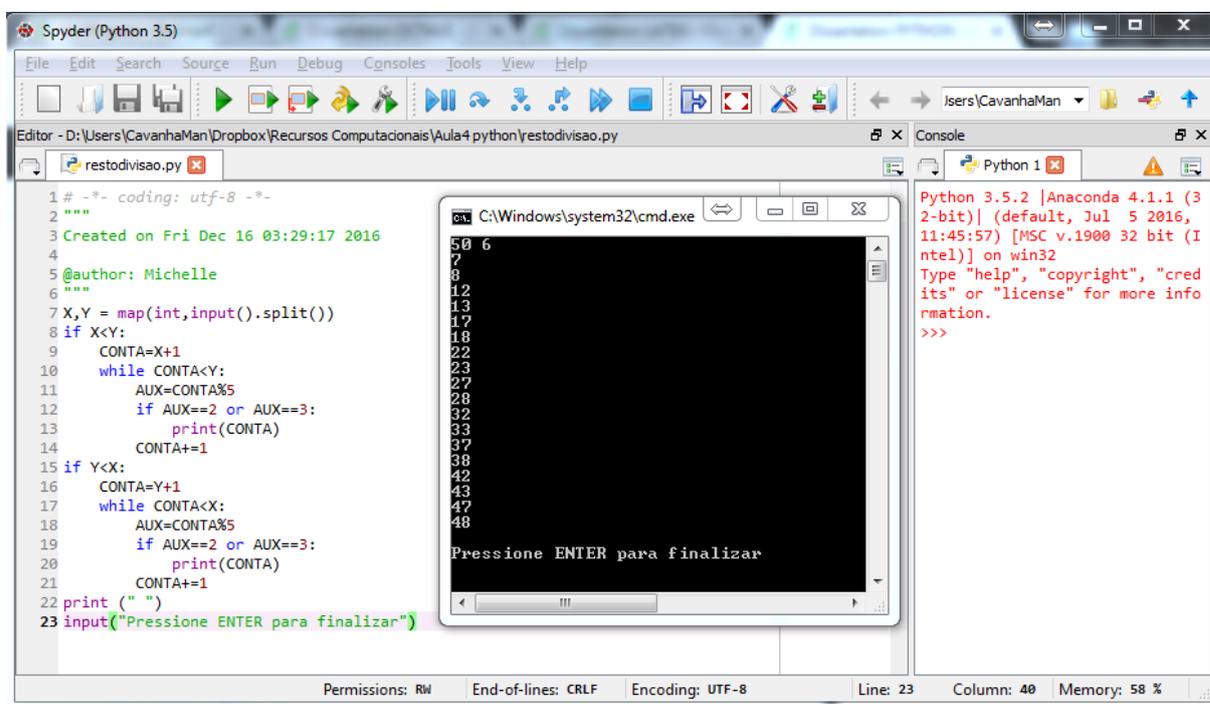
A tipagem de Python é forte, pois os valores e objetos têm tipos bem definidos e não sofrem coerções como em outras linguagens. São disponibilizados diversos tipos de dados nativos:

- *str, unicode* – uma cadeia de caracteres imutável
- *list* – lista heterogênea mutável

- *tuple* – tupla imutável (lista ordenada de n elementos)
- *set*, *frozenset* – conjunto não ordenado, não contem elementos duplicados
- *dict* – conjunto associativo
- *int* – número de precisão fixa
- *float* – ponto flutuante
- *complex* – número complexo
- *bool* – booleano (1 ou 0, verdadeiro ou falso)

Python também permite a definição dos tipos de dados próprios através de classes.

Tela do compilador Python Spyder do Anaconda:



Palavras Reservadas

O Python 2.5.2 define as seguintes 31 palavras reservadas, que não podem ser utilizadas como um identificador por fazerem parte da gramática da linguagem:

| | | | | | | | |
|-----------|-------------|------------|--------------|---------------|--------------|-----------------|----------------|
| <i>in</i> | <i>or</i> | <i>def</i> | <i>else</i> | <i>assert</i> | <i>exec</i> | <i>raise</i> | <i>finally</i> |
| <i>is</i> | <i>try</i> | <i>not</i> | <i>pass</i> | <i>global</i> | <i>yield</i> | <i>continue</i> | <i>except</i> |
| <i>if</i> | <i>elif</i> | <i>del</i> | <i>with</i> | <i>break</i> | <i>class</i> | <i>lambda</i> | <i>import</i> |
| <i>as</i> | <i>for</i> | <i>and</i> | <i>while</i> | <i>from</i> | <i>print</i> | <i>return</i> | |

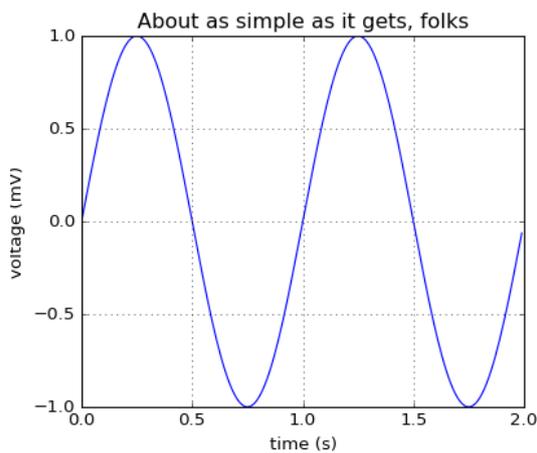
2.1 MATPLOTLIB

Matplotlib é uma biblioteca de plotagem 2D, ou seja é uma coleção de sub-rotinas e funções utilizadas em **Python** no desenvolvimento de gráficos em duas dimensões. Essa biblioteca consegue gerar figuras de qualidade em uma grande variedade de formatos e de ambientes interativos em diversas plataformas.

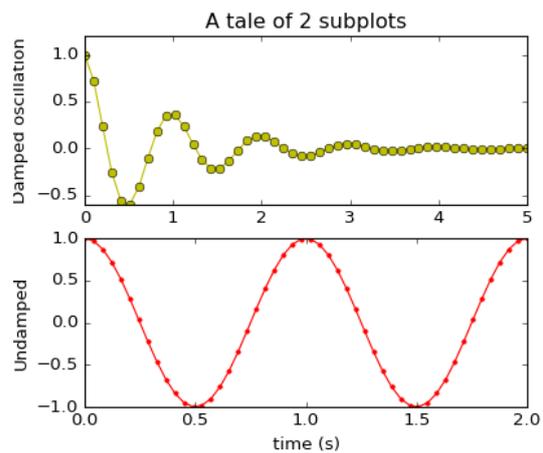
O objetivo da biblioteca é tornar fácil as tarefas rotineiras e tornar possíveis as tarefas mais difíceis. Ela permite gerar inúmeros modelos computacionais com apenas algumas linhas de código. Sua interface é simples e semelhante ao Matlab.

Segue a seguir alguns exemplos de plotagem geradas em **matplotlib**.

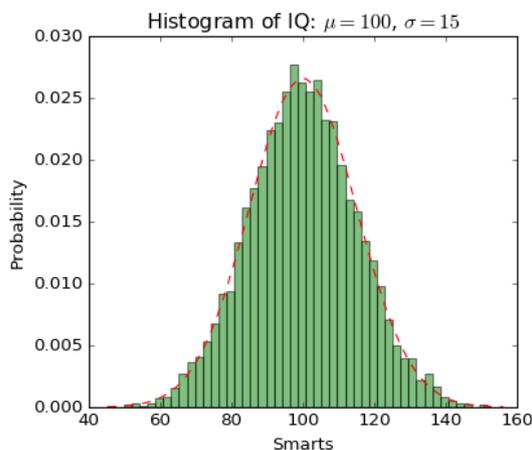
Plotagem simples:



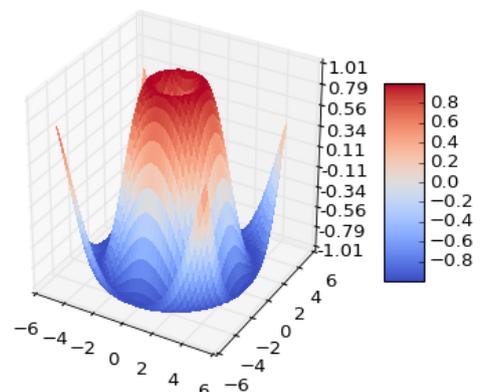
Plotagem com múltiplos eixos gerado pelo comando `subplot()`:



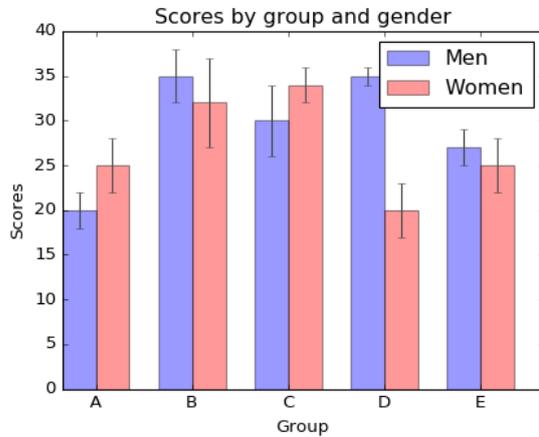
Plotagem de histograma (distribuição de frequência) gerada com o comando `hist()`:



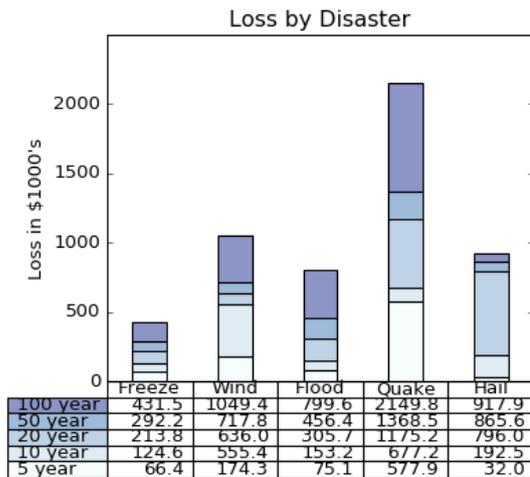
Gráficos 3D simples utilizando a ferramenta `mplot3d`:



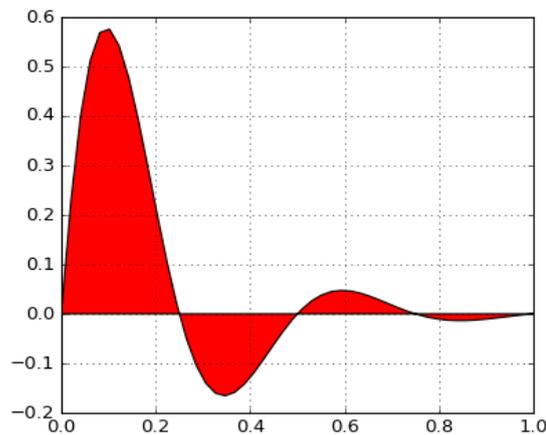
Gráficos em barras – que são facilmente gerados com o comando `bar()`:



Gráficos com tabelas – gerados com o comando `table()`:



Plotagem de curvas e polígonos preenchidos com o comando `fill()`:



Gráficos em pizza (ou torta) – gerados facilmente com o comando `pie()`:

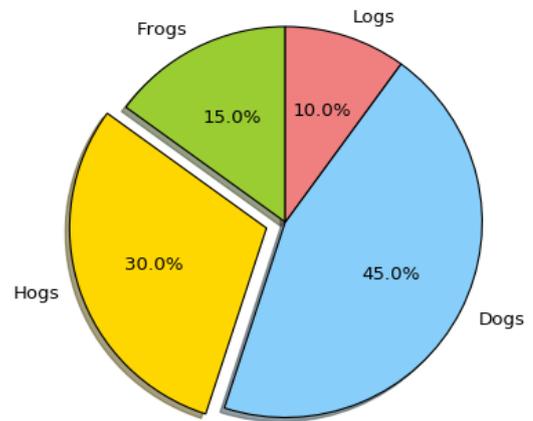
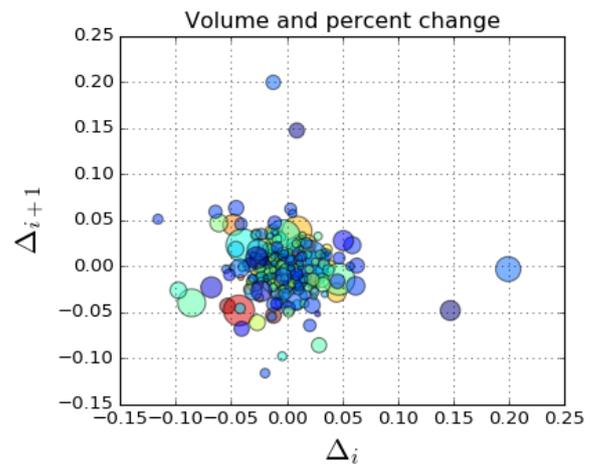
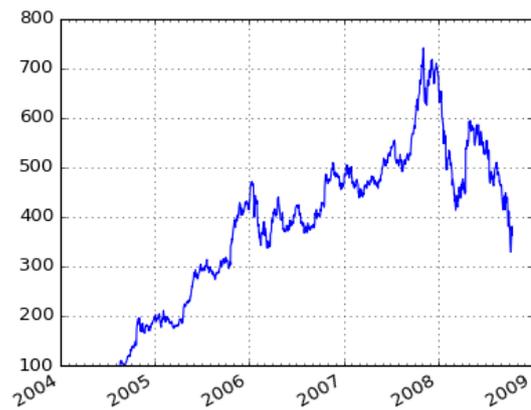


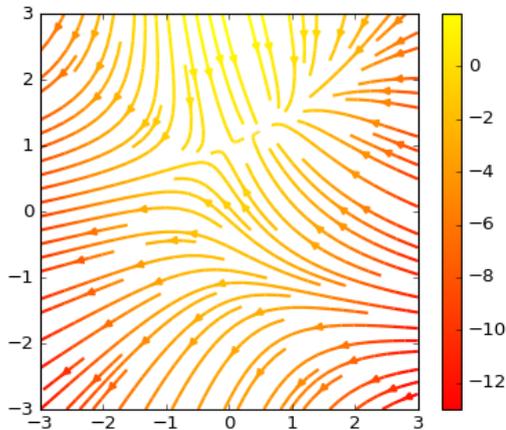
Gráfico de dispersão, gerado com o comando `scatter()`:



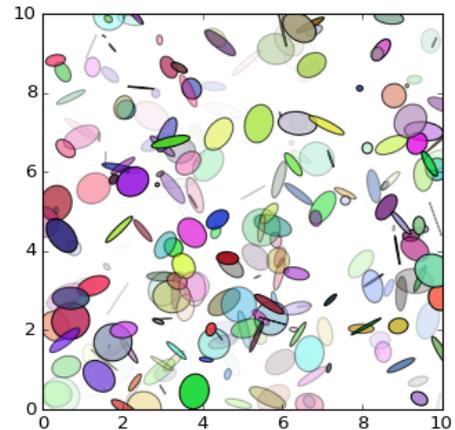
Plotagem de dados de data com intervalos maiores ou menores:



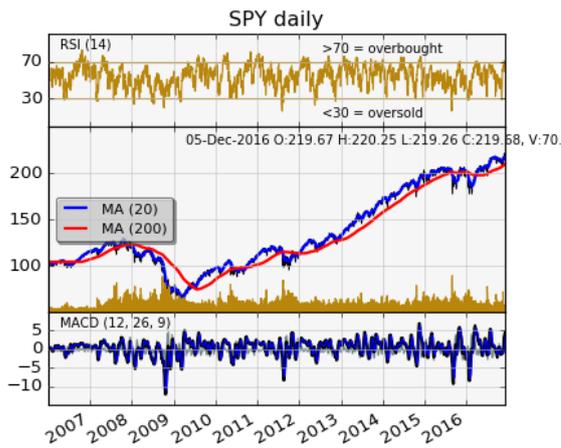
Linha de corrente gerada com o comando `streamplot()`:



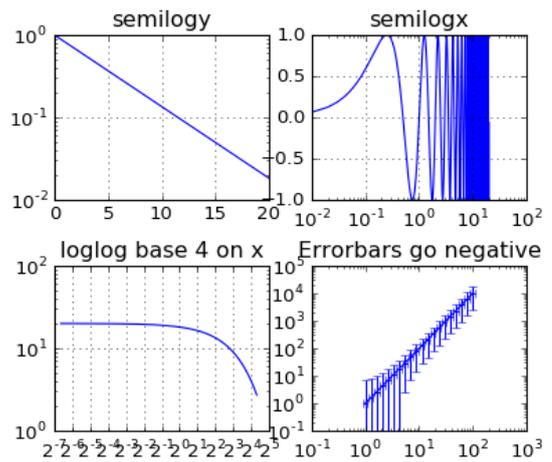
Elipses:



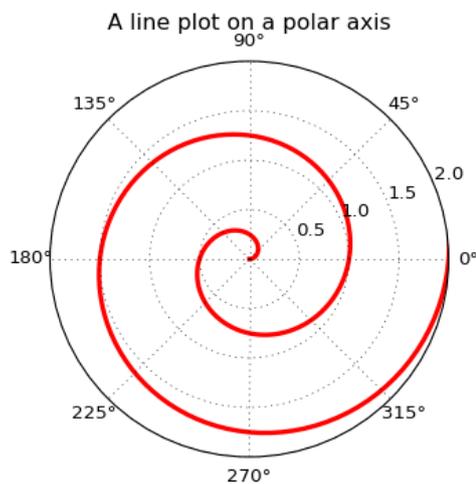
Gráficos financeiros:



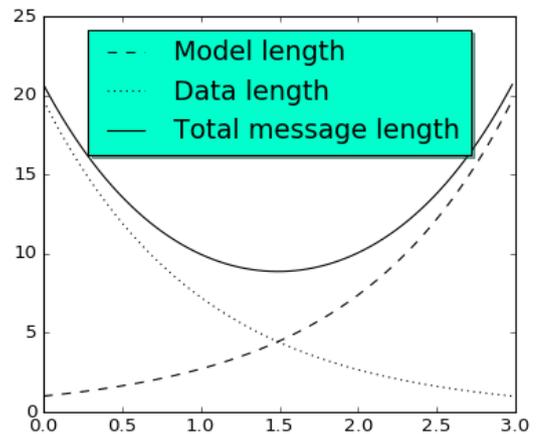
Funções simples para criação de plotagens logarítmicas:



Gráficos polares com o comando `polar()`:



Comando `legend()` para gerar automaticamente legendas:



2.2 MATH

Este módulo está sempre disponível e provê acesso às funções matemáticas definidas pela *linguagem C*.

As funções são:

- Funções da Teoria dos Números e de Representação
- Funções de Potência e Logarítmicas
- Funções Trigonométricas
- Conversão Angular
- Funções Hiperbólicas
- Além de Funções Especiais e Constantes

Em Python estas funções não utilizarão números complexos. Neste caso utilizaremos as mesmas funções porém na biblioteca *cmath*.

3 LATEX

LaTeX é um conjunto de macros para o programa de diagramação de textos **TeX**, utilizado amplamente na produção de textos matemáticos e científicos, devido a sua alta qualidade tipográfica.

A large, stylized logo of the word "LATEX" in a serif font. The letters are black and have a classic, elegant appearance. The 'L' and 'A' are particularly prominent, with the 'L' having a long vertical stem and the 'A' having a wide, flat top. The 'T' is tall and narrow, and the 'E' has a distinct horizontal bar. The 'X' is composed of two 'L' shapes mirrored horizontally.

Entretanto, também é utilizado para produção de cartas pessoais, artigos e livros sobre assuntos diversos.

O sistema LaTeX fornece ao usuário um conjunto de comandos de alto nível, facilitando dessa forma sua utilização por iniciantes. Possui abstrações para lidar com bibliografias, citações, formatos de páginas, referência cruzada e tudo mais que não seja relacionado ao conteúdo do documento em si.

Podemos dividir os programas de processamento de texto em duas classes. Com os chamados processadores de texto, existe um menu na tela apresentando os recursos, que podem ser usados no processamento do texto, que por sua vez podem ser selecionados com o uso do mouse. Depois de selecionado um recurso, o texto é digitado e aparece na tela exatamente como vai ser impresso no papel. O usuário pode ver logo no estágio de entrada do texto, se o texto será impresso como esperado. Este método é chamado "*what-you-see-is-what-you-get*" ou simplesmente **WYSWYG**, ou seja, "*o que você vê é o que você obtém*".

A segunda classe, que é a que pertence o LaTeX, o processamento do texto é feito em duas etapas distintas. O texto a ser impresso e os comandos de formatação são escritos em um arquivo fonte com o uso de um editor de textos, isto é, um programa que escreve textos em meio magnético. Em seguida o arquivo fonte é submetido a um programa formador de textos, no nosso caso o LaTeX, que gera um arquivo de saída, que pode ser impresso ou visualizado na tela.

Programas deste tipo podem parecer inicialmente mais complicados do que os do outro tipo, mas apresentam uma série de vantagens em relação aos processadores de texto, como por exemplo:

- Mudanças na formatação do texto inteiro com apenas a mudança de alguns comandos.
- Escrita de fórmulas complexas usando apenas comandos;
- Numeração automática de fórmulas, seções, definições, exemplos e teoremas, o que permite que você faça mudanças na ordem do texto sem que seja necessário trocar os números dos itens.
- As citações a fórmulas, seções, definições, exemplos, teoremas além de citações bibliográficas também podem ser automatizadas, de forma que mudanças no texto não produzem erros nas citações.

Além das vantagens citadas acima, não podemos deixar de ressaltar que o LaTeX é um freeware, com implementações disponíveis para todos os sistemas operacionais mais famosos. Assim, tudo o que fizermos aqui no curso você pode e deve fazer no seu computador, além de usá-lo para treinar com outros textos (por exemplo, um exercício que você fez na aula de Teoria dos Números ou um teorema de Cálculo). A próxima sessão será sobre a instalação dos programas.

Curiosidade: O nome TEX na verdade é formado pelas letras gregas tau, épsilon, chi, que são as iniciais da palavra tecnologia. A pronúncia correta é téqui, latéqui, etc.

2. Estrutura Básica de um Arquivo LaTeX

Um arquivo LaTeX contém o texto a ser processado e comandos que indicam como fazê-lo. Em geral um comando inicia-se com \backslash .

A entrada mais simples para um arquivo LaTeX é mostrado abaixo:

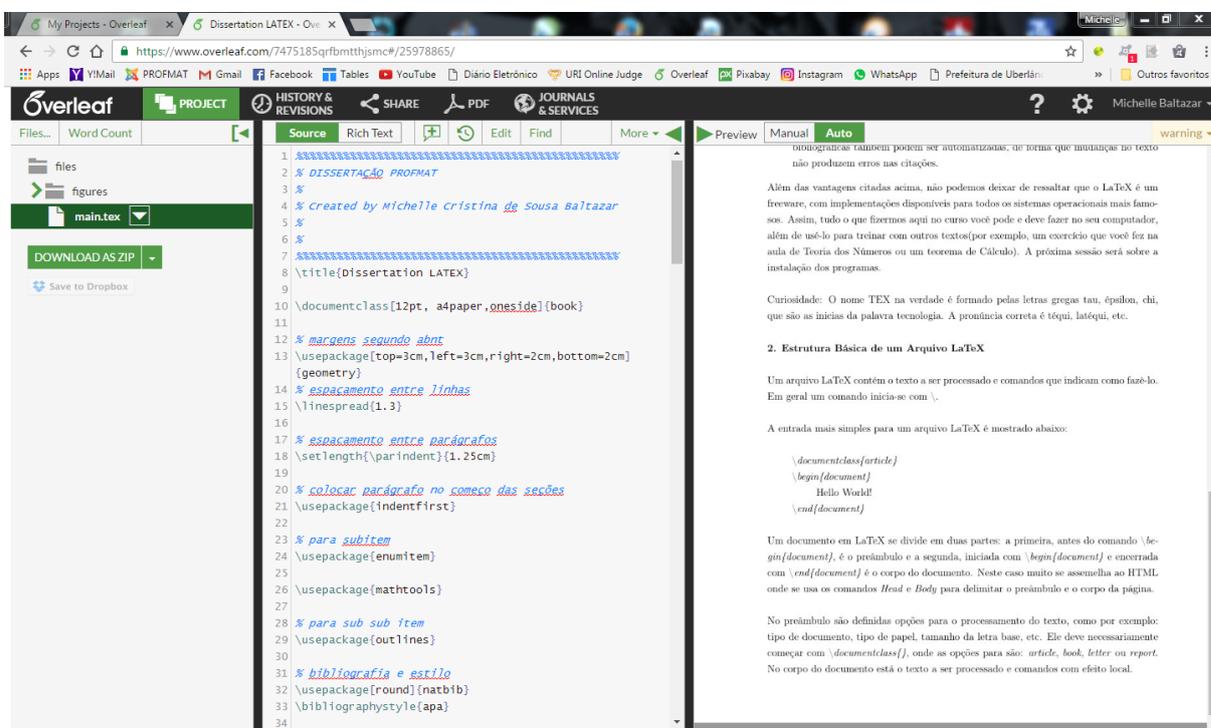
```
 $\backslash documentclass\{article\}$   
 $\backslash begin\{document\}$   
    Hello World!  
 $\backslash end\{document\}$ 
```

Um documento em LaTeX se divide em duas partes: a primeira, antes do comando $\backslash be-$

`gin{document}`, é o preâmbulo e a segunda, iniciada com `\begin{document}` e encerrada com `\end{document}` é o corpo do documento. Neste caso muito se assemelha ao HTML onde se usa os comandos *Head* e *Body* para delimitar o preâmbulo e o corpo da página.

No preâmbulo são definidas opções para o processamento do texto, como por exemplo: tipo de documento, tipo de papel, tamanho da letra base, etc. Ele deve necessariamente começar com `\documentclass{}`, onde as opções para são: *article*, *book*, *letter* ou *report*. No corpo do documento está o texto a ser processado e comandos com efeito local.

A seguir temos a tela de um ótimo editor de LaTeX online, o Overleaf:



O Beamer é uma classe do LATEX para criação de apresentações no formato PDF, o que as torna altamente portáteis. Sua estrutura é a mesma do LATEX, com algumas características específicas do Beamer. É possível desenvolver apresentações dinâmicas com sobreposições e transições animadas entre os quadros.

As vantagens de usar o Beamer ao invés do MS-PowerPoint e outras ferramentas, são as mesmas de usar o LATEX para desenvolver documentos:

- Os comandos principais do LATEX funcionam também no Beamer. Alguém capaz de escrever um documento em LATEX, é perfeitamente capaz de escrever uma apresentação em Beamer.
- Pode-se facilmente criar camadas, sobreposições e efeitos dinâmicos.

- Um sumário dinâmico é criado automaticamente, com links para cada seção e subseção que você fizer.
- Podem-se escolher temas que melhor se adequem ao propósito de sua apresentação. Os temas são desenvolvidos para serem legíveis e úteis, de forma a facilitar a compreensão da audiência e dar uma aparência mais profissional à apresentação.
- Existe controle sobre o layout, cores e fontes, que podem ser alterados para todo o documento, o que permite modificações de última hora sem complicação.
- A característica multiplataforma dos formatos LATEX e PDF.

4 OCTAVE

GNU Octave é uma linguagem computacional, desenvolvida para computação matemática. Possui uma interface em linha de comando para a solução de problemas numéricos, lineares e não-lineares, também é usada em experimentos numéricos.

Faz parte do projeto **GNU**, é um software livre sob os termos da licença **GPL**.

É compatível com *MATLAB*, possuindo um grande número de funções semelhantes e é atualmente uma das maiores alternativas gratuitas ao *MATLAB* (As outras opções são *FreeMat* e *Scilab*).

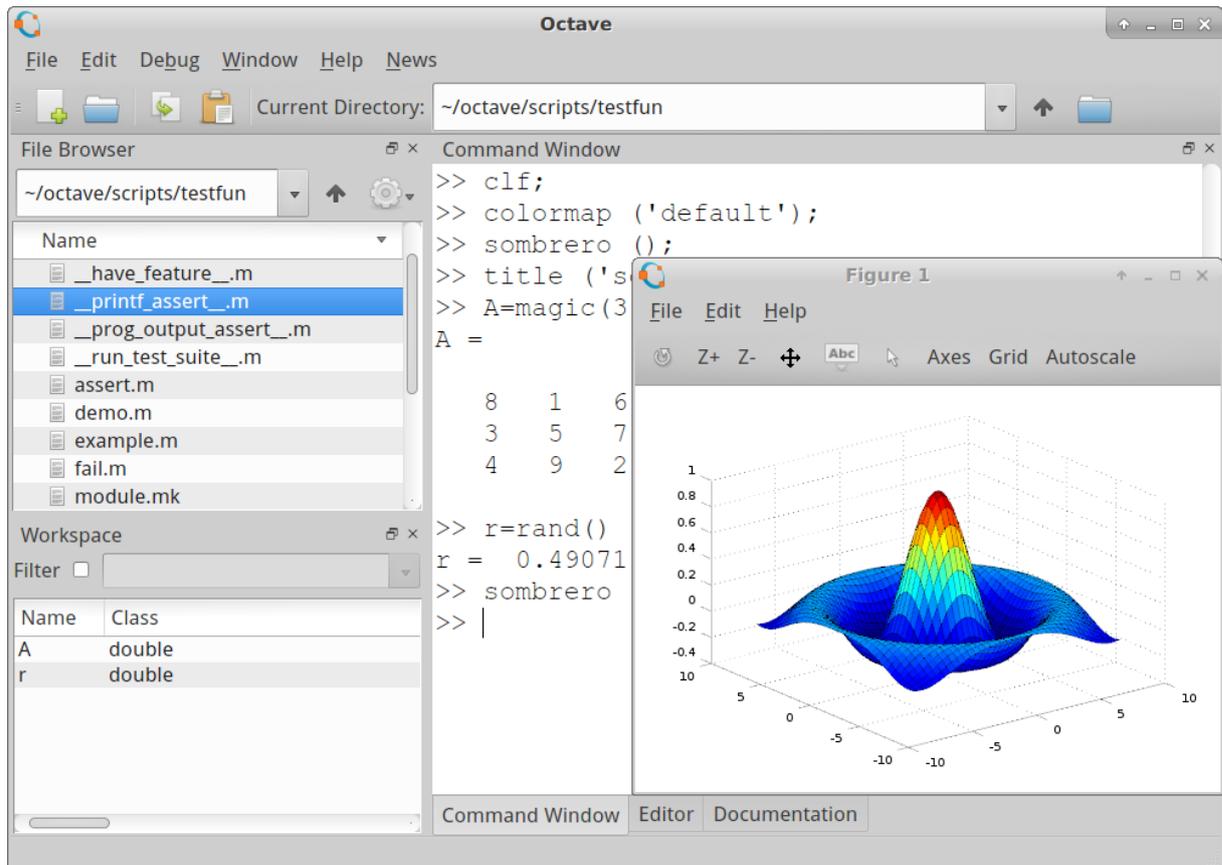
GNU é um sistema operacional e uma extensa coleção de softwares e aplicativos, todos gratuitos sob a licença *GPL* supracitada - *General Public License*, ou Licença Pública Geral. É a licença com maior utilização por parte de projetos de software livre.

O GNU/Octave conta com interação através de uma linguagem de programação de alto nível, criada inicialmente para cálculo numérico. Fornece uma linha de comando conveniente resolvendo problemas lineares e não-lineares numericamente, e executando outras tarefas numéricas usando uma linguagem que é bastante compatível com o *MATLAB*. Pode também ser usado como uma linguagem orientada a grupos.

Possui ferramentas extensivas para a resolução de problemas lineares numéricos comuns de álgebra, cálculo aproximado de raízes de equações não-lineares, funções ordinárias, polinômios, cálculo de integrais, e integração numérica de equações diferenciais ordinárias e diferenciais-algébricas. Pode ser facilmente expandido e adaptado através de funções definidas pelo utilizador, escritas na própria linguagem presente ou usando módulos di-



namicamente carregados escritos em outras linguagens como C++, C e Fortran.

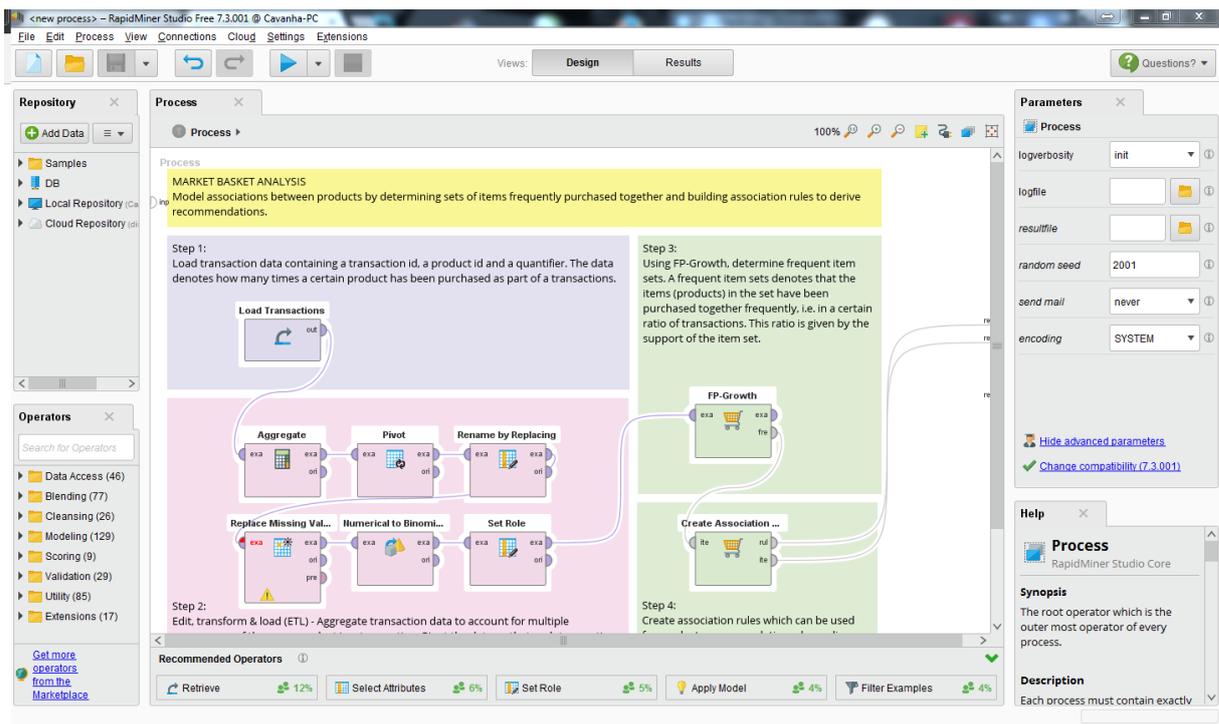


5 RAPIDMINER

Rapidminer é uma plataforma de análise poderosa e colaborativa com um rico conjunto de ferramentas para acelerar a criação, entrega e manutenção de análises preditivas de alto valor.



A Plataforma RapidMiner oferece mais funções do que qualquer outra solução visual, além de ser aberta e extensível para suportar todas as suas necessidades de análise de dados. RapidMiner fornece um ambiente integrado para a aprendizagem de máquina (inteligência artificial), mineração de dados, mineração de texto, análise preditiva e análise de negócios. É utilizado para aplicações empresariais e comerciais, bem como para pesquisa, educação, treinamento, prototipagem rápida e desenvolvimento de aplicativos e suporta todas as etapas do processo de mineração de dados, incluindo preparação de dados, visualização de resultados, validação e otimização.

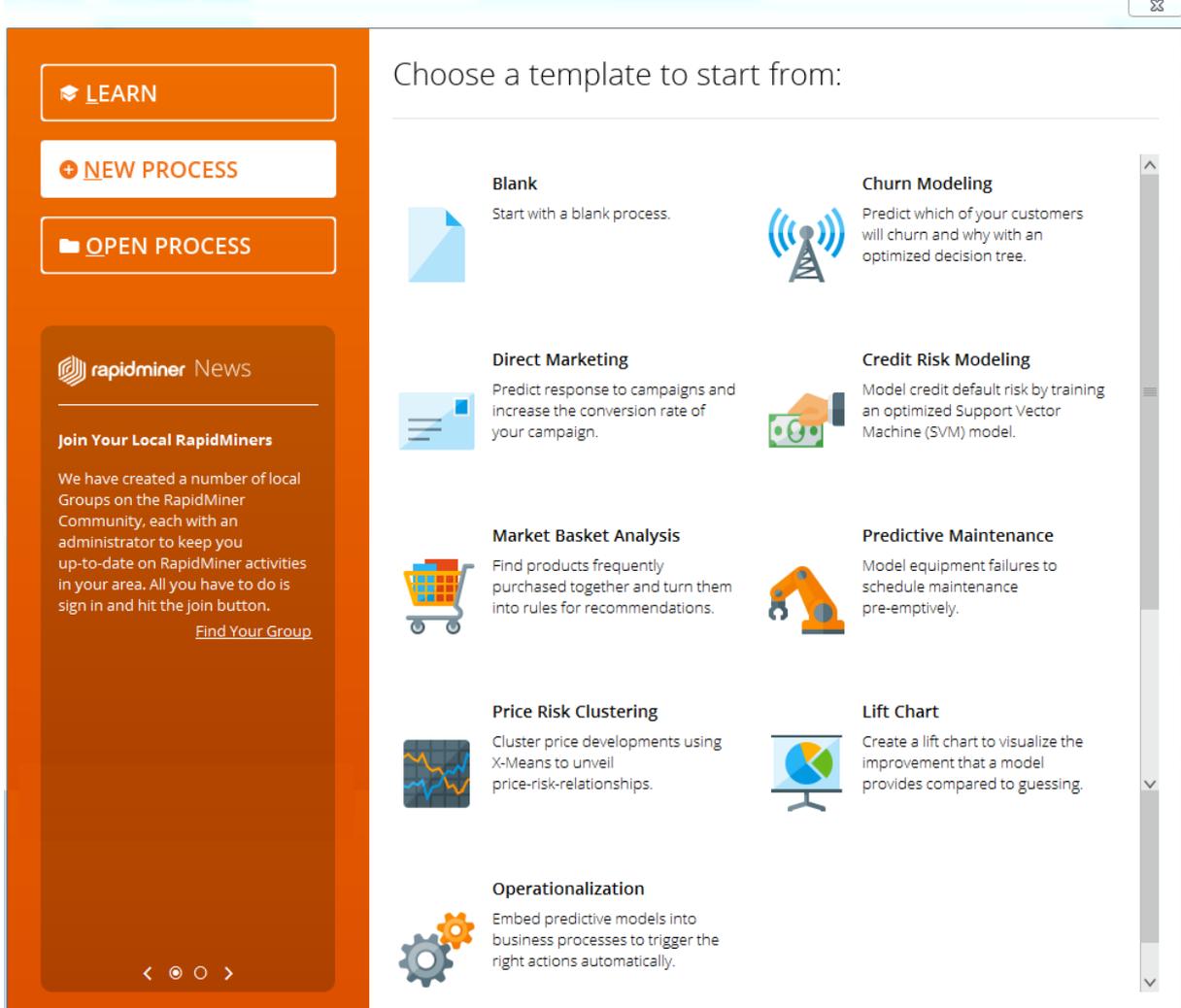


Atualmente a mineração de dados é usada onde quer que haja dados digitais disponíveis. Exemplos notáveis de mineração de dados podem ser encontrados em todo tipo de negócios: medicina, ciência, engenharia, genética, vigilância, comércio, política, etc.

Um bom exemplo de utilização de mineração de dados e de texto é o que acontece com o Facebook que, além de ser uma plataforma de comunicação é amplamente utilizado para conhecer gostos e peculiaridades da população, agregando por idade, sexo, religião, região, escolaridade, etc.

O RapidMiner usa um modelo cliente-servidor com o servidor sendo oferecido como software, como um Serviço ou em uma estrutura de nuvem, atendendo a diversas plataformas e necessidades.

A tela inicial permite escolher o tipo de modelo que será utilizado baseado na necessidade do usuário:



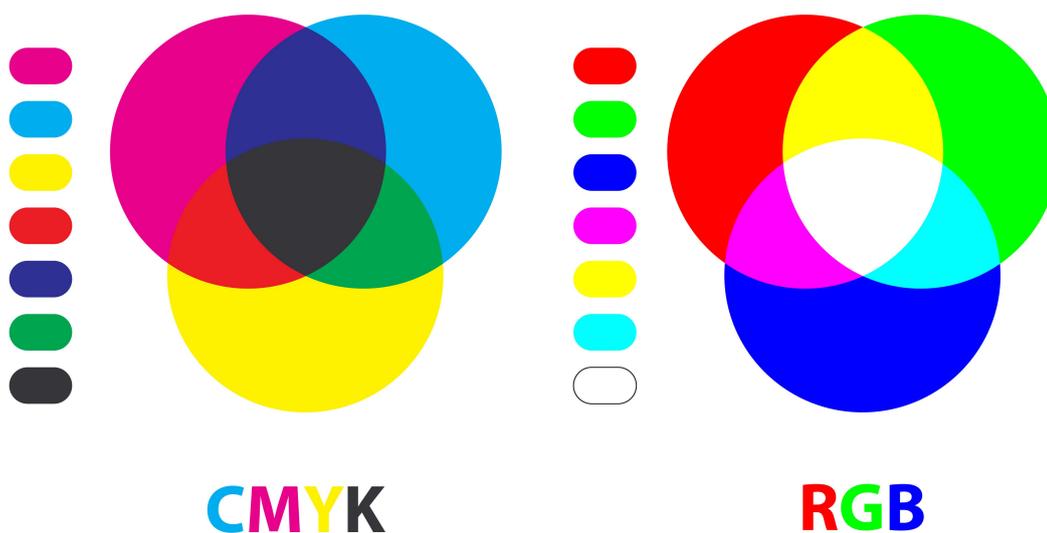
O RapidMiner é fácil de usar, utilizando diversos operadores que são separados por classes, para diferentes tipos de análises de dados e processos, apenas arrastando e soltando os operadores, além de conferir constantemente o processo que está sendo construído.

6 GIMP

Gimp é um programa de códigos aberto voltado principalmente para edição de imagens e em menor escala também para desenho vetorial. Possui recursos para criação e manipulação de imagens e fotografias, também redimensionar fotos, alterar cores, e estas podem ser no modelo RGB ou CMYK.



RGB é formado pelas cores vermelho, verde e azul, a união destas cores duas a duas ou as três, formam as demais cores. Também temos o modelo CMYK formado pelas cores ciano, magenta, amarelo e preto.

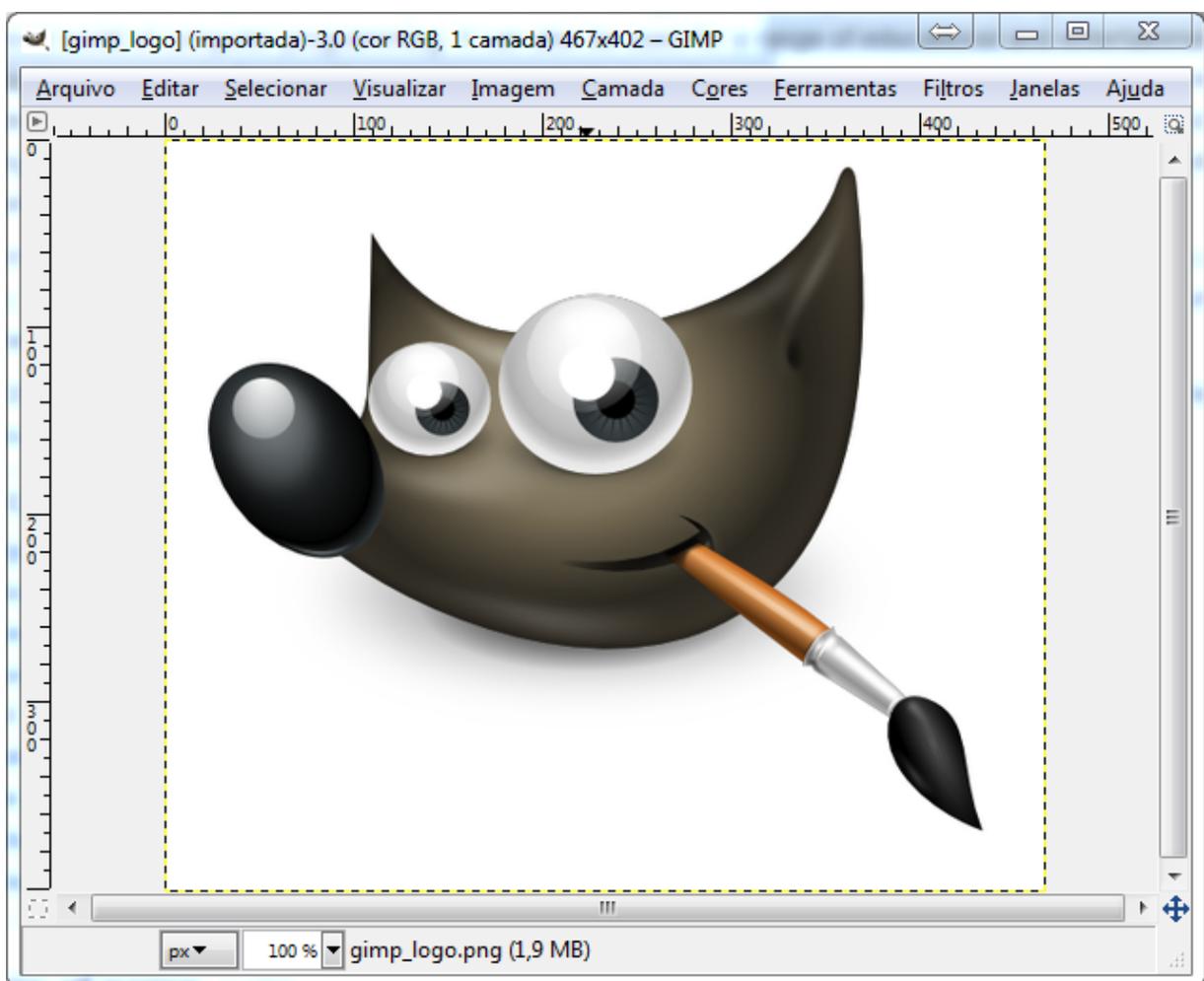


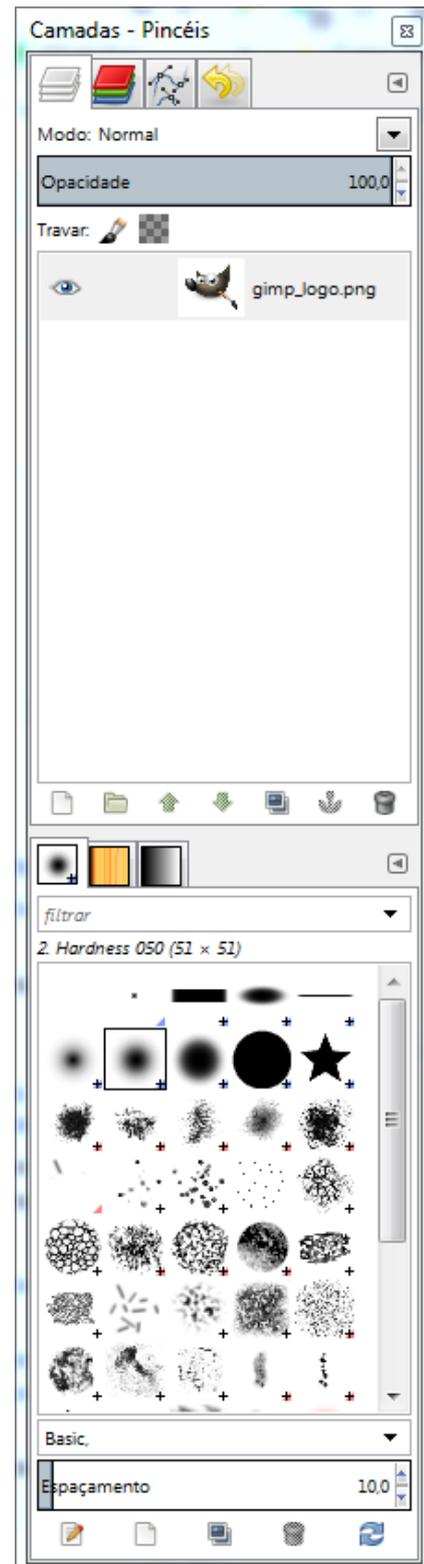
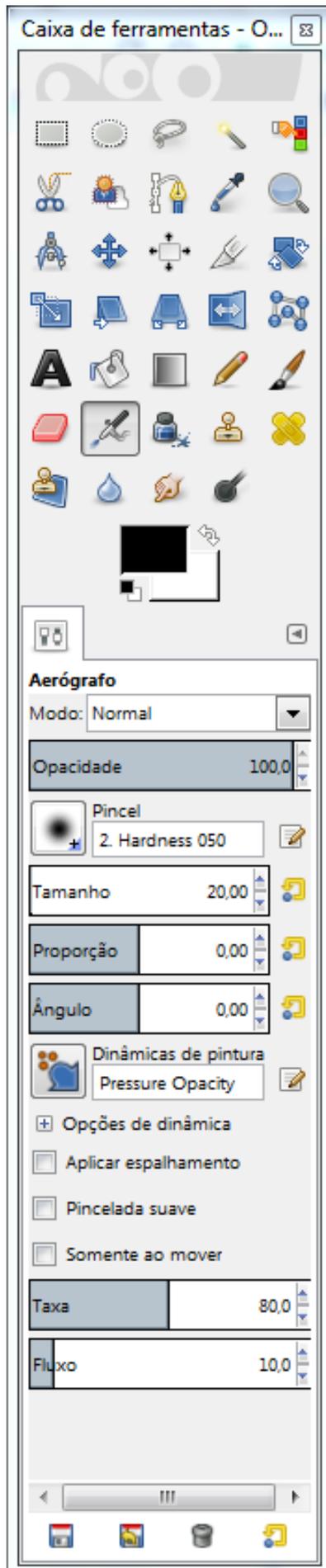
O programa permite salvar a imagem em diversas extensões, lembrando que algumas podem ter perda na qualidade ou limita a quantidade de cores na sua compressão.

O Gimp Faz parte do projeto **GNU** (coleção de softwares e aplicativos gratuitos), é um software livre sob os termos da licença **GPL** - *General Public License*, ou Licença Pública Geral. O seu código fonte pode ser alterado e distribuído livremente.

O programa está disponível para Windows e Linux e outros sistemas operacionais.

O Gimp possui três janelas distintas, são elas: a janela inicial onde encontramos a funcionalidades do programa, a janela de ferramentas as quais usamos para tratar e modificar a imagem; e a janela de camadas que permite trabalhar com as camadas e canais de cores e vetores.





7 INKSCAPE

Scratch é uma linguagem de programação criada em 2007 pelo Media Lab do *MIT* (*Massachusetts Institute of Technology*).

O Inkscape é um software livre (*open source*) utilizado para editoração eletrônica de imagens e documentos vetoriais tais como ilustrações, diagramas, artes de linha, gráficos, logotipos e pinturas complexas.

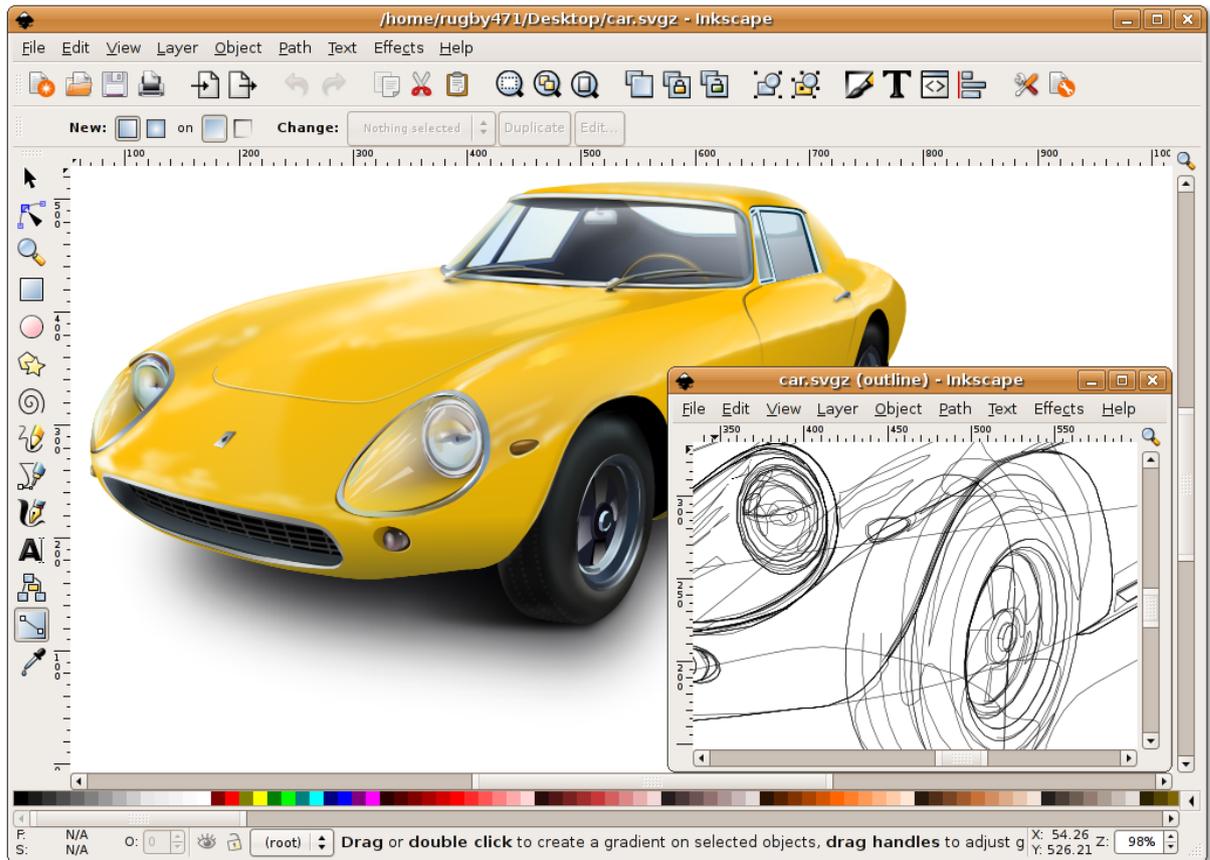


Estas imagens podem ser redimensionadas sem perda de qualidade - gera imagens a partir de um caminho de pontos definindo suas coordenadas, de forma transparente ao usuário, têm maior aplicação em desenho técnico ou artístico.

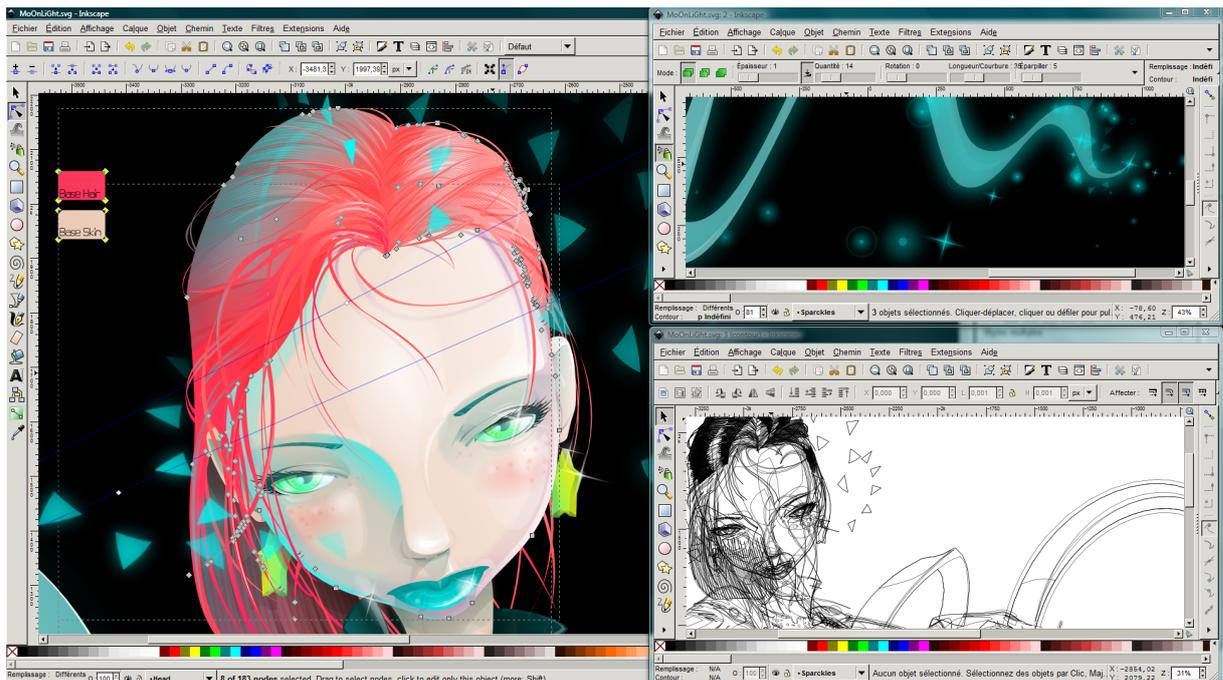
O Inkscape pode renderizar formas vetoriais primitivas (por exemplo, retângulos, elipses, polígonos, arcos, espirais, estrelas e caixas 3D) e texto. Esses objetos podem ser preenchidos com cores sólidas, padrões, gradientes de cores radiais ou lineares e suas bordas podem ser manipuladas, inclusive com transparência ajustável.

Incorporação e traçado opcional de gráficos de varredura também é suportado, permitindo que o editor para criar gráficos vetoriais de fotos e outras fontes. As formas criadas podem ser manipuladas com transformações como mover, girar, dimensionar e enviesar.

O Inkscape trabalha nativamente com o formato *SVG* (*Scalable Vectorial Graphics*), um formato aberto de imagens vetoriais, O aplicativo também exporta para o popular formato da Internet PNG e importa vários formatos vetoriais ou bitmap, como por exemplo: GIF, JPG, entre outros.



Rasterização, é a tarefa de converter uma imagem vetorial em uma imagem raster (pixels ou pontos) para a saída em vídeo ou impressora. O termo Rasterização também é utilizado para converter uma imagem formada por vetores para um arquivo de formato bitmap (SVG para PNG).



8 SCRATCH

Scratch é uma linguagem de programação criada em 2007 pelo Media Lab do *MIT* (*Massachusetts Institute of Technology*).

O Scratch é uma linguagem de programação que possibilita a criação de histórias interativas, animações, simulações, jogos e músicas, e a partilha dessas criações na Web.

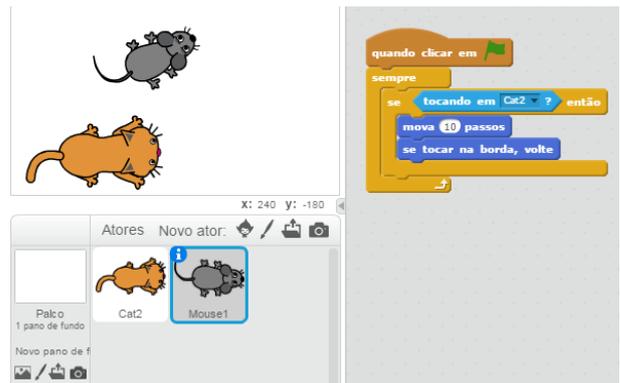


Auxilia numa aprendizagem interativa, de forma criativa, desenvolvendo a criatividade, imaginação e raciocínio.

Por não exigir o conhecimento prévio de outras linguagens de programação, ele é ideal para pessoas que estão começando a programar e foi desenvolvida para ajudar pessoas acima de 8 anos e oferece uma linguagem de programação extremamente simples para a criação de projetos que auxiliam os infantes a aprenderem e desenvolverem suas habilidades matemáticas e computacionais.

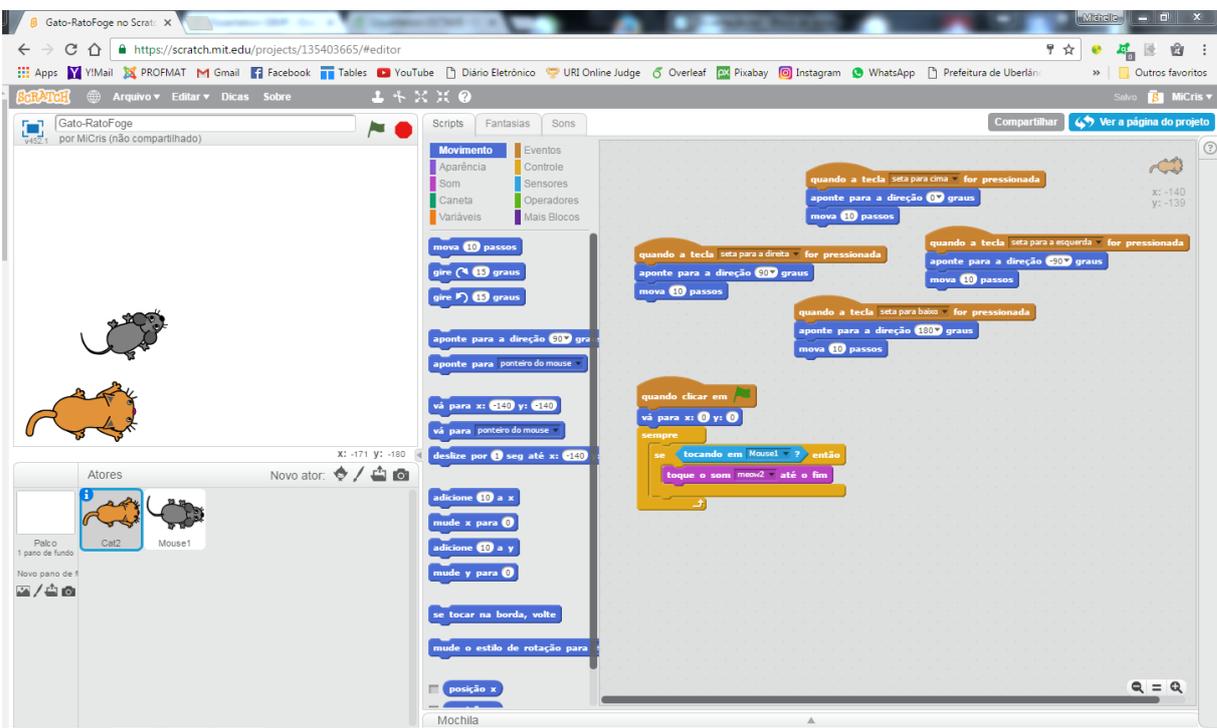
Além disso, o programa complementa e enriquece a criatividade das crianças ensinando-as a trabalhar de maneira colaborativa, como uma equipe formada com seus amigos. A plataforma online permite que usuário interajam entre si, critiquem e aprendam com os projetos dos outros (remixing).

Para escrever programas em Scratch, encaixam-se blocos gráficos uns nos outros, formando empilhamentos ordenados usando blocos de montar, lembrando o brinquedo Lego. Os blocos são feitos para se encaixarem apenas de forma que faça sentido sintaticamente, assim não há erros de sintaxe, permitindo realizar várias tarefas diferentes.

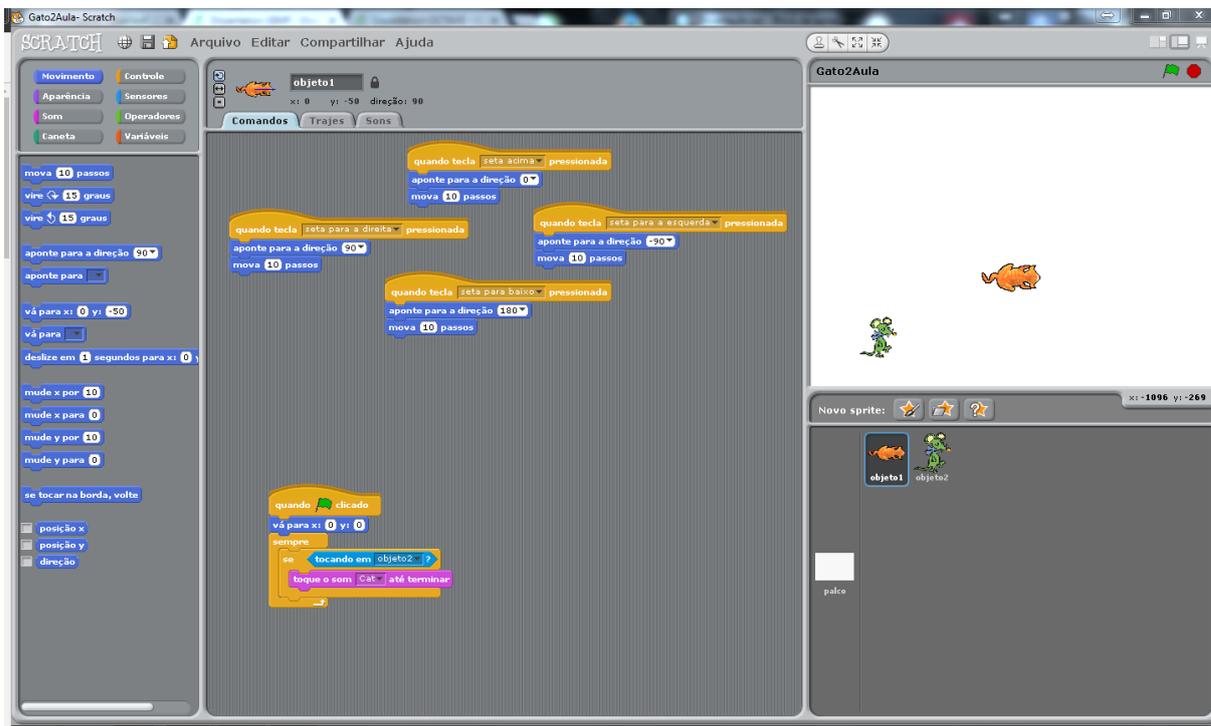


Cada bloco da linguagem contém um comando em separado, que podem ser agrupados livremente caso se encaixem. E os comandos podem ser modificados através de menus barra de snirks. O Scratch se inspirou na forma como os DJs fazem a mixagem de sons para criarem novas músicas. Mas essa linguagem consegue mixar diversos tipos de mídias, como imagens, sons e outros programas.

O programa pode ser utilizado diretamente através do editor online no site:



Ou também pode ser baixado e utilizado offline no computador:



Referências Bibliográficas

- Corrêa, J. W. L. A. (2016). Tutorial beamer. Acessado em: 15/12/2016.
- CreativeCommons (2016a). Cascading style sheets em inglês. Acessado em: 13/12/2016.
- CreativeCommons (2016b). Cascading style sheets em português. Acessado em: 13/12/2016.
- CreativeCommons (2016c). Gnu octave em inglês. Acessado em: 14/12/2016.
- CreativeCommons (2016d). Gnu octave em português. Acessado em: 14/12/2016.
- CreativeCommons (2016e). Gnu octave programming tutorial. Acessado em: 14/12/2016.
- CreativeCommons (2016f). Gnu octave wiki. Acessado em: 14/12/2016.
- CreativeCommons (2016g). Latex em inglês. Acessado em: 15/12/2016.
- CreativeCommons (2016h). Latex em português. Acessado em: 15/12/2016.
- CreativeCommons (2016i). Prezi - rapidminer. Acessado em: 15/12/2016.
- CreativeCommons (2016j). Wikipedia gimp em inglês. Acessado em: 15/12/2016.
- CreativeCommons (2016k). Wikipedia gimp em português. Acessado em: 15/12/2016.
- CreativeCommons (2016l). Wikipedia html em inglês. Acessado em: 15/12/2016.
- CreativeCommons (2016m). Wikipedia html em português. Acessado em: 15/12/2016.
- CreativeCommons (2016n). Wikipedia inkscape. Acessado em: 15/12/2016.
- CreativeCommons (2016o). Wikipedia python em inglês. Acessado em: 15/12/2016.
- CreativeCommons (2016p). Wikipedia python em português. Acessado em: 15/12/2016.
- CreativeCommons (2016q). Wikipedia rapidminer em inglês. Acessado em: 15/12/2016.
- CreativeCommons (2016r). Wikipedia scratch em inglês. Acessado em: 15/12/2016.

- CreativeCommons (2016s). Wikipedia scratch em português. Acessado em: 15/12/2016.
- CreativeCommons (2016t). Wikipediamatplotlib. Acessado em: 15/12/2016.
- da Internet no Brasil, C. G. (2016). Guia de referência css. Acessado em: 13/12/2016.
- Diversos (2016a). Css tutorial. Acessado em: 13/12/2016.
- Diversos (2016b). Gimp.org. Acessado em: 12/12/2016.
- Diversos (2016c). Inkscape org em inglês. Acessado em: 15/12/2016.
- Diversos (2016d). Inkscape org em português. Acessado em: 15/12/2016.
- Diversos (2016e). The latex project. Acessado em: 13/12/2016.
- Diversos (2016f). Matplotlib.org. Acessado em: 15/12/2016.
- Diversos (2016g). Python org docs. Acessado em: 11/12/2016.
- Diversos (2016h). Python org docs - math v.2. Acessado em: 15/12/2016.
- Diversos (2016i). Python org docs - math v.3. Acessado em: 15/12/2016.
- Diversos (2016j). Python org em português. Acessado em: 15/12/2016.
- Diversos (2016k). Python pymotw. Acessado em: 15/12/2016.
- Diversos (2016l). Pytnon org. Acessado em: 15/12/2016.
- Diversos (2016m). Rapidminer. Acessado em: 15/12/2016.
- Diversos (2016n). Stack exchange. Acessado em: 15/12/2016.
- Diversos (2016o). w3schools. Acessado em: 15/12/2016.
- Eaton, J. W. (2016). Gnu octave org. Acessado em: 14/12/2016.
- Junior, E. (2016). Curso de python. Acessado em: 15/12/2016.
- Leo Balter, B. M. e. T. Q. (2016). Aprenda layout com css. Acessado em: 13/12/2016.
- MIT (2016). Scrath edu. Acessado em: 15/12/2016.
- Paixao, V. (2016). Prezi aula inkscape. Acessado em: 15/12/2016.
- Vasconcelos, J. (2016). Prezi gimp. Acessado em: 15/12/2016.
- Ventorini, A. (2016). Prezi scratch. Acessado em: 15/12/2016.