FIAP

Bot Calculadora Científica

São Paulo, SP.

FIAP

Elias Sandri Figueiredo RM 335953

Wellington Rodrigo Nonato RM 335826

Bot Calculadora Científica

Sumário

[1. Apresentação do projeto 4](#_Toc32079668)

[1.1. Acesso ao projeto 4](#_Toc32079669)

[2. Componentes do projeto 4](#_Toc32079670)

[2.1. Framework Maven 4](#_Toc32079671)

[2.2. Framework java A.P.I for Telegram 4](#_Toc32079672)

[2.3. Framework Mx Parser. 4](#_Toc32079673)

[2.4. Framework Deriv 5](#_Toc32079674)

[3. Estrutura do projeto 5](#_Toc32079675)

[3.1. Apresentação 5](#_Toc32079676)

[3.2. Pacotes 7](#_Toc32079677)

[4. Diagrama uml do projeto 8](#_Toc32079678)

[5. Funcionalidade do bot 10](#_Toc32079679)

[5.1. Condições técnicas de funcionamento. 10](#_Toc32079680)

[5.2. Explanação do funcionamento. 10](#_Toc32079681)

[6. Explanação das equações polinomiais 14](#_Toc32079682)

# Apresentação do projeto

O projeto bot calculadora cientifica foi inspirado nos problemas e na dificuldade de aprender e similar o cálculo de nível superior.

No intuito no de ajudar os alunos de engenharia diversas, estudantes de licenciatura e entre outros, o projeto fornece resultados de integrais e derivadas com seus valores e resultado simbólico da derivada de primeira ordem.

Com mérito supracitado, nos focamos em uma interface simples que auxilie o usuário em seus compreendimentos no universo da ciência da matemática.

## Acesso ao projeto

Neste link está o projeto para acesso: <https://github.com/erius11/chat_bot_calculadora_cientifica>

# Componentes do projeto

## Framework Maven

O Maven é uma ferramenta para gerenciamento das bibliotecas e plug-ins de um ou mais repositórios de forma dinâmica utilizando um XML para descrever o projeto de software sendo construído.

Documentação do componente: <https://maven.apache.org/guides/>

## Framework java A.P.I for Telegram

O Java API for Telegram possibilita o desenvolvimento de um bot personalizado conforme a necessidade do projeto.

Website da API:

<https://github.com/pengrad/java-telegram-bot-api>

<https://core.telegram.org/bots/api>

## Framework Mx Parser.

Segue uma prevê descrição do framework:

**“O mXparser** é uma biblioteca de**analisador de expressões matemáticas super fácil**,**rica**,**rápida**e**altamente flexível** (analisador e**avaliador** de expressões / fórmulas matemáticas fornecidas como**texto sem formatação / string**). O software é fácil de usar para JAVA, Android e C # .NET / MONO (compatível com Common Language Specification: F #, Visual Basic, C ++ / CLI)”.

Com intuito no projeto, esse framework é responsável pela os cálculos com resultado de valor.

Site para documentação: <http://mathparser.org/>

## Framework Deriv

Esse framework é o responsável pelo cálculo simbólico das derivadas polinomiais, ele em vez de retornar um valor retorna uma expressão matemática da derivada de primeira ordem.

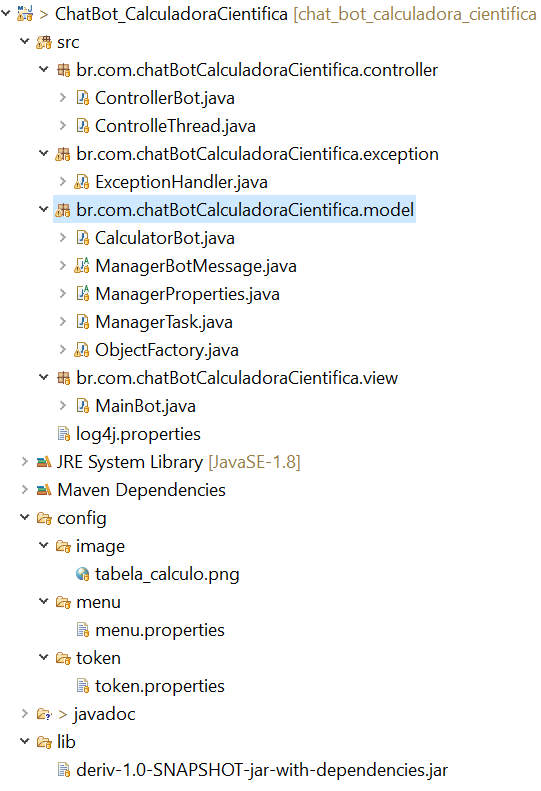
Segue uma prevê descrição do framework: “Comecei este projeto querendo fazer uma calculadora derivada simbólica que pudesse diferenciar funções arbitrárias da forma f: R ^ n -> R. E fiz isso. Mas, depois de dedicar algum tempo para refletir sobre o projeto, percebi que acabei definindo as bases para algo muito mais interessante. Por que me limitar a funções apenas da forma descrita acima? Por que não tentar diferenciar funções com valor vetorial, ou, melhor ainda, funções com valor tensorial?”

Site para documentação: <https://github.com/horeilly1101/deriv>

# Estrutura do projeto

## Apresentação

Segue abaixo uma breve apresentação da estrutura do projeto.



Na pasta **src** temos os pacotes com as classes(códigos) responsáveis pelo funcionamento do projeto. O projeto foi definido uma arquitetura com a utilização de **threads**, modelados nos seguintes designs pathers que são: MVC, Fabric e Singleton,

## Pacotes

O pacote “**br.com.chatBotCalculadoraCientifica.controller**”, tem a funcionalidade de controlar o processo, somente repassará a classe específica qual tarefa(comando) que deve ser exercido."

Pacote “**br.com.chatBotCalculadoraCientifica.model**” tem a reponsabilidade de fazer as ações necessárias do processo, esse pacote recebe sua tarefa da classe controller e desenvolve o resultado necessário.

O pacote “**br.com.chatBotCalculadoraCientifica.view**” é a camada de inicialização do projeto, nela se encontra o método Main para rodar aplicação.

Pacote “**br.com.chatBotCalculadoraCientifica.exception**” é a camada das exceções, caso alguma thread sinalize algum erro, essa camada irá trata-lo.

A pasta **config** contempla todas as configurações do projeto. Nela consta os subníveis:

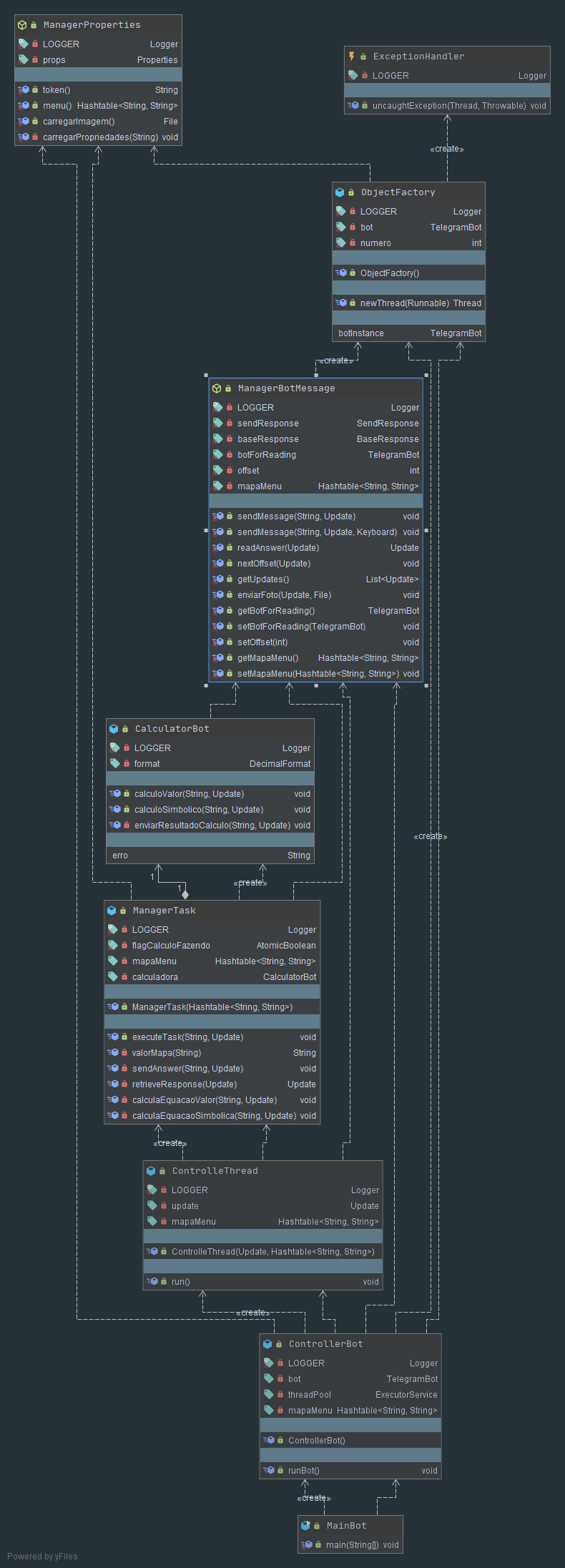
* Image: local aonde está a imagem das expressões matemáticas;
* Menu: local do arquivo de propriedades do menu, nele tem as descrições dos comandos do menu do chat bot;
* Token: chave de inicialização da tecnologia do TelegramBot, arquivo de propriedade com o id(chave) do bot específico;

Pasta **javadoc** contém toda a documentação que explicita o funcionamento do projeto.

A pasta **lib** abrangeo jar do framework deriv, para mais informações ver capítulo componentes do projeto.

# Diagrama uml do projeto





# Funcionalidade do bot

## Condições técnicas de funcionamento.

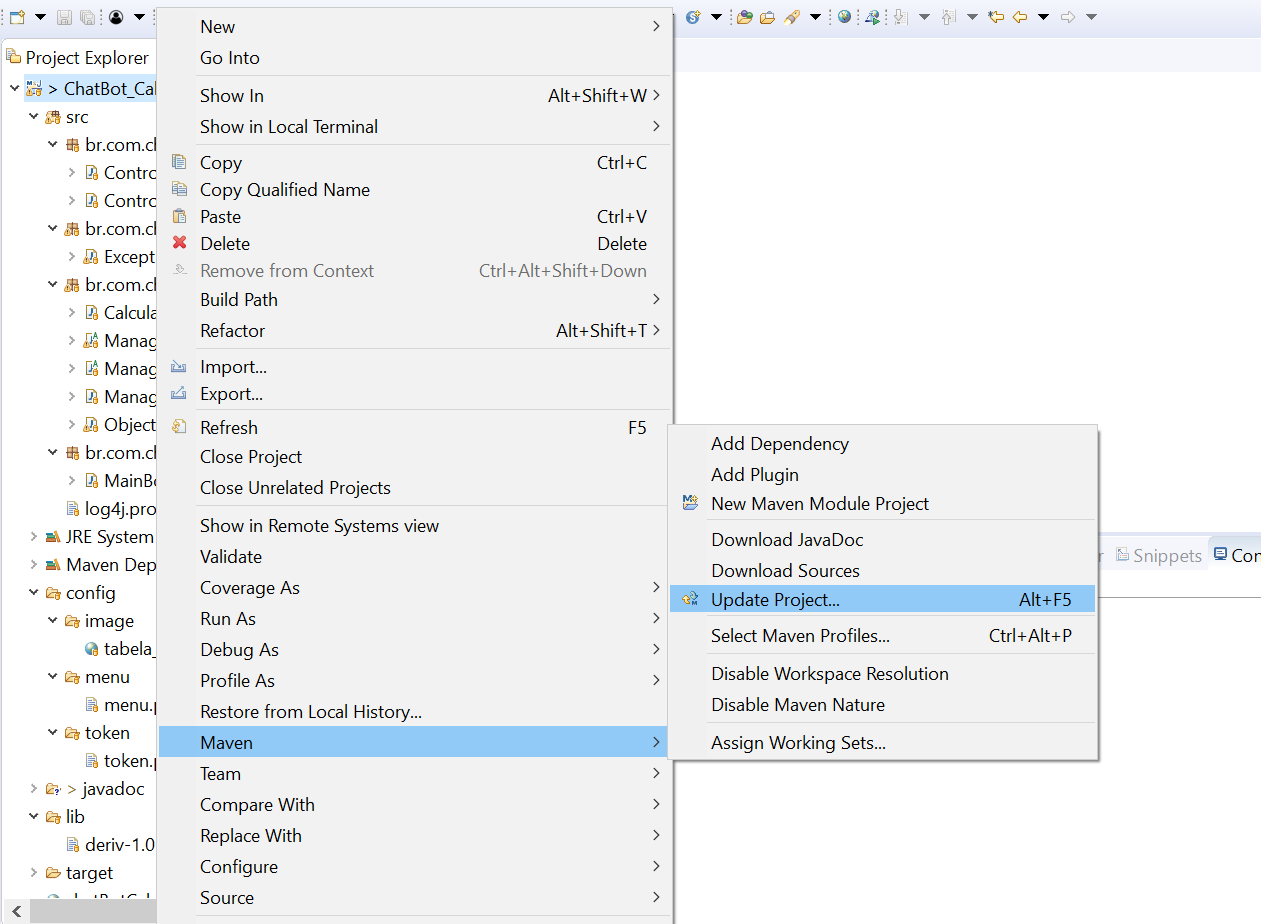
As tecnologias necessárias para rodar o projeto são:

* Para executar o projeto é necessário ter o JRE 8 ou JDK 8 instalado;
* Utilizar uma ide para execução. (Eclipse é recomendado);
* Maven instalado na máquina para build do projeto e atualização do repositório local.

Para iniciar a aplicação é necessário inicialmente fazer o clone do projeto do repositório git, mais informações ver capítulo de acesso ao projeto. Depois do clone, exportar para o eclipse (IDE do seu interesse) o projeto e esperar o build da sua área de trabalho.

Caso de algum erro no build, fazer os seguintes passos:

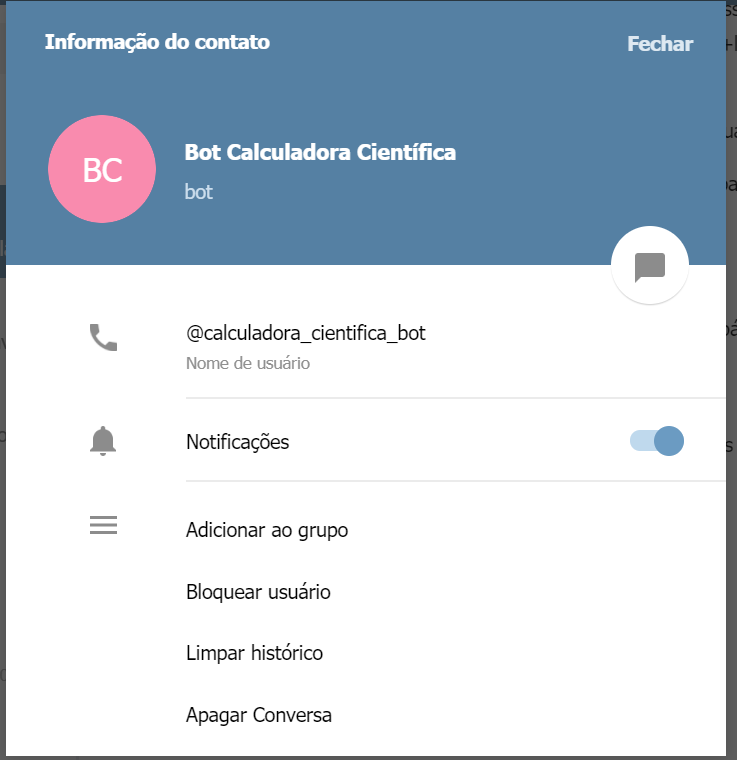
* Fazer um update do projeto (conforme imagem abaixo), esse procedimento irá atualizar seu repositório com as dependências do projeto.



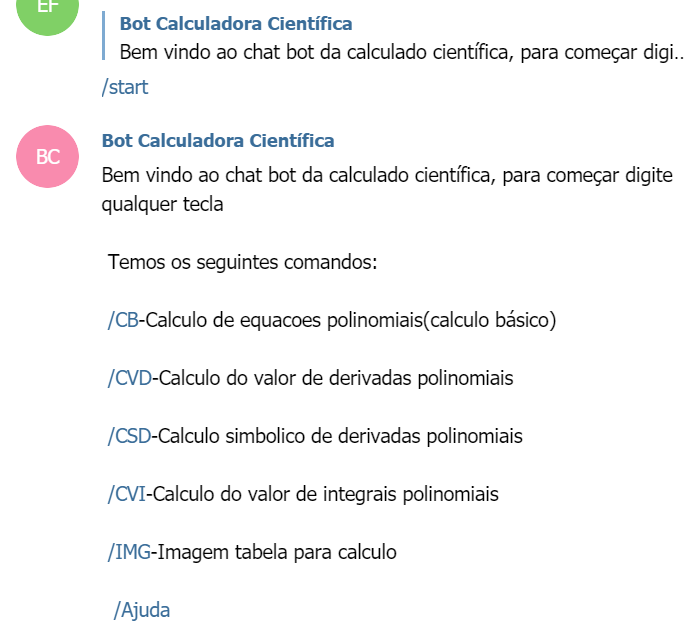
* Depois abrir o terminal e nas pasta dentro do projeto( onde está o arquivo pom.xml) fazer o seguinte comando: mvn clean install -U.

## Explanação do funcionamento.

O funcionamento do bot será explano nesse capítulo, inicialmente temos os dados para encontrar o mesmo no telegrama.

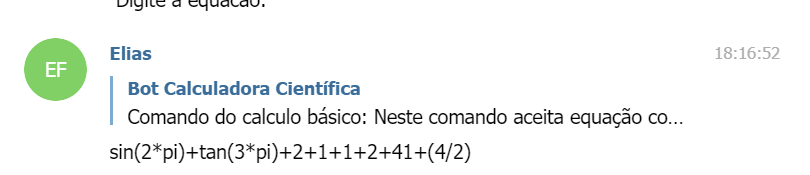


Quando a aplicação estiver iniciada, na tela do bot será necessário digitar qualquer palavra ou /start, esse comando devolverá a tela inicial de menu.

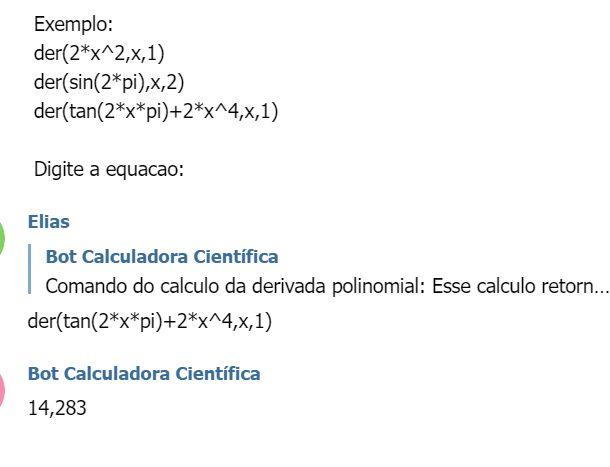


O menu supracitado contempla comandos de cálculo, imagem e ajuda, sendo explanados abaixo:

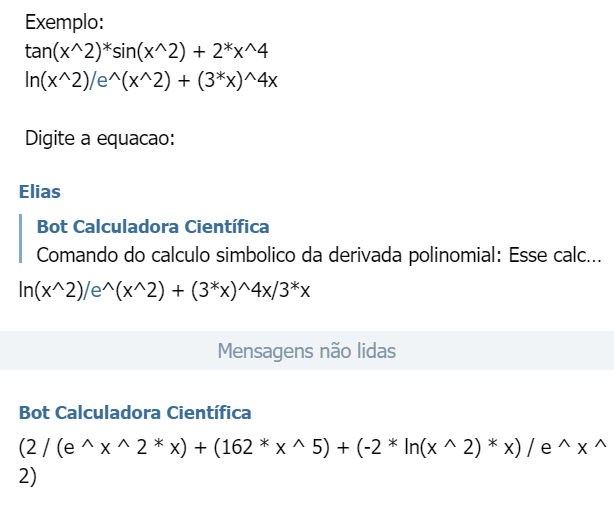
* Comando **/CB** é o comando para cálculos diversos, nele é necessário digitar a equação do cálculo desejado, conforme exemplo;



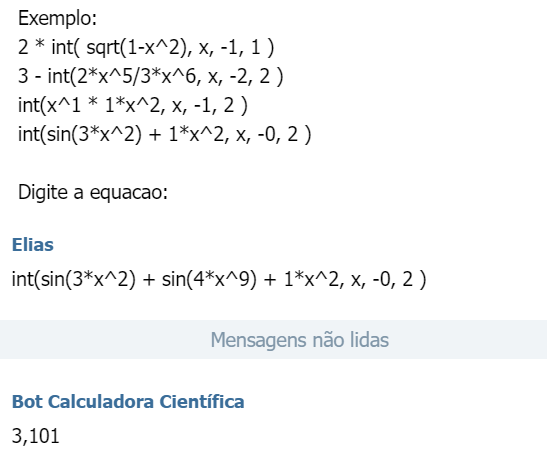
* O comando **/CVD** faz o cálculo de derivadas polinomiais, com intuito de devolver o valor da derivada de primeira ordem.
* Para esse comando é necessário digitar a palavra **der**(derivada), nisso a equação que deseja derivar, depois a variável que será derivada e finalmente o valor da variável, conforme exemplo:



* Comando **/CSD** faz o calculo da derivada simbólica de primeira ordem, ou seja ele retorna a primeira equação derivada. A equação deve ser digita em função da variável x, conforme exemplo:



* Comando **/CVI** tem o foco no cálculo de integrais polinomiais, retornando o valor da integral desejada. A equação necessita ser digitada com a palavra **int**(integral) na frente, depois ‘, x,' valor do limite máximo e o valor do limite mínimo. Exemplo:



* Comando **/IMG**, retornará uma imagem com as expressões matemática para calcular derivadas e integrais.
* Comando **/Ajuda** é uma breve explanação de como usar as funções existente na matemática, item que iremos abordar no capítulo seguinte:

# Explanação das equações polinomiais

Para efetuar os cálculos no bot é necessário digitar a equação a ser calculada, ele aceita as seguintes funções:

* 2+2-Soma;
* 2-2-Subtração;
* 2\*2-Multiplicação;
* 2/2-Divisão;
* 2^2-Exponenciação;
* 2^2-Exponenciação;
* sin(x)-Função seno;
* cos(x)-Função cosseno;
* tan(x)-Função tangente;
* log2(4)-Função logaritma na base 2;
* log10(100)-Função logaritma na base 10;
* sqrt(2)-Função da raiz 2;

As funções trigonométricas podem ser feitas em graus ou radianos, mas para uma precisão melhor digitá-las em radianos. O radiano deverá ser digitado como **pi**, conforme exemplo:

* sem(2\*pi), ou cos(3\*pi/2), ou tan(pi).

Segue uma imagem das conversões de graus para radianos.

