

## SUMÁRIO - AULA 02

- ❑ Histórico e Motivação, Contextualização
- ❑ Paradigmas
  - ❑ O que são?
  - ❑ Importância
- ❑ Atividade/debate
  - ❑ Quanto vale/custa um algoritmo?



## SOBRE O INVENTÁRIO DA TURMA



- ❑ **Python e Javascript**, muito comentadas (aprender)
- ❑ **Futuro**
  - ❑ cada vez mais automatizado, simplificado
  - ❑ o mundo está cada vez mais usando programação pra resolver problemas, o surgimento de novas linguagens e criação de aplicações crescerá tornando necessário a aprimoramento de suas habilidades como programador.
  - ❑ web e mobile, com linguagens de programação web ganhando cada vez mais espaço.

## Me learning programing languages



66

## SOBRE O INVENTÁRIO DA TURMA

- ❑ Não sabem outra forma de desenvolver software a não ser codificar
- ❑ linguagem **perfeita**: que se programa sozinha
- ❑ **restrição** imposta por LP:
  - ❑ não poder usar variável global
  - ❑ não poder sobrecarregar métodos em C
  - ❑ função em C para retornar mais que um valor

67

## HISTÓRICO E MOTIVAÇÃO

❑ Viver é a arte de **resolver problemas...**

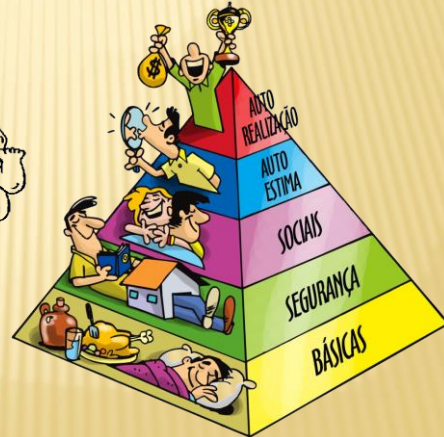
❑ Para sobreviver:

- ❑ Comer
- ❑ Se vestir
- ❑ Dormir



❑ Outros:

- ❑ Se locomover
- ❑ Estudar
- ❑ Etc.



Pirâmide de Maslow

## HISTÓRICO E MOTIVAÇÃO

❑ Além dos problemas **perceptíveis** (concretos) nos deparamos com **outros** problemas:

❑ Quanto tenho que **pagar** de ICMS?

❑ Minha loja está gerando **lucro**?

❑ Qual a minha **média** na disciplina X?

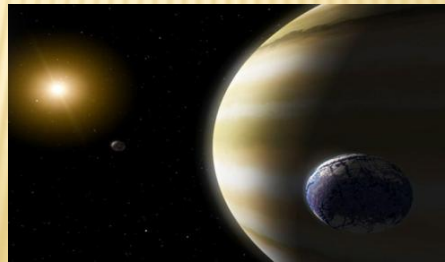




## HISTÓRICO E MOTIVAÇÃO



- ❑ **Charles Babbage** (1820), o “pai” da Computação”, irritou-se com a maneira **mecânica** e **repetitiva** com que os **cálculos** da posição de estrelas e planetas eram feitos na *Royal Astronomical Society*, onde trabalhava.

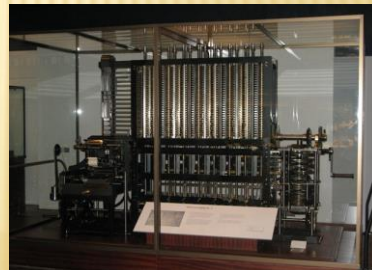


70

## HISTÓRICO E MOTIVAÇÃO



- ❑ Ele então escreveu o artigo “*Observations on the Application of Machinery to the Computatiuon of Mathematical Tables*”, onde **propunha** que estes algoritmos fossem realizados por **máquinas**.



Máquina Diferencial de Babbage

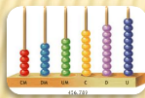
71

## HISTÓRICO E MOTIVAÇÃO

- Computação visa **ampliar** a **capacidade** do homem em **resolver** problemas.



- Delegar **tarefas repetitivas** às máquinas.

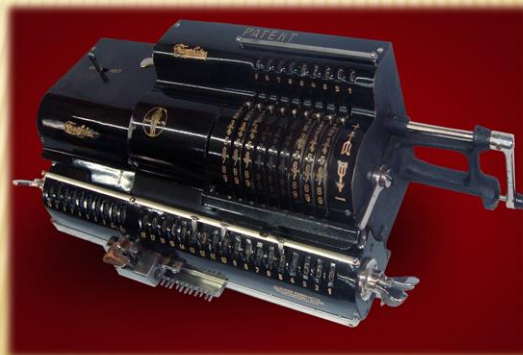


72

## HISTÓRICO E MOTIVAÇÃO

- Primeiras máquinas eram “**engessadas**” e só resolviam problemas **específicos**:

- Calculadora mecânica



73

## HISTÓRICO E MOTIVAÇÃO

- ❑ Ao longo do tempo as máquinas foram tornando-se mais **generalistas** a ponto de **executarem instruções programáveis** via:
  - ❑ Alavancas, botões, linguagem de máquina
    - ❑ Aí é que surgem os **computadores** e os **softwares**.
  - ❑ Nossa **salvação** para a realização de algumas atividades...
  - ❑ ...mas criando a necessidade de ter que **aprender** a usá-los.

74

## ELABORAÇÃO

- ❑ E o que isso tem a ver com **Computação**?
  - ❑ Hoje resolvemos **problemas** do mundo real escrevendo **instruções** (codificando algoritmos em alguma linguagem de programação) que após **compiladas** são **executadas** pelos circuitos lógicos dos **computadores**.





## ELABORAÇÃO

- E o que isso tem a ver com **Computação**?
- Sim, escrevemos **textos coloridos**, nada poéticos e muito **melindrosos** que até a falta de qualquer ponto-e-vírgula faz diferença...



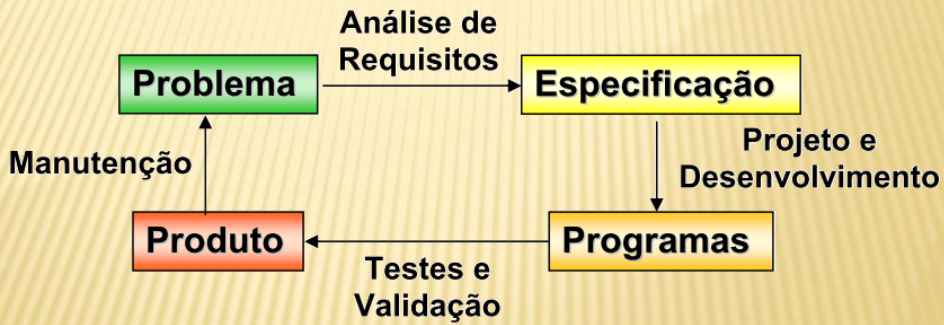
## CONTEXTUALIZAÇÃO

- Fases no desenvolvimento de sistemas:



## CONTEXTUALIZAÇÃO

- Fases no desenvolvimento de sistemas:



79

## CONTEXTUALIZAÇÃO

- Desenvolver **software** é uma tarefa:

- Complexa
- Perene...
- Repetitiva
- Intelectual e artística



80



## CONTEXTUALIZAÇÃO

- ❑ Várias **camadas** (linguagens) foram criadas para se lidar com a **complexidade**:
  - ❑ Desde *Assembly* até Genexus<sup>1</sup>, BPM, etc...



- ❑ Reformulando:

- ❑ “A computação é um jardim de cebolas.”

<sup>1</sup> <http://www.genexus.com/global/inicio?pt>



## ENTENDENDO O PROCESSO DE COMPILAÇÃO

- ❑ Linguagem de **Alto Nível** (Java, C++)

- ❑ `total = parcela + juro;` *ok, entendi!*

Sequência de  
códigos simbólicos

- ❑ Linguagem **Assembly** (montagem)

- ❑ `r1 = a[16] + a[32]` *!?!?!?!?*

- ❑ Linguagem de **Máquina**

- ❑ `01110011101001` *??????*



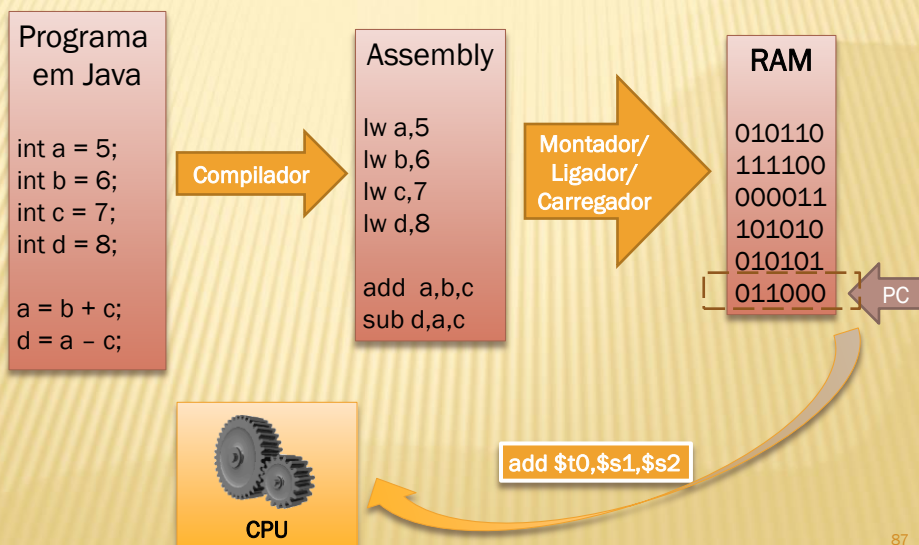
## COMO OS SOFTWARES FUNCIONAM?

- Linguagens de **alto nível** permitem ao **programador** pensar em termos da **estrutura** do **problema** (for, while, if ?) a ser resolvido ao invés da **estrutura** do **computador**.



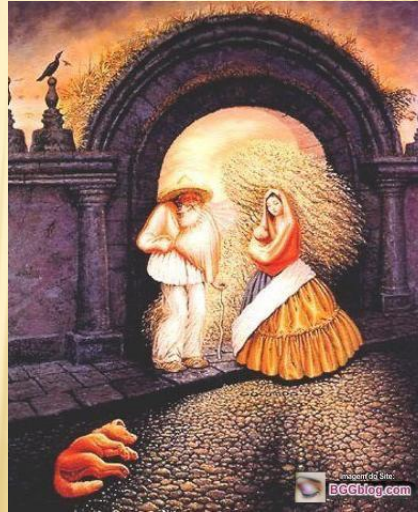
85

## COMO OS SOFTWARES FUNCIONAM?



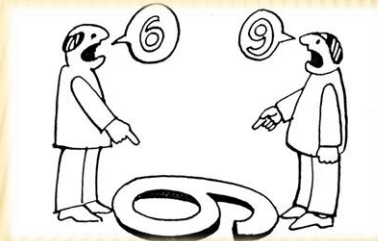
87

## O QUE VOCÊS ENXERGAM AQUI?



## O QUE É UM PARADIGMA?

- Um paradigma é o que determina o **ponto de vista** da realidade e como se **atua** sobre ela.

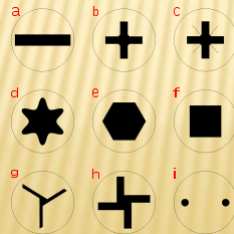


- Cada qual determina uma **forma particular de abordar** os **problemas** e de formular respectivas **soluções**.



## O QUE É UM PARADIGMA?

- Uma linguagem de programação pode **combinar dois** ou **mais** paradigmas para **potencializar** as análises e soluções.
- Devemos **escolher** o paradigma **mais adequado** para analisar e resolver cada **problema**.



## HELLO WORLD ORIENTADO A OBJETOS

```
package Hello;

public class HelloWorld{

    private String mensagem;

    public HelloWorld(String mensagem){
        this.mensagem = mensagem;
    }

    public void show() {
        System.out.println("mensagem: " + mensagem);
    }

    public static void main(String [] args){
        HelloWorld obj = new HelloWorld("ola mundo");
        obj.show();
    }
}
```

gambiarra\_mode = ON

## O QUE É UM PARADIGMA?

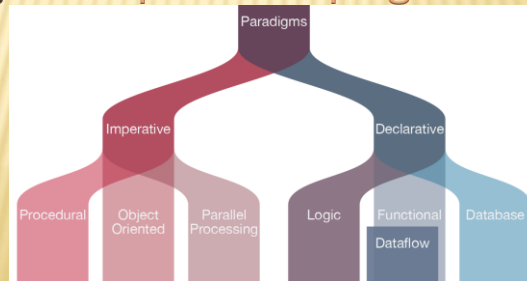
- ❑ É um **modelo**.
- ❑ É a representação de um **padrão** a ser seguido.
- ❑ É uma **teoria**, um **conhecimento** que origina o estudo de um campo científico.

## QUEBRAR PARADIGMAS...



## O QUE É UM PARADIGMA DE PROGRAMAÇÃO?

- ❑ Modelo, padrão ou **estilo de programação** suportado por linguagens que agrupam certas **características comuns**.
- ❑ Forma geral de **pensar ao programar**.



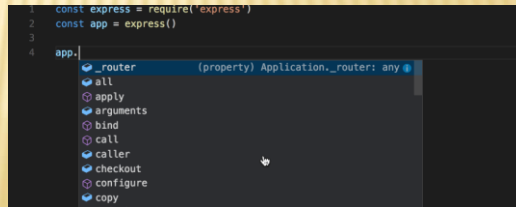
## O QUE É UM PARADIGMA DE PROGRAMAÇÃO?

- ❑ Linguagens seguem/suportam **um ou mais** paradigmas. Ex:
  - ❑ Java: imperativa + OO
- ❑ Há **centenas** de linguagens, mas **poucos** paradigmas.
  - ❑ Elas é que se encaixam nos paradigmas



## O QUE É IMPORTANTE PARA O PROGRAMADOR?

- ❑ Boa **lógica de programação**;
- ❑ Compreensão dos **conceitos** fundamentais das LPs
- ❑ **Ambiente** com suporte adequado



## POR QUÊ?

- ❑ Quais as **razões** para se estudar os paradigmas de programação?
  - 1) Aumentar a capacidade de expressar ideias
  - 2) Aumentar o conhecimento para escolher linguagens de programação apropriadas
  - 3) Melhorar a habilidade de aprender novas linguagens

## POR QUÊ?

- ❑ Quais as **razões** para se estudar os paradigmas de programação?
- 4) Melhorar o entendimento da importância da implementação
- 5) Melhorar a habilidade de projetar novas linguagens
- 6) Avanço da Computação como um todo

## 1) AUMENTAR A CAPACIDADE DE EXPRESSAR IDEIAS

- ❑ A profundidade com que **pensamos** é influenciada pelo **poder expressivo** da linguagem com a qual nos **comunicamos**;
  - ❑ Pensamos em **português**?
  - ❑ E quando ainda **não** sabemos falar?
  - ❑ Apenas expressamos o pensamento em português?



## 1) AUMENTAR A CAPACIDADE DE EXPRESSAR IDEIAS

❑ É difícil para as pessoas **expressar estruturas** que elas **não conseguem descrever**;

❑ **Programadores sofrem** muito com este problema!

❑ Ex: esta aula, como modelar ela?

❑ Arrays, matrizes?



## 2) AUMENTAR O CONHECIMENTO PARA ESCOLHER LINGUAGENS DE PROGRAMAÇÃO APROPRIADAS

❑ Programadores tendem **a continuar** usando a linguagem que lhes é **mais familiar**, mesmo que esta **não seja** a mais adequada.



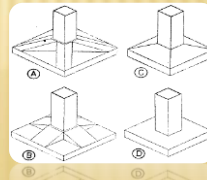
❑ **Conhecer** características de **outras** linguagens permite fazer **escolhas** adequadas para desenvolvimento de **novos** projetos.



### 3) MELHORAR A HABILIDADE DE APRENDER NOVAS LINGUAGENS

- Computação está em **evolução** contínua
- Conhecer conceitos fundamentais de LPs permite **aprender** mais facilmente **novas** linguagens.

- A essência é a mesma...



### 4) MELHORAR O ENTENDIMENTO DA IMPORTÂNCIA DA IMPLEMENTAÇÃO

- Entender melhor o **projeto** de implementação de uma LP pode **otimizar** o uso de seus **recursos**.

- Ex: LISP é útil com listas!



- Permite usar a linguagem mais **inteligentemente**, pois estará avaliando as **consequências** de suas decisões.

## 5) MELHORAR A HABILIDADE DE PROJETAR NOVAS LINGUAGENS

- ❑ Talvez um dia você precisará **projetar** uma linguagem.
- ❑ Entender o projeto de uma LP pode **facilitar** o **planejamento** de projetos de sistemas complexos.
  - ❑ Grande desafio implementar uma LP.

## 6) AVANÇO DA COMPUTAÇÃO COMO UM TODO

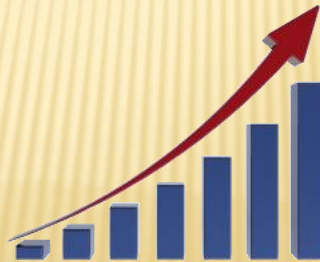
- ❑ Nem sempre a linguagem **mais utilizada** é a **melhor...**



- ❑ Se as pessoas responsáveis pela escolha de LPs forem **melhor informadas**, talvez **linguagens melhores** sejam utilizadas.

## 6) AVANÇO DA COMPUTAÇÃO COMO UM TODO

- ❑ Saber avaliar o projeto de LPs pode permitir que a escolha seja feita com base em **fundamentos sólidos** e não somente na **popularidade** de uma LP.



## QUEM?

- ❑ E qual que é a linguagem **mais popular**?





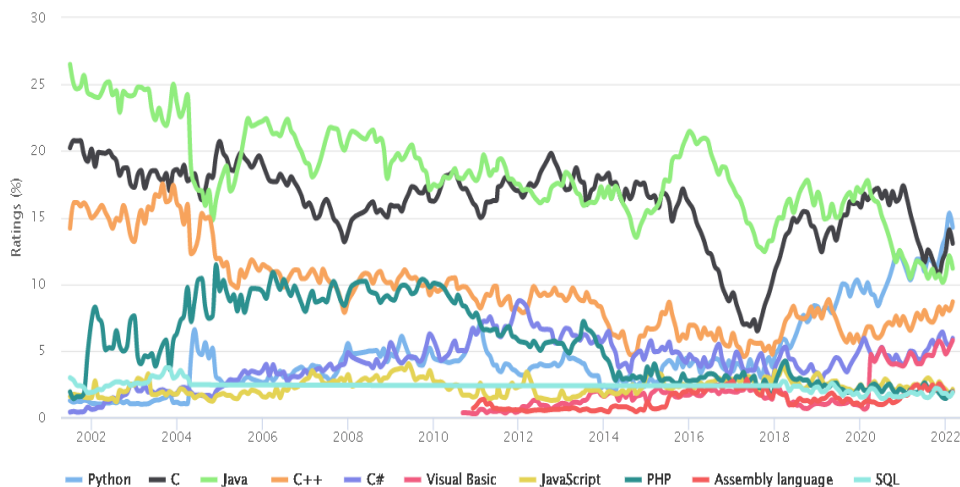
## TIOBE PROGRAMMING COMMUNITY INDEX

- ❑ Indicador da popularidade de linguagens de programação.
  - ❑ atualizado uma vez por mês.
- ❑ Classificações baseadas:
  - ❑ número de **engenheiros** qualificados no mundo
  - ❑ **cursos** e fornecedores terceirizados
  - ❑ mecanismos de **busca**
    - ❑ Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube e Baidu
- ❑ **não é sobre a melhor** linguagem de programação
- ❑ nem em qual linguagem foram **escritas** mais linhas de código.

## TIOBE PROGRAMMING COMMUNITY INDEX





















TIOBE Programming Community Index

Source: [www.tiobe.com](http://www.tiobe.com)



fonte: <https://www.tiobe.com/tiobe-index>

fonte: <https://www.tiobe.com/tiobe-index>

	Mar 2022	Mar 2021	Change	Programming Language	Ratings	Change
1	3		▲	 Python	14.26%	+3.95%
2	1		▼	 C	13.06%	-2.27%
3	2		▼	 Java	11.19%	+0.74%
4	4			 C++	8.66%	+2.14%
5	5			 C#	5.92%	+0.95%
6	6			 Visual Basic	5.77%	+0.91%
7	7			 JavaScript	2.09%	-0.03%
8	8			 PHP	1.92%	-0.15%
9	9			 Assembly language	1.90%	-0.07%
10	10			 SQL	1.85%	-0.02%
11	13		▲	 R	1.37%	+0.12%
12	14		▲	 Delphi/Object Pascal	1.12%	-0.07%
13	11		▼	 Go	0.98%	-0.33%
14	19		▲	 Swift	0.90%	-0.05%
15	18		▲	 MATLAB	0.80%	-0.23%
16	16			 Ruby	0.66%	-0.52%
17	12		▼	 Classic Visual Basic	0.60%	-0.66%
18	20		▲	 Objective-C	0.59%	-0.31%
19	17		▼	 Perl	0.57%	-0.58%
20	38		▲	 Lua	0.56%	+0.23%

## LINGUAGEM DE PROGRAMAÇÃO



- ❑ Linguagens de programação são usadas para **descrever algoritmos**
  - ❑ Sequencias de passos que levam à solução de um problema.
- ❑ Precisam suportar a definição de **ações** e prover meios para especificar **operações básicas** de computação

# LINGUAGEM DE PROGRAMAÇÃO



- Permite que os usuários **especifiquem** como estes passos devem ser **sequenciados** para resolver um problema.
- **Notação** que pode ser usada para **especificar algoritmos** com precisão.



## HELLO WORLD

```

1 print("Hello World!")

```

```

//COBUCLG JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//HELWRLD EXEC COBUCLG,PARM=COB='MAP,LIST,LET'
//COB.SYSIN DD *
001 IDENTIFICATION DIVISION.
002 PROGRAM-ID. 'HELLO'.
003 ENVIRONMENT DIVISION.
004 CONFIGURATION SECTION.
005 SOURCE-COMPUTER. IBM-360.
006 OBJECT-COMPUTER. IBM-360.
006S SPECIAL-NAMES.
0066 CONSOLE IS CNSL.
007 DATA DIVISION.
008 WORKING-STORAGE SECTION.
009 77 HELLO-CONST PIC X(12) VALUE 'HELLO, WORLD'.
075 PROCEDURE DIVISION.
090 000-DISPLAY.
100 DISPLAY HELLO-CONST UPON CNSL.
110 STOP RUN.
//LKED.SYSLIB DD DSN=SYS1.COBLIB,DISP=SHR
// DD DSN=SYS1.LINKLIB,DISP=SHR
//GO.SYSPRINT DD SYSOUT=A
//

```

```

1 #include <stdio.h>
2
3 main(){
4     printf("hello world");
5 }

```

```

/*
Java Example for 'hello world!'
daveti
*/
import java.lang.*;

public class HelloWorld {
    public static void main( String[] args) {
        System.out.println( "hello world!");
    }
}

```

113



## SEM CÓDIGO?!

- ❑ Microsoft “Project Siena”
  - ❑ Aplicativos acessando fonte de dados mas sem programação
    - ❑ <https://www.youtube.com/watch?v=e4BB0mGCulk>
    - ❑ <https://www.youtube.com/watch?v=dcQcLGy2hSO>
- ❑ Gratuito
- ❑ No Windows 8



## SEM CÓDIGO?!

- ❑ GeneXus
  - ❑ <http://www.genexus.com/produtos/genexus?pt>
  - ❑ Modelando o sistema desejado, ele cria:
    - ❑ o banco de dados
    - ❑ o código dos aplicativos
    - ❑ a interface do usuário para o cliente
    - ❑ e os serviços necessários para o servidor
  - ❑ Vídeos
    - ❑ <https://www.youtube.com/watch?v=nONdyzCyKoE>
    - ❑ <https://www.youtube.com/watch?v=jngLbOZJce8>

**GeneXus™**

## ATIVIDADE

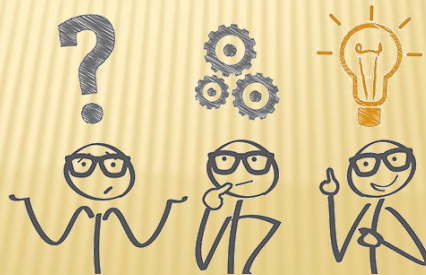


- ❑ Quanto **vale** um **algoritmo**?
- ❑ <http://www.duopo.com.br/blog/a-tecnologia-aliada-a-matematica-e-a-inteligencia-de-mercado-ajuda-muito>
- ❑ Leia o artigo. Comente no vídeo da aula.
- ❑ **United Parcel Service** é uma das maiores empresas de logística do mundo, distribuindo diariamente mais de 14 milhões de encomendas em mais de 200 países

117

## ATIVIDADE

- ❑ Agora imagine outros **problemas reais** que um bom **algoritmo** resolveria trazendo benefícios **enormes** para a sociedade.
- ❑ Ele poderia ser criado? O que falta?
  - ❑ Debata com **seus colegas** e depois **na turma**.



118

## RESPOSTA

### Química combinatória

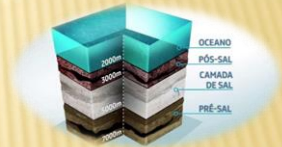
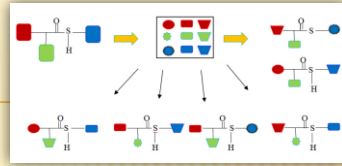
#### Combinatorial chemistry

[https://franz.com/success/customer\\_apps/bioinformatics/mdl\\_store.y.html](https://franz.com/success/customer_apps/bioinformatics/mdl_store.y.html)

### Petrobrás e Pré-Sal

#### Big Data e Inteligência Artificial

<https://revistaoeste.com/economia/inteligencia-artificial-permitiu-nova-descoberta-no-pre-sal-informa-petrobras>



### Calendário de Aulas

Segunda-feira				Terça-feira			
1	7/3/2022	M	QUIM - QUIM - LPOD - LPOD	8/3/2022	M	MATEI - MATEI - LPOD - EDPD - EDPD	8/3/2022
2	14/3/2022	T	QUIM - QUIM - LPOD - LPOD	15/3/2022	T	MATEI - MATEI - LPOD - EDPD - EDPD	15/3/2022
3	21/3/2022	M	QUIM - QUIM - LPOD - LPOD	22/3/2022	M	MATEI - MATEI - LPOD - EDPD - EDPD	22/3/2022
4	28/3/2022	T	QUIM - QUIM - LPOD - LPOD	29/3/2022	T	MATEI - MATEI - LPOD - EDPD - EDPD	29/3/2022
5	4/4/2022	M	QUIM - QUIM - LPOD - LPOD	5/4/2022	M	MATEI - MATEI - LPOD - EDPD - EDPD	5/4/2022
6	11/4/2022	T	QUIM - QUIM - LPOD - LPOD	12/4/2022	T	MATEI - MATEI - LPOD - EDPD - EDPD	12/4/2022
7	18/4/2022	M	QUIM - QUIM - LPOD - LPOD	19/4/2022	M	MATEI - MATEI - LPOD - EDPD - EDPD	19/4/2022
8	25/4/2022	T	QUIM - QUIM - LPOD - LPOD	26/4/2022	T	MATEI - MATEI - LPOD - EDPD - EDPD	26/4/2022
9	2/5/2022	M	QUIM - QUIM - LPOD - LPOD	3/5/2022	M	MATEI - MATEI - LPOD - EDPD - EDPD	3/5/2022
10	9/5/2022	T	QUIM - QUIM - LPOD - LPOD	10/5/2022	T	MATEI - MATEI - LPOD - EDPD - EDPD	10/5/2022

## ATIVIDADE

### Pesquisa sobre a Máquina Analítica e a Máquina Diferencial de Babbage

