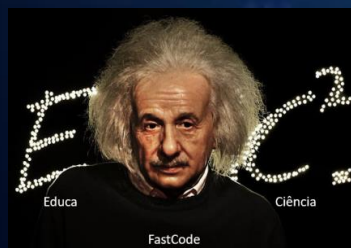


Java: Entendendo Extends & Implements

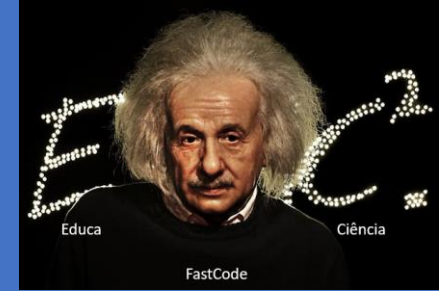
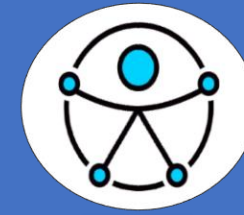


2022

Fábio Perucello



Palestrante



+

JAVA: ENTENDENDO EXTENDS E IMPLEMENTS

<

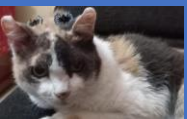
Palestrante: Fábio Perucello

Trailer

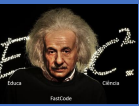
Trailer

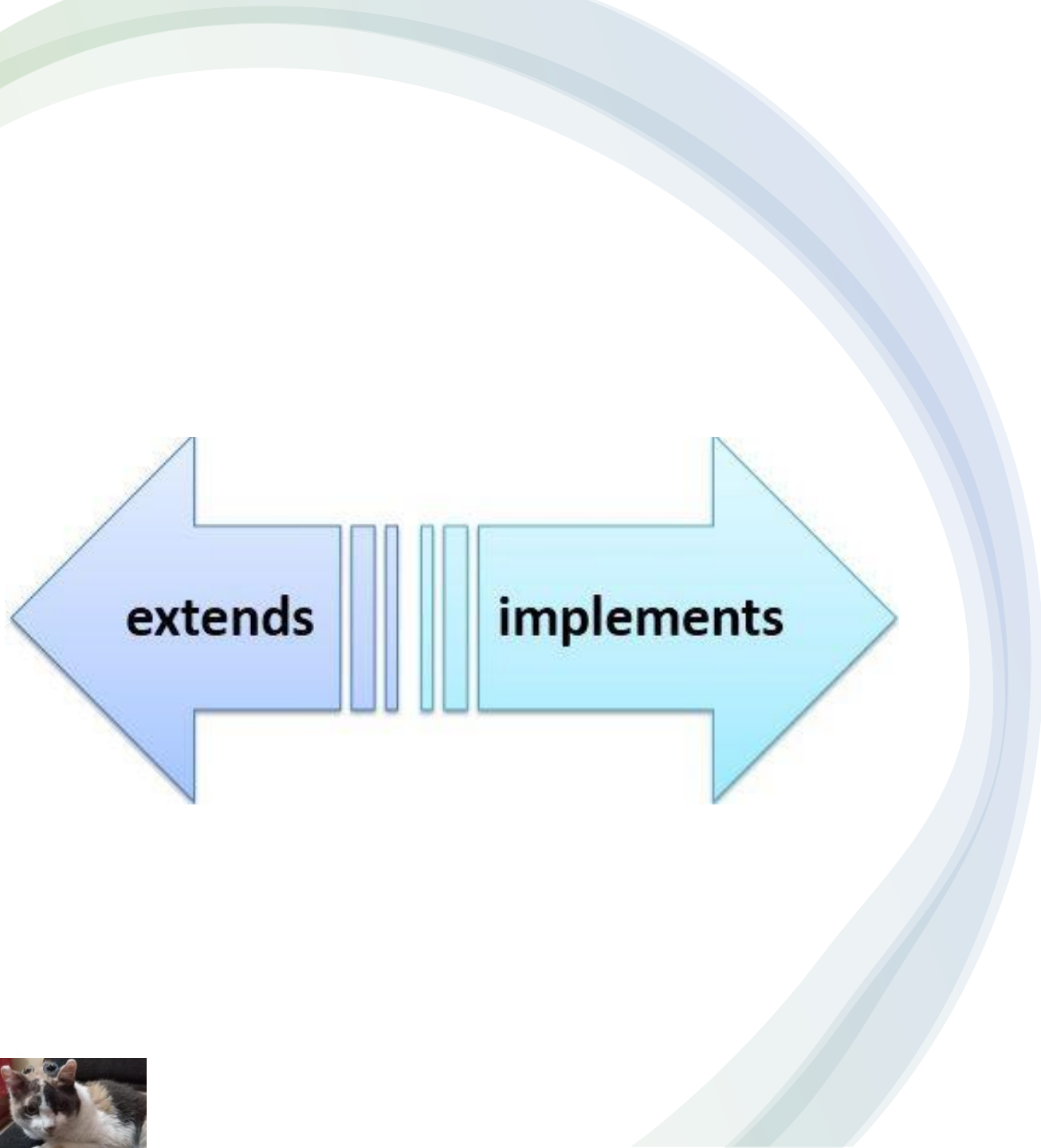
EXPLORANDO TI

+

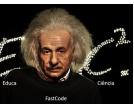


Fábio Perucello





Vamos falar de Extends
e Implements ?

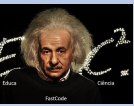
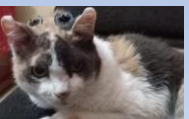


Java: Extends

Extends – podemos resumir como “herança”, ocorre principalmente quando você quer que uma determinada “classe” estenda as suas funcionalidades ou seja, você estende a funcionalidade original para outra classe.

A “herança” é um dos elementos mais importantes da Programação orientada à objetos, pois permite a criação de classificação hierárquica.

Nesse cenário é que você pode criar uma classe genérica em que serão definidos os traços mais comuns entre elas, depois criar uma classe “específica” em que receberá como herança “estes” traços criados.



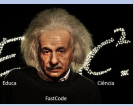
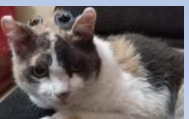
Java: Extends

SUPERCLASSE – é a classe herdada

SUBCLASSE – é a classe que faz a “herança”

Podemos então afirmar que uma SUBCLASSE é uma versão especializada de uma SUPERCLASSE.

Tendo este conceito, para herdar uma classe, basta incorporar a definição de uma classe à outra através da palavra chave “EXTENDS”



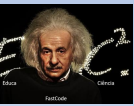
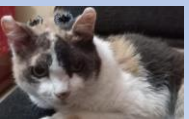
Java: Extends

Exemplo:

```
Veiculo_Caracteristicas.java
1 package com.educaciencia.usingExtends;
2
3 public class Veiculo_Caracteristicas {
4
5     /** Caracteristicas Comuns */
6     public String nome;
7     public String fabricante;
8     public String cor;
9     public String motor;
10    public String combustivel;
11
12    /** Metodos Contrutores */
13    public Veiculo_Caracteristicas() {
14        super();
15    }
16
17    public Veiculo_Caracteristicas(String nome, String fabricante, String cor, String motor, String combustivel) {
18        super();
19        this.nome = nome;
20        this.fabricante = fabricante;
21        this.cor = cor;
22        this.motor = motor;
23        this.combustivel = combustivel;
24    }
25
26    /** Metodos Getters Setters */
27    public String getNome() {
28        return nome;
29    }
30
31    public void setNome(String nome) {
32        this.nome = nome;
33    }
34
35    public String getFabricante() {
36        return fabricante;
37    }
38
39    public void setFabricante(String fabricante) {
40        this.fabricante = fabricante;
41    }
42
43    public String getCor() {
44        return cor;
45    }
46
47    public void setCor(String cor) {
48        this.cor = cor;
49    }
50
51    public String getMotor() {
52        return motor;
53    }
54
55    public void setMotor(String motor) {
56        this.motor = motor;
57    }
58
59    public String getCombustivel() {
60        return combustivel;
61    }
62
63    public void setCombustivel(String combustivel) {
64        this.combustivel = combustivel;
65    }
66
67    public String toString() {
68        return "Veiculo_Caracteristicas [nome=" + nome + ", fabricante=" + fabricante + ", cor=" + cor + ", motor=" + motor + ", combustivel=" + combustivel + "]";
69    }
70
71 }
```

```
Veiculo_de_Passeio.java
1 package com.educaciencia.usingExtends;
2
3 public class Veiculo_de_Passeio extends Veiculo_Caracteristicas {
4
5     public String portas;
6     public String lugares;
7
8     public Veiculo_de_Passeio() {
9         super();
10    }
11
12    public Veiculo_de_Passeio(String nome, String fabricante, String cor, String motor, String combustivel) {
13        super(nome, fabricante, cor, motor, combustivel);
14    }
15
16    public Veiculo_de_Passeio(String nome, String fabricante, String cor, String motor, String combustivel, String portas, String lugares) {
17        super(nome, fabricante, cor, motor, combustivel);
18        this.portas = portas;
19        this.lugares = lugares;
20    }
21
22    public String getPortas() {
23        return portas;
24    }
25
26    public void setPortas(String portas) {
27        this.portas = portas;
28    }
29
30    public String getLugares() {
31        return lugares;
32    }
33
34    public void setLugares(String lugares) {
35        this.lugares = lugares;
36    }
37
38    public String toString() {
39        return "Veiculo_de_Passeio [nome=" + nome + ", fabricante=" + fabricante + ", cor=" + cor + ", motor=" + motor + ", combustivel=" + combustivel + ", portas=" + portas + ", lugares=" + lugares + "]";
40    }
41
42 }
```

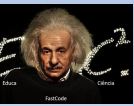
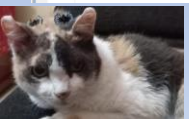
```
Veiculo_de_Carga.java
1 package com.educaciencia.usingExtends;
2
3 public class Veiculo_de_Carga extends Veiculo_Caracteristicas {
4
5     public String eixos;
6     public String comprimento;
7
8     public Veiculo_de_Carga(String nome, String fabricante, String cor, String motor, String combustivel) {
9         super(nome, fabricante, cor, motor, combustivel);
10    }
11
12    public Veiculo_de_Carga(String nome, String fabricante, String cor, String motor, String combustivel, String eixos, String comprimento) {
13        super(nome, fabricante, cor, motor, combustivel);
14        this.eixos = eixos;
15        this.comprimento = comprimento;
16    }
17
18    public String getEixos() {
19        return eixos;
20    }
21
22    public void setEixos(String eixos) {
23        this.eixos = eixos;
24    }
25
26    public String getComprimento() {
27        return comprimento;
28    }
29
30    public void setComprimento(String comprimento) {
31        this.comprimento = comprimento;
32    }
33
34    public String toString() {
35        return "Veiculo_de_Carga [nome=" + nome + ", fabricante=" + fabricante + ", cor=" + cor + ", motor=" + motor + ", combustivel=" + combustivel + ", eixos=" + eixos + ", comprimento=" + comprimento + "]";
36    }
37
38 }
```



Java: Extends

Exemplo:

```
Veiculo_Caracteristicas.java  Veiculo_de_Carga.java  Veiculo_de_Passeio.java  Main.java
3 public class Main {
4
5     public static void main(String[] args) {
6
7         // Testando Veiculo
8         String veic_caract_fabricante = "Volks";
9         String veic_caract_combustivel = "Gasolina";
10        String veic_caract_cor = "Branco";
11
12        Veiculo_Caracteristicas vcaracteristica = new Veiculo_Caracteristicas();
13        vcaracteristica.setFabricante(veic_caract_fabricante);
14        vcaracteristica.setCombustivel(veic_caract_combustivel);
15        vcaracteristica.setCor(veic_caract_cor);
16        System.out.println("Testando Veiculo");
17        System.out.println(vcaracteristica.getFabricante());
18        System.out.println(vcaracteristica.getCombustivel());
19        System.out.println(vcaracteristica.getCor());
20        System.out.println("-----");
21
22        // testando Veiculo Passeio
23        String veic_passeio_fabricante = "Ford";
24        String veic_passeio_combustivel = "Alcool";
25        String veic_passeio_cor = "Azul";
26        String veic_passeio_portas = "4";
27        String veic_passeio_lugares = "5";
28
29        Veiculo_de_Passeio vpasseio = new Veiculo_de_Passeio();
30        vpasseio.setFabricante(veic_passeio_fabricante);
31        vpasseio.setLugares(veic_passeio_lugares);
32        vpasseio.setCor(veic_passeio_cor);
33        vpasseio.setCombustivel(veic_passeio_combustivel);
34        vpasseio.setPortas(veic_passeio_portas);
35
36        System.out.println("testando Veiculo Passeio");
37        System.out.println(vpasseio.getFabricante());
38        System.out.println(vpasseio.getLugares());
39        System.out.println(vpasseio.getPortas());
40        System.out.println("-----");
41
42        // testando Veiculo Carga
43        String vcarga_fabricante = "Mercedes";
44        String vcarga_combustivel = "Diesel";
45        String vcarga_cor = "Rosa";
46        String vcarga_eixos = "8";
47        String vcarga_comprimento = "15 metros";
48
49        Veiculo_de_Carga vcarga = new Veiculo_de_Carga();
50        vcarga.setEixos(vcarga_eixos);
51        vcarga.setComprimento(vcarga_comprimento);
52        vcarga.setFabricante(vcarga_fabricante);
53        vcarga.setCor(vcarga_cor);
54        vcarga.setCombustivel(vcarga_combustivel);
55
56        System.out.println("testando Veiculo Carga");
57        System.out.println(vcarga.getFabricante());
58        System.out.println(vcarga.getEixos());
59        System.out.println(vcarga.getComprimento());
60        System.out.println("-----");
61
62    }
63
64 }
65
```

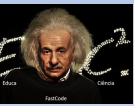
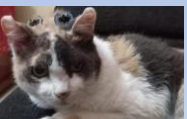


Java: Implements

Implements basicamente podemos dizer que teremos que seguir um “CONTRATO” , isso mesmo.

Além de se implementar apenas “INTERFACES” , podemos dizer que firmaremos um **contrato de usabilidade dos “métodos ou funções” criados.**

Na interface , teremos apenas a definição ou seja , nossa função ou método, também podemos dizer que na interface nunca tem implementação ou seja , “NÃO” se usa Implements em interfaces , ou pelo menos não deveria pois ela nunca será derivada de uma classe.

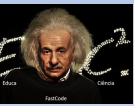
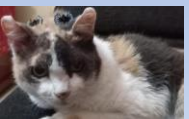


Java: Implements

Uma vez que você tenha a interface criada e definida, uma ou mais classes podem se beneficiar ou seja , podem implementar esta interface.

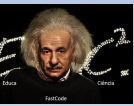
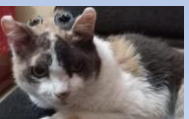
É habitual e permitido e até como que as classes que implementem interfaces , definam membros adicionais e exclusivos.

Em resumo, podemos dizer que as “interfaces” são projetadas para suportar a resolução dinâmica de métodos durante a execução.



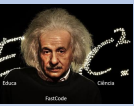
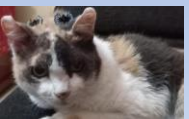
Java: Implements

```
Veiculo_Acao.java x Veiculo.java
1 package com.educaciencia.usingImplements;
2
3
4 public interface Veiculo_Acao {
5
6     public void Ligar();
7     public void Desligar();
8
9 }
10
```



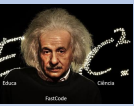
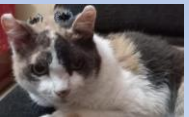
Java: Implements

```
Veiculo_Acao.java  Veiculo.java x
1 package com.educaciencia.usingImplements;
2
3 public class Veiculo implements Veiculo_Acao {
4
5     @Override
6     public void Ligar() {
7         // TODO Auto-generated method stub
8         System.out.println("Veiculo Ligado");
9     }
10
11
12     @Override
13     public void Desligar() {
14         // TODO Auto-generated method stub
15         System.out.println("Veiculo Desligado");
16     }
17 }
18
19
20
```



Java: Implements

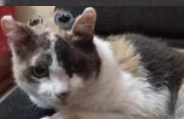
```
Veiculo_Acao.java  Veiculo.java  Main.java x
1 package com.educaciencia.usingImplements;
2
3 public class Main {
4
5     public static void main(String[] args) {
6
7         Veiculo veiculo = new Veiculo();
8         veiculo.Ligar();
9         veiculo.Desligar();
10
11     }
12
13 }
14
```



Então,

vamos à

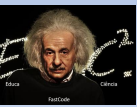
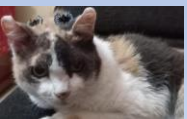
prática!



Java: Extends & Implements

Lembre-se

<code>extends</code>	Quando uma subclasse (filha) <code>extends</code> uma classe base (pai), permite que a subclasse herde o código definido na classe base.	Uma classe pode herdar outra classe. Uma interface pode herdar outra interface usando a palavra-chave <code>extends</code> .	Uma classe pode estender apenas uma classe.
<code>implements</code>	A palavra-chave <code>implements</code> é usada para implementar uma interface. Uma interface contém métodos abstratos. O corpo do método é definido na classe que o implementa.	Usando a palavra-chave <code>implements</code> , uma classe pode implementar uma interface.	Mais de uma interface pode ser implementada por uma classe.



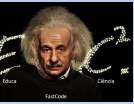
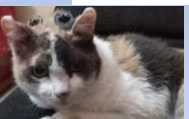
Java: Extends & Implements

Veiculo_Caracteristicas_Extends_Implements.java

```
1 package com.educaciencia.usingExtendsImplements;
2
3 public class Veiculo_Caracteristicas_Extends_Implements {
4
5     /** Caracteristicas Comuns */
6     public String nome;
7     public String fabricante;
8     public String cor;
9     public String motor;
10    public String combustivel;
11
12    /** Metodos Contrutores */
13    public Veiculo_Caracteristicas_Extends_Implements() {
14        super();
15    }
16
17
18    public Veiculo_Caracteristicas_Extends_Implements(String nome, St
19        super();
20        this.nome = nome;
21        this.fabricante = fabricante;
22        this.cor = cor;
23        this.motor = motor;
24        this.combustivel = combustivel;
25    }
26
27    /** Metodos Getters Setters*/
28    public String getNome() {
```

Veiculo_Acao_Extends_Implements.java

```
1 package com.educaciencia.usingExtendsImplements;
2
3
4 public interface Veiculo_Acao_Extends_Implements {
5
6     public void Ligar();
7     public void Desligar();
8
9 }
10
```



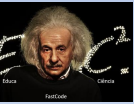
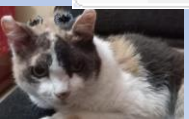
Java: Extends & Implements

Veiculo_de_Carga_Extends_Implements.java

```
1 package com.educaciencia.usingExtendsImplements;
2
3 public class Veiculo_de_Carga_Extends_Implements extends Veiculo_Caracteristicas_Extends_Implements implements Veiculo_Acao_Extends_Implements {
4
5     public String eixos;
6     public String comprimento;
7
8     public Veiculo_de_Carga_Extends_Implements(String nome, String fabricante, String cor, String motor, String combustivel) {
9         super(nome, fabricante, cor, motor, combustivel);
10    }
11
12    public Veiculo_de_Carga_Extends_Implements() {
13        super();
14    }
15
16    public String getEixos() {
```

Veiculo_de_Passeio_Extends_Implements.java

```
1 package com.educaciencia.usingExtendsImplements;
2
3 public class Veiculo_de_Passeio_Extends_Implements extends Veiculo_Caracteristicas_Extends_Implements implements Veiculo_Acao_Extends_Implements{
4
5     public String portas;
6     public String lugares;
7
8     public Veiculo_de_Passeio_Extends_Implements() {
9         super();
10    }
11
12    public Veiculo_de_Passeio_Extends_Implements(String nome, String fabricante, String cor, String motor, String combustivel) {
13        super(nome, fabricante, cor, motor, combustivel);
14    }
15
16    public Veiculo_de_Passeio_Extends_Implements(String nome, String fabricante, String cor, String motor, String combustivel,
```

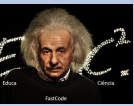


Java: Extends & Implements

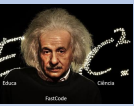
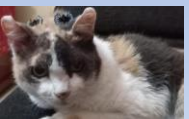
```
1 package com.educaciencia.usingExtendsImplements;
2
3 public class Main {
4
5     public static void main(String[] args) {
6
7         // testando Veiculo Passeio
8         String veic_passeio_fabricante = "Ford";
9         String veic_passeio_combustivel = "Alcool";
10        String veic_passeio_cor = "Azul";
11        String veic_passeio_portas = "4";
12        String veic_passeio_lugares = "5";
13
14        Veiculo_de_Passeio_Extends_Implements vpasseio = new Veiculo_de_Passeio_Extends_Implements();
15        vpasseio.setFabricante(veic_passeio_fabricante);
16        vpasseio.setLugares(veic_passeio_lugares);
17        vpasseio.setCor(veic_passeio_cor);
18        vpasseio.setCombustivel(veic_passeio_combustivel);
19        vpasseio.setPortas(veic_passeio_portas);
20
21        System.out.println("testando Veiculo Passeio");
22        vpasseio.Ligar();
23        System.out.println(vpasseio.getFabricante());
24        System.out.println(vpasseio.getLugares());
25        System.out.println(vpasseio.getPortas());
26        vpasseio.Desligar();
27        System.out.println("-----");
28
29        // -----
30    }
```

```
31
32    // testando Veiculo Carga
33    String vcarga_fabricante = "Mercedes";
34    String vcarga_combustivel = "Diesel";
35    String vcarga_cor = "Rosa";
36    String vcarga_eixos = "8";
37    String vcarga_comprimento = "15 metros";
38
39    Veiculo_de_Carga_Extends_Implements vcarga = new Veiculo_de_Carga_Extends_Implements();
40    vcarga.setEixos(vcarga_eixos);
41    vcarga.setComprimento(vcarga_comprimento);
42    vcarga.setFabricante(vcarga_fabricante);
43    vcarga.setCor(vcarga_cor);
44    vcarga.setCombustivel(vcarga_combustivel);
45
46    System.out.println("testando Veiculo Carga");
47    vcarga.Ligar();
48    System.out.println(vcarga.getFabricante());
49    System.out.println(vcarga.getEixos());
50    System.out.println(vcarga.getComprimento());
51    vcarga.Desligar();
52    System.out.println("-----");
53
54 }
55 }
56
```

```
Problems Javadoc Declaration Console
<terminated> Main (2) [Java Application] C:\Users\FabioPerucelloArante\AppData\Local\Programs\AdoptOpenJDK\jdk-11.0.9.11-hotspot\bin\javaw.exe
testando Veiculo Passeio
Veiculo_de_Passeio ligado
Ford
5
4
Veiculo_de_Passeio desligado
-----
testando Veiculo Carga
Veiculo_de_Carga ligado
Mercedes
8
15 metros
Veiculo_de_Carga desligado
-----
```



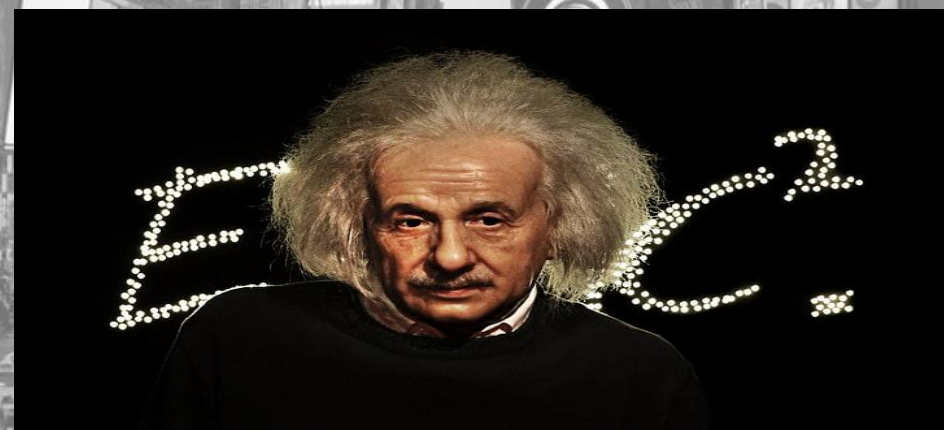
Ainda teremos:



AGRADECIMENTOS aos Apoiadores!!!



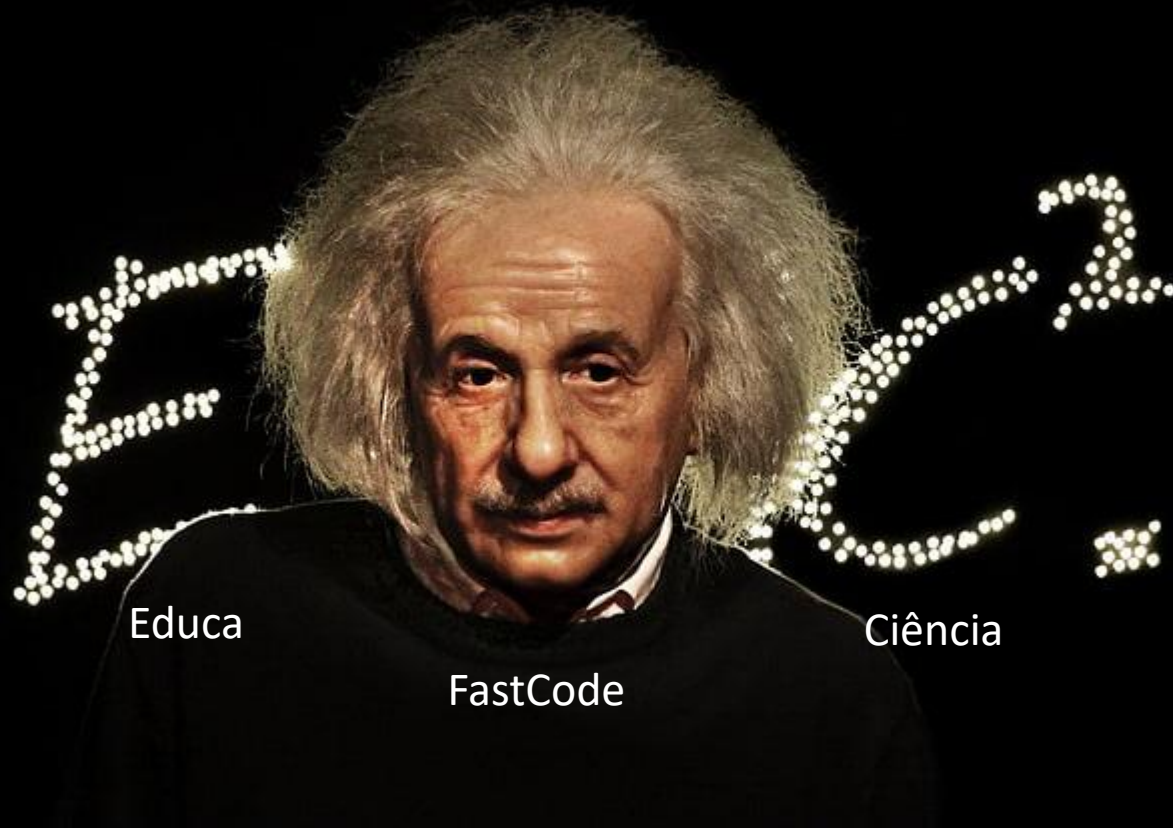
EXPLORANDO TI



SysConecte
Sistemas inteligentes para seu negócio

Fábio Perucello





Educa

FastCode

Ciência

EMAIL

EDUCACIENCIA-FASTCODE@OUTLOOK.COM.BR

EXPLORANDO TI

[HTTPS://WWW.EXPLORANDOTI.COM.BR/](https://www.explorandoti.com.br/)

[HTTPS://WWW.YOUTUBE.COM/EXPLORANDOTI](https://www.youtube.com/explorandoti)

GITHUB

[HTTPS://GITHUB.COM/PERUCELLO](https://github.com/perucello)

ARTIGOS

[HTTPS://GITHUB.COM/PERUCELLO/ARTIGOS-EDUCACIENCIA FASTCODE](https://github.com/perucello/artigos-educaciencia-fastcode)

LIVES

[HTTPS://WWW.YOUTUBE.COM/PLAYLIST?LIST=PLG_DRIR73C6OGRAJXXQD0XJ677LU_ZFGM](https://www.youtube.com/playlist?list=PLG_DRIR73C6OGRAJXXQD0XJ677LU_ZFGM)

Fábio Perucello

