

# POO – Classe - Objeto

Jose.wellington@uniceub.br

# Calendário

◀ agosto de 2013 ▶

D	S	T	Q	Q	S	S
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

◀ setembro de 2013 ▶

D	S	T	Q	Q	S	S
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

◀ outubro de 2013 ▶

D	S	T	Q	Q	S	S
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2

◀ novembro de 2013 ▶

D	S	T	Q	Q	S	S
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

◀ dezembro de 2013 ▶

D	S	T	Q	Q	S	S
24	25	26	27	28	29	30
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

# Agenda

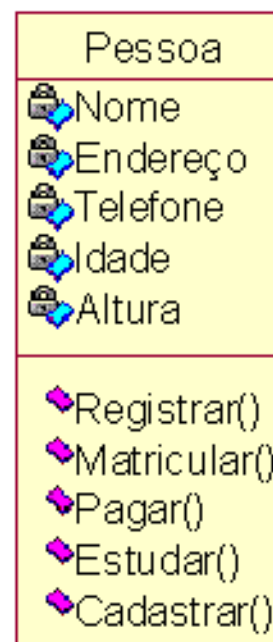
- **Classe**
- **Objeto**
- **Scanner**
- **Exercício**

# Classe

# Objeto

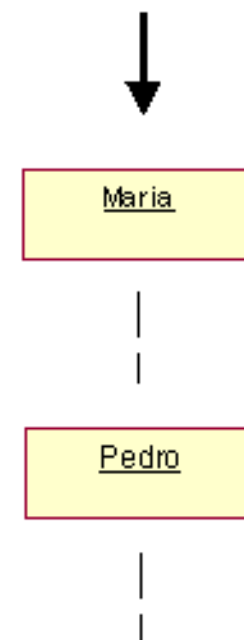
- Um objeto é capaz de armazenar estados através de seus atributos.
- Exemplo de objetos da classe Humanos: João, José, Maria

## Classe



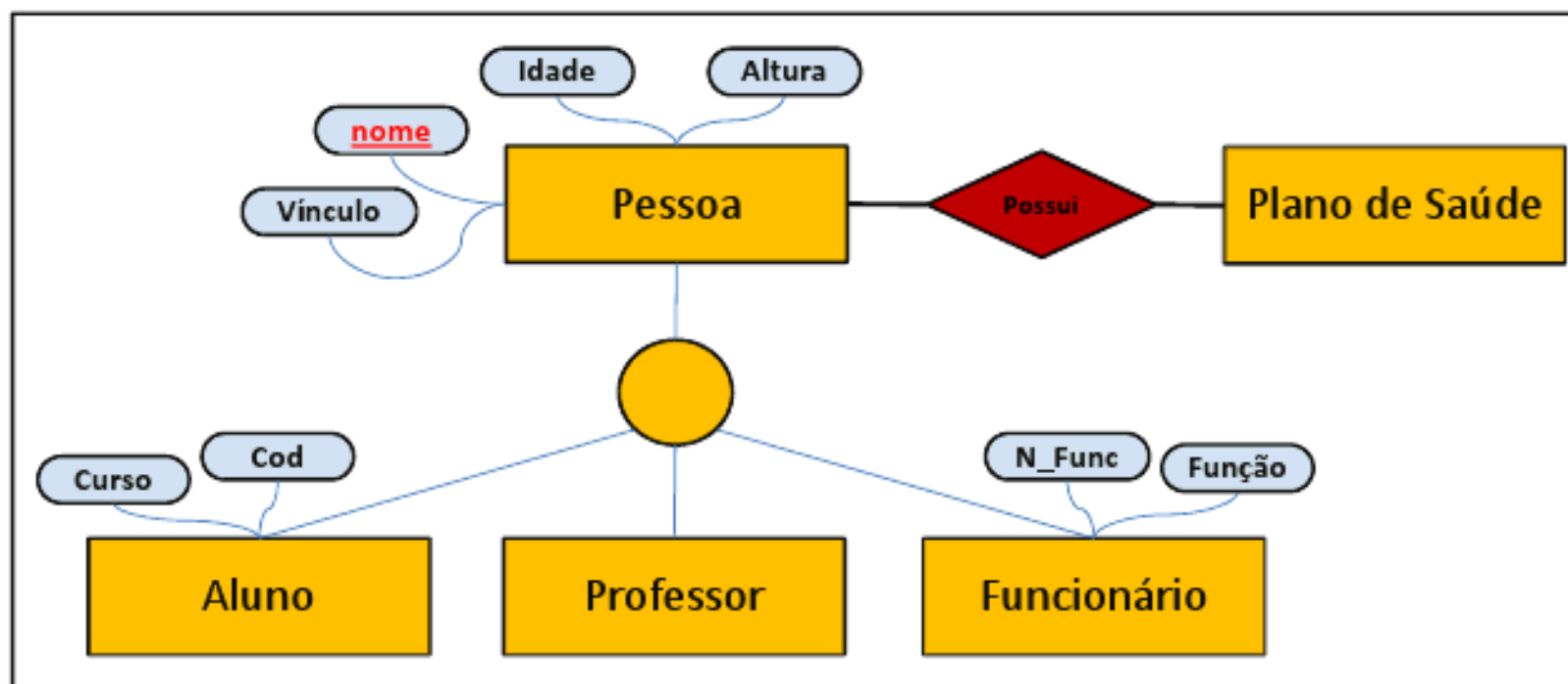
A  
T  
R  
I  
B  
U  
T  
O  
S  
  
M  
É  
T  
O  
D  
O  
S

## Objetos



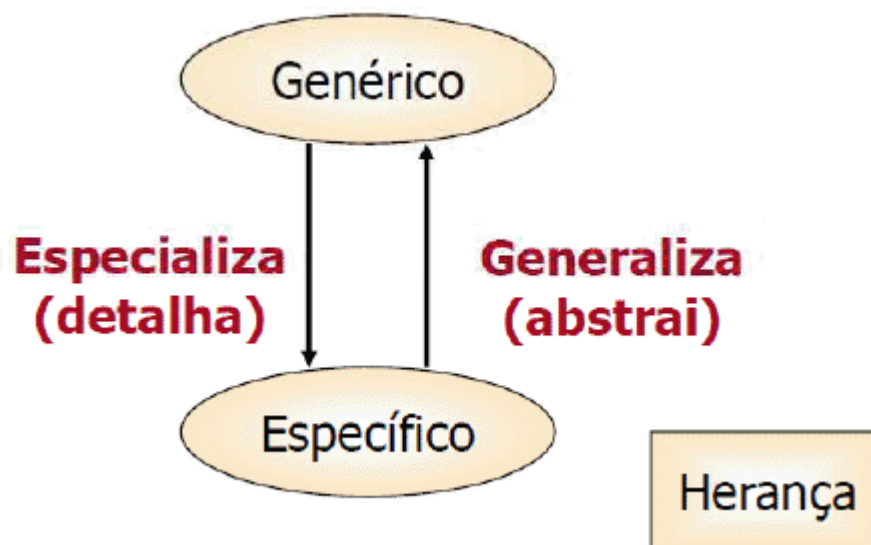
# Herança

- Conjuntos de entidades mais específicos herdam **todos os atributos** do conjunto de entidades mais genérico;



## Generalização e Especialização

Dado que é associado  
a cada ocorrência de  
uma entidade ou de  
um relacionamento



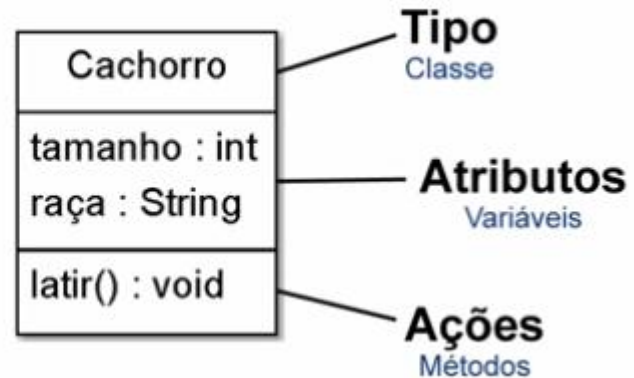
# Programação Orientado a Objeto



# Orientação a Objetos

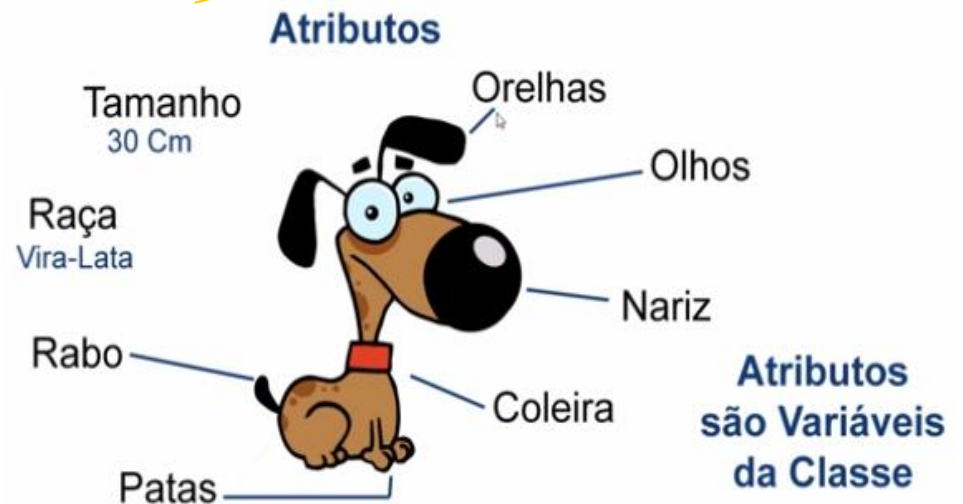
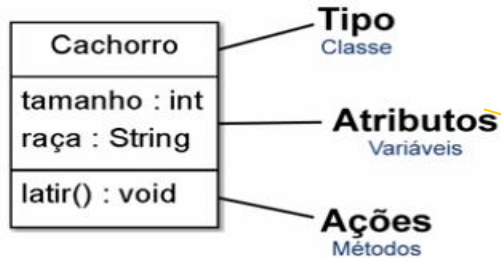
- Propósitos da Programação Orientada a Objetos
  - ⇒ Prover mecanismos para visualizar a complexidade do desenvolvimento de software da mesma forma que visualizamos a complexidade do mundo real;
  - ⇒ Acelerar o desenvolvimento de softwares com base na modularidade e acoplamento;
  - ⇒ Melhorar a qualidade do software desenvolvido.

# Classes - conceitos

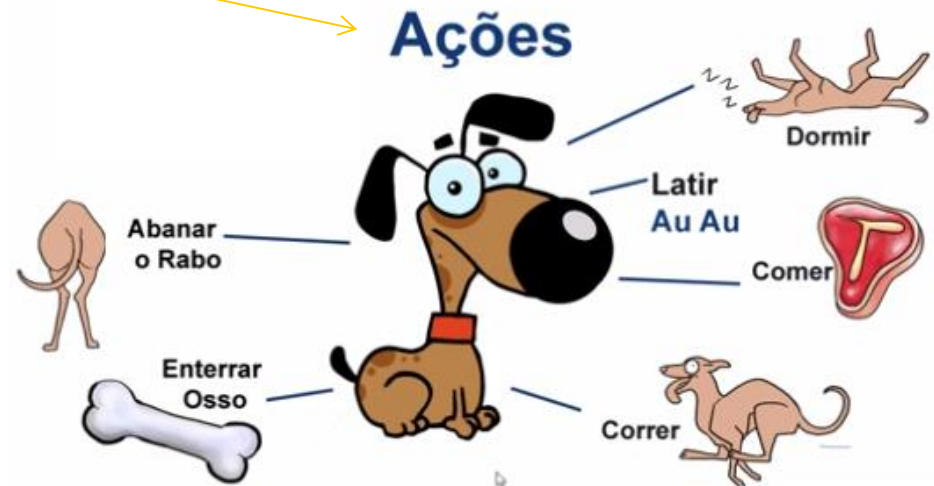
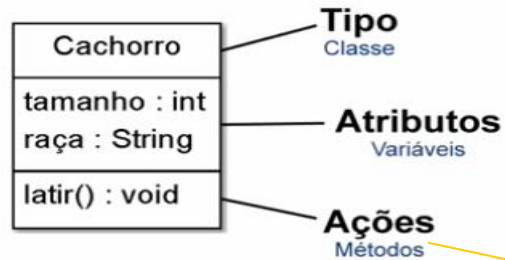


- definem a estrutura e o comportamento de um tipo de objeto;
- atuam como templates;
- permitem a instanciação de um ou mais objetos de um mesmo tipo.

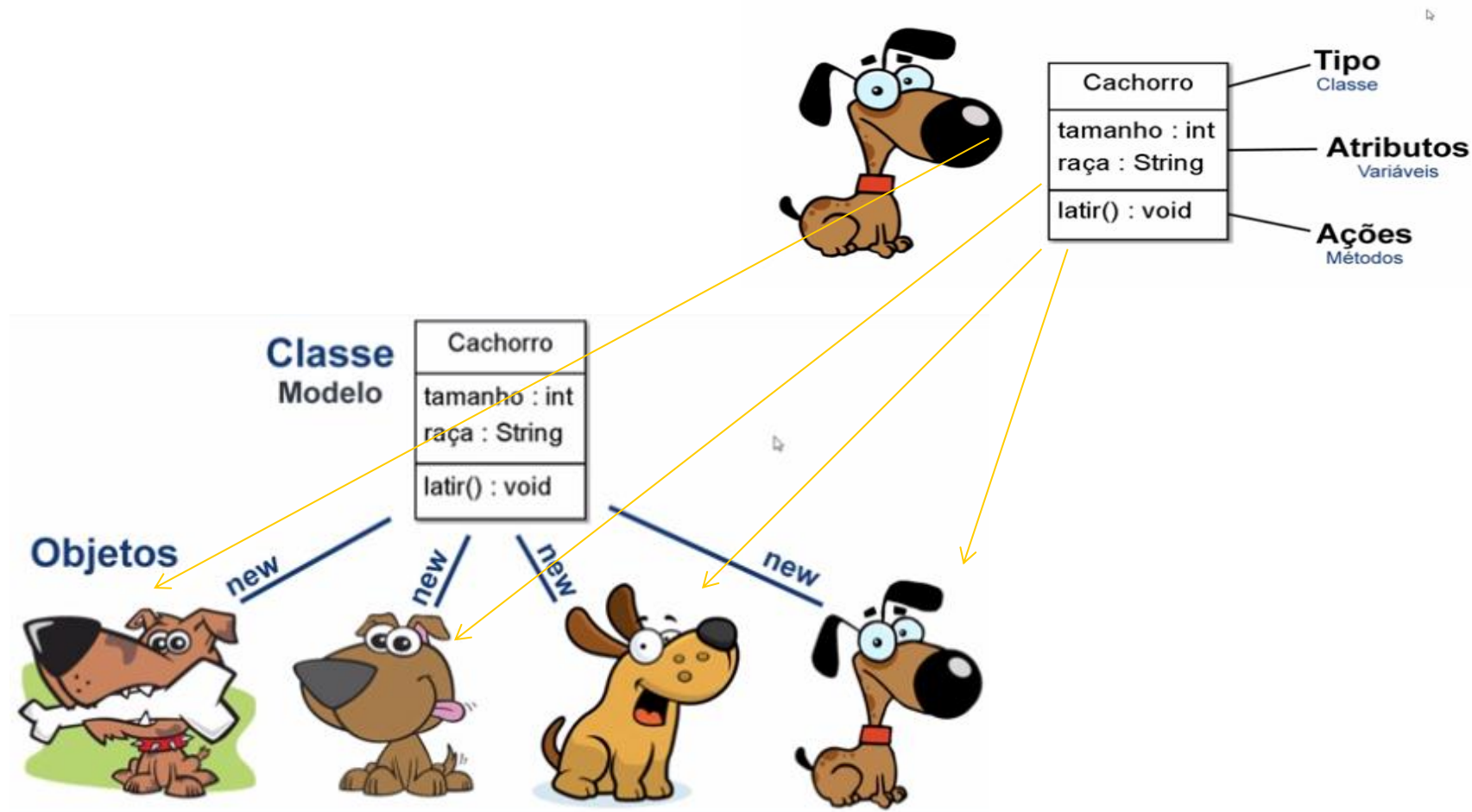
# Classes - Atributos



# Classes - Métodos



# Objetos – Instância de Objeto

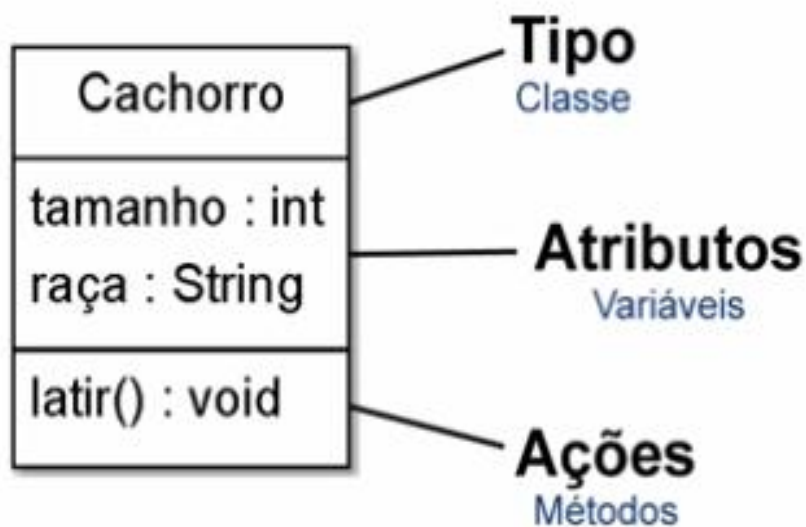


# Criando Objetos em Java

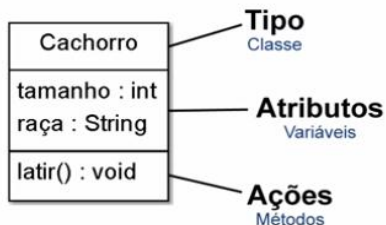
- Objetos são criados através da declaração `new`, seguida de um método construtor. Exemplo:  

```
Automovel gol = new Automovel();
```
- Uma classe pode ter construtores especializados ou somente o construtor default (gerado pelo compilador).

# Criando Objetos em Java



# Criando Objetos em Java

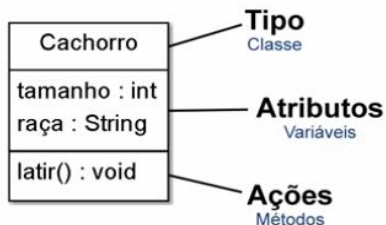


```
package canil;
```

```
class Cachorro {  
    int tamanho;  
    String raca;  
  
    void latir() {  
        System.out.println("Au Au Au!!!");  
    }  
}
```



# Criando Objetos em Java



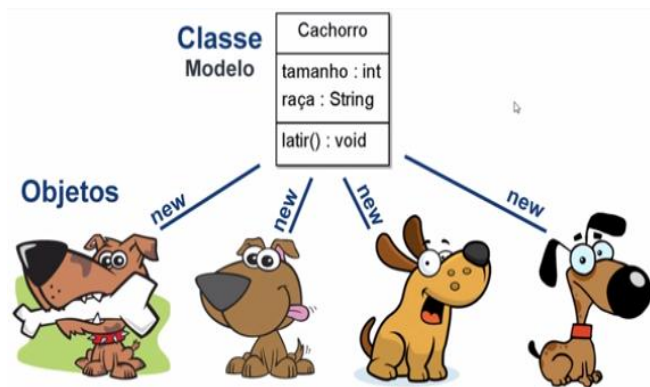
```
public class Canil {  
  
    public static void main(String[] args) {
```

```
        Cachorro pitbul = new Cachorro();
```

```
        pitbul.raça = "Pitbull";  
        pitbul.tamanho = 40;  
        pitbul.latir();
```

```
        Cachorro viralata = new Cachorro();  
        viralata.raça = "Viralata";  
        viralata.tamanho=50;  
        viralata.latir();
```

```
    }  
}
```



# Scanner – Entrada de Dados

# Scanner - Classe

- Recebendo dados do usuário: `new Scanner(System.in)`
- Para receber dados do usuário, temos que usar a classe Scanner, que faz parte do **pacote 'java.util'**.

Vamos dizer ao Java que usaremos essa classe na nossa aplicação

Para isso, adicione essa linha no **começo do programa**:

```
import java.util.Scanner;
```

# Scanner - Classe

- Bom, temos a classe. Vamos declarar o nosso objeto do tipo Scanner.  
Vamos chamá-lo de 'entrada'. Sua declaração é feita da seguinte maneira:

***Scanner **entrada** = new Scanner(System.in);***

# Scanner - Classe

```
import java.util.Scanner;
```

```
public class entrada{
```

```
    public static void main(String[] args) {  
        Scanner entrada = new Scanner(System.in);  
        int idade;
```

```
        System.out.println("Digite sua idade: ");  
        idade = entrada.nextInt();
```

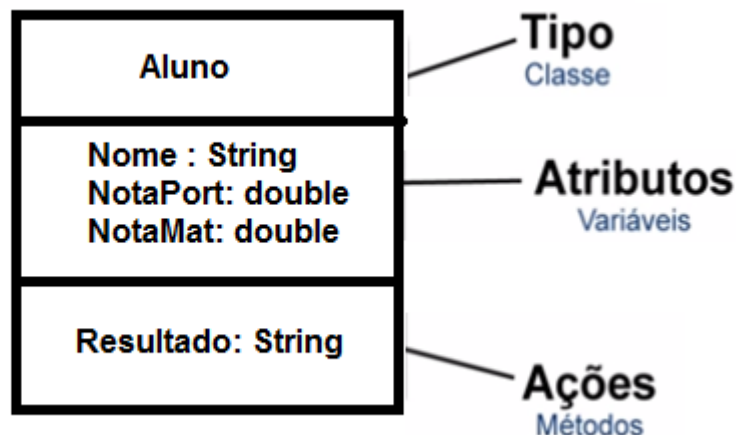
```
        System.out.printf("Sua idade é " + idade + "\n");
```

```
    }
```

```
}
```

# Exercício

# Criando Objetos em Java



- Maria
- Nota Mat = 8
- Nota Port = 7

- Joao
- Nota Mat = 4
- Nota Port = 3

- Carlos
- Nota Mat = 9
- Nota Port = 8

```
Maria Aprovado 7.0  
João Reprovado 3.0  
Carlos Aprovado 8.0
```

# Agenda

- **Classe**
- **Objeto**
- **Scanner**
- **Exercício**

Jose.wellington@uniceub.br