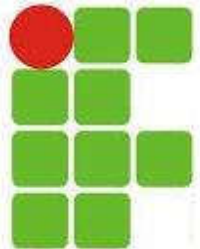


---

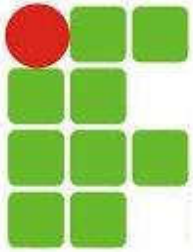
# ALGORITMOS

**Professor: Diego Oliveira**



**Aula 06 -  
Tipos Primitivos de Dados**

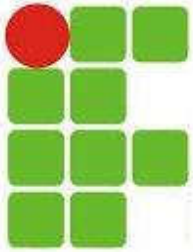




# Netbeans

- Para testar nossos exemplos criaremos uma classe no Netbeans
- Nesta disciplina não será visto nada de orientação a objetos, trabalharemos apenas dentro do método **main** que é criado automaticamente pelo Netbeans no momento da criação do projeto
- Vamos ver como fazer isso nos próximos slides





# Netbeans

- Abra o Netbeans, que estará na área de trabalho do computador do laboratório

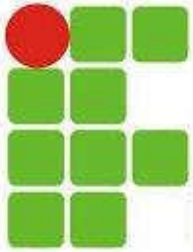


**NetBeans**

[Imagem do ícone do programa Netbeans. É um cubo prateado]

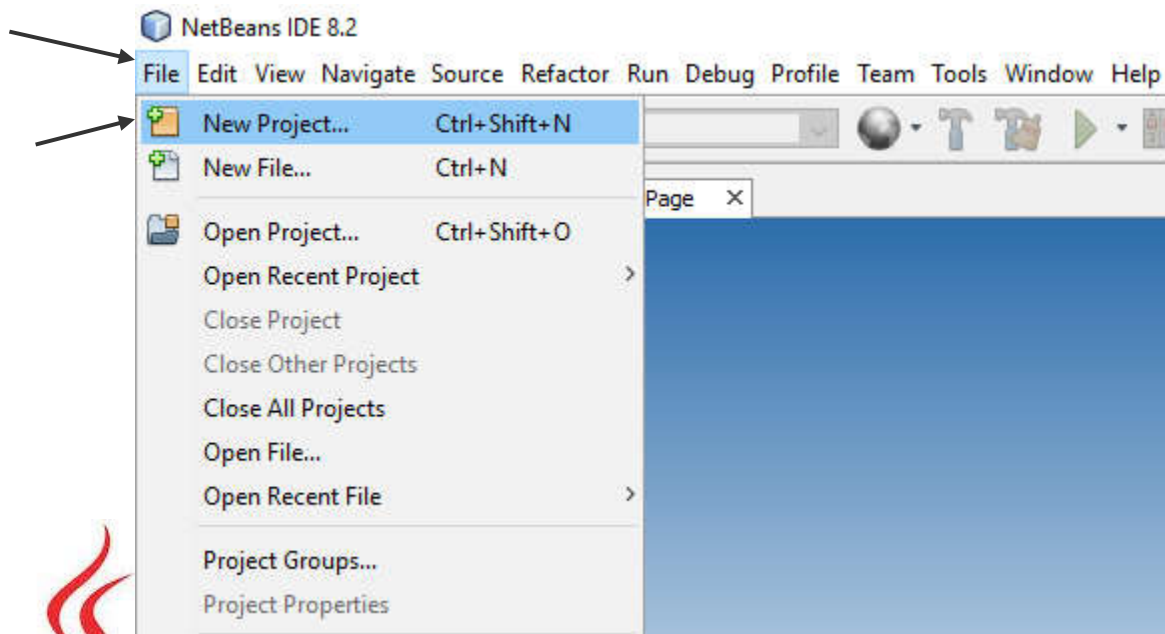
- Aguarde até que a tela principal do programa seja inicializada





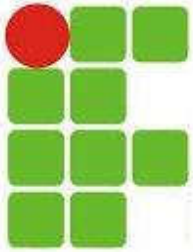
# Netbeans

- Depois clique em Arquivo (File) e Novo Projeto (New Project):



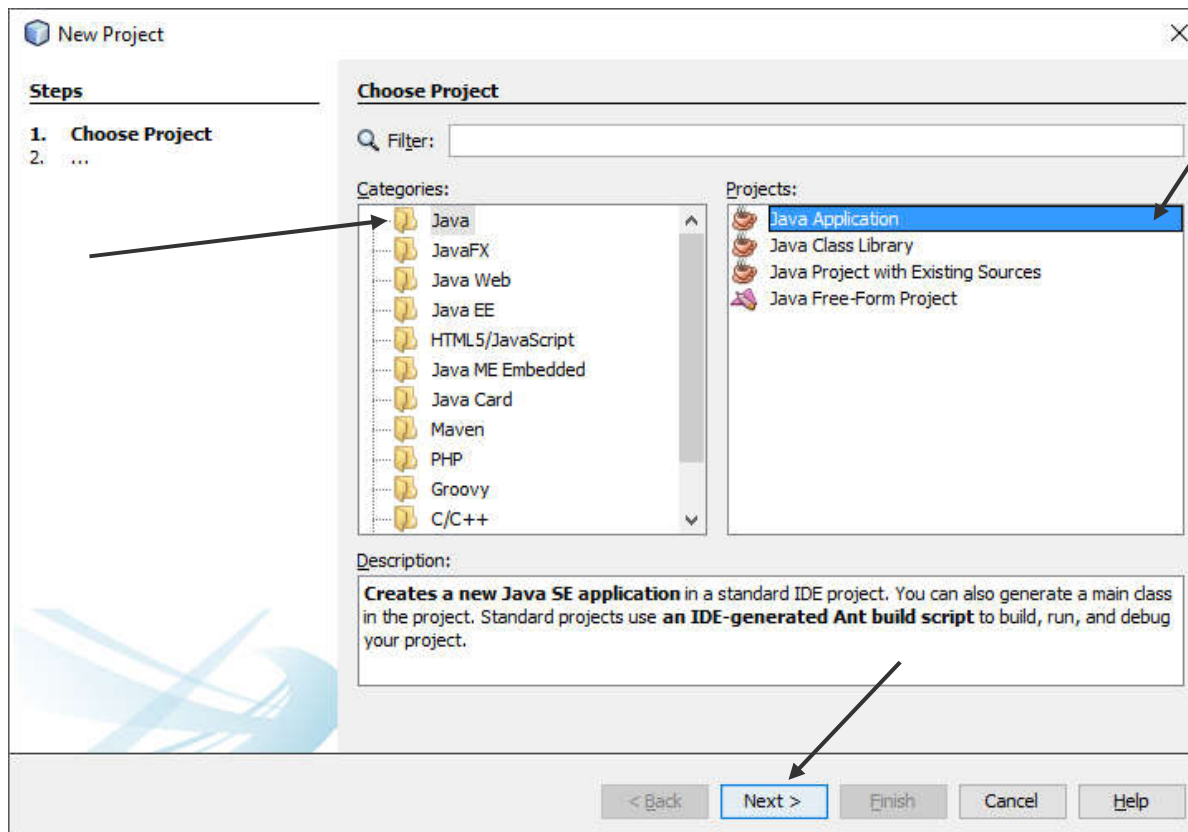
[Imagem da tela inicial do Netbeans. Foi clicado no item de menu Arquivo (File) e depois em Novo Projeto (New Project)]



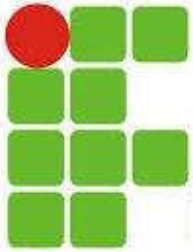


# Netbeans

- Escolha a pasta Java, depois Aplicação Java (Java Application) e clique em Próximo (Next):



[Imagem da tela de criação de novo projeto no Netbeans. Está selecionada a pasta Java e depois Aplicação Java (Java Application) ]



# Netbeans

- Escolha um nome para seu projeto e deixe marcado **Criar Classe Principal** (Create Main Class):

**New Java Application**

**Steps**

1. Choose Project
2. **Name and Location**

**Name and Location**

Project Name:

Project Location:

Project Folder:

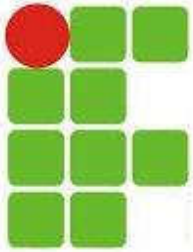
Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

Create Main Class

[Nesta tela há um campo para se digitar o nome do projeto, outro para a localização do projeto no computador e no final há um botão para se criar a classe principal automaticamente, que está marcado (create main class)]



# Netbeans

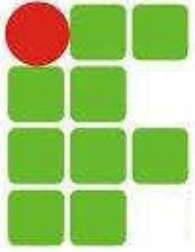
- Será criado automaticamente um código como o mostrado abaixo:

```
package olamundo;
```

```
public class Olamundo {  
    public static void main(String [] args) {  
        //este é um comentário, o código será escrito aqui  
    }  
}
```

- Este código ainda não faz nada, vamos alterá-lo para imprimir algo.





# Netbeans

- A linha 5 deve ser alterada e o código ficará assim:

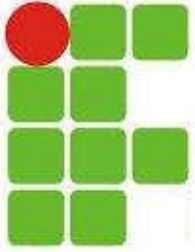
```
package olamundo;

public class Olamundo {
    public static void main(String [] args) {
        System.out.println("Olá Mundo!"); //linha 5
    }
}
```

- Para executar o código aperte F6 e observe a saída na aba Saída (Output) do Netbeans na parte inferior







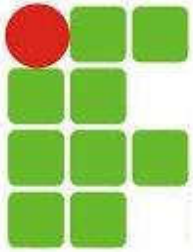
# Netbeans

- Observe a saída:

```
OlaMundo - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Navigator
OlaMundo
  Source Packages
  olamundo
  OlaMundo.java
  Libraries
OlaMundo.java
1 package olamundo;
2
3 public class OlaMundo {
4     public static void main(String[] args) {
5         System.out.println("Olá Mundo!");
6     }
7 }
HTTP Server Monitor Output - OlaMundo (run) Search Results Usages
run:
Olá Mundo!
BUILD SUCCESSFUL (total time: 0 seconds)
8:1 | INS
```

[A imagem mostra a tela do Netbeans com a saída Olá Mundo!]

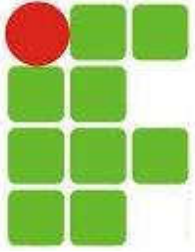




# Netbeans

- O `System.out.println()` é um método que imprime um texto
- Esse texto está entre aspas duplas e aparece na cor laranja
- Também é possível imprimir números e valores de variáveis, como veremos a seguir
- Para usar variáveis precisamos declará-las e para tal devemos escolher um **tipo**, um nome e um valor

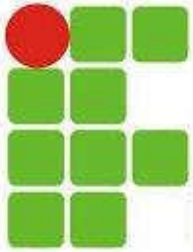




# Tipos Primitivos

- Cada linguagem de programação tem seus tipos primitivos, no Java são 8:
  - boolean
  - char
  - byte
  - short
  - int
  - long
  - float
  - double





## int

- Nesse caso foi criada uma variável chamada 'idade' com valor '30' do tipo 'inteiro':

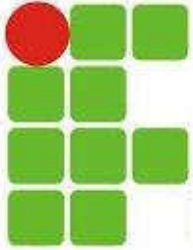
```
package olamundo;
```

```
public class Olamundo {  
    public static void main(String [] args) {  
        int idade = 30;  
        System.out.println("A idade é " + idade);  
    }  
}
```

- Ao executar este código a saída será 'A idade é 30'

- Podemos declarar valores reais, com casas decimais utilizando os tipos float ou double





# double

- Vamos criar uma variável que representa um **salário com seus respectivos centavos**

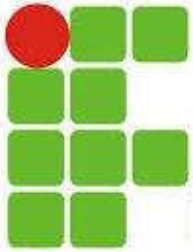
```
package olamundo;
```

```
public class Olamundo {  
    public static void main(String [] args) {  
        double salário = 954.53;  
        System.out.println("O salário é " + salário);  
    }  
}
```

- Lembrando que em inglês a separação de casas decimais é representada pelo PONTO e não pela vírgula, como no português



- E o verdadeiro e falso professor???



# boolean

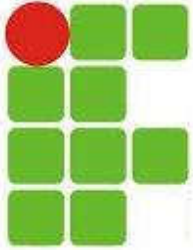
- Neste caso foi criada uma variável boolean com valor verdadeiro (true)

```
package olamundo;
```

```
public class Olamundo {  
    public static void main(String [] args) {  
        boolean isProfessor = true;  
        System.out.println("Diego é professor?" + isProfessor);  
    }  
}
```

- Java utiliza CamelCase para representar variáveis com mais de uma palavra, a segunda fica com a inicial maiúscula: meuNome, minhaldade, salarioMinimo...





# char

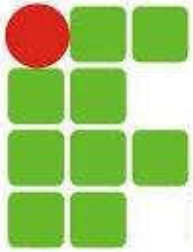
- Para representar um caractere utilizamos o tipo 'char':

```
package olamundo;
```

```
public class Olamundo {  
    public static void main(String [] args) {  
        char gênero = 'm';  
        System.out.println("O gênero é " + gênero);  
    }  
}
```

- Não se coloca acento em nome de variável, o Java permite, porém não é uma boa prática de programação!





# String

- E para salvar um texto? Neste caso o Java utiliza a classe String, que é um conjunto de caracteres:

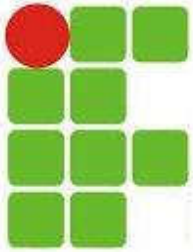
```
package olamundo;
```

```
public class Olamundo {  
    public static void main(String [] args) {  
        String meuNome = "Diego Oliveira";  
        System.out.println("O nome do professor é " + meuNome);  
    }  
}
```

- O tipo primitivo 'char' só permite um caractere enquanto a String permite um texto completo, porém **não é um tipo primitivo!**





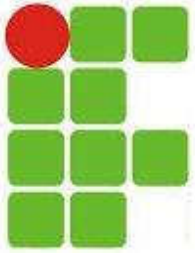


# Exemplo com vários tipos

- Podemos criar diversas variáveis e imprimir seus valores todos de uma vez:

```
package olamundo;
public class Olamundo {
    public static void main(String [] args) {
        String meuNome = "Diego Oliveira";
        int minhaIdade = 30;
        double meuPeso = 77.5;
        boolean isLegal = true;
        char gênero = 'm';
        System.out.println("O nome é" + meuNome);
        System.out.println("A idade é" + minhaIdade);
        System.out.println("O peso é" + meuPeso);
        System.out.println("É legal?" + isLegal);
        System.out.println("O gênero é" + gênero);
    }
}
```





# Exemplo com operações

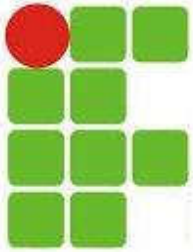
- Também podemos realizar operações aritméticas com os valores das variáveis:

```
package olamundo;
public class OlaMundo {
    public static void main(String [] args) {
        double salario = 1200.00;
        System.out.println("Em um ano esta pessoa ganha " +
salario * 12);
        System.out.println("Metade do salário desta pessoa é " +
salario / 2);
    }
}
```

- Neste caso será impresso o valor ganho em um ano, ou seja o salário vezes 12 e a metade do salário



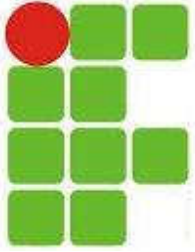
- As operações de + e - também são possíveis



# Exercício

- Crie um projeto com o nome MeusDados e salve na sua pasta do IFRN
- Edite o código dentro do método MAIN para que ele imprima 10 informações sobre você
- Utilize 10 variáveis de pelo menos 5 tipos diferentes, como mostrado na aula
- O texto impresso deve ser claro!
- Imprima a quantidade de dias, horas, minutos e segundos de sua vida utilizando a variável idade como base





---

# Perguntas?

---

