



Java WebDeveloper - Aula 04b - Noite Seg, Qua, Sex - Noite



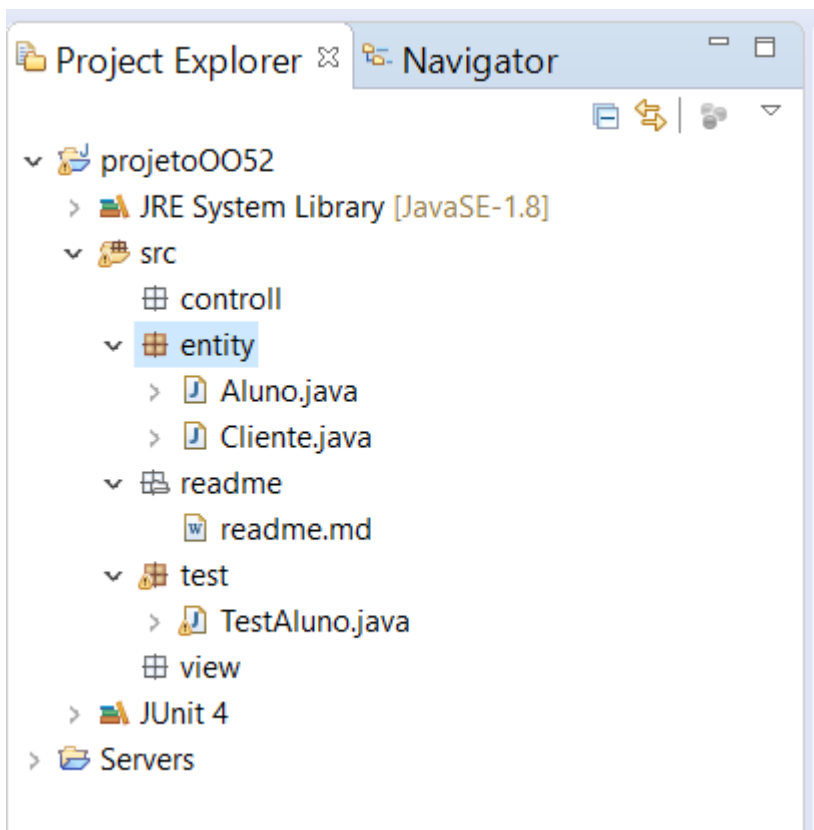
Tema da aula:

Orientação a Objetos, Vetor, Validação dos dados, Calculo, JUnit4, Classe de Test, Expect, AssertTrue, DateTimeFormatter, SimpleDateFormat, LocalDate, Period

— Prof Edson Belém - profedsonbelem@gmail.com (mailto:profedsonbelem@gmail.com) 🕒 Quarta, Abr 01, 2020



Estrutura do projeto depois de finalizado:



Readme.md (<http://Readme.md>)

```

package entity;

import java.util.Arrays;

public class Aluno {
//Java Bean (Padrão Puro)

    //Regra de Negocio será validando as notas
    // verificando o resultado da Media..
    private Integer id;
    private String nome;
    private String disciplina;
    //comeco com vetor depois faco lista ...
    private Double notas[];
    private Double media=0.;
    //escopo global ... Atributo
    //obj.setNome("belem");
    //detalhe (static)
    public Aluno() {
    }

    public Aluno(Integer id, String nome, String disciplina, Double... notas)
{
        this.id = id;
        this.nome = nome;
        this.disciplina = disciplina;
        this.notas = notas;
    }

    @Override
    public String toString() {
        return "Aluno [id=" + id + ", nome=" + nome + ", disciplina=" +
        disciplina + ", notas=" + Arrays.toString(notas) + ", media=" +
        media + "]";
    }

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public String getDisciplina() {
        return disciplina;
    }

    public void setDisciplina(String disciplina) {
        this.disciplina = disciplina;
    }
}

```

```

public Double[] getNotas() {
    return notas;
}
//varargs ou vetor
public void setNotas(Double... notas) {
    this.notas = notas;
}
public Double getMedia() {
    return media;
}
public void setMedia(Double media) {
    this.media = media;
}

//alt + s (generate getter and setter)

public static void print() {
    //dentro do método static sã³ objeto
    Aluno obj = new Aluno();
    obj.setNome("belem");
    //nome = "belem";
    System.out.println(obj.getNome());
}

public void gerarMedia() {
    //escopo local
    double soma = 0;
    for (Double pos : notas) {
        soma += pos;
    }
    this.media = (soma)/notas.length;
}

public static void main(String args[]) {
    Aluno a =new Aluno();
    a.setId(10);
    a.setNome("bele");
    a.setDisciplina("quimica");
    a.setNotas(7d, 6d, 8.);

    a.gerarMedia();

    System.out.println(a);
}
}

```



Aluno.java

```
1 package entity;
2
3 import java.util.Arrays;
4
5 public class Aluno {
6     /*
7      * Regra de Negocio Validar as notas gerarMedia()
8      * verificando a media
9      */
10    private Integer id;
11    private String nome;
12    private String disciplina;
13    private Double notas[];
14    private Double media = 0.;
15
16    public Aluno() {
17    }
18
19    public Aluno(Integer id, String nome, String disciplina,
20    Double... notas) {
21        this.id = id;
22        this.nome = nome;
23        this.disciplina = disciplina;
24        this.notas = notas;
25    }
26
27    @Override
28    public String toString() {
29        return "Aluno [id=" + id + ", nome=" + nome +
30        ", disciplina=" + disciplina + ", notas=" +
31        Arrays.toString(notas) + ", media=" + media + "];"
32    }
33
34    public Integer validarNotas() {
35        for (Double pos : getNotas()) {
36            if (pos < 0. | pos > 10.) {
37                throw new IllegalArgumentException
38                ("Nota Invalida:" + pos);
39            }
40        }
41        return 1;
42    }
43
44    public Integer getId() {
45        return id;
46    }
47
48    public void setId(Integer id) {
49        this.id = id;
50    }
51
52    public String getNome() {
53        return nome;
54    }
```

```

55
56     public void setNome(String nome) {
57         this.nome = nome;
58     }
59
60     public String getDisciplina() {
61         return disciplina;
62     }
63
64     public void setDisciplina(String disciplina) {
65         this.disciplina = disciplina;
66     }
67
68     public Double[] getNotas() {
69         return notas;
70     }
71
72     // varargs ou vetor
73     public void setNotas(Double... notas) {
74         this.notas = notas;
75     }
76
77     public Double getMedia() {
78         return media;
79     }
80
81     public void setMedia(Double media) {
82         this.media = media;
83     }
84
85     //IMPRIMINDO O NOME
86     public static void print() {
87         Aluno obj = new Aluno();
88         obj.setNome("belem bisonho");
89         System.out.println(obj.getNome());
90     }
91
92     //CALCULANDO A MEDIA DO ALUNO
93     public void gerarMedia() {
94         double soma = 0;
95         for (Double pos : notas) {
96             soma += pos;
97         }
98         this.media = (soma) / notas.length;
99     }
100
101     public static void main(String args[]) {
102         Aluno a = null;
103
104         try {
105             a = new Aluno();
106             a.setId(10);
107             a.setNome("bele");
108             a.setDisciplina("quimica");
109             a.setNotas(12d, 6d, 8.);

```

```

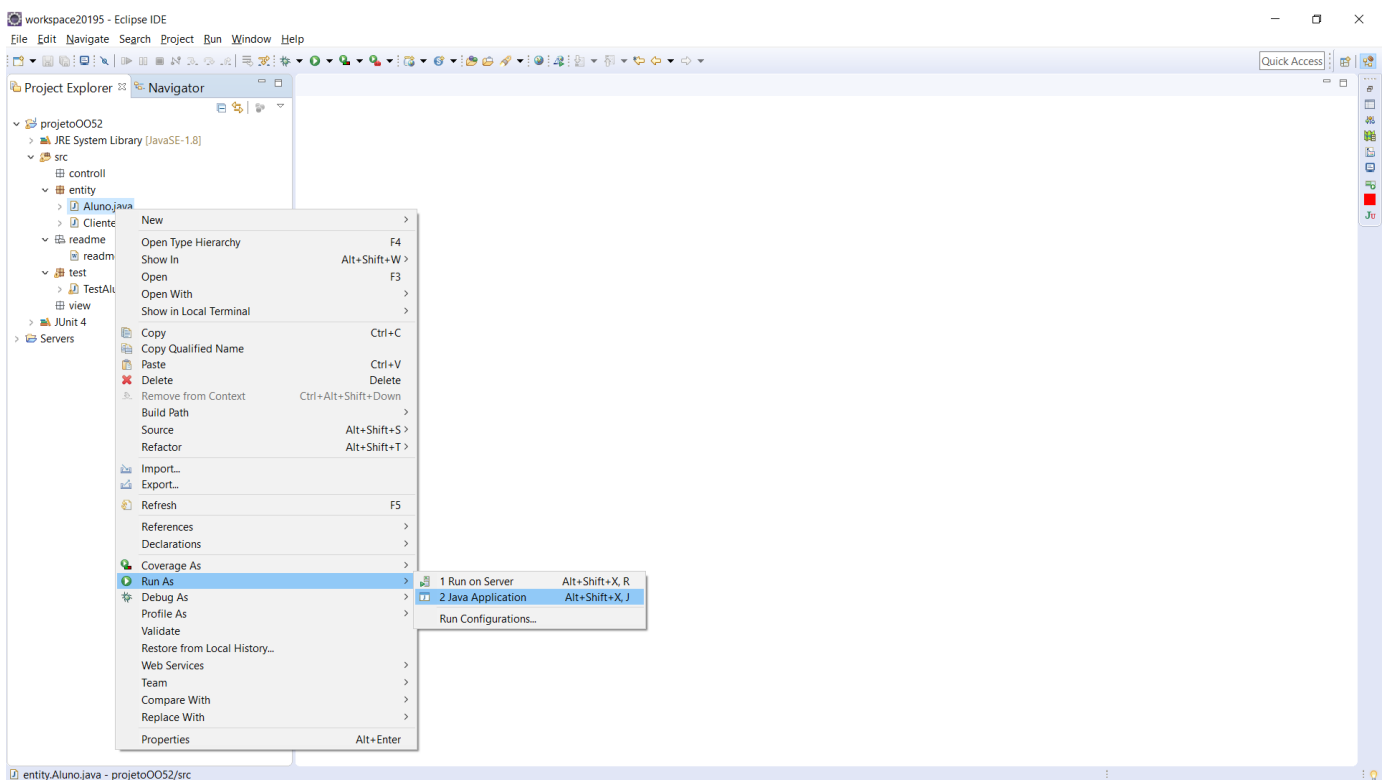
110         a.validarNotas();
111         a.gerarMedia();
112
113         System.out.println(a);
114     } catch (Exception e) {
115         System.out.println("Error :" + e.getMessage());
116         e.printStackTrace();
117     }
118 }
119 }

```

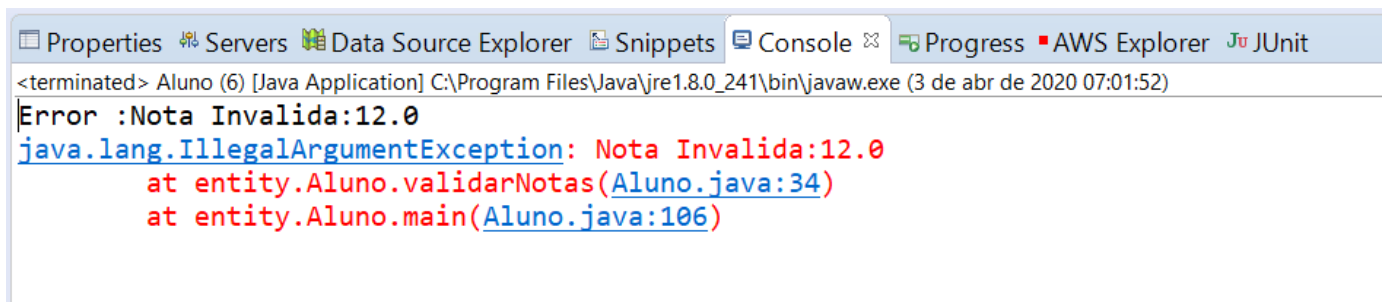


Para rodar a classe:

Clique na classe com o botão direito → run as → java application



Resultado no console:



Digitar uma nota valida (10)



Resultado no console:

```
Properties Servers Data Source Explorer Snippets Console Progress AWS Explorer JUnit
<terminated> Aluno (6) [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (3 de abr de 2020 07:02:39)
Aluno [id=10, nome=bele, disciplina=quimica, notas=[10.0, 6.0, 8.0], media=8.0]
```

JUnit (<https://junit.org/junit5/>)



O **JUnit** é um framework open-source, criado por Erich Gamma e Kent Beck, com suporte à criação de testes automatizados na linguagem de programação Java.

Esse framework facilita a criação de código para a automação de testes com apresentação dos resultados. Com ele, pode ser verificado se cada método de uma classe funciona da forma esperada, exibindo possíveis erros ou falhas podendo ser utilizado tanto para a execução de baterias de testes como para extensão.

Com **JUnit**, o programador tem a possibilidade de usar esta ferramenta para criar um modelo padrão de testes, muitas vezes de forma automatizada.

O teste de unidade testa o menor dos componentes de um sistema de maneira isolada. Cada uma dessas unidades define um conjunto de estímulos (chamada de métodos), e de dados de entrada e saída associados a cada estímulo. As entradas são parâmetros e as saídas são o valor de retorno, exceções ou o estado do objeto. Tipicamente um teste unitário executa um método individualmente e compara uma saída conhecida após o processamento da mesma.

O **JUnit** permite a realização de testes de unidades, conhecidos como “caixa branca”, facilitando assim a correção de métodos e objetos.

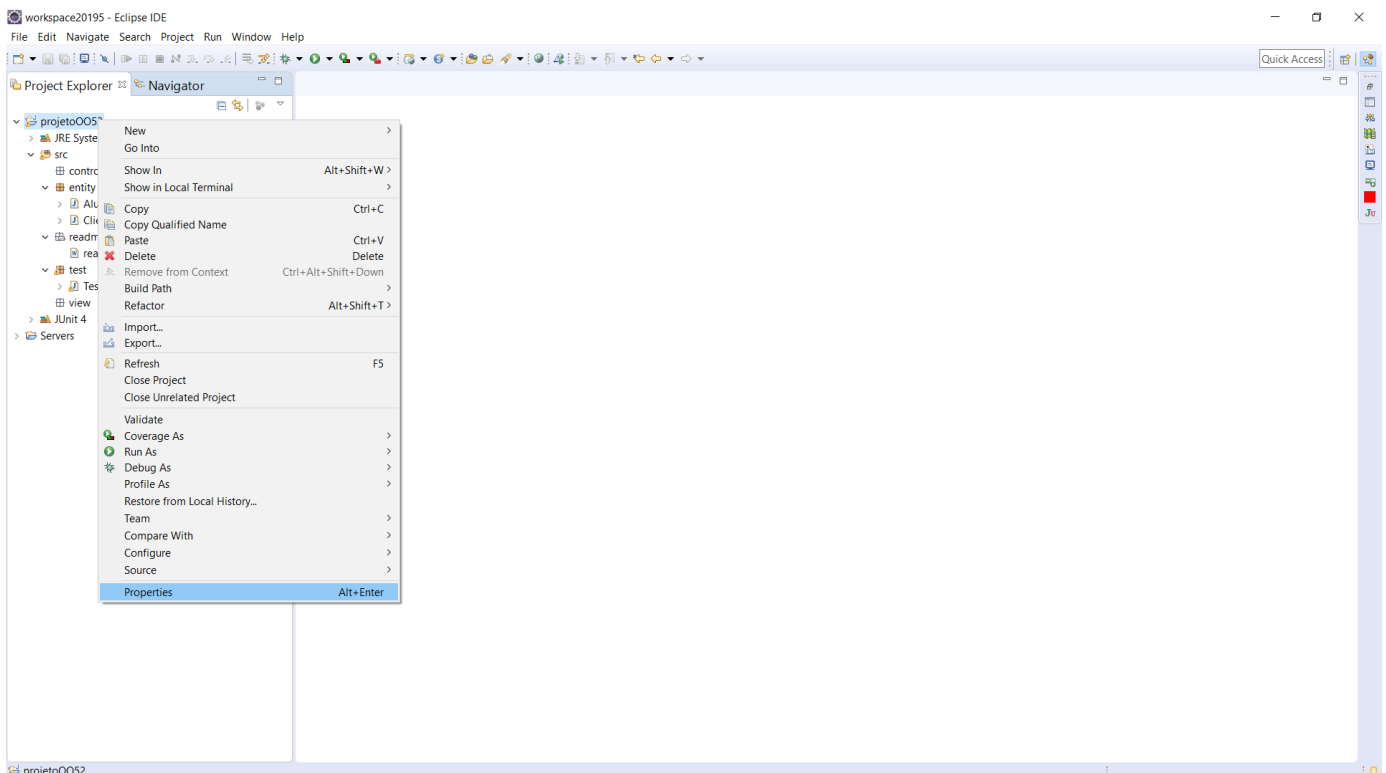
Algumas vantagens de se utilizar **JUnit**:

- Permite a criação rápida de código de teste enquanto possibilita um aumento na qualidade do sistema sendo desenvolvido e testado;
- Não é necessário escrever o próprio framework;
- Amplamente utilizado pelos desenvolvedores da comunidade código-aberto, possuindo um grande número de exemplos;
- Uma vez escritos, os testes são executados rapidamente sem que, para isso, seja interrompido o processo de desenvolvimento;
- JUnit checa os resultados dos testes e fornece uma resposta imediata;
- Pode-se criar uma hierarquia de testes que permitirá testar apenas uma parte do sistema ou todo ele;
- Escrever testes com JUnit permite que o programador perca menos tempo depurando seu código;
- JUnit é LIVRE.

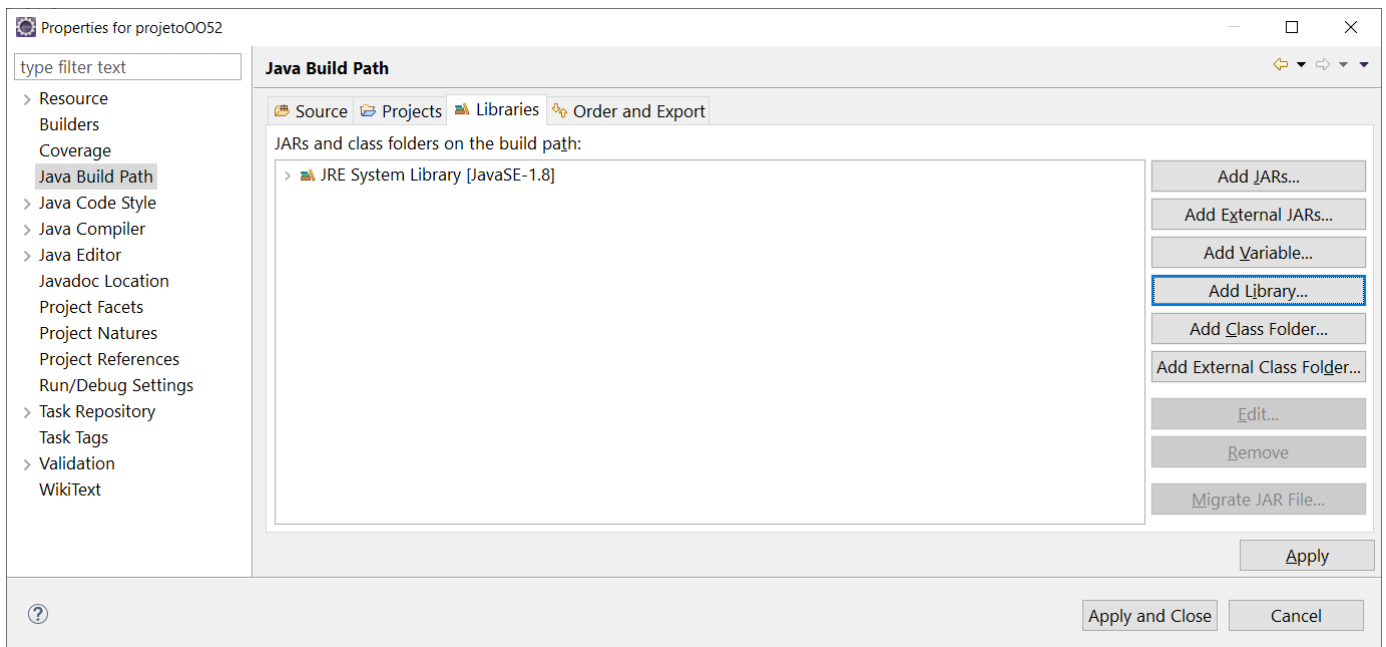
A experiência adquirida com o **JUnit** tem sido importante na consolidação do Test Driven Development (desenvolvimento direcionado a testes). Além disso, ele foi adaptado a outras linguagens, tais como C# (NUnit), Python, Fortran, e C++.

Habilitando JUnit no projeto

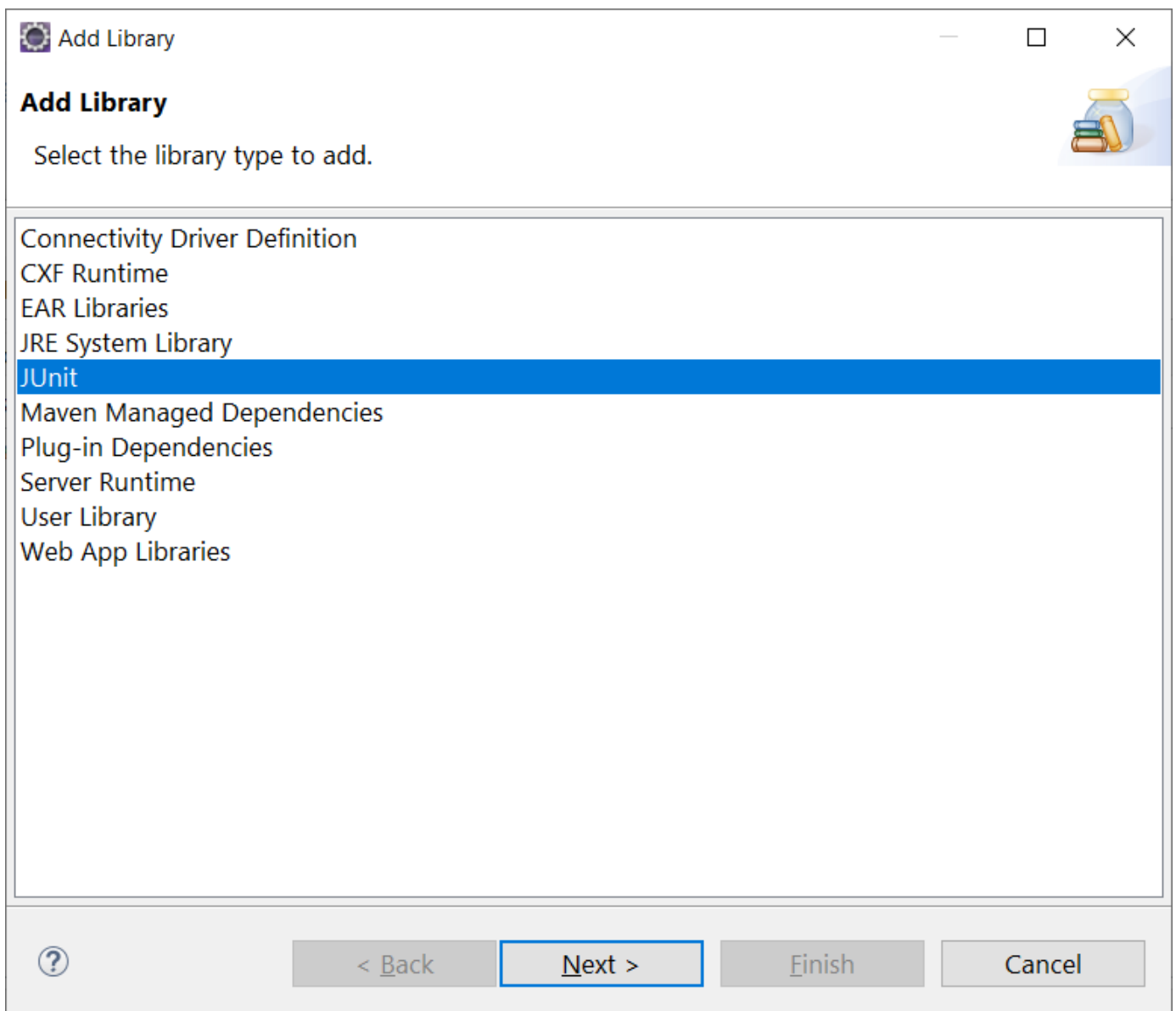
Clicar com o botão direito no projeto → Properties



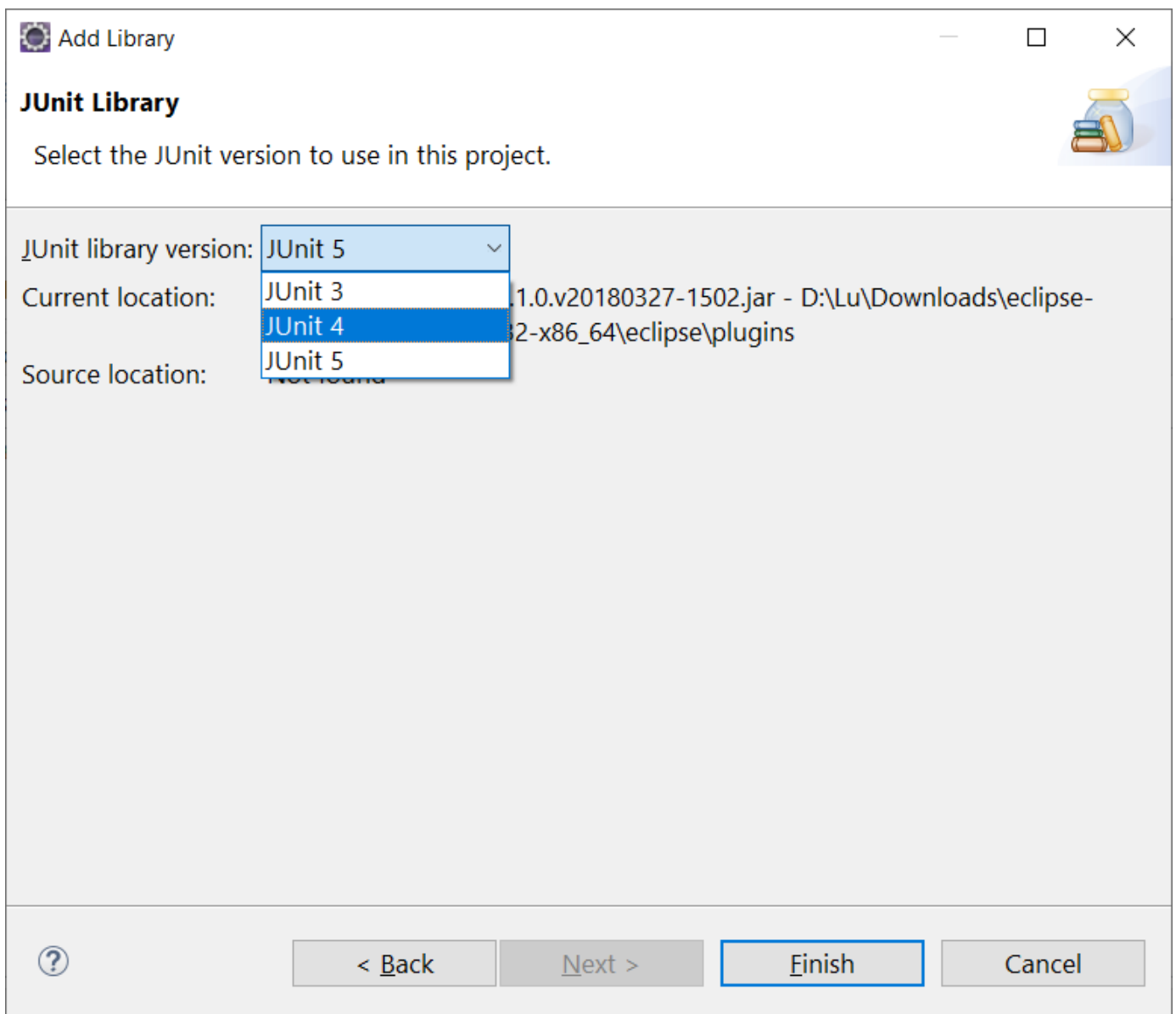
Clicar em Java Build path → Aba Libraries → Add Libraries



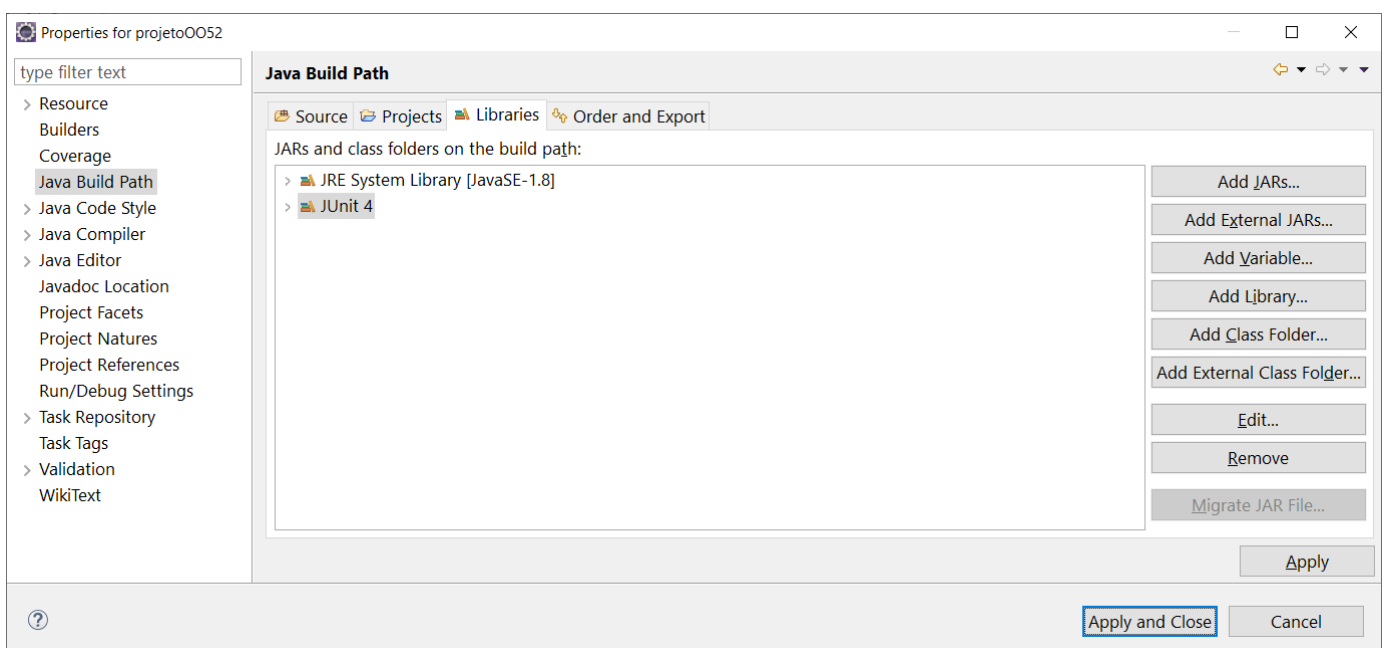
Marcar JUnit ➡ Next



Marcar JUnit4 ➡ Finish



Apply and close





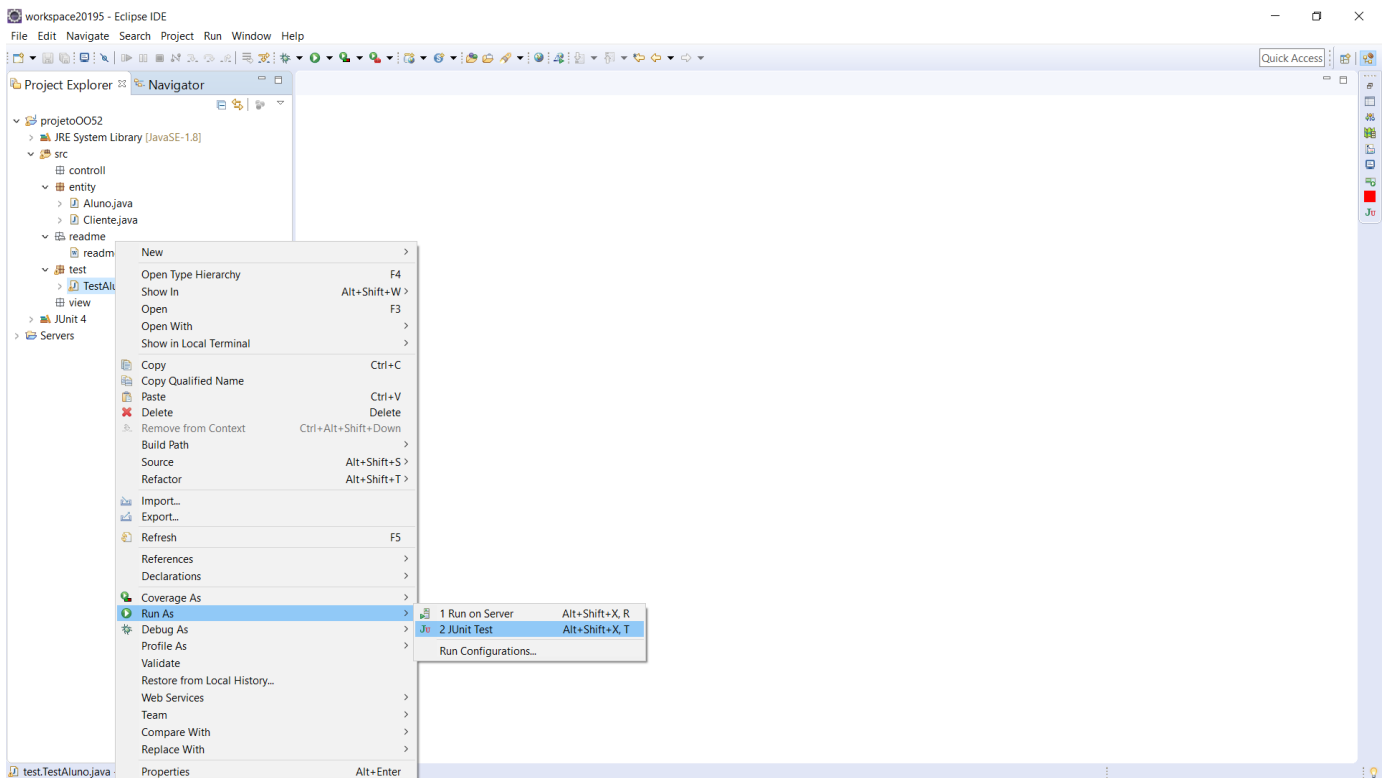
TestAluno.java

```
1  package test;
2
3  import static org.junit.Assert.assertTrue;
4
5  import org.junit.Before;
6  import org.junit.Test;
7
8  import entity.Aluno;
9
10 public class TestAluno {
11
12     // Teste JUNIT
13     // iniciar todos os atributos
14     Aluno aluno;
15
16     @Before
17     public void inicio() {
18         aluno = new Aluno();
19         aluno.setId(10);
20         aluno.setNome("luciana");
21         aluno.setDisciplina("java");
22         // alimento as variáveis que nao irão mudar
23     }
24
25     @Test
26     public void testeNotasValidas() {
27         aluno.setNotas(7., 8., 9.);
28         int numero = aluno.validarNotas();
29         assertTrue(numero == 1);
30     }
31
32     @Test(expected = IllegalArgumentException.class)
33     public void testeNotasInvalidas() {
34         aluno.setNotas(7., 11., 9.);
35         int numero = aluno.validarNotas();
36     }
37
38     @Test
39     public void TestCalculoMedia() {
40         double nota1 = 6.;
41         double nota2 = 7.;
42         double nota3 = 8.;
43         aluno.setNotas(6., 7., 8.);
44         int numero = aluno.validarNotas();
45         aluno.gerarMedia();
46         assertTrue(aluno.getMedia().
47             equals((nota1 + nota2 + nota3) / 3));
48     }
49 }
```

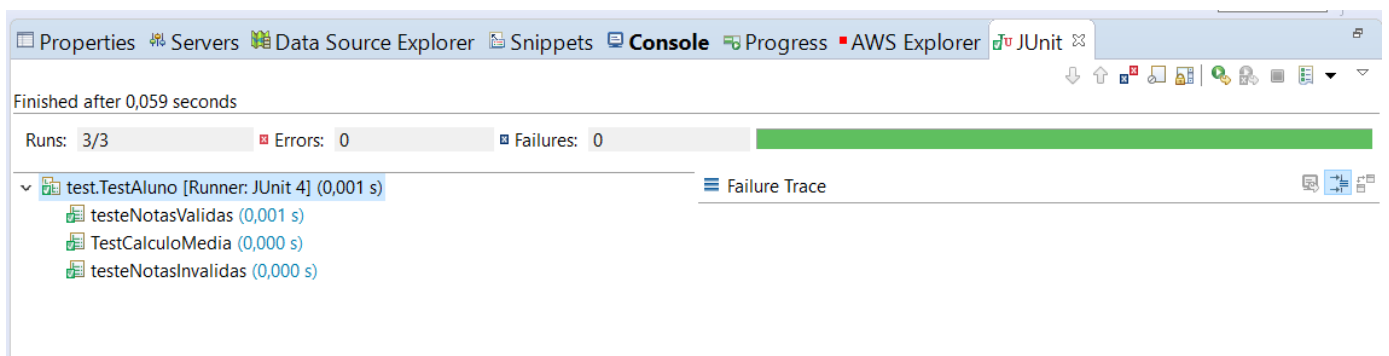


Para testar a classe com o JUnit

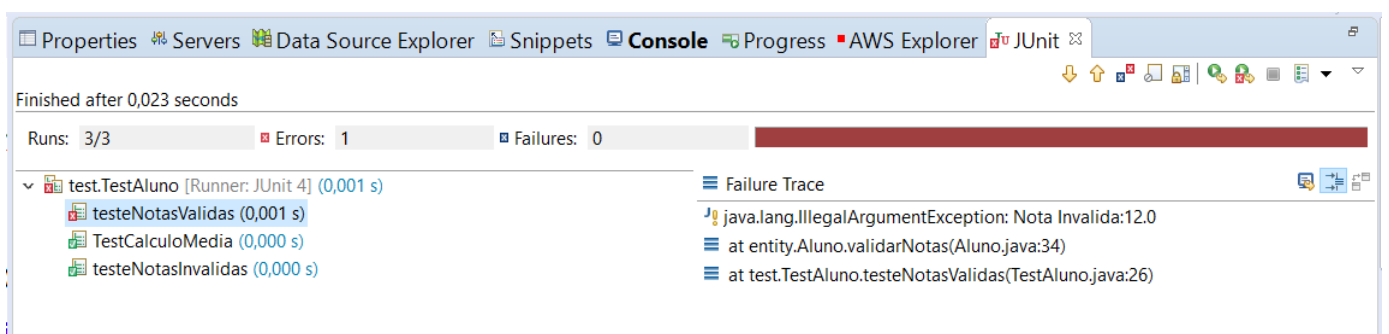
Clique na classe com o botão direito ➡ run as ➡ JUnit test



Resultado na aba do JUnit



Digitar uma nota inválida (12) e testar novamente.





Cliente.java

```
1 package entity;
2
3 import java.text.SimpleDateFormat;
4 import java.time.LocalDate;
5 import java.time.Period;
6 import java.time.format.DateTimeFormatter;
7
8 public class Cliente {
9
10     /*
11      * Calcular a Idade
12      */
13
14     static SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
15
16     //Formata Date
17     // Date
18     static DateTimeFormatter dtf = DateTimeFormatter.
19     ofPattern("dd-MM-yyyy");
20
21     private Integer id;
22     private String nome;
23     private LocalDate dataNascimento;
24
25     public Cliente() {
26     }
27
28     public Cliente(Integer id, String nome, LocalDate dataNascimento) {
29         this.id = id;
30         this.nome = nome;
31         this.dataNascimento = dataNascimento;
32     }
33
34     public Integer retornarIdade() {
35         LocalDate dataAtual = LocalDate.now(); // data Atual
36         Period period = Period.between(dataNascimento, dataAtual);
37         int anos = period.getYears();
38         return anos;
39     }
40
41     @Override
42     public String toString() {
43         return "Cliente [id=" + id + ", nome=" + nome +
44         ", dataNascimento=" + dtf.format(dataNascimento) + "]";
45     }
46
47     public Integer getId() {
48         return id;
49     }
50
51     public void setId(Integer id) {
52         this.id = id;
53     }
54 }
```

```

55     public String getName() {
56         return nome;
57     }
58
59     public void setName(String nome) {
60         this.nome = nome;
61     }
62
63     public LocalDate getDataNascimento() {
64         return dataNascimento;
65     }
66
67     public void setDataNascimento(LocalDate dataNascimento) {
68         this.dataNascimento = dataNascimento;
69     }
70
71     public static void main(String[] args) {
72
73         Cliente obj = new Cliente();
74         try {
75             obj.setId(10);
76             obj.setName("belem");
77             obj.setDataNascimento(LocalDate.of(1973, 1, 28));
78             System.out.println(obj);
79             System.out.println("Idade:" + obj.retornarIdade());
80         } catch (Exception ex) {
81             System.out.println("Error :" + ex.getMessage());
82         }
83     }
84 }

```



Para rodar a classe:

Clique na classe com o botão direito ➡ run as ➡ java application

👁 Resultado no console.

```

<terminated> Aluno (6) [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (3 de abr de 2020 07:19:34)
Aluno [id=10, nome=bele, disciplina=quimica, notas=[10.0, 6.0, 8.0], media=8.0]

```