

Fundamentos de Desenvolvimento com Java

Professor: arménio cardoso



Kaike Torres da silva

30.03.2025

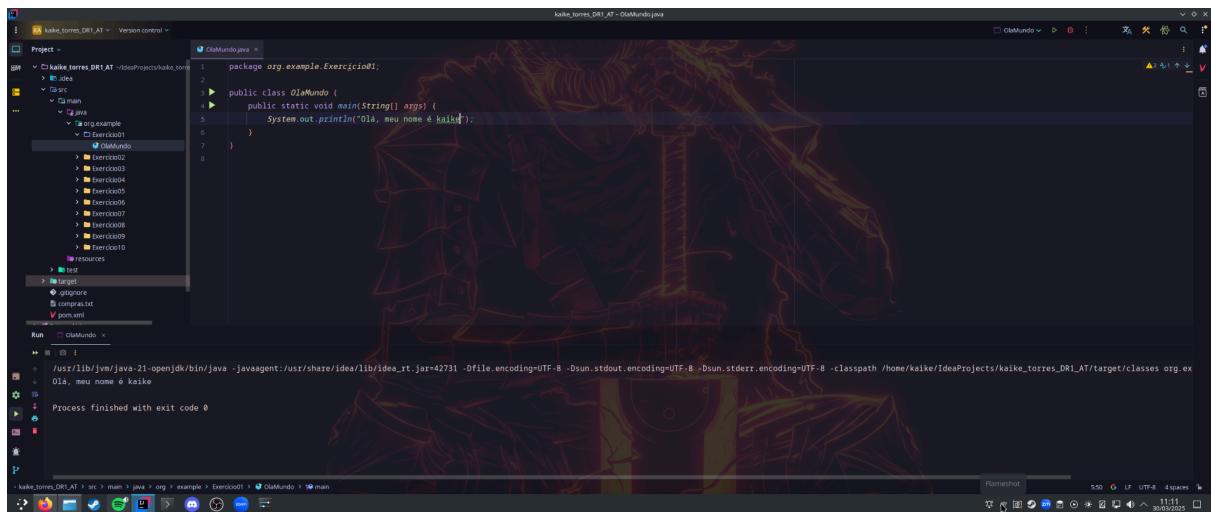
Análise e Desenvolvimento de Sistemas (ADS)

INTRODUÇÃO

A criação de programas em Java envolve o entendimento tanto de ferramentas de desenvolvimento quanto dos conceitos fundamentais da linguagem. Ao longo deste exercício, você passará por uma série de tarefas práticas que abrangem desde a configuração do ambiente até a implementação de sistemas mais complexos, utilizando diferentes conceitos de programação, como manipulação de arquivos, orientação a objetos, e interação com o usuário.

Este exercício tem como objetivo preparar você para a configuração do ambiente de desenvolvimento, utilizando o JDK e a IDE IntelliJ IDEA, além de fornecer uma introdução aos principais conceitos da linguagem Java. Além disso, o exercício permitirá que você implemente sistemas que envolvem lógica de negócios, como validação de senhas, cálculos de impostos, e a simulação de um sistema bancário. Ao longo do caminho, você também aprenderá a trabalhar com arrays, arquivos e implementações de objetos.

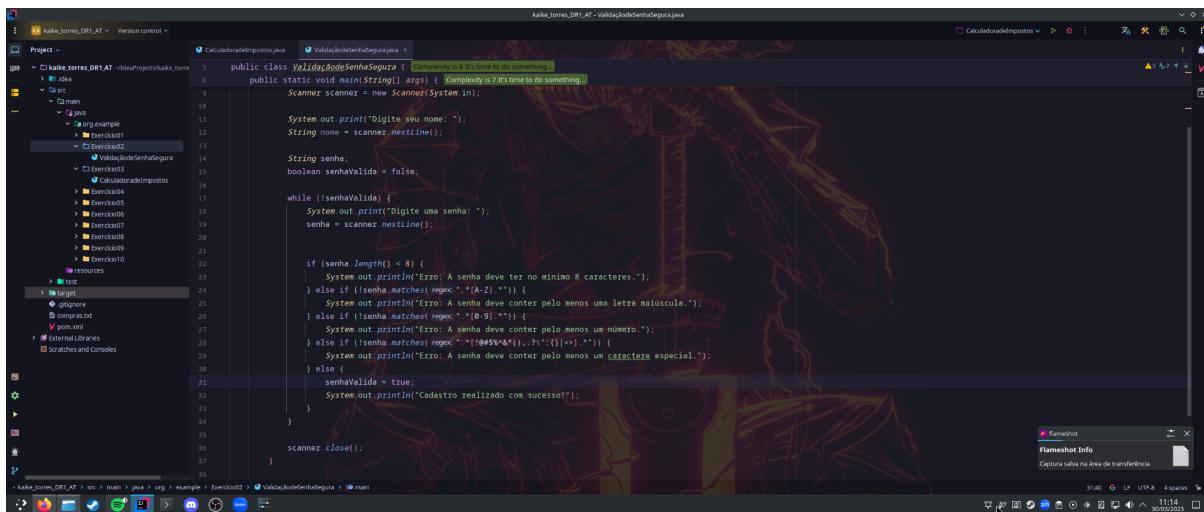
Exercício 1 - Instalando e configurando o ambiente Java



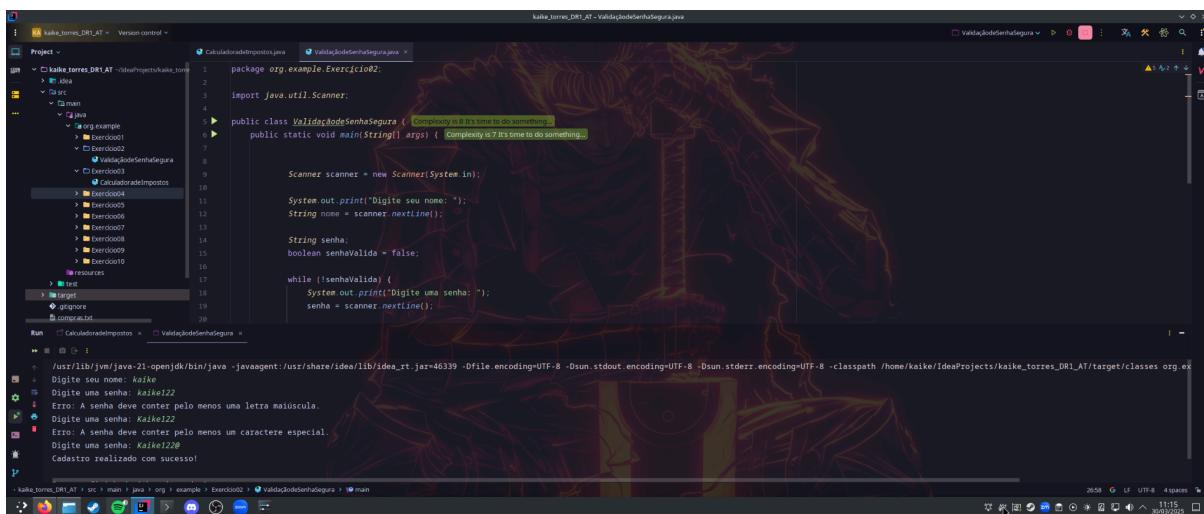
```
package org.example.Exercicio01;
public class OlaMundo {
    public static void main(String[] args) {
        System.out.println("Olá, meu nome é kaike");
    }
}

/usr/lib/jvm/java-21-openjdk/bin/java -javaagent:/usr/share/idea/lib/idea_rt.jar=42731:file.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath /home/kaike/IdeaProjects/kaike_torres_DR1_AT/target/classes org.example.Exercicio01
Olá, meu nome é kaike
Process finished with exit code 0
```

Exercício 2 - Validação de Senha Segura



```
public class ValidacaoDeSenhaSegura {    public static void main(String[] args) {        Scanner scanner = new Scanner(System.in);        System.out.print("Digite seu nome: ");        String nome = scanner.nextLine();        String senha;        boolean senhaValida = false;        while (!senhaValida) {            System.out.print("Digite uma senha: ");            senha = scanner.nextLine();            if (senha.length() < 8) {                System.out.println("Erro: A senha deve ter no minimo 8 caracteres.");            } else if (!senha.matches("[a-zA-Z]*")) {                System.out.println("Erro: A senha deve conter pelo menos uma letra maiúscula.");            } else if (!senha.matches("[0-9]*")) {                System.out.println("Erro: A senha deve conter pelo menos um numero.");            } else if (!senha.matches("[!@#$%^&{}(),.:;~`|<>]*")) {                System.out.println("Erro: A senha deve conter pelo menos um caractere especial.");            } else {                senhaValida = true;                System.out.println("Cadastro realizado com sucesso!");            }        }        scanner.close();    }}
```



```
package org.example.Exercicio02;import java.util.Scanner;public class ValidacaoDeSenhaSegura {    public static void main(String[] args) {        Scanner scanner = new Scanner(System.in);        System.out.print("Digite seu nome: ");        String nome = scanner.nextLine();        String senha;        boolean senhaValida = false;        while (!senhaValida) {            System.out.print("Digite uma senha: ");            senha = scanner.nextLine();            if (senha.length() < 8) {                System.out.println("Erro: A senha deve conter pelo menos 8 caracteres.");            } else if (!senha.matches("[a-zA-Z]*")) {                System.out.println("Erro: A senha deve conter pelo menos uma letra maiúscula.");            } else if (!senha.matches("[0-9]*")) {                System.out.println("Erro: A senha deve conter pelo menos um número.");            } else if (!senha.matches("[!@#$%^&{}(),.:;~`|<>]*")) {                System.out.println("Erro: A senha deve conter pelo menos um caractere especial.");            } else {                senhaValida = true;                System.out.println("Cadastro realizado com sucesso!");            }        }        scanner.close();    }}
```

```
/usr/lib/jvm/java-21-openjdk/bin/java -javaagent:/usr/share/idea/lib/idea_rt.jar=46339:file.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath /home/kaike/IdeaProjects/kaike_torres_DR1_AT/target/classes org.example.Exercicio02Digite seu nome: kaikeDigite uma senha: Kiske123Erro: A senha deve conter pelo menos uma letra maiúscula.Digite uma senha: Kiske123Erro: A senha deve conter pelo menos um caractere especial.Digite uma senha: Kiske123#Cadastro realizado com sucesso!
```

Exercício 3 - Calculadora de Impostos

```
package org.example.Exercicio03;
import java.util.Scanner;

public class CalculadoraDeImpostos { Complexity is 17 You must be kidding
    public static void main(String[] args) { Complexity is 16 You must be kidding

        Scanner scanner = new Scanner(System.in);

        System.out.print("Digite seu nome: ");
        String nome = scanner.nextLine();

        System.out.print("Digite seu salário mensal: R$ ");
        double salarioMensal = scanner.nextDouble();

        double salarioAnual = salarioMensal * 12;

        double imposto = 0.0;

        double salarioLiquido = salarioAnual;

        if (salarioAnual > 22847.76) {
            imposto = 0.08;
        } else if (salarioAnual <= 33919.80) {
            imposto = salarioAnual * 0.075;
        } else if (salarioAnual <= 45812.60) {
            imposto = salarioAnual * 0.15;
        } else {
            imposto = salarioAnual * 0.275;
        }

        salarioLiquido = salarioAnual - imposto;

        System.out.println("Nome: " + nome);
        System.out.println("Salário Anual: R$ " + salarioAnual);
        System.out.println("Imposto de Renda: R$ " + imposto);
        System.out.println("Salário Líquido: R$ " + salarioLiquido);

        scanner.close();
    }
}
```

```
keike_torres_DR1_AT>src>main>java>org>example>Exercicio03>CalculadoraDeImpostos>main
Digite seu nome: keike
Nome: keike
Salário Anual: R$ 240000.0
Imposto de Renda: R$ 66000.0
Salário Líquido: R$ 174000.0
Process finished with exit code 0
```

Exercício 4 - Simulador de Empréstimo Bancário

```

kake_torres_DRT_AT - Version control
Project
src
  java
    org.example
      Exercicio01
      Exercicio02
      Exercicio03
      Exercicio04
        SimuladordeEmprestimoBancario
      Exercicio05
      Exercicio06
      Exercicio07
      Exercicio08
      Exercicio09
      Exercicio10
    CalculadoraImpostos
    test
      gignore
      compact
    .classpath
    .project
  External Libraries
  Scratches and Consoles

SimuladordeEmprestimoBancario.java
public class SimuladordeEmprestimoBancario {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Digite seu nome:");
        String nome = input.nextLine();

        System.out.print("Digite o valor do empréstimo:");
        double emprestimo = input.nextDouble();

        System.out.print("Quantas parcelas voce deseja pagar:");
        int qntParcelas = input.nextInt();

        calcJuros(emprestimo, qntParcelas);
    }

    public static void calcJuros(double emprestimo, int qntParcelas) {
        double jurosMensal = emprestimo * 0.03 * qntParcelas;

        double valorParcelaMensual = (emprestimo + jurosMensal) / qntParcelas;

        System.out.println("Juros mensal fixo: R$" + jurosMensal / qntParcelas);

        System.out.println("Total a Pagar: R$" + (emprestimo + jurosMensal));

        System.out.println("Valor da parcela mensal: R$" + valorParcelaMensual);
    }
}

24.62  LF  UTF-8  4 spaces  11.20  2023-05-29  24.62  LF  UTF-8  4 spaces  11.20  2023-05-29

kake_torres_DRT_AT - Version control
Project
src
  java
    org.example
      Exercicio01
      Exercicio02
      Exercicio03
      Exercicio04
        SimuladordeEmprestimoBancario
      Exercicio05
      Exercicio06
      Exercicio07
      Exercicio08
      Exercicio09
      Exercicio10
    resources
    test
      gignore
      compact
    .classpath
    .project

Run  CalculadoraImpostos  SimuladordeEmprestimoBancario

  1. Dígite seu nome:Kakê
  2. Dígite o valor do empréstimo:1000
  3. Quantas parcelas voce deseja pagar:5
  4. Juros mensal fixo: R$30.0
  5. Total a Pagar: R$1150.0
  6. Valor da parcela mensal: R$230.0

Process finished with exit code 0
  24.62  LF  UTF-8  4 spaces  11.20  2023-05-29

```

Exercício 5 - Criando um Programa CGI em Java

```

kake_torres_DRT_AT - Version control
Project
src
  java
    org.example
      Exercicio01
      Exercicio02
      Exercicio03
      Exercicio04
        SimuladordeEmprestimoBancario
      Exercicio05
        CriandomProgramaCGI.java
      Exercicio06
      Exercicio07
      Exercicio08
      Exercicio09
      Exercicio10
    CalculadoraImpostos
    test
      gignore
      compact
    .classpath
    .project

CriandomProgramaCGI.java
package org.example.Exercicio05;
public class CriandomProgramaCGI {
    public static void main(String[] args) {
        System.out.println("Content-Type: text/html\n");

        System.out.println("<html>");
        System.out.println("  <head><title>Saudação CGI</title></head>");
        System.out.println("  <body>");
        System.out.println("    <h1>Olá, Terráqueos!</h1>");
        System.out.println("  </body>");
        System.out.println("</html>");

    }
}

<html>
<head><title>Saudação CGI</title></head>
<body>
<h1>Olá, Terráqueos!</h1>
</body>
</html>

Process finished with exit code 0
  13:39  LF  UTF-8  4 spaces  11.21  2023-05-29

```

Exercício 6 - Cadastro de Veículos

The screenshot shows the Java code for the `CadastrarVeiculos` class. The code creates two `Veiculo` objects, registers them with 150 km and 300 km respectively, and then prints their details. The code is as follows:

```
package org.example.Exercicio06;
public class CadastrarVeiculos {
    public static void main(String[] args) {
        Veiculo veiculo01 = new Veiculo("ABC-1234", "Fusca", 1975, 120000.5);
        Veiculo veiculo02 = new Veiculo("XYZ-5678", "Civic", 2020, 30000.2);

        veiculo01.registrarViagem(150);
        veiculo01.exibirDetalhes();

        veiculo02.registrarViagem(300);
        veiculo02.exibirDetalhes();
    }
}
```

The screenshot shows the implementation of the `Veiculo` class. It has a constructor that initializes place, modelo, anoFabricacao, and quilometragem. It also has methods to print details and register a trip. The code is as follows:

```
public class Veiculo {
    private String place;
    private String modelo;
    private int anoFabricacao;
    private double quilometragem;

    public Veiculo(String place, String modelo, int anoFabricacao, double quilometragem) {
        this.place = place;
        this.modelo = modelo;
        this.anoFabricacao = anoFabricacao;
        this.quilometragem = quilometragem;
    }

    public void exibirDetalhes() {
        System.out.println("-----");
        System.out.println("Detalhes do Carro:");
        System.out.println("Place: " + place);
        System.out.println("Modelo: " + modelo);
        System.out.println("Ano de Fabricação: " + anoFabricacao);
        System.out.println("Quilometragem: " + quilometragem + " km");
        System.out.println("-----");
    }

    public void registrarViagem(double km) {
        if (km > 0) {
            this.quilometragem += km;
            System.out.println("Viagem registrada com sucesso! Quilometragem atual: " + quilometragem + " km");
        } else {
            System.out.println("Quilometragem inválida! A quantidade de km deve ser positiva.");
        }
    }
}
```

The screenshot shows the execution results of the program. It prints the details of the two vehicles and then registers a trip for each. The output is as follows:

```
Viagem registrada com sucesso! Quilometragem atual: 120150.5 km
-----  
Detalhes do Carro:  
Place: ABC-1234  
Modelo: Fusca  
Ano de Fabricação: 1975  
Quilometragem: 120150.5 km  
  
Viagem registrada com sucesso! Quilometragem atual: 30300.2 km
-----  
Detalhes do Carro:  
Place: XYZ-5678  
Modelo: Civic  
Ano de Fabricação: 2020  
Quilometragem: 30300.2 km
```

Exercício 7 - Gerenciador de Aluno

Aluno.java

```

package org.example.Exercicio07;

public class Aluno {
    String nome;
    String matricula;
    double nota1;
    double nota2;
    double nota3;
}

public Aluno(String nome, String matricula, double nota1, double nota2, double nota3) {
    this.nome = nome;
    this.matricula = matricula;
    this.nota1 = nota1;
    this.nota2 = nota2;
    this.nota3 = nota3;
}

public double calcularMedia() {
    return (nota1 + nota2 + nota3) / 3;
}

public void verificarAprovacao() {
    double media = calcularMedia();
    if (media >= 7) {
        System.out.println(nome + " foi aprovado com média: " + media);
    } else {
        System.out.println(nome + " foi reprovado com média: " + media);
    }
}

```

Alunos.java

```

package org.example.Exercicio07;

import java.util.Scanner;

public class Alunos {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Digite o nome do aluno: ");
        String nome = scanner.nextLine();

        System.out.print("Digite a matrícula do aluno: ");
        String matricula = scanner.nextLine();

        System.out.print("Digite a primeira nota do aluno: ");
        double nota1 = scanner.nextDouble();

        System.out.print("Digite a segunda nota do aluno: ");
        double nota2 = scanner.nextDouble();

        System.out.print("Digite a terceira nota do aluno: ");
        double nota3 = scanner.nextDouble();

        Aluno aluno = new Aluno(nome, matricula, nota1, nota2, nota3);
        aluno.verificarAprovacao();
    }
}

```

Alunos.java

```

public class Alunos {
    public static void main(String[] args) {
        System.out.print("Digite a matrícula do aluno: ");
        String matricula = scanner.nextLine();

        System.out.print("Digite a primeira nota do aluno: ");
        double nota1 = scanner.nextDouble();

        System.out.print("Digite a segunda nota do aluno: ");
        double nota2 = scanner.nextDouble();

        System.out.print("Digite a terceira nota do aluno: ");
        double nota3 = scanner.nextDouble();

        Aluno aluno = new Aluno(nome, matricula, nota1, nota2, nota3);
        aluno.verificarAprovacao();
    }
}

Process finished with exit code 0

```

Exercício 8 - Sistema de Funcionários

```
package org.example.Exercicio08;

class Estagario extends Funcionario {
    public Estagario(String nome, double salarioBase) {
        super(nome, salarioBase);
    }

    @Override 2 usages
    public double calcularSalario() { Complexity is 4 Everything is cool!
        return salarioBase * 0.9;
    }
}
```

```
package org.example.Exercicio08;

class Funcionario {
    String nome; 3 usages 2 inheritors
    double salarioBase; 4 usages

    public Funcionario(String nome, double salarioBase) {
        this.nome = nome;
        this.salarioBase = salarioBase;
    }

    public double calcularSalario() { 2 usages 2 overrides
        return salarioBase;
    }
}
```

```
package org.example.Exercicio08;

public class Gerente extends Funcionario {
    public Gerente(String nome, double salarioBase) {
        super(nome, salarioBase);
    }

    @Override 3 usages
    public double calcularSalario() { Complexity is 3 Everything is cool!
        return salarioBase * 1.2;
    }
}
```

```

keike_torres_DR1_AT - Main.java
package org.example.Exercicio08;
public class Main {
    public static void main(String[] args) {
        Funcionario gerente = new Gerente("Carlos", salarioBase: 5000.0);
        Funcionario estagiario = new Estagiario("Ana", salarioBase: 1500.0);
        System.out.println(gerente nome + " - Salário Final: R$ " + gerente calcularSalario());
        System.out.println(estagiario nome + " - Salário Final: R$ " + estagiario calcularSalario());
    }
}

```

Process finished with exit code 0

Exercício 9 - Conta Bancária com Encapsulamento

```

keike_torres_DR1_AT - ContaBancaria.java
public class ContaBancaria {
    private String titular; // 2 usages
    private double saldo; // 5 usages

    public ContaBancaria(String titular, double saldoInicial) {
        this.titular = titular;
        this.saldo = saldoInicial;
    }

    public void depositar(double valor) { // 1 usage
        if (valor > 0) {
            saldo += valor;
            System.out.println("Depósito de R$ " + valor + " realizado com sucesso.");
        } else {
            System.out.println("Valor de depósito inválido.");
        }
    }

    public void sacar(double valor) { // 2 usages
        if (valor > 0 & valor < saldo) {
            saldo -= valor;
            System.out.println("Saque de R$ " + valor + " realizado com sucesso.");
        } else {
            System.out.println("Saldo insuficiente ou valor inválido.");
        }
    }

    public void exibirSaldo() { // 2 usages
        System.out.println("Saldo atual da conta de " + titular + ": R$ " + saldo);
    }
}

```

```

keike_torres_DR1_AT - Main.java
package org.example.Exercicio09;
public class Main {
    public static void main(String[] args) {
        ContaBancaria conta = new ContaBancaria(titular: "João", saldoInicial: 1000.0);

        conta.exibirSaldo();
        conta.depositar(valor: 500.0);
        conta.sacar(valor: 200.0);
        conta.sacar(valor: 2000.0);
        conta.exibirSaldo();
    }
}

```

Process finished with exit code 0

Exercício 10 - Registro de Compras em Arquivo

Exercício 10 - Loja de Produtos

```

public class Loja {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        String arquivo = "compras.txt";

        try {
            FileWriter writer = new FileWriter(arquivo);

            for (int i = 1; i <= 3; i++) {
                System.out.print("Digite o nome do produto de compra " + i + ": ");
                String produto = scanner.nextLine();

                System.out.print("Digite a quantidade do produto: ");
                int quantidade = scanner.nextInt();

                System.out.print("Digite o preço unitário do produto: ");
                double precoUnitario = scanner.nextDouble();
                scanner.nextLine();

                writer.write("Produto: " + produto + ", Quantidade: " + quantidade + ", Preço Unitário: R$ " + precoUnitario + "\n");
            }

            writer.close();
        }

        System.out.println("\nCompras registradas no arquivo!");
    }
}

```

Output:

```

Produto: xbox, Quantidade: 6, Preço Unitário: R$ 1200.0
Produto: pc, Quantidade: 2, Preço Unitário: R$ 3000.0
Produto: teclado, Quantidade: 56, Preço Unitário: R$ 140.0

```

Exercício 11 - Simulação de Loteria

Exercício 11 - Simulação de Loteria

```

public class SimulacaoLoteria {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Random random = new Random();

        int[] numerosSorteados = new int[6];
        for (int i = 0; i < 6; i++) {
            int numeroSorteado;
            do {
                numeroSorteado = random.nextInt(60) + 1;
            } while (jáFoiSorteados(numerosSorteados, numeroSorteado, i));
            numerosSorteados[i] = numeroSorteado;
        }

        int[] numerosUsuario = new int[6];
        System.out.println("Digite 6 números entre 1 e 60:");
        for (int i = 0; i < 6; i++) {
            System.out.print("Digite o " + (i + 1) + "º número: ");
            numerosUsuario[i] = scanner.nextInt();
        }

        int acertos = 0;
        for (int i = 0; i < 6; i++) {
            for (int j = 0; j < 6; j++) {
                if (numerosUsuario[i] == numerosSorteados[j]) {
                    acertos++;
                    break;
                }
            }
        }
    }
}

```

```
public class SimulacaoLoteria { Complexity is 20 You must be kidding
    public static void main(String[] args) { Complexity is 19 You must be kidding
        for (int i = 0; i < limite; i++) {
            if (sorteados[i] == numero) {
                return true;
            }
        }
        return false;
    }
}
```

```
Digitte 6 numeros entre 1 e 60:
Digite o 1º numero: 40
Digite o 2º numero: 21
Digite o 3º numero: 3
Digite o 4º numero: 43
Digite o 5º numero: 1
Digite o 6º numero: 7

Numeros sorteados:
22 10 36 48 58 6

Seus numeros:
40 21 3 43 1 7

Voce acertou @ numero(s).
```

Exercício 12 - Sistema de Chat Simples com Arrays

```
public class SistemaChatSimple { Complexity is 7 It's time to do something...
    public static void main(String[] args) { Complexity is 6 It's time to do something...
        Scanner scanner = new Scanner(System.in);

        System.out.print("Digite o nome do primeiro usuário: ");
        String usuario1 = scanner.nextLine();
        System.out.print("Digite o nome do segundo usuário: ");
        String usuario2 = scanner.nextLine();

        String[] mensagens = new String[10];

        int index = 0;
        String mensagem;

        for (int i = 0; i < 5; i++) {
            System.out.print(usuario1 + ", digite sua mensagem: ");
            mensagem = scanner.nextLine();
            mensagens[index++] = usuario1 + ":" + mensagem;

            System.out.print(usuario2 + ", digite sua mensagem: ");
            mensagem = scanner.nextLine();
            mensagens[index++] = usuario2 + ":" + mensagem;
        }

        System.out.println("\n***** Histórico de Mensagens *****");
        for (String msg : mensagens) {
            System.out.println(msg);
        }
    }
}
```

Conclusão

Ao final deste exercício, você terá configurado seu ambiente de desenvolvimento, criado programas básicos em Java e utilizado conceitos fundamentais como classes, objetos, e manipulação de dados. Você também será capaz de validar entradas, realizar cálculos lógicos e simular sistemas interativos. Esses conhecimentos são essenciais para avançar no aprendizado de Java e para a criação de soluções reais e eficientes em ambientes de desenvolvimento profissional.