Tarefa 1

***Informações Importantes:***

* Anexar o arquivo .zip no SIGAA até a data estabelecida no sigaa
* Não será possível reabrir o SIGAA para reenvio de atividades, então, fiquem atentos aos prazos.

Questão 1) Analise os códigos abaixo e compare-os:

# Exemplo1 de Código Procedural em C: Função para cadastrar um usuário e imprimir seu nome e email

#include <stdio.h>

void cadastrarUsuario(char nome[], char email[]) { printf("Cadastrando usuário: %s\n", nome); printf("Email: %s\n", email);

}

int main() {

char nome[] = "João";

char email[] = "[joao@example.com](mailto:joao@example.com)";

cadastrarUsuario(nome, email); return 0;

}

# Exemplo2 de Código Procedural em Java: Cadastro de Usuário e imprimir seu nome e email

public class CadastroUsuario {

public static void main(String[] args) { String nome = "João";

String email = "[joao@example.com](mailto:joao@example.com)"; cadastrarUsuario(nome, email);

}

public static void cadastrarUsuario(String nome, String email) { System.out.println("Cadastrando usuário: " + nome);

System.out.println("Email: " + email);

}

}

# Exemplo3 de Código Orientado a Objetos em Java: Cadastro de Usuário e imprimir seu nome e email

public class Usuario { private String nome; private String email;

public Usuario(String nome, String email) { this.nome = nome;

this.email = email;

}

public void cadastrar() { System.out.println("Cadastrando usuário: " + nome); System.out.println("Email: " + email);

}

}

public class CadastroUsuario {

public static void main(String[] args) {

Usuario usuario = new Usuario("João", "[joao@example.com](mailto:joao@example.com)"); usuario.cadastrar();

}

}

Responda às seguintes perguntas:

1. Quais são as principais diferenças entre os 3 códigos (procedural e POO)?
   * A Diferença entre os 3 códigos é que o método de procedural se concentra em procedimentos ou metodos específicos, para realizar uma tarefa especifica, e o programa e dividido em partes especificas, mas as variaveis sao globais e podem ser acessadas por qualquer método. Também é mais fácil de entender. Já em POO, algumas partes podem ser reutilizadas no decorrer do codigo como em “Usuário”.
2. Como a POO melhora a organização e a escalabilidade do sistema?
   * O POO na organização melhora em 3 condições, encapsulamento que é onde pode juntar dados e metodos em objetos, tornando o codigo mais organizado. Nas abstrações, onde permite abstrair os detalhes de implementação, focando mais na funcionalidade e propriedade dos objetos. Nas hierarquias de classes, onde uma classe herda as propriedades e comportamentos das classes mais gerais. Já na escalabilidade, as partes do codigo podem ser reaproveitadas, criando objetos que podem ser usados em diferentes partes do sistema. Há também uma flexibilidade, onde objetos podem ser facilmente estendidos e modificados sem prejuízo para o resto do sistema. Alem de que é de fácil manutenção para o desenvolvedor.
3. Quais são os benefícios de usar objetos no lugar de funções simples?
   * Os Benefícios de usar objetos no lugar de funções simples, são as manutenções e escalabilidade, a reutilização do código sem que haja repetição, assim o sistema não precisa se esforçar tanto. Flexibilidade, sendo mais fácil de estender e modificar sem afetar o sistema, alem da facilidade na manutenção. Alem de tudo, é muito mais fácil de organizar de forma logica, como também tem melhor legibilidade, tornando mais fácil de entender. Há também a abstração, que permite ocultar detalhes na implementação, concentrando nas funcionalidades e propriedades.

Questão 2) Modifique o código POO Exemplo3 para incluir um segundo usuário, criando um método para alterar o e-mail de um usuário.

Link para a atividade no GitHub: https://github.com/ElsonJr01/AtividadesJava/tree/main/Atividade%202%20Primeira%20Tarefa%20POO

Figura 1:

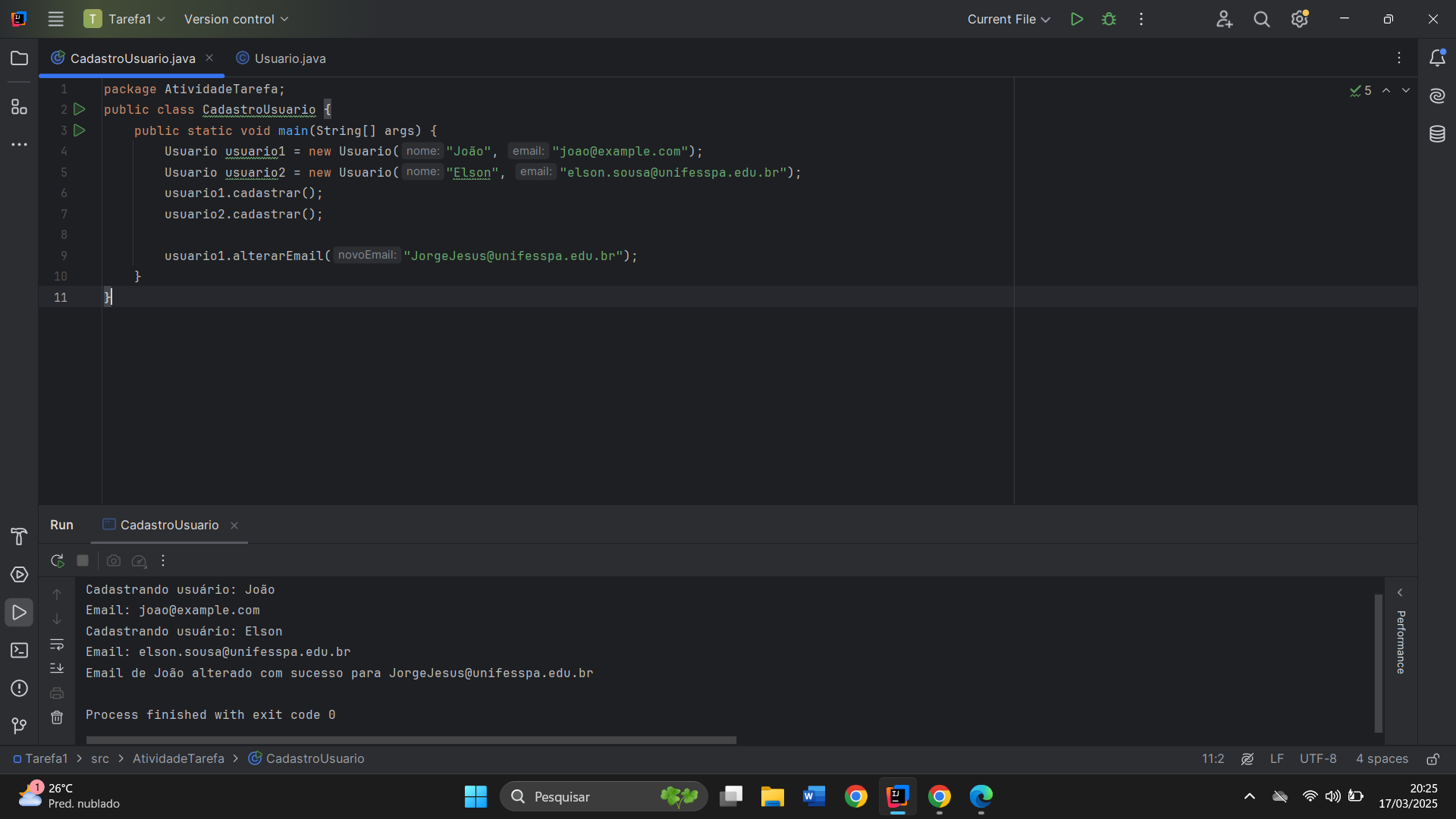
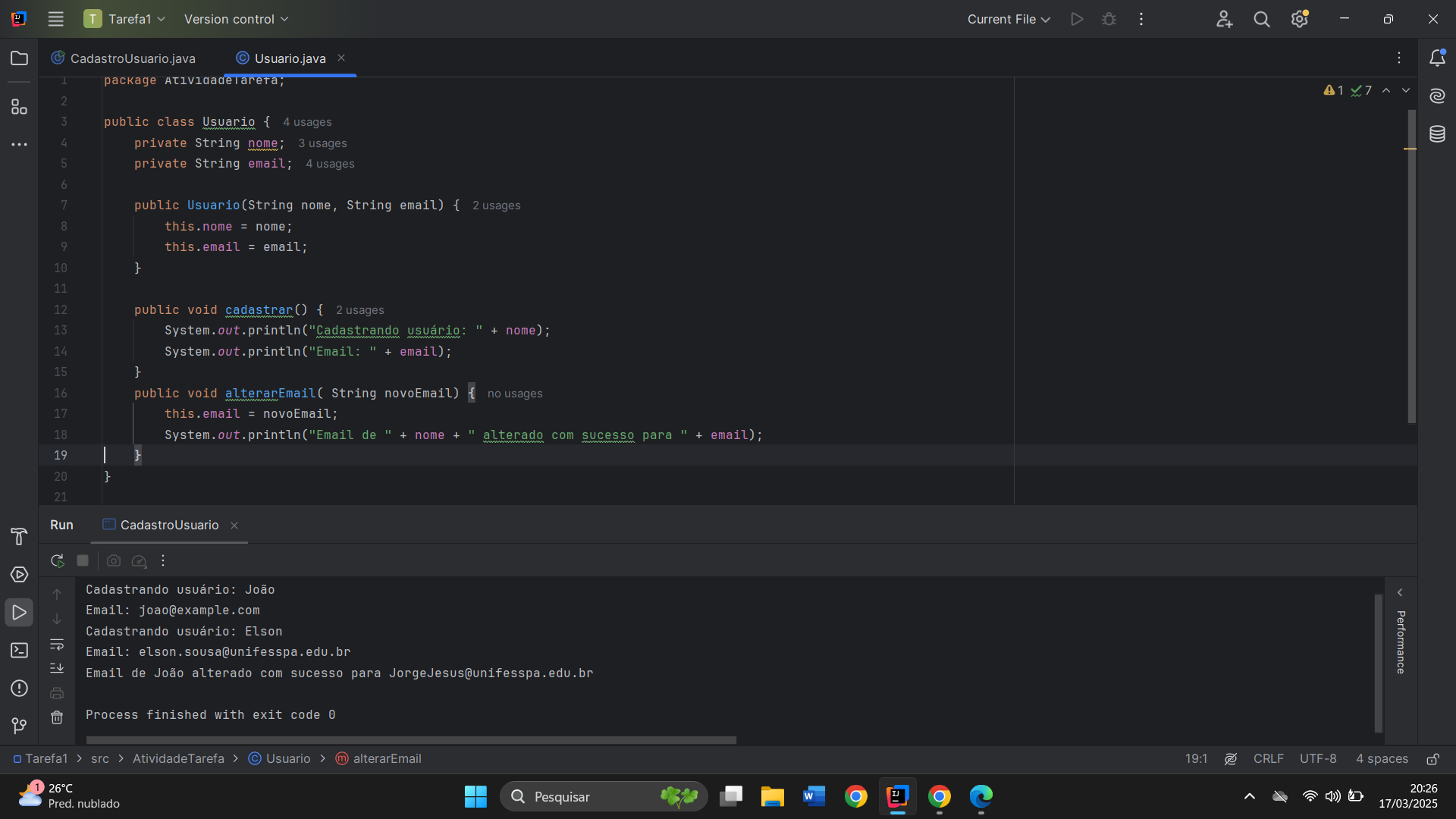


Figura 2:



Questão 3)Explique os tipos de dados primitivos em Java. Dê exemplos de como utilizar cada um desses tipos no código, e descreva as diferenças entre tipos de dados inteiros,

ponto flutuante, booleanos e caracteres. Além disso, explique como a conversão entre tipos de dados (casting) pode ser realizada em Java, dando um exemplo prático.

R: Em java assim como em outras linguagens, existem os tipos primitivos que podem ser Int, para números inteiros sem virgula ou casas decimais, floats que são totalmente o contrario de int. Long que assim como Int, tem uma unica diferença é que suporta mais dados, e double que suporta mais dados com casas decimais. Também temos Boolean que pode operar como atribuição logica, true ou false. Alem disso, existem char que trabalha com um único caracter e String que pode trabalhar com textos, mas não é um tipo primitivo, e sim uma classe de atribuição.

Link para a atividade no GitHub: https://github.com/ElsonJr01/AtividadesJava/tree/main/Questao3PrimeiraTarefaPOO

Figura com exemplos de tipos primitivos:

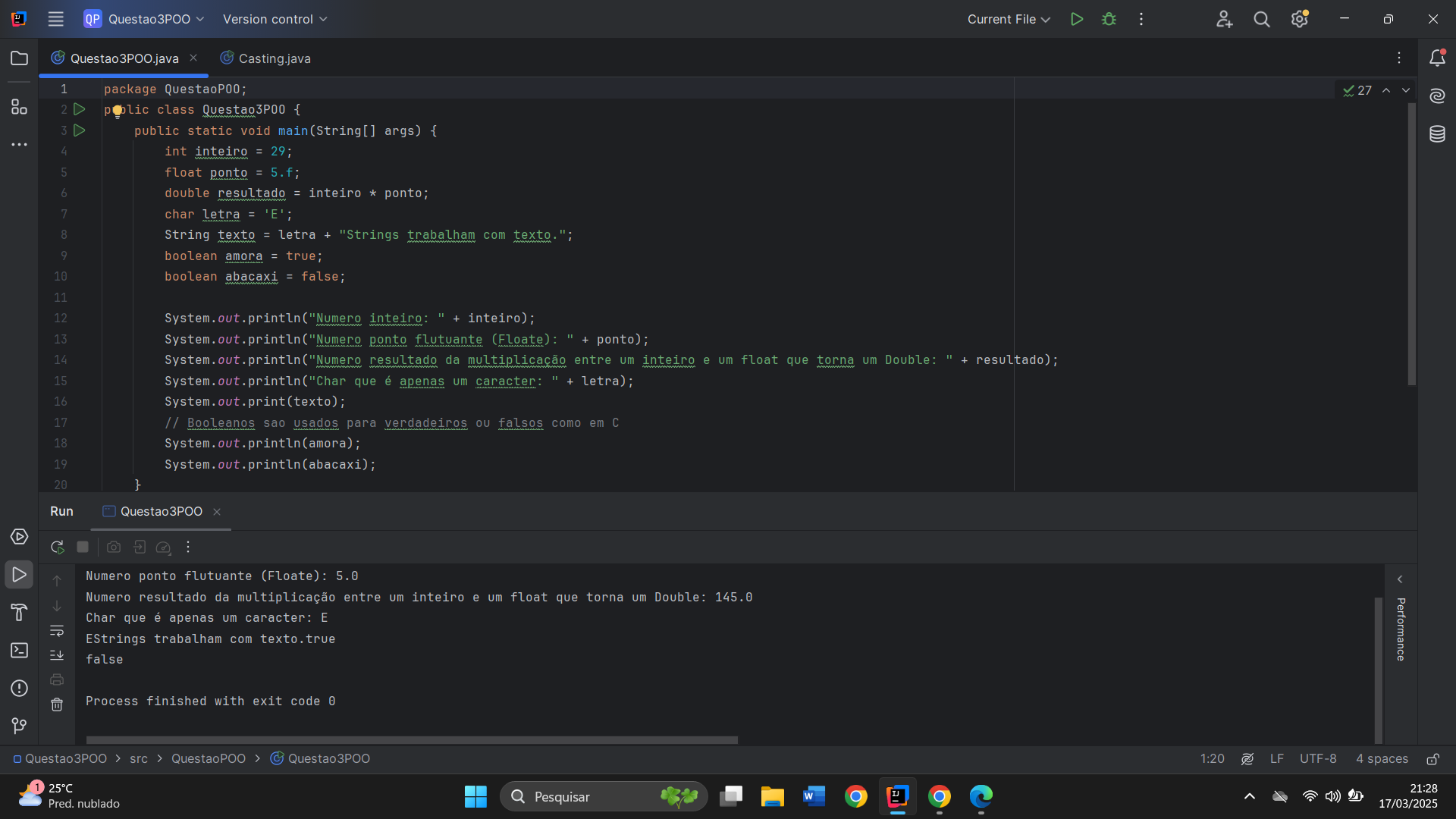
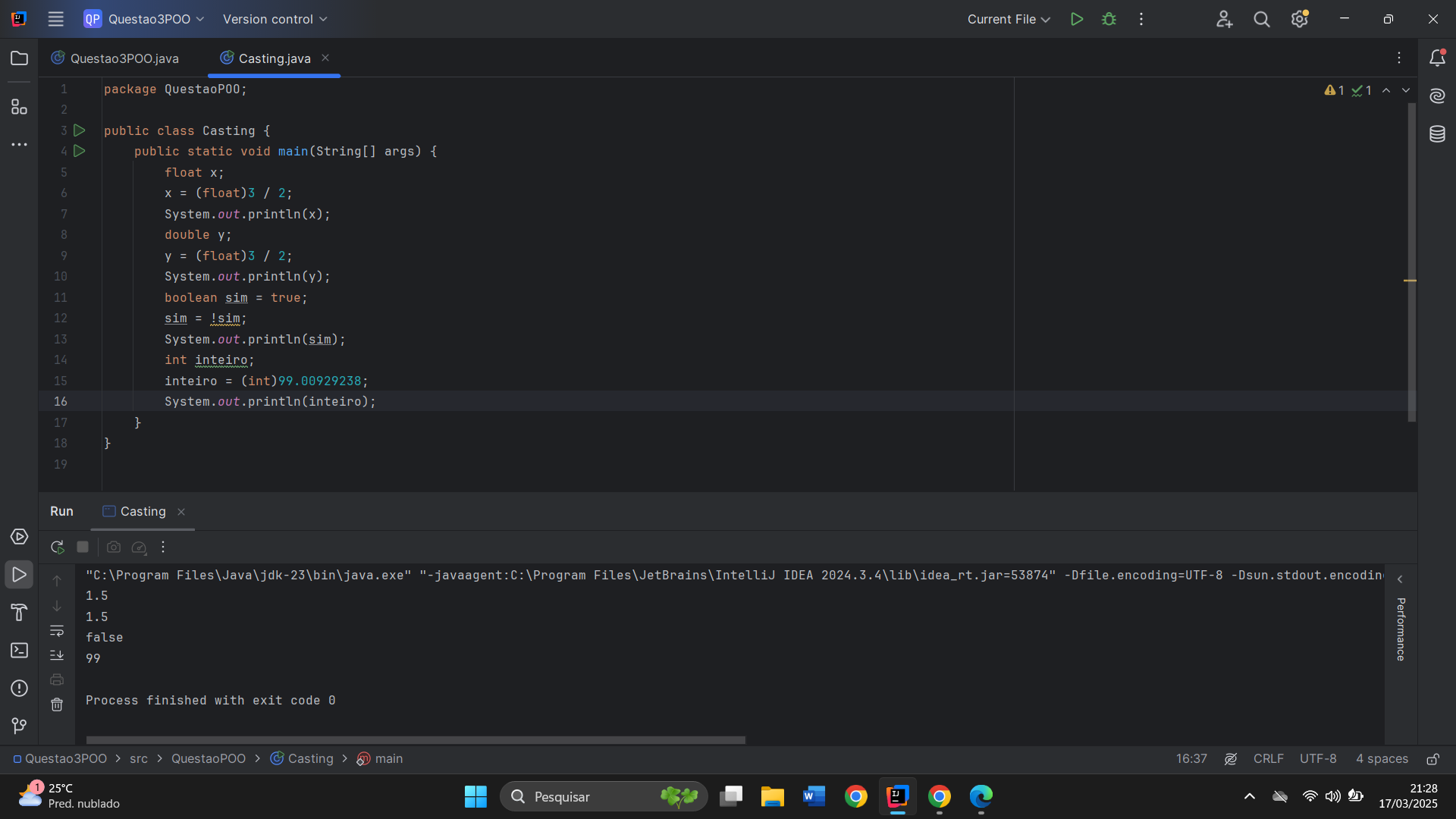


Figura com exemplos de Casting:

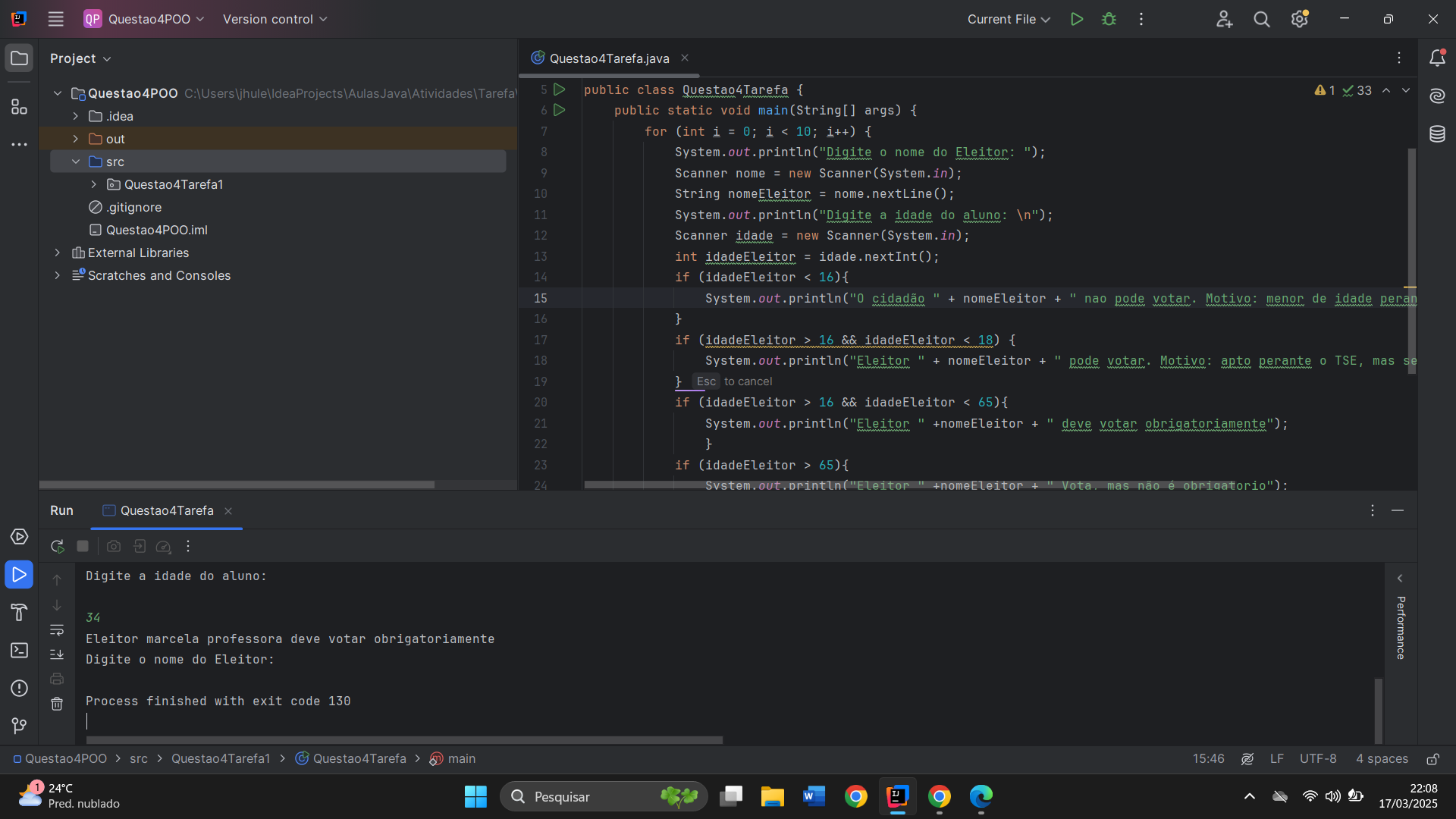


Questão 4) Explique o funcionamento das estruturas condicionais if, else if e else em Java. Em seguida, crie um exemplo de código onde um programa decide, com base na idade do usuário, se ele pode votar ou não. A idade mínima para votar é 16 anos, mas o voto é obrigatório apenas para maiores de 18 anos.

R: a estrutura condicional If e Else, é uma estrutura que me dá a opção de fazer comparações entre diferentes elementos, seja booleano, inteiro, float ou qualquer outro tipo primitivo.

Link para a questão 4 salva no GitHub: https://github.com/ElsonJr01/AtividadesJava/tree/main/Questao4Tarefa1

Figura que mostra o algoritmo feito:

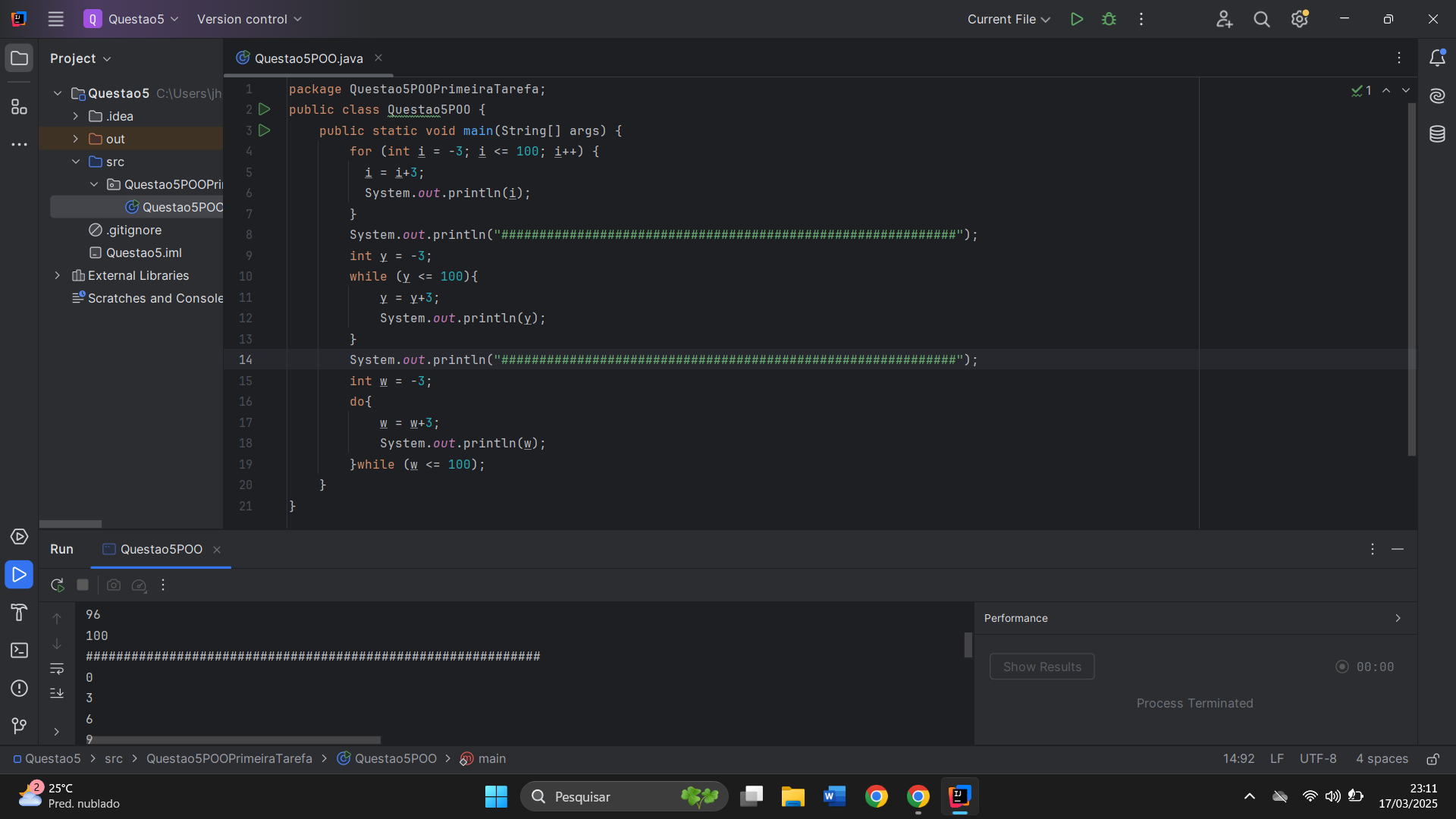


Questão 5) Discuta os diferentes tipos de laços de repetição em Java: for, while e do-while. Em seguida, escreva um programa que, utilizando um laço, exiba os números de 1 a 100 que são divisíveis por 3. Explique qual seria a diferença caso você usasse um laço for em vez de um while para resolver este problema.

R: Os laços de repetição são muito úteis em várias situações, dependendo do que a gente precisa fazer. O for, por exemplo, é ótimo para trabalhar com matrizes e outras estruturas onde a repetição tem um número definido de vezes, já que ele conta e incrementa automaticamente. O while é mais flexível e funciona bem quando a repetição depende de uma condição lógica, mas se essa condição for falsa desde o início, ele nem chega a rodar. Já o do while garante que o bloco de código seja executado pelo menos uma vez, mesmo que a condição já seja falsa logo de cara.

Link do GitHub para o acesso ao repositório com a questão 5: <https://github.com/ElsonJr01/AtividadesJava/tree/main/Questao5POOPrimeiraTarefa>

Imagem da tela com a atividade feita:



Questão 6) Explique como o tipo String funciona em Java. Discuta suas características, como imutabilidade, e as principais operações que podem ser realizadas com strings, como concatenar e comparar. Escreva um código que leia uma palavra do usuário e verifique se ela contém a letra "a" (em qualquer posição). Se a letra "a" for encontrada, o programa deve exibir "A letra 'a' foi encontrada!" e, caso contrário, "A letra 'a' não foi encontrada!".

R: Em Java, String é um tipo de dado especial, representando uma sequência de caracteres. Ele é uma classe, e não um tipo primitivo, mas pode ser usado de forma semelhante a tipos primitivos devido a otimizações internas da linguagem. Strings não podem ser alteradas, qualquer alteração que seja gera um novo objeto ao invés de alterar.

Link para o código pedido na questão: https://github.com/ElsonJr01/AtividadesJava/tree/main/Questao6Tarefa1POO

Imagem da atividade sendo executada:

