

1. Resposta Desafio de Conversor

```
public class ConversorHorario {
    public static String converterHorario12Para24(String horario12) {
        // Divide o horário em partes: hora, minuto, segundo e AM/PM
        String[] partes = horario12.split(":");
        int hora = Integer.parseInt(partes[0]);
        int minuto = Integer.parseInt(partes[1]);
        int segundo = Integer.parseInt(partes[2].substring(0, 2)); // Remove "AM" ou "PM" e
        pega os dois primeiros caracteres

        // Verifica se é AM ou PM
        String periodo = partes[2].substring(2); // Pega "AM" ou "PM"
        if (periodo.equals("PM") && hora != 12) {
            hora += 12;
        } else if (periodo.equals("AM") && hora == 12) {
            hora = 0;
        }

        // Formata o horário de 24 horas
        String horario24 = String.format("%02d:%02d:%02d", hora, minuto, segundo);

        return horario24;
    }

    public static void main(String[] args) {
        String horario12 = "06:15:25PM";
        String horario24 = converterHorario12Para24(horario12);
        System.out.println(horario24); // Deve imprimir "18:15:25"
    }
}
```

2. Resposta Desafio: Quantidade de palavras

```
public class CountCamelCaseWords {
    public static void main(String[] args) {
        String input =
"façaMercadoNolfoodEntregamosTudoOQueVocêPrecisaNaPortaDaSuaCasa";
        int wordCount = countCamelCaseWords(input);
        System.out.println("Número de palavras: " + wordCount);
    }

    public static int countCamelCaseWords(String input) {
        if (input == null || input.isEmpty()) {
```

```

        return 0;
    }

    int wordCount = 1; // Começamos com 1, pois a primeira palavra começa em
    minúscula.

    for (int i = 0; i < input.length(); i++) {
        char currentChar = input.charAt(i);
        if (Character.isUpperCase(currentChar)) {
            wordCount++;
        }
    }

    return wordCount;
}

```

Número de palavras: 9

3. Desafio: Número Solitário

```

import java.util.*;

public class EncontreNumeroSolitario {
    public static void main(String[] args) {
        int[] numeros = {12, 17, 15, 19, 22, 17, 19, 12};
        int[] numerosSolitarios = encontrarNumerosSolitarios(numeros);

        System.out.println("Números Solitários:");
        for (int numero : numerosSolitarios) {
            System.out.println(numero);
        }
    }

    public static int[] encontrarNumerosSolitarios(int[] numeros) {
        Map<Integer, Integer> contador = new HashMap<>();

        for (int numero : numeros) {
            contador.put(numero, contador.getDefault(numero, 0) + 1);
        }

        List<Integer> numerosSolitarios = new ArrayList<>();
        for (Map.Entry<Integer, Integer> entry : contador.entrySet()) {
            if (entry.getValue() == 1) {

```

```

        numerosSolitarios.add(entry.getKey());
    }
}

int[] resultado = new int[numerosSolitarios.size()];
for (int i = 0; i < numerosSolitarios.size(); i++) {
    resultado[i] = numerosSolitarios.get(i);
}

return resultado;
}
}

```

Números Solitários:

15

22

4. Resposta: Qual é a saída do código a seguir?

A saída do código será:

e. 1 1 2 1 2 3 1 2 3 4

5. Qual é o resultado da expressão da booleana abaixo?

a. true

6. Qual o valor da saída do código abaixo?

b) 2