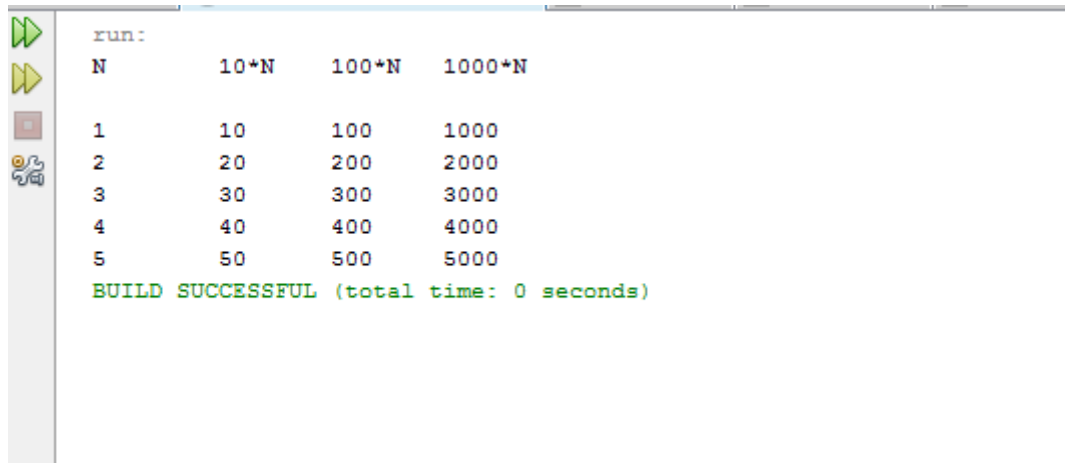


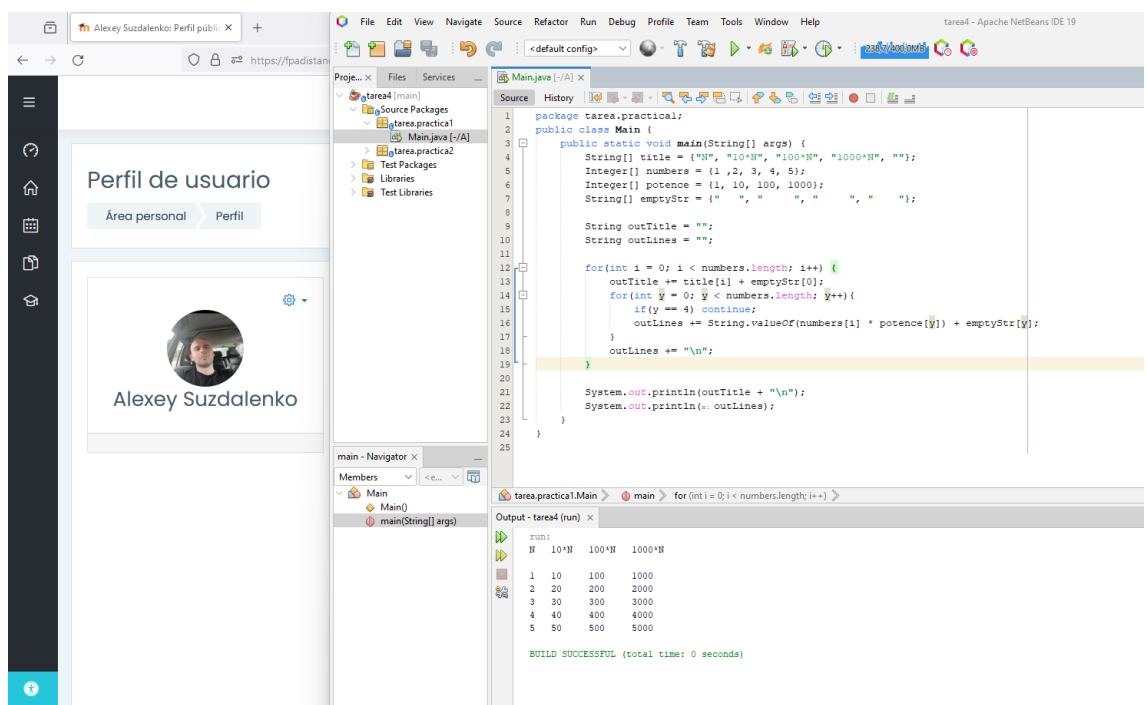
PROGRAMACIÓN 1º DESARROLLO DE APLICACIONES MULTIPLATAFORMA

Parte práctica

1. Escriba una aplicación en Java que utilice ciclos para imprimir la siguiente tabla de valores **(1,6 puntos)**



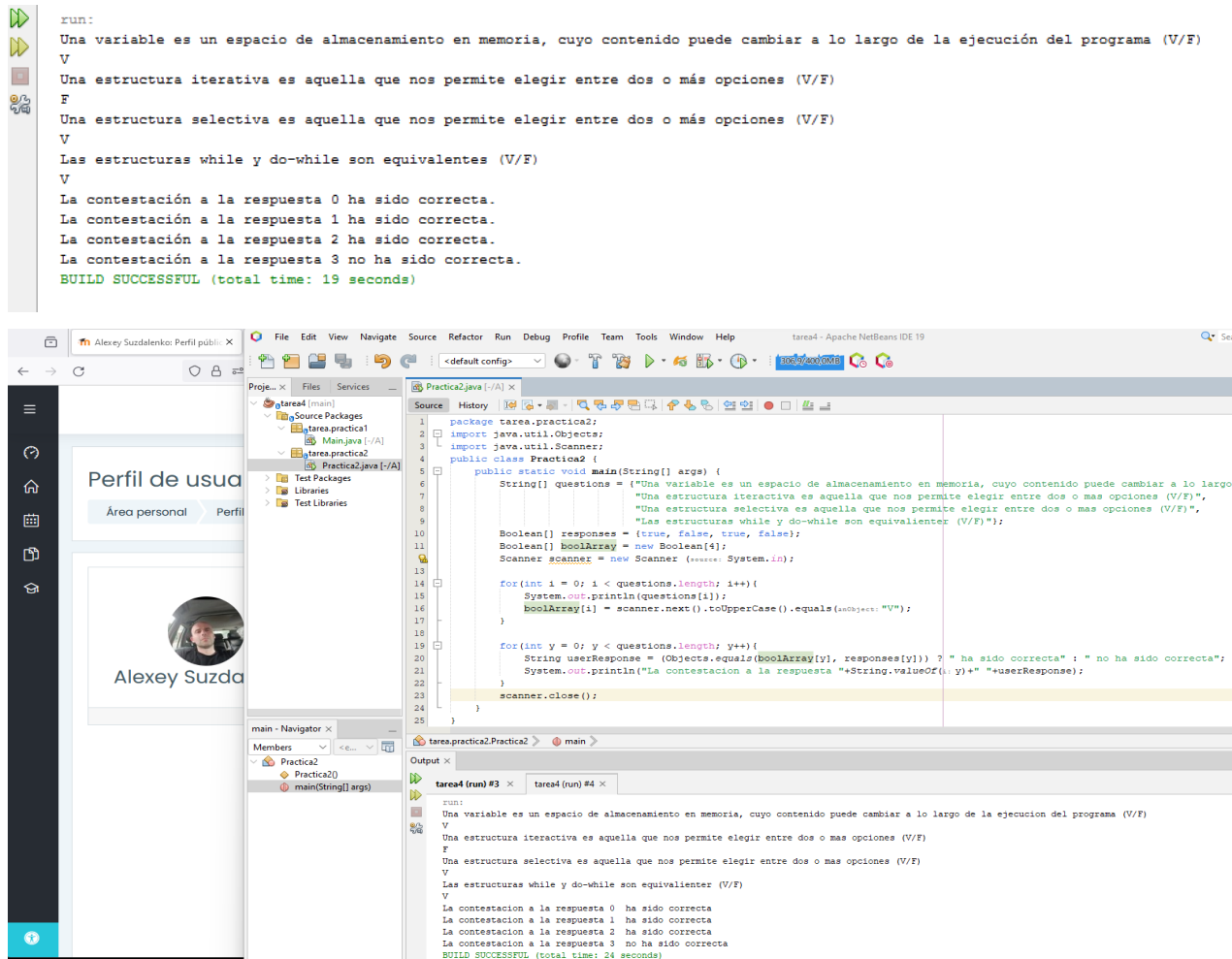
```
run:
N      10*N    100*N   1000*N
1      10      100     1000
2      20      200     2000
3      30      300     3000
4      40      400     4000
5      50      500     5000
BUILD SUCCESSFUL (total time: 0 seconds)
```



2. Realice una aplicación en Java que realice cuatro preguntas que deban responderse con Verdadero/Falso (V/F), bien mediante el carácter o bien mediante la cadena. Una vez leído si la respuesta es verdadera o falsa, el programa deberá almacenar en un **array booleano** si se contestó bien o mal (true para bien, false para mal) la

pregunta para después mostrarlo por pantalla (**mediante un bucle**).
(1,6 puntos)

Ejemplo de ejecución (última pregunta contestada mal aposta para comprobar que el programa funciona)



```
run:
Una variable es un espacio de almacenamiento en memoria, cuyo contenido puede cambiar a lo largo de la ejecución del programa (V/F)
V
Una estructura iterativa es aquella que nos permite elegir entre dos o más opciones (V/F)
F
Una estructura selectiva es aquella que nos permite elegir entre dos o más opciones (V/F)
V
Las estructuras while y do-while son equivalentes (V/F)
V
La contestación a la respuesta 0 ha sido correcta.
La contestación a la respuesta 1 ha sido correcta.
La contestación a la respuesta 2 ha sido correcta.
La contestación a la respuesta 3 no ha sido correcta.
BUILD SUCCESSFUL (total time: 19 seconds)
```

```
package tarea.practica2;
import java.util.Objects;
import java.util.Scanner;
public class Practica2 {
    public static void main(String[] args) {
        String[] questions = {"Una variable es un espacio de almacenamiento en memoria, cuyo contenido puede cambiar a lo largo de la ejecución del programa (V/F)",
                                "Una estructura iterativa es aquella que nos permite elegir entre dos o más opciones (V/F)",
                                "Una estructura selectiva es aquella que nos permite elegir entre dos o más opciones (V/F)",
                                "Las estructuras while y do-while son equivalentes (V/F)"};
        Boolean[] responses = {true, false, true, false};
        Boolean[] boolArray = new Boolean[4];
        Scanner scanner = new Scanner(System.in);
        for (int i = 0; i < questions.length; i++) {
            System.out.println(questions[i]);
            boolArray[i] = scanner.next().toUpperCase().equals("V");
        }
        for (int y = 0; y < questions.length; y++) {
            String userResponse = (Objects.equals(boolArray[y], responses[y])) ? "ha sido correcta" : "no ha sido correcta";
            System.out.println("La contestación a la respuesta " + String.valueOf(y) + " es " + userResponse);
        }
        scanner.close();
    }
}
```

3. Una empresa grande paga a sus vendedores mediante comisiones. Los vendedores reciben 200 Euros por semana, más el 9% de sus ventas brutas durante la semana. Por ejemplo, un vendedor que vende 5000 Euros de mercancía en una semana, recibe 200 euros más el 9% de 5000, o un total de 650 Euros. Usted acaba de recibir una lista de los artículos vendidos por cada vendedor. No hay límite en cuanto al número de artículos que un vendedor puede vender.
(1,6 puntos)

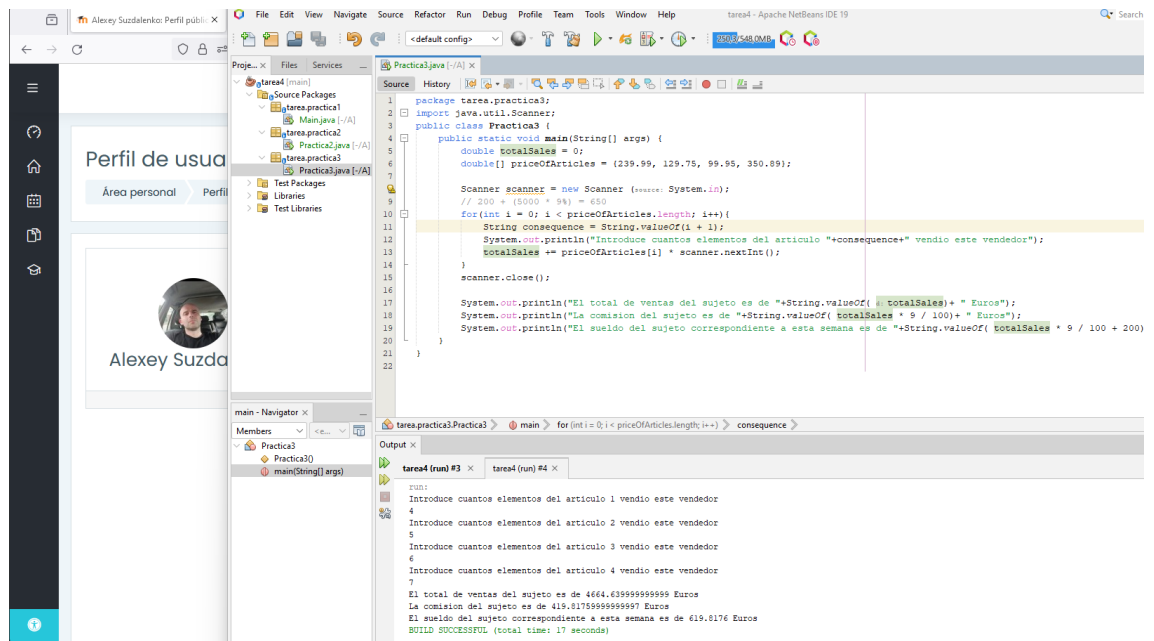
Artículo	Valor
Artículo 1	239,99
Artículo 2	129,75
Artículo 3	99,95
Artículo 4	350,89

Ejemplo de ejecución del programa:

```

run:
Introduce cuántos elementos del artículo 1 vendió este vendedor
4
Introduce cuántos elementos del artículo 2 vendió este vendedor
5
Introduce cuántos elementos del artículo 1 vendió este vendedor
6
Introduce cuántos elementos del artículo 4 vendió este vendedor
7
El total de ventas del sujeto es de 4664.6399999999999 Euros
La comisión del sujeto es de 419.81759999999997 Euros
El sueldo del sujeto correspondiente a esta semana es de 619.8176 Euros
BUILD SUCCESSFUL (total time: 7 seconds)

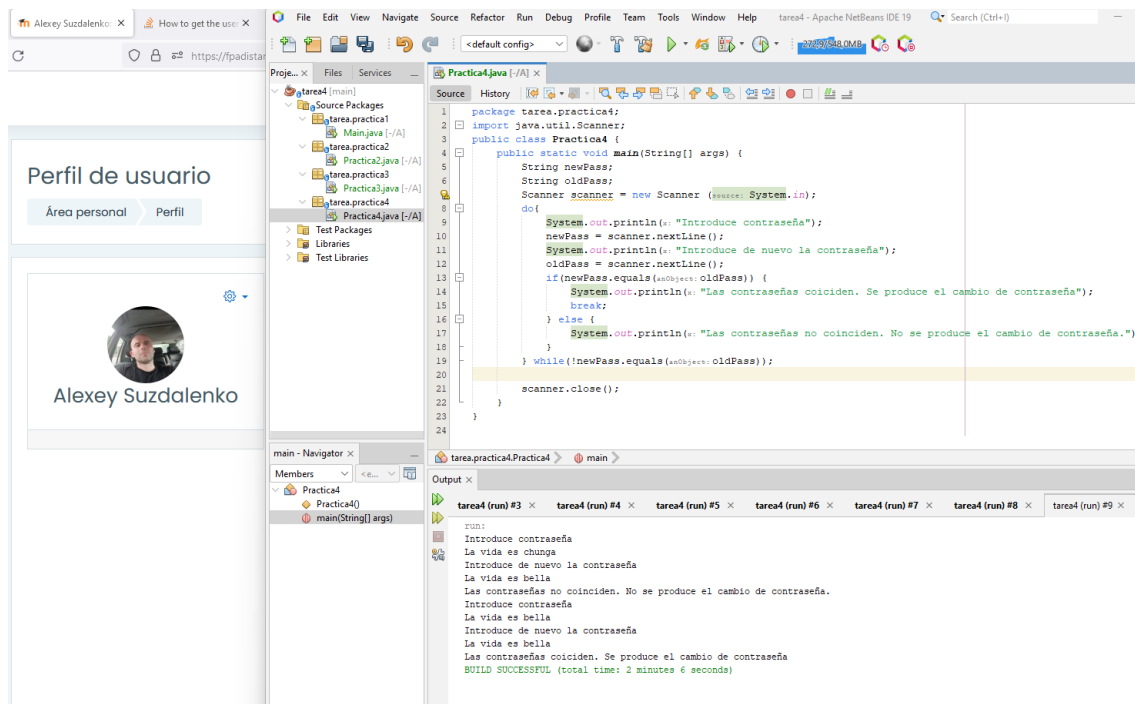
```



4. Escriba una aplicación que me pida una contraseña, después pida volver a introducirla para confirmarla y, por último, compare ambas

cadenas. Si el resultado es que las dos cadenas son iguales, deberá sacar un mensaje informándonos de ello. Si no, deberá volver a pedirnos la contraseña y su confirmación (**1.6 puntos**). Ejemplo de ejecución:

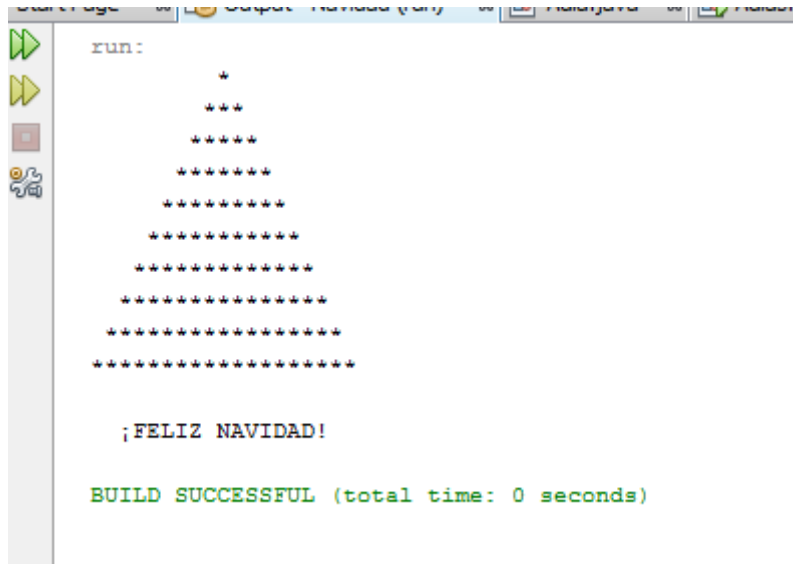
```
run:
introduce contraseña
La vida es chungu
Introduce de nuevo la contraseña
La vida es bella
Las contraseñas no coinciden. No se procede al cambio de contraseña
introduce contraseña
La vida es bella
Introduce de nuevo la contraseña
La vida es bella
Las contraseñas coinciden. se procede al cambio de contraseña
BUILD SUCCESSFUL (total time: 23 seconds)
```



5. Escriba una aplicación que me saque por pantalla la siguiente figura de felicitación navideña. Puede utilizar instrucciones de salida que impriman un solo asterisco (*), una sola línea o un solo carácter de

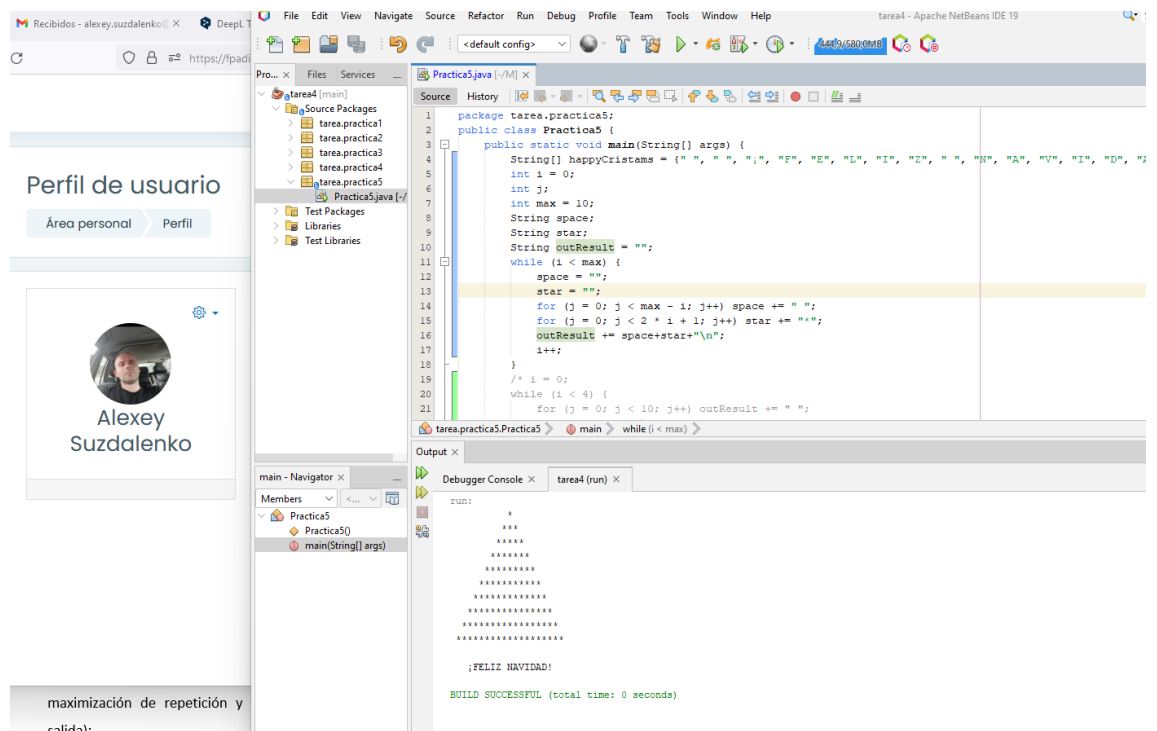
nueva línea. Maximice el uso de la repetición (con instrucciones **for** anidadas), y minimice el número de instrucciones de salida (**1 punto**).

Ejemplo de ejecución



```
run:
  *
 ***
 *****
 *******
          *
         **
        ***
       ****
      *****
     *******
    *********
   **********
  **********
 *FELIZ NAVIDAD!

BUILD SUCCESSFUL (total time: 0 seconds)
```



Ahora voy a modificar el código para ajustarme a la segunda imagen (versión) del árbol de navidad

```
run:  
  
      *  
    ***  
   *****  
  *********  
 ****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
  
      *  
      *  
      *  
      *  
*****  
* ;FELIZ NAVIDAD! *  
*****  
  
BUILD SUCCESSFUL (total time: 0 seconds)
```

