Tarea PSP01

Yanet López Rodríguez
29/11/2023

Indice

Introducción

En esta actividad deberemos realizar las siguientes tres partes:

Primera parte: implementa una aplicación que ordena un conjunto indeterminado de números que recibe a través de su entrada estándar; y muestra el resultado de la ordenación en su salida estándar. La aplicación se llamará 'ordenarNumeros'.

Segunda parte: implementa una aplicación, llamada 'aleatorios', que genere al menos 40 números aleatorios (entre 0 y 100), y que los escriba en su salida estándar.

Tercera parte: Realiza un pequeño manual (tipo "¿Cómo se hace?" o "HowTo"), utilizando un editor de textos (tipo word o writer) en el que indiques, con pequeñas explicaciones y capturas, cómo has probado la ejecución de las aplicaciones que has implementado en este ejercicio. Entre las pruebas que hayas realizado, debes incluir una prueba en la que utilizando el operador "|" (tubería) redirijas la salida de la aplicación 'aleatorios' a la entrada de la aplicación 'ordenarNumeros'.

En este archivo haremos un manual en el que explicaremos cada código y los probaremos.

Primera parte: ***implementa una aplicación que ordena un conjunto indeterminado de números que recibe a través de su entrada estándar; y muestra el resultado de la ordenación en su salida estándar. La aplicación se llamará 'ordenarNumeros'.***

Para realizar esta actividad lo primero que haremos será crear un nuevo proyecto, en el tendremos nuestra clase *Main* y en ella tendremos *main*(), *ordenarNumeros*() y *pedirNumeros*(). Desde *main*() lo único que haremos será llamar a la función *ordenarNumeros*() pasándole *pedirNumeros*() como parámetro ya que esta función retornará el array de enteros y poner un salto de línea para que al ejecutarlo en la terminal quede más bonito.

```
public class Main {
    public static void main(String[] args) {
        //DE ESTA MANERA LLAMAREMOS A LA FUNCIÓN pedirNumeros() PARA PEDIR LOS DATOS POR TECLARO
        //Y LUEGO LOS ORDENAREMOS USANDO LA FUNCIÓN ordenarNumeros()
        ordenarNumeros(pedirNumeros());

        //PONEMOS UN SALTO DE LÍNEA PARA QUE QUEDE MAS BONICO
        System.out.println("\n");
    }
```

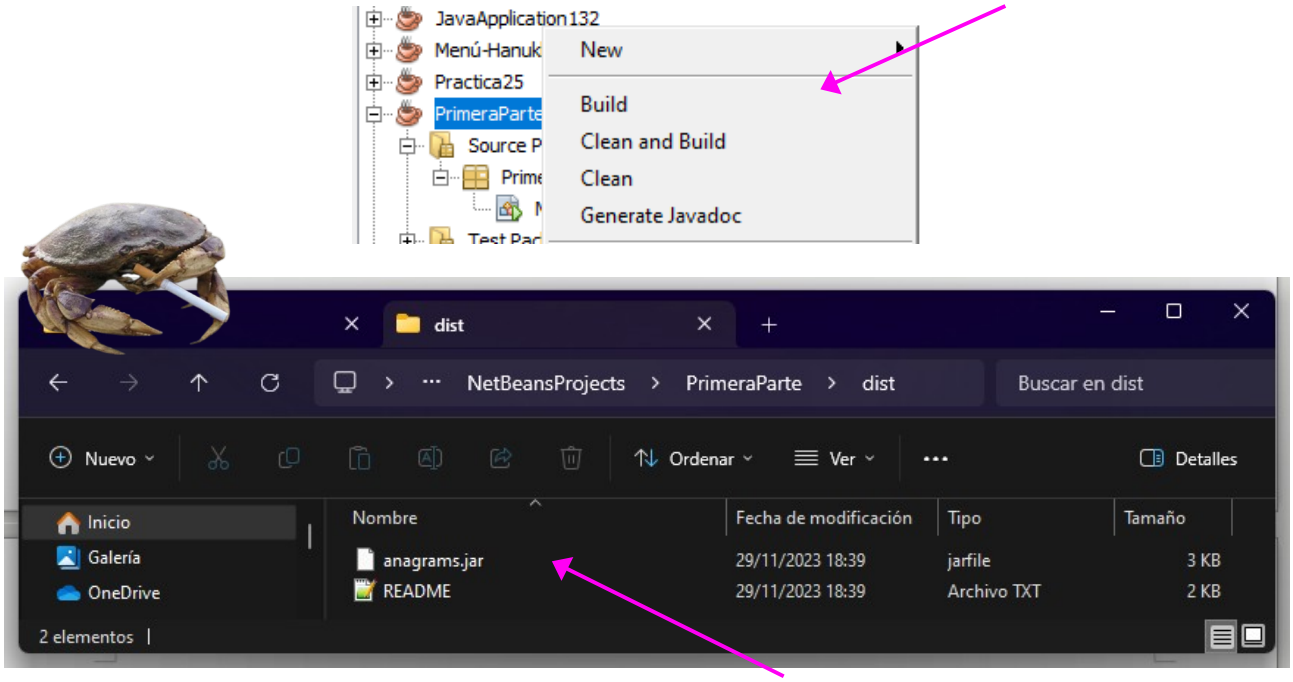
La primera función que haremos será *pedirNumeros*(). En ella pediremos por teclado una lista de números separados por espacios, esto se guardará en un *String* el cual dividiremos usando el espacio como separador y almacenando los distintos *Strings* en un *array* de *Strings*. Luego crearemos un *array* de enteros con el mismo largo que el *array* de *Strings* y haremos un bucle *for* que convertirá los *Strings* en enteros y los irá almacenando en el nuevo *array*. Por último retornaremos el *array*.

```
public static int[] pedirNumeros() {
    //PEDIMOS LOS NÚMEROS Q DESEAMOS ALMACENAR Y LOS GUARDAMOS EN EL STRING numerosString
    System.out.println("Inserte los números que quiera ordenar separados por espacios:");
    Scanner scannerGuai = new Scanner(System.in);
    String numerosString = scannerGuai.nextLine();

    //DIVIDIMOS EL STRING USANDO EL ESPACIO COMO SEPARADOR Y ALMACENANDO LOS STRING
    //INDEPENDIENTES EN UN ARRAY DE STRING
    String[] numeros = numerosString.split("\\s+");

    //CREAMOS UN ARRAY DE ENTEROS CON EL LARGO DEL ARRAY DE STRING
    int[] numerosArray = new int[numeros.length];

    //HACEMOS UN BUCLE FOR QUE CONVIERTE LOS STRING EN ENTEROS Y LOS ALMACENA EN EL NUEVO ARRAY
    for (int i = 0; i < numeros.length; i++) {
        numerosArray[i] = Integer.parseInt(numeros[i]);
    }

    //Y POR ÚLTIMO RETORNAMOS EL ARRAY DE ENTEROS
    return numerosArray;
}
```

La segunda función será *ordenarNumeros*(), esta recogerá a la función *pedirNumeros*() como parámetro ya que retorna el *array* de entero que ordenaremos. En esta función usaremos el método de la *burbuja* para ordenar los enteros de nuestro super *array*, recorriendolo y revisando cada elemento con el siguiente e intercambiándolos si están en el orden equivocado. Pror último imprimiremos por pantalla los números ordenados.

```
public static void ordenarNumeros(int[] array) {
    //ESTE ES EL METODO DE LA BUSBUJA UWU RECORRE EL ARRAY REVISANDO CADA ELEMENTO DE LA LISTA CON EL SIGUENTE
    //INTERCAMBIÁNDOLOS DE POSICIÓN SI ESTÁN EN EL ORDEN EQUIVOCADO
    boolean uwu;
    for (int i = 0; i < array.length - 1; i++) {
        uwu = false;

        for (int j = 0; j < array.length - i - 1; j++) {
            if (array[j] > array[j + 1]) {
                int temp = array[j];
                array[j] = array[j + 1];
                array[j + 1] = temp;
                uwu = true;
            }
        }
        if (!uwu) {
            break;
        }
    }

    //POR ÚLTIMO IMPRIMIMOS POR PANTALLA LOS NÚMEROS ORDENADOS
    System.out.print("\nNúmeros ordenados: ");
    for (int numero : array) {
        System.out.print(numero + " ");
    }
}
```

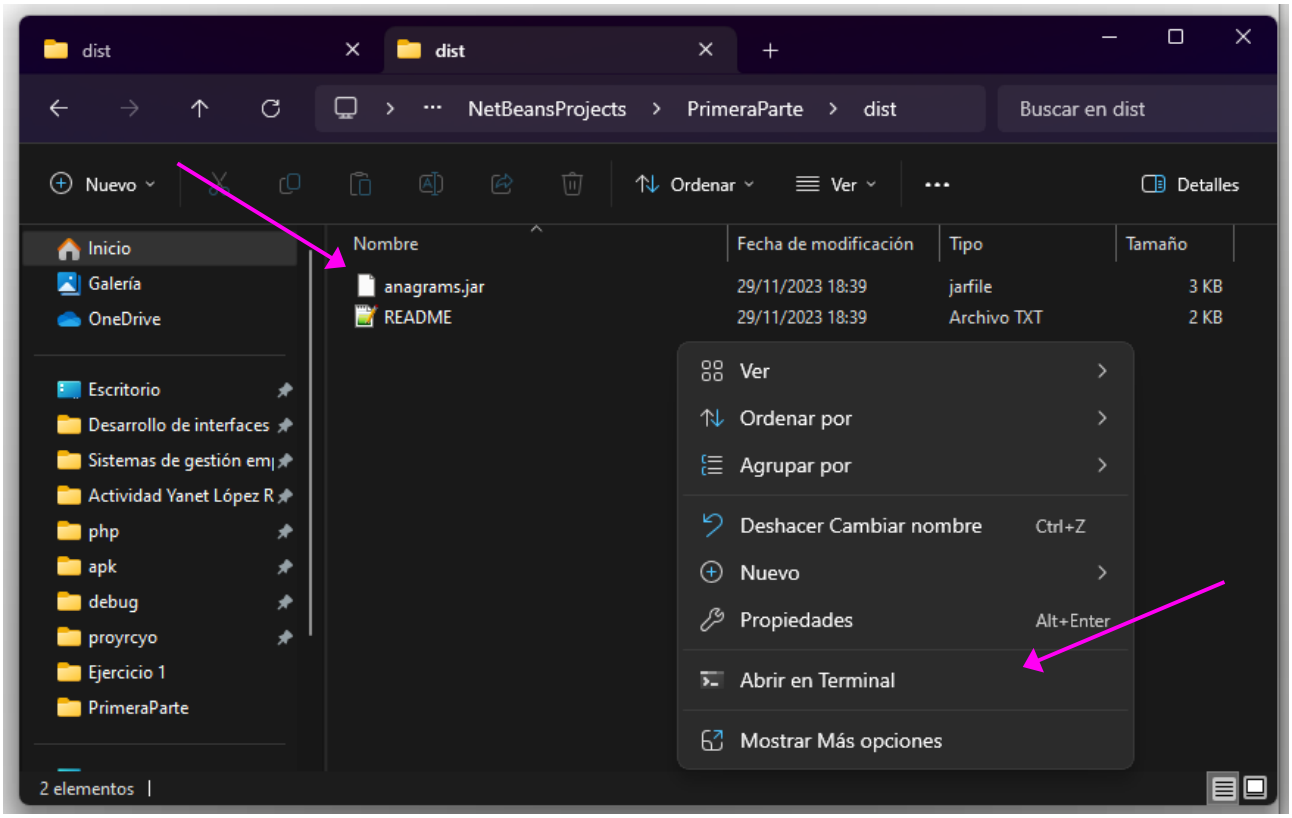
Por último haremos click derecho en nuestro proyecto y pulsaremos en "*Build*", esto generará un archivo .jar en una carpeta llamada dir de nuestro proyecto, en mi caso la ruta de mi jar estará en: *C:\Users\yanet\Documents\NetBeansProjects\PrimeraParte\dist*



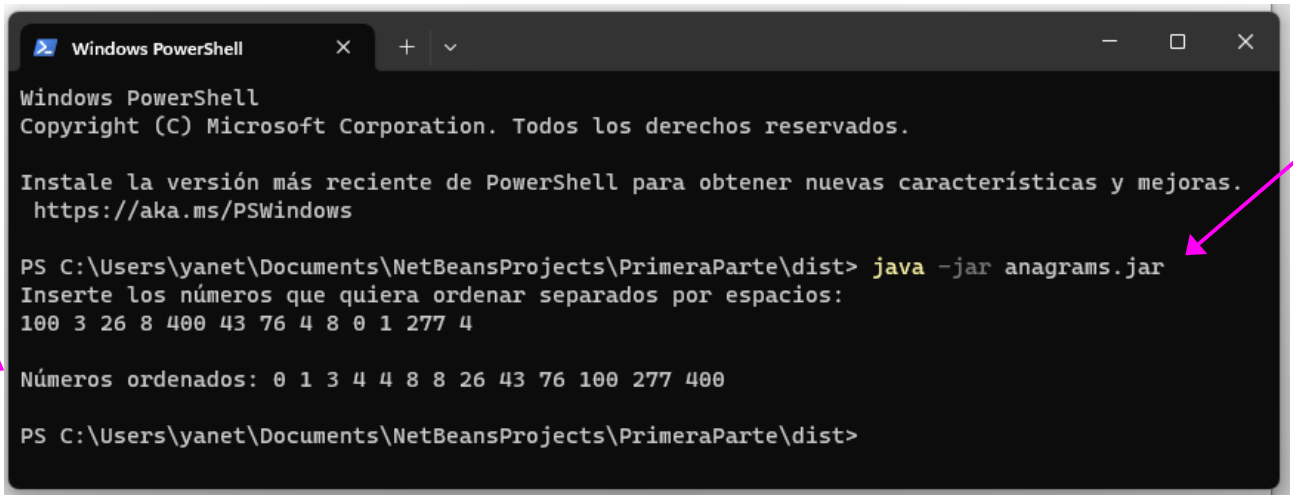
Pruebas

Para probar nuestro programa iremos a la carpeta dir donde encontraremos nuestro hermoso .jar
C:\Users\yanet\Documents\NetBeansProjects\PrimeraParte\dist

Aquí deberemos hacer click derecho y escoger la opción de “Abrir terminal”, esto nos abrirá el cmd ubicándonos en la ruta en la que estamos.



Ejecutamos en el terminal ***java -jar anagrams.jar*** y comprobamos que nuestro código funciona como queremos, haremos varias pruebas con distintos números para probar que no hay fallos en nuestra aplicación.ç

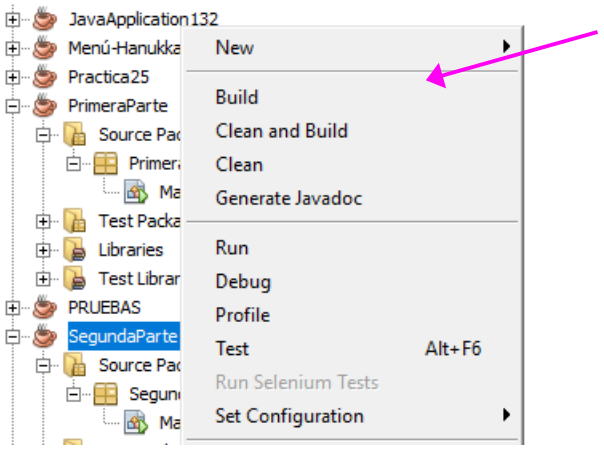


Segunda parte: ***implementa una aplicación, llamada 'aleatorios', que genere al menos 40 números aleatorios (entre 0 y 100), y que los escriba en su salida estándar.***

Para realizar esta actividad lo primero que haremos será crear un nuevo proyecto, en el tendremos nuestra clase *Main* y en ella tendremos *main*() y *aleatorios*(). Desde *main*() lo único que haremos será llamar a la función *aleatorios*() y poner un salto de línea para que al ejecutarlo en la terminal quede más bonito.
En la función *aleatorios*() empezaremos pidiendo por pantalla la cantidad de números que deseamos generar, luego crearemos una instancia en la clase *Random*() y un array de enteros con el mismo tamaño que cantidad de números desea el usuario, ahora hacemos un bucle for que inserte un número aleatorio en cada posición del array y lo imprimimos por pantalla.

```
public class Main {
    public static void main(String[] args) {
        //LLAMAMOS A LA FUNCIÓN aleatorios() PARA QUE NOS GENERE LA LISTA DE NÚMEROS
        aleatorios();

        //PONEMOS UN SALTO DE LÍNEA PARA QUE QUEDE MAS BONICO
        System.out.println("\n");
    }

    public static void aleatorios() {
        //PEDIMOS LA CANTIDAD DE NÚMEROS QUE DESEAMOS GENERAR
        System.out.println("¿Cuántos números aleatórios desea generar?");
        Scanner sc = new Scanner(System.in);
        int cantidad = sc.nextInt();

        // CREAMOS UNA INSTANCIA EN LA CLASE RANDOM
        Random random = new Random();

        //CREAMOS UN ARRAY CON EL TAMAÑO INDICADO POR EL USUARIO
        int[] array = new int[cantidad];

        //CREAMOS UN BUBLE FOR QUE INSERTE NÚMEROS ALEATÓRIOS EN LA ARRAY
        for (int i = 0; i < cantidad; i++) {
        //GENERA UN NÚMERO ALEATÓRIO ENTRE 0 Y 100 Y LO INSERTA EN LA POSICIÓN i DE LA ARRAY
            array[i] = random.nextInt(101);
        }

        //IMPRIMIMOS EL CONTENIDO DE LA ARRAY POR PANTALLA
        System.out.print("\nNúmeros: \n");
        for (int i = 0; i < cantidad; i++) {
            System.out.print (array[i] +  " ");
        }
    }
}
```

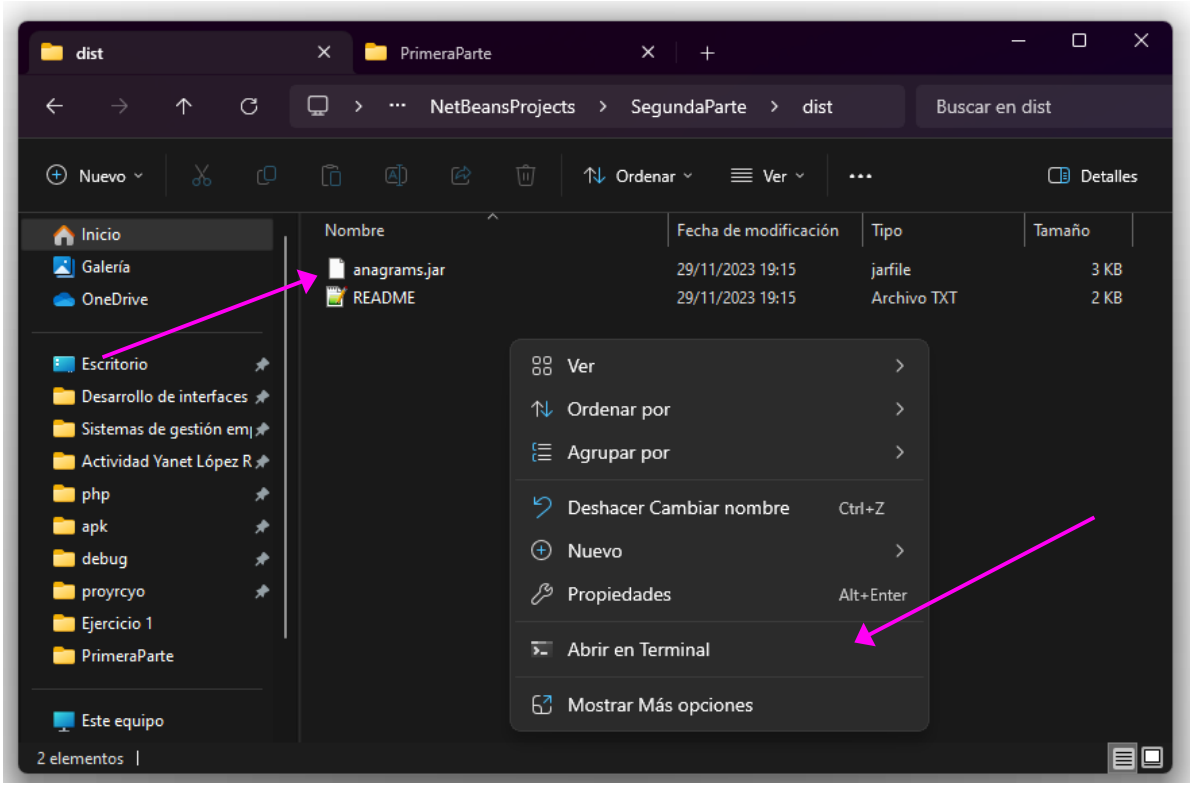
Por último haremos click derecho en nuestro proyecto y pulsaremos en "*Build*", esto generará un archivo .jar en una carpeta llamada dir de nuestro proyecto, en mi caso la ruta de mi jar estará en:
C:\Users\yanet\Documents\NetBeansProjects\SegundaParte\dist



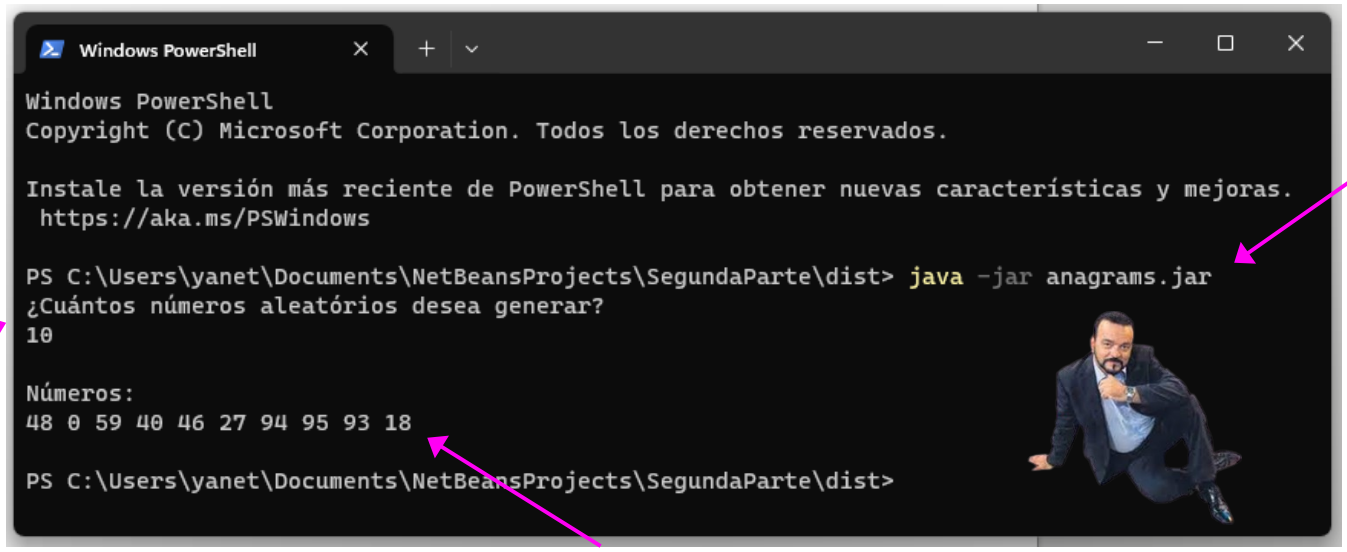
Pruebas

Para probar nuestro programa iremos a la carpeta dir donde encontraremos nuestro hermoso .jar
C:\Users\yanet\Documents\NetBeansProjects\SegundaParte\dist

Aquí deberemos hacer click derecho y escoger la opción de “Abrir terminal”, esto nos abrirá el cmd ubicándonos en la ruta en la que estamos.



Ejecutamos en el terminal ***java -jar anagrams.jar*** y comprobamos que nuestro código funciona como queremos, haremos varias pruebas con distintos números para probar que no hay fallos en nuestra aplicación.



Tercera parte: ***entre las pruebas que hayas realizado, debes incluir una prueba en la que utilizando el operador "|" (tubería) redirijas la salida de la aplicación 'aleatorios' a la entrada de la aplicación 'ordenarNumeros'.***

Para poder redirigir la salida de *aleatorios* a *ordenarNumeros* deberemos hacer algunos cambios en el código para que nos permita llamarlos y que no nos de ningún error.

Usaremos ***java -jar ordenar.jar | java -jar nums.jar***, donde ordenar lo primero que hará será recoger la salida de nums para ejecutar su código.

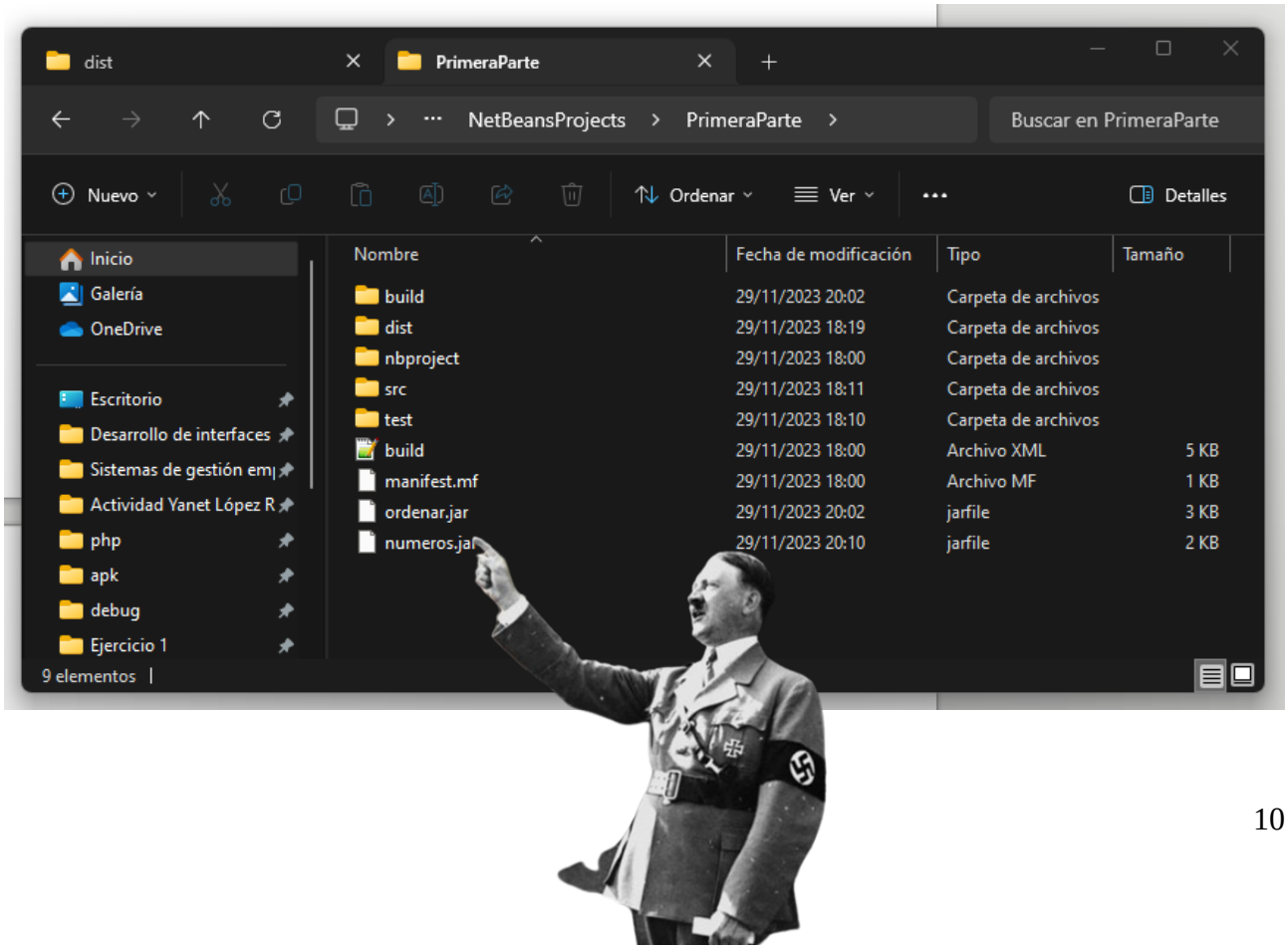
En el código de la primera parte (*ordenarNumeros*) lo que he cambiado ha sido que he puesto todo en el *main* para simplificarlo y le he quitado el texto en el que pedía la lista de números. Y quedaría así:

```
public class Main {
    public static void main(String[] args) {
        //RECOGEMOS LOS NÚMEROS DE aleatorios
        Scanner scannerGuai = new Scanner(System.in);
        String numerosString = scannerGuai.nextLine();

        //DIVIDIMOS EL STRING USANDO EL ESPACIO COMO SEPARADOR Y ALMACENANDO LOS STRING
        //INDEPENDIENTES EN UN ARRAY DE STRING
        String[] numeros = numerosString.split("\\s+");

        //CREAMOS UN ARRAY DE ENTEROS CON EL LARGO DEL ARRAY DE STRING
        int[] array = new int[numeros.length];

        //HACEMOS UN BUCLE FOR QUE CONVIERTE LOS STRING EN ENTEROS Y LOS ALMACENA EN EL NUEVO ARRAY
        for (int i = 0; i < numeros.length; i++) {
            array[i] = Integer.parseInt(numeros[i]);
        }
        //ESTE ES EL METODO DE LA BUSBUJA UWU RECORRE EL ARRAY REVISANDO CADA ELEMENTO DE LA LISTA CON EL SIGUENTE
        //INTERCAMBIÁNDOLOS DE POSICIÓN SI ESTÁN EN EL ORDEN EQUIVOCADO
        boolean uwu;
        for (int i = 0; i < array.length - 1; i++) {
            uwu = false;

            for (int j = 0; j < array.length - i - 1; j++) {
                if (array[j] > array[j + 1]) {
                    int temp = array[j];
                    array[j] = array[j + 1];
                    array[j + 1] = temp;
                    uwu = true;
                }
            }
            if (!uwu) {
                break;
            }
        }

        //POR ÚLTIMO IMPRIMIMOS POR PANTALLA LOS NÚMEROS ORDENADOS
        System.out.print("\nNúmeros ordenados: ");
        for (int numero : array) {
            System.out.print(numero + " ");
        }
    }
}
```

En la segunda parte (*aleatorios*) , he puesto todo en el *main* para simplificarlo también, pero aquí he quitado por completo las líneas donde pedíamos la cantidad de números por teclado, ahora se las especificaremos desde el código, en este caso he puesto 50.

```
public class Main {
    public static void main(String[] args) {
        //LE DAMOS LA CAMTIDAD DE NUMEROS QUE DESEAMOS GENERAR
        int cantidad = 50;

        // CREAMOS UNA INSTANCIA EN LA CLASE RANDOM
        Random random = new Random();

        //CREAMOS UN ARRAY CON EL TAMAÑO INDICADO POR EL USUARIO
        int[] array = new int[cantidad];

        //CREAMOS UN BUBLE FOR QUE INSERTE NÚMEROS ALEATÓRIOS EN LA ARRAY
        for (int i = 0; i < cantidad; i++) {
        //GENERA UN NÚMERO ALEATÓRIO ENTRE 0 Y 100 Y LO INSERTA EN LA POSICIÓN i DE LA ARRAY
            array[i] = random.nextInt(101);
        }

        //IMPRIMIMOS EL CONTENIDO DE LA ARRAY POR PANTALLA
        for (int i = 0; i < cantidad; i++) {
            System.out.print (array[i] +  " ");
        }
    }
}
```

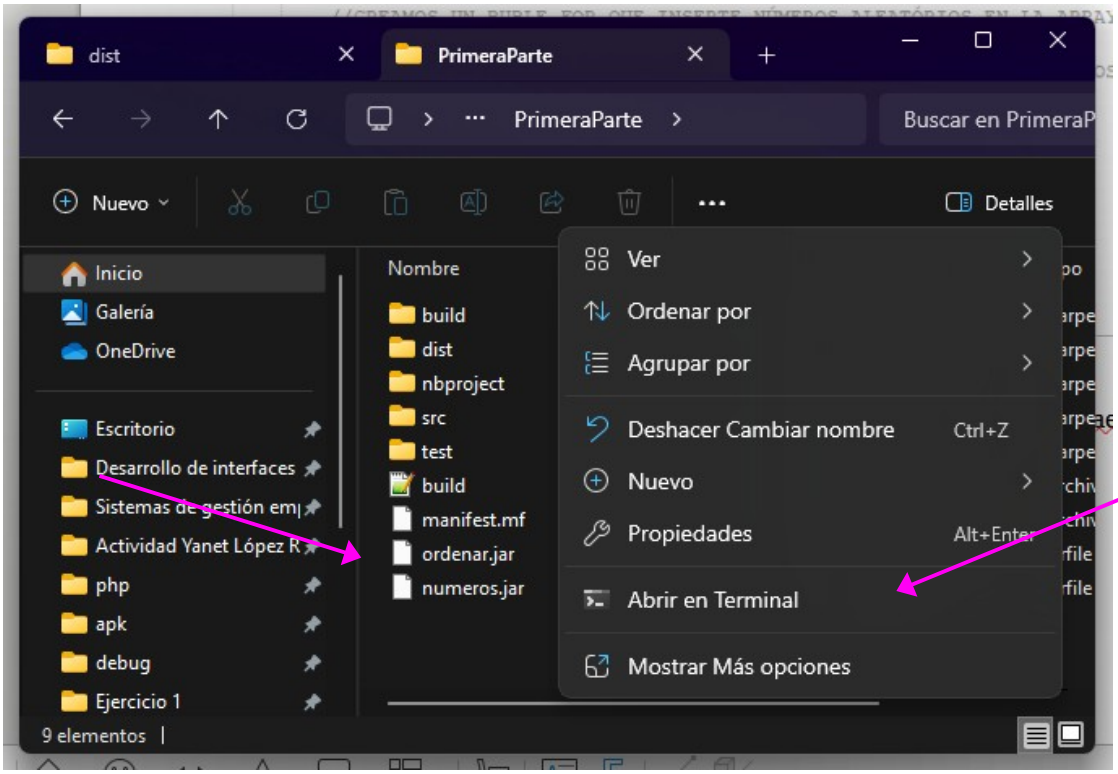
Ahora deberemos copiar los dos .jar en una misma carpetas y renombrarlos para saber cual es cual, yo los he llamado “ordenar” y “numeros” super original, lo sé.



Pruebas

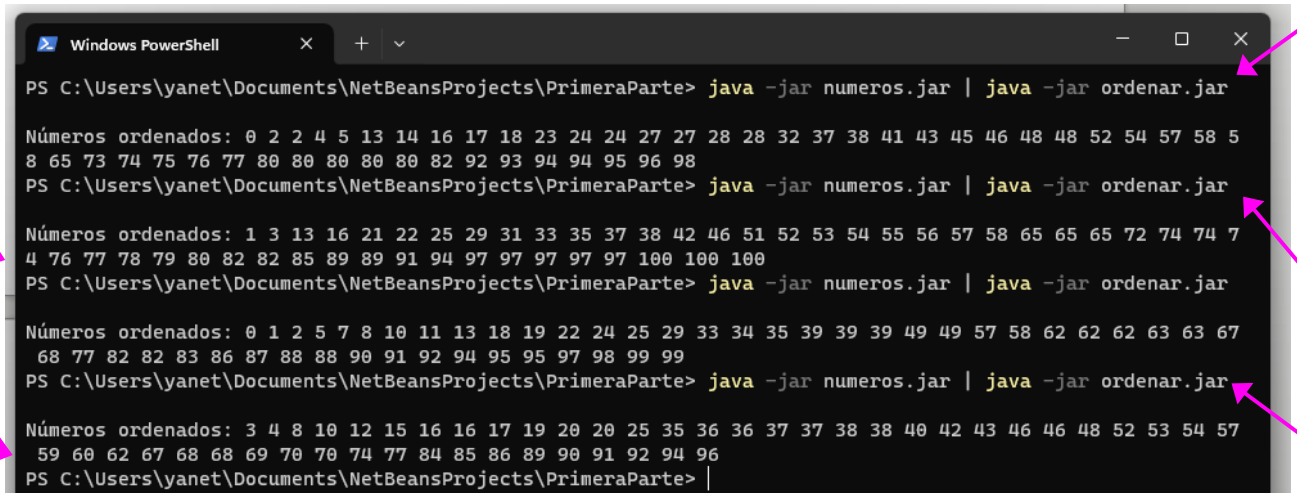
Para probar nuestro programa iremos a la carpeta donde tenemos nuestros dos .jar en mi caso es: *C:\Users\yanet\Documents\NetBeansProjects\PrimeraParte*

Aquí deberemos hacer click derecho y escoger la opción de “Abrir terminal”, esto nos abrirá el *cmd* ubicándonos en la ruta en la que estamos.



Ejecutamos en el terminal ***java -jar ordenar.jar | java -jar numeross.jar*** y comprobamos que nuestro código funciona como queremos, haremos varias pruebas con distintos números para probar que no hay fallos en nuestra aplicación.

Aquí vemos como siempre nos retorna 50 números aleatorios (puedes contarlos si quieres xd) bien ordenaditos.



Conclusiones.

En conclusión, he realizado la implementación de dos aplicaciones: 'ordenarNumeros' y 'aleatorios'.

La primera aplicación, 'ordenarNumeros', se encarga de ordenar un conjunto indeterminado de números que se reciben a través de la entrada estándar y muestra el resultado de la ordenación en la salida estándar.

La segunda aplicación, 'aleatorios', genera al menos 40 números aleatorios en un rango del 0 al 100 y los escribe en la salida estándar.

Para probar la ejecución de estas aplicaciones, he realizado diferentes pruebas. Una de las pruebas fue utilizar el operador "|" (tubería) para redirigir la salida de la aplicación 'aleatorios' a la entrada de la aplicación 'ordenarNumeros'. Esto me permitió generar números aleatorios y ordenarlos automáticamente.

En general, las aplicaciones funcionaron correctamente, ordenando los números de manera ascendente y generando los números aleatorios esperados. Pude comprobar esto mediante la salida en la consola o la redirección de la salida hacia un archivo de texto.

En resumen, la implementación de estas aplicaciones me permitió ordenar números de manera eficiente y generar números aleatorios de forma rápida y sencilla.

Recursos.

Recursos necesarios para realizar la Tarea:

- IDE NetBeans.
- Contenidos de la unidad.
- Ejemplos expuestos en el contenido de la unidad.

Bibliografía.


https://www.biblia.es/biblia-online.php


https://www.churchofjesuschrist.org/study/scriptures?lang=spa


https://pastoralsj.org/biblia


https://www.biblija.net/biblija.cgi?l=es


https://www.bibliatodo.com/la-biblia