

NOMBRE:	Juan Luis Gómez Ruiz
Asignatura	Programación de Servicios y Procesos

## EJERCICIO 1 – MULTIPROCESOS

La actividad 1 se compone en 4 partes:

**Primera Parte:** Implementa una aplicación, llamada 'cadenas', que genere cadenas formadas por caracteres del alfabeto (a-z). La cantidad de cadenas será indicada por el usuario al ejecutar la aplicación y su longitud será aleatoria (máximo 10 caracteres). Las cadenas se deben escribir mediante tubería establecida con la aplicación 'frecuencia'.

En esta primera clase usamos el propio argumento del método main, que mandará como un array de String a la otra clase el número de cadenas que el usuario quiere que genere.

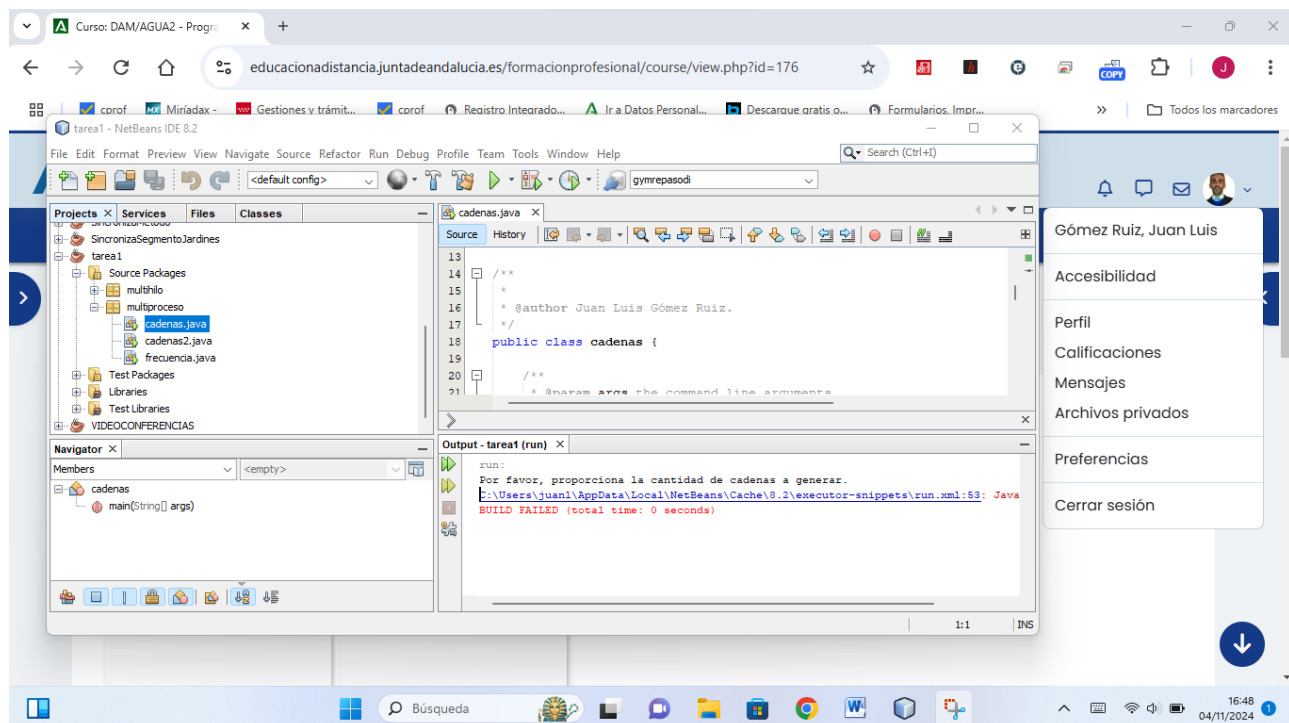
Si el otro programa o clase no recibe el número de cadenas como argumento terminará.

Ahora creamos una variable entera cadenas para guardar el número de argumentos pasado. Este argumento tiene que parsearse a entero para poder operar con él. Este lo metemos en un bloque try-catch para evitar que el argumento no sea un número válido.

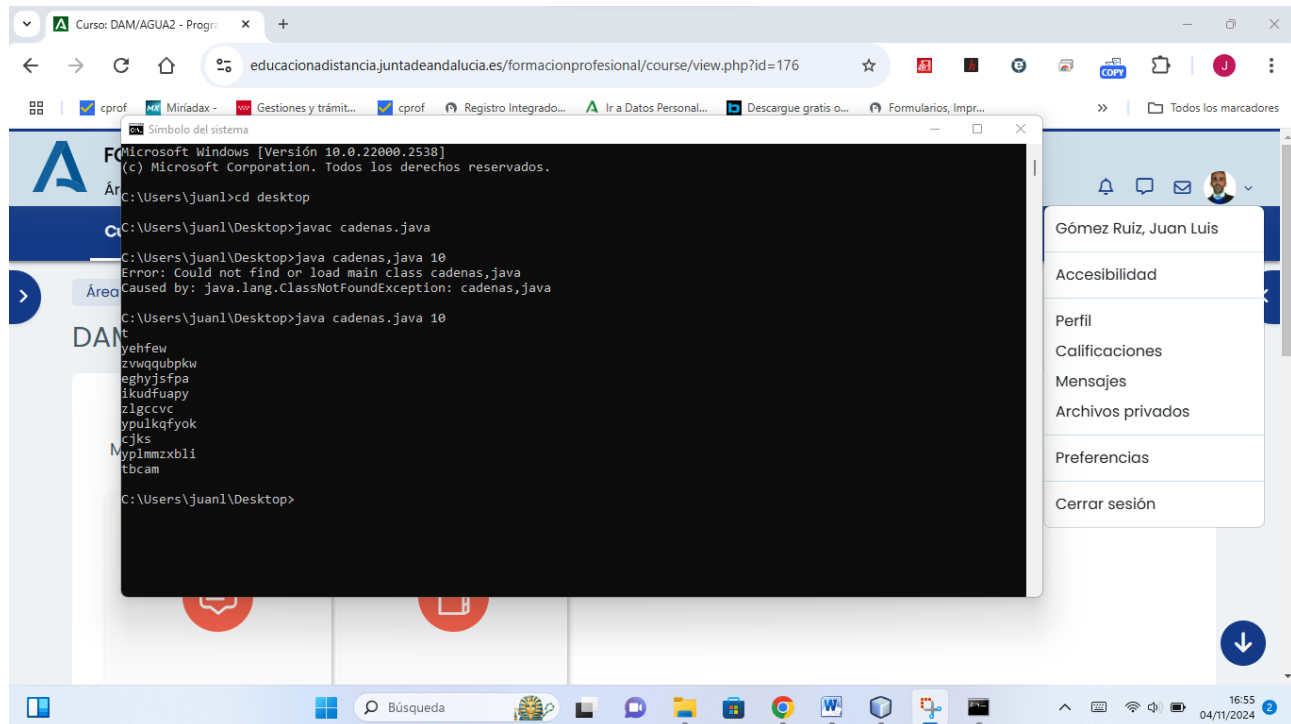
Ahora con un doble bucle for creamos un número aleatorio con un máximo de 10 unidades y lo hacemos cadena de String con el largo que nos haya dado ese número aleatorio.

El otro bucle for lo que nos dará las piezas que serán las 26 letras minúsculas del abecedario para formar las cadenas que vamos a mandar al otro programa. Yo finalmente como control he puesto un sout para poder imprimir las cadenas generadas en consola.

En esta imagen podemos ver la clase cadenas que si no recibe el argumento el programa finaliza abruptamente.



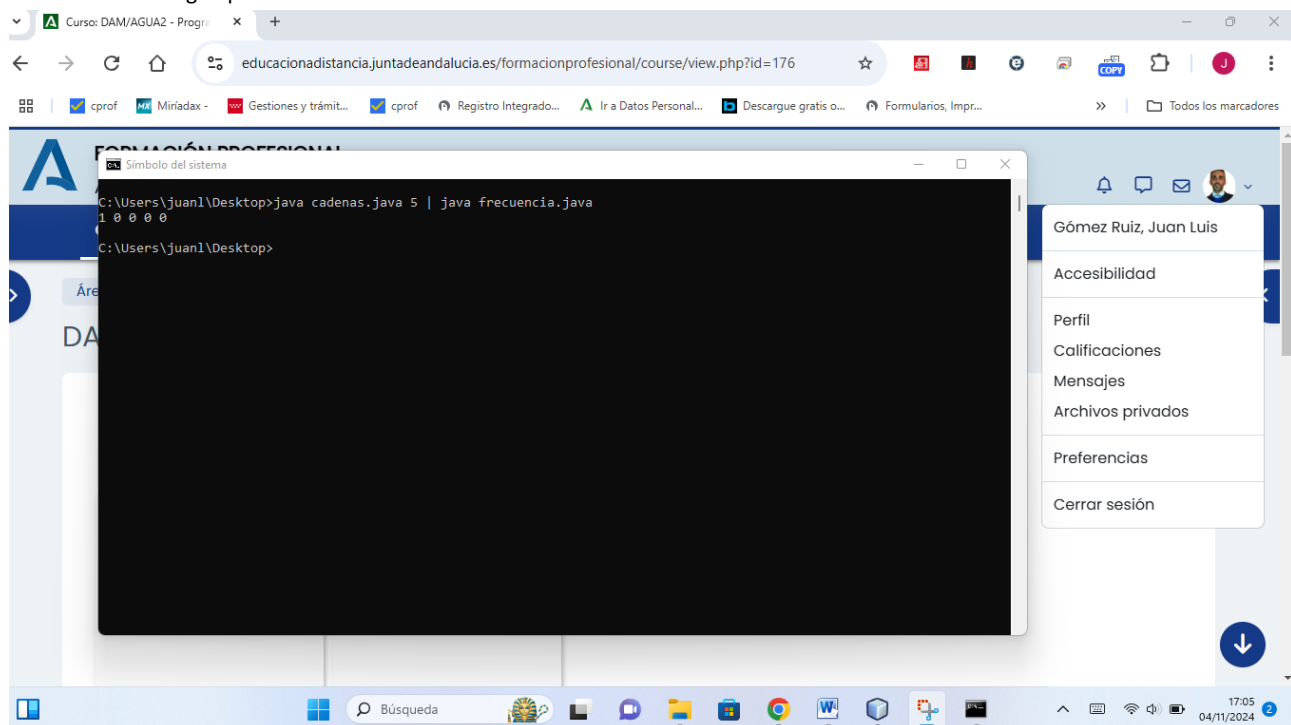
En la siguiente imagen podemos ver como cadenas nos genera las cadenas que le pasamos por parámetro:



Segunda Parte: Implementa una aplicación, llamada 'frecuencia', que dada una cadena de texto recibida a través de la tubería obtenga la frecuencia absoluta de cada una de las vocales, es decir, la cantidad de veces que aparece cada vocal. En el caso de que una de las vocales no aparezca la frecuencia será 0. Por ejemplo: para la palabra "caramelo" el resultado debe ser 2 1 0 1 0.

Creamos un array de enteros donde vamos a meter las posiciones de las vocales de las cadenas, luego con Buffered recibimos la cadena a través de la tubería y la leemos en forma de String, convirtiendo toda la cadena en minúsculas. Finalmente contamos la frecuencia de vocales en las cadenas y las imprimimos por la consola.

Ahora en esta imagen podemos ver el uso de la tubería:



**Tercera Parte:** Para generar palabras parecidas al español y que no salgan combinaciones extrañas de muchas consonantes seguidas, podéis utilizar un array de sílabas de dos o tres letras que siempre contengan una vocal:

```
String[] SILABAS = {"ba", "be", "bi", "bo", "bu", "ca", "ce", "ci", "co", "cu", "da"....., "zo", "zu"}
```

Cuanto más completo creéis el array, más combinaciones posibles. Lo normal es que se generen cadenas que no tengan ningún significado pero que si tengan un parecido al español.

Creamos un array de enteros donde vamos a meter las posiciones de las vocales de las cadenas, luego con Buffered recibimos la cadena a través de la tubería y la leemos en forma de String, convirtiendo toda la cadena en minúsculas. Finalmente contamos la frecuencia de vocales en las cadenas y las imprimimos por la consola.

