

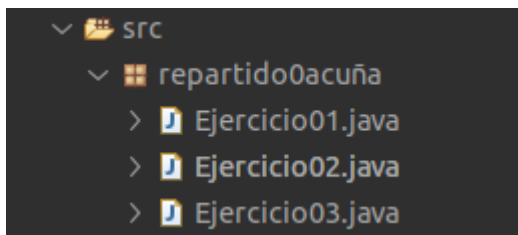
Repartido 0

El objetivo es el repaso de lo trabajado en el curso de Programación 1. Haremos un breve pasaje por

- tipos de datos primitivos
- *Strings*
- colecciones
- lectura de datos desde el teclado
- imprimir en pantalla
- estructuras de control

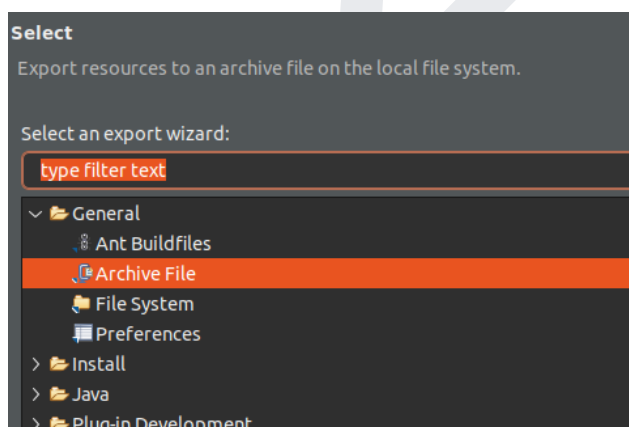
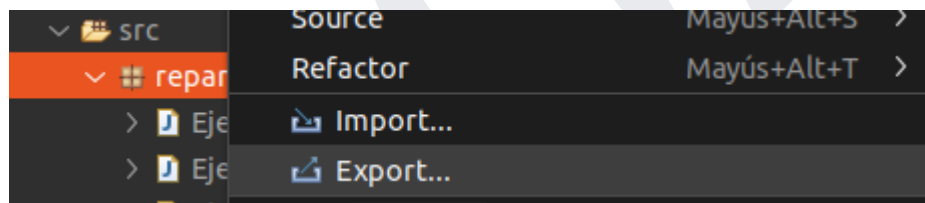
Forma de entrega

Cada ejercicio deberá quedar guardado como un archivo, nombrándolo de la forma “Ejercicio01.java”. Y todos estos archivos deben estar en el paquete “repartido0apellido”.



En el código se pueden hacer las aclaraciones que se desee por medio de comentarios `//` o por medio de *Javadoc* `/**`

Al finalizar, se entregará el paquete comprimido.



Ejercicio 01

Escribir un programa que a partir de los números 30 y 514:

- A. Los sume y muestre el resultado en pantalla.
- B. Los reste y muestre el resultado en pantalla.
- C. Los divida y muestre su resultado en pantalla.
- D. Los multiplique y muestre su resultado en pantalla.

Ejercicio 02

El programa debe realizar las mismas operaciones que en el ejercicio 01. Pero, los números deben ser ingresados por el usuario y almacenados en las variables **a** y **b**.

Es decir, el usuario podrá ingresar cualquier valor que desee y el programa deberá realizar las operaciones.

El programa debe correr solo una vez.

Ejercicio 03

Al programa Ejercicio02 se le debe agregar una funcionalidad: la posibilidad de que el usuario pueda continuar usando el programa para operar con otros números. Es decir, el programa debe correr más de una vez.

Debe finalizar cuando se ingresen los valores 0, 0.

Ejercicio 04

Crear un programa que:

- A. imprima todos los números de 0 a 255.
- B. imprima todos los números de 255 a 0.

Ejercicio 05

- A. En un array, se deberán guardar números. Estos serán los mismos del ejercicio 04 parte A
- B. En otro array, se deberán guardar los mismos números del ejercicio 04 parte B, pero a cada uno se le multiplicará por 7 antes de guardarlo.

Ejercicio 06

Mostrar en pantalla el contenido del *array* del ejercicio 05. Es decir, recorrer el *array* y mostrar sus elementos.

Ejercicio 07

Se da este *String*: "Hola Carlos"

- A. ¿Qué largo tiene? Imprimir la respuesta en pantalla
- B. ¿Es igual a este otro? "Hola Carla". Imprimir la respuesta en pantalla

Ejercicio 08

Se proporciona [este String](#). Realizar un programa que:

- A. Imprima en pantalla el String tal cual está.
- B. Pase todo a minúsculas y lo imprima en pantalla.
- C. Pase todo a mayúsculas y lo imprima en pantalla.
- D. Reemplace las letras 'e' por un '3' y lo imprima en pantalla.

Ejercicio 09

Se da el [String q](#). ¿Qué pasó? ¿Hay un texto ahí dentro? Realizar las operaciones necesarias para limpiar ese texto e imprimirlo en pantalla.

Ejercicio 10

Se cuenta con el [String p](#). ¡Es enorme! Y se pide que: se busque si en él, hay un número de teléfono:

- A. 4332 1234
- B. 2714 2714

El programa debe imprimir en pantalla “Número de teléfono encontrado” o “Número de teléfono no encontrado” según corresponda.

Ejercicio 11

- A. Crear un *array* de *Strings* que almacene las palabras de esta frase “Escuela Técnica Canelones”. Cada palabra será un elemento del array.
- B. Imprimir en pantalla el contenido del array.

Ejercicio 12

Se da [este String](#). Crear un programa que:

- A. lo divida en partes más pequeñas (lo que a veces se define como *tokenizar*), sabiendo que el delimitador entre estas partes es el símbolo “#”.
- B. cuente la cantidad de tokens y lo imprima en pantalla.
- C. imprima en pantalla cada token (o partecita), uno por línea.

Ejercicio 13

Revisa todos los ejercicios anteriores y trata de que cada operación (o ítem del ejercicio) quede contenida en un método.

Ejercicio 14

Revisa todos los ejercicios anteriores y agrega una explicación de su funcionamiento por medio de *Javadoc*.

Ejercicio 15

Revisa todos los ejercicios anteriores y asegúrate de que cumplen con el estándar de codificación que definimos para el curso: [Google Java Style Guide](#).

¿TODO PRONTO? Bien hecho. Ahora, envía tu trabajo.