# Fundamentos de Programación (ST0242)

## Taller número 9

*En el taller 9, usted resolverá varios problemas de Arreglos en Codingbat. Su cuenta de estudiante será* <codigoDeEstudiante>s20182g037t9@hotmail.com (no es necesario que la cuenta exista)*. Debe agregar al profesor para que se pueda calificar el taller. La cuenta del profesor es:* [Edwin.Duque@outlook.com](mailto:Edwin.Duque@outlook.com) *. La entrega se hace por Codingbat y los puntos 3, 4 y 5 por Eafit InterctivaVirtual, en la fecha que diga la aplicación. Para los puntos 3, 4 y 5 se debe enviar un archivo .zip con todo el proyecto de BlueJ.*

1. **Array-1**:

* maxEnd3
* middleWay
* has23
* makeLast
* fix23
* biggerTwo
* front11

1. **Array-2**:

* sum28
* fizzArray
* fizzArray2
* isEverywhere
* matchUp
* has12
* haveThree
* fizzBuzz

1. Desarrolle un generador de nombres de forma aleatoria combinando uno o dos nombres y dos apellidos. Se sugiere tener una lista de nombres y una lista de apellidos. Los apellidos se pueden repetir en un nombre de persona, pero cuando el nombre es compuesto por dos nombres, estos deben ser diferentes, es decir, no es válido María María López López, pero si es válido Luz María López López.

Para hacer pruebas en una aplicación, se requiere un array de objetos de persona, donde cada persona tiene un nombre, un sexo, una edad, una talla y un peso. Genere un programa que solicite la longitud del array que requiere generar y retorne el arrayList con las personas generadas de forma aleatoria y autónoma.

1. La editorial “No lea todo lo que cree”, cuenta con un arreglo en el cual se guardan los apellidos de los autores de los artículos que se han publicado. Escriba un programa que implemente la siguiente funcionalidad:
   1. Llene el arreglo por código.
   2. Un método debe recibir el apellido de un autor y contar cuántos artículos han sido escritos por autores con ese apellido
   3. Un método que cree un arreglo con los apellidos de los autores, pero sin repeticiones. Luego de esto, el método cuenta cuántos apellidos comienzan con una vocal.
2. Desarrolle un programa que permita el manejo de una agenda de teléfonos. Para cada contacto, se almacena el nombre y el teléfono del contacto en un objeto de la clase Contacto. En todo momento, la agenda se mantendrá ordenada por el nombre del contacto. Se debe implementar un método para cada una de las siguientes funciones “CRUD” (Create, Read, Update, Delete):
   1. El método crear() recibe un nombre y un teléfono. Si no existe un contacto con ese nombre, se agrega a la agenda.
   2. El método leer() recibe el nombre de un contacto y lo imprime.
   3. El método actualizar() recibe un nombre y un teléfono. Si el nombre existe en la agenda, se cambia el número del teléfono.
   4. El método borrar() recibe un nombre. Si el nombre existe en la agenda, se retira.

La agenda se almacena en una clase que tiene un arreglo con 100 referencias a contactos.