



Esta guía está basada en la realizada para la semana 7, por lo tanto si el estudiante en una tutoría anterior desarrollo la guía, avancen a las secciones marcadas como **Anexo**.

Sección 1 – Ciclos y objetos dentro de arreglos. (Realizar en hojas blancas).

Etapa 1: Concepto de arreglos

Escenario:

Debido a las nuevas restricciones que se han puesto sobre la música en Internet, ud. y sus compañeros de APO1 han decidido realizar un reproductor de música Web. El reproductor tiene un conjunto de máximo 5 canciones y cada canción tiene múltiples calificaciones. Cada canción tiene su nombre y duración, mientras que la calificación tiene el correo de quién la calificó y la calificación numérica (1 a 5).

1. Elabore el diagrama de clases del problema planteado. Recuerde incluir la constante necesaria para indicar el tamaño del arreglo de tamaño fijo. La duración de la canción puede modelarse como un String.

2. Complete el siguiente cuadro que representa el arreglo de tamaño fijo de canciones, con las siguientes canciones. La **posición** indica el lugar de la canción dentro del arreglo de tamaño fijo:
 - Canción 1
 - Nombre: Rude
 - Duración: 3:45
 - Canción 2
 - Nombre: All About That Bass
 - Duración: 3:10
 - Canción 3
 - Nombre: Stay With Me

- Duración: 3:29
- Canción 4
 - Nombre: Break Free
 - Duración: 3:34
- Canción 5
 - Nombre: Fancy
 - Duración: 3:24

posición:	posición:	posición:	posición:	posición:
nombre:	nombre:	nombre: "Stay With me"	nombre:	nombre:
duracion:	duracion:	duracion: "3:29"	duracion:	duracion:

3. Diga cuál es la cantidad de calificaciones que tiene una canción, dado que el último elemento en el arreglo de calificaciones es el siguiente:

posicion: 11
correo: "fake123@hola.com"
valorCalificacion: 4

Etapa 2: Creación e inicialización de arreglos en Java

Escenario:

Debido a las nuevas restricciones que se han puesto sobre la música en Internet, ud. y sus compañeros de APO1 han decidido realizar un reproductor de música Web. El reproductor tiene un conjunto de máximo 5 canciones y cada canción tiene múltiples calificaciones. Cada canción tiene su nombre y duración, mientras que la calificación tiene el correo de quién la calificó y la calificación numérica (1 a 5).

1. Cree el atributo arreglo de tamaño fijo en la clase **ReproductorWeb** e inicialicelo. Recuerde también crear la constante que modela el tamaño del arreglo.

```
/**
 * Clase que representa la clase principal del Reproductor web
 *
 */
public class ReproductorWeb
{
    //-----
    // Constantes
    //-----

    //-----
    // Atributos
    //-----
}
```

```

//-----
// Constructores
//-----

/**
 * Método constructor de la clase ReproductorWeb
 */
private ReproductorWeb()
{

}

}

```

2. Cree el atributo de tamaño variable en la clase **Cancion** para almacenar las calificaciones.

```

/**
 * Clase que representa una canción dentro del Reproductor web
 */
public class Cancion
{
//-----
// Constantes
//-----

//-----
// Atributos
//-----

//-----
// Constructores
//-----

/**
 * Método constructor de la clase Cancion
 */
private Cancion()
{

}

}
}

```

Etapas 3: Concepto de ciclos en arreglos

Escenario:

Suponga que tiene varias cajas con tapa, todas en fila y enumeradas del 0 al 5. Dentro de cada caja hay una pelota de un color.

1. Describa los pasos que usaría para saber dónde se encuentra la pelota azul.

2. Ahora suponga que las pelotas pueden estar repetidas. Describa los pasos necesarios para contar el número de pelotas azules.

Anexo: Recorridos sobre asociaciones 0...1

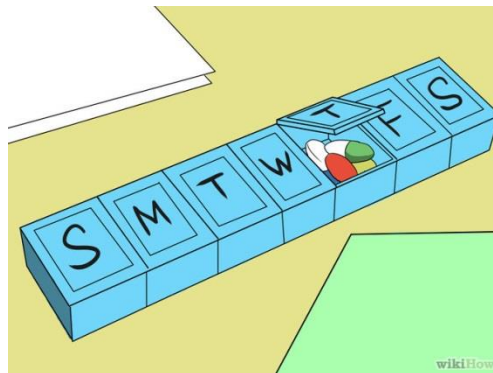
3. Suponiendo que puede que las cajas pueden o no tener pelota adentro. Describa los pasos necesarios para saber dónde se encuentra la pelota azul.

4. Suponiendo que puede que las cajas tengan o no pelotas adentro. Describa los pasos necesarios para contar el número de pelotas azules.

Anexo: Etapa 4. Dobles recorridos

Escenario:

Suponga que tiene varias cajas con tapa, todas en fila y enumeradas del 0 al 5. Dentro de cada caja hay un recipiente que guarda pastillas con el día de la semana que debe tomarse, como la mostrada en la figura. Dentro de las cajas puede haber o no pastillas y sólo puede haber una pastilla.



Fuente: <http://es.wikihow.com/mantener-un-registro-de-tus-medicamentos>

1. Describa los pasos que usaría para encontrar la pastilla de aspirina. Suponga que sólo hay una pastilla de aspirina dentro de todos los contenedores.

2. Ahora suponga que puede haber pastillas repetidas dentro de todos los contenedores, es decir que puede haber 2 o más pastillas de aspirina dentro del sistema. Describa los pasos que usaría para conocer la cantidad de pastillas de aspirina que hay dentro de todos los contenedores.

Sección 2 – Ciclos y objetos dentro de arreglos. (Realizar en el ejemplo Avión).

Para la parte práctica, vamos a implementar una funcionalidad que nos permita rifar premios entre los pasajeros registrados en clase económica. Para ello, cada pasajero debe sacar un número aleatorio entre 1 y 3. Aquellos pasajeros que saquen el número 1, serán ganadores de fabulosos premios que ofrece la Aerolínea.

Para implementar lo anterior, es necesario:

En la clase Pasajero:

1. Crear un nuevo atributo para el pasajero, en el cual podamos guardar el número aleatorio que sacó dicho pasajero.
2. Crear un método **public void** asignarNumeroAleatorio(**int** numeroAleatorio) que permita asignar el valor del parámetro al atributo definido en el punto 1.
3. Crear un método **public int** darNumeroAleatorio() que retorne el valor del atributo definido en el punto 1.

En la clase Avión:

1. Crear un método **public void** realizarRifaClaseEconomica() que recorra las sillas económicas. Por cada silla debe mirar si la silla tiene un pasajero asignado. Si la silla tiene un pasajero asignado, se debe asignar a dicho pasajero un número aleatorio entre 1 y 3. Para generar el número aleatorio, utilice la siguiente instrucción:

```
numeroAleatorio = (int) ((Math.random() * 3) + 1);
```

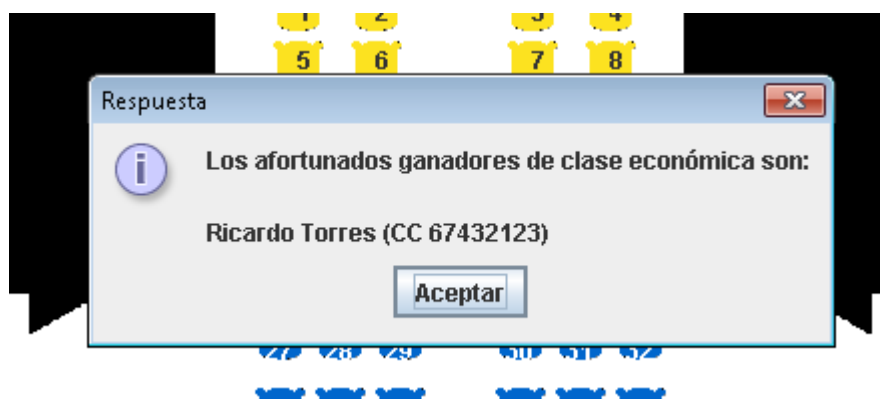
Recuerde utilizar la instrucción for-each para realizar el recorrido.

2. Cree el método **public boolean** hayDosGanadoresMismaUbicacionClaseEconomica() que informa si hubo al menos dos pasajeros ganadores en clase económica y que estén en la misma ubicación (entiéndase por ubicación si la silla está en VENTANA, CENTRAL o PASILLO). Utilice un doble recorrido para cumplir con este requerimiento.
3. Modificar el método **public** String metodo2() para que invoque al método realizado en el punto anterior. En caso que retorne **true**, muestre el mensaje "Hay al menos dos

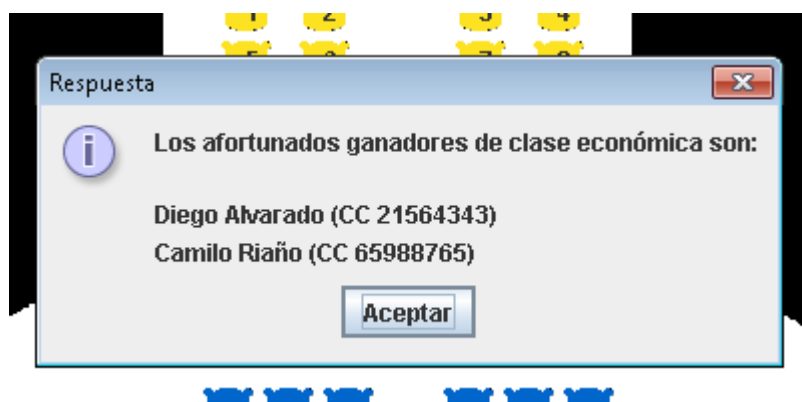
ganadores en clase económica que tienen la misma ubicación”; en caso contrario debe mostrar el mensaje “No hay dos ganadores en clase económica con la misma ubicación”.

4. Crear un método **public** `ArrayList<Pasajero> darGanadoresClaseEconomica()` que recorra las sillas económicas. Por cada silla debe mirar si la silla tiene un pasajero asignado. Si la silla tiene un pasajero asignado, se debe mirar si el número aleatorio de dicho pasajero es 1. Si el número del pasajero es 1, el pasajero debe ser agregado a un `ArrayList` donde se guarden los pasajeros ganadores. El método retorna al final el `ArrayList`. Recuerde utilizar la instrucción `for-each` para realizar el recorrido.
5. Modificar el método **public** `String metodo1()` para que invoque a los métodos:
`realizarRifaClaseEconomica()`
`darGanadoresClaseEconomica()`

y si hay ganadores en clase económica, recorra el `ArrayList` reuniendo sus nombres y cédulas en una variable tipo `String` para mostrar un resultado como el siguiente:



Note que al hacer clic de nuevo sobre el botón Opción 1, cambia el listado de ganadores



Si no hubo ganadores, debe mostrar el siguiente mensaje:

