

## UD4 – Actividad 02 – Colecciones y Estructuras de Datos

### Proyecto Taller

Se ha de diseñar una aplicación que permita crear un taller y a través de él gestionar los coches que en él se irán almacenando. Se distribuirá el código en tres clases:

#### Clase Coche:

Atributos: color y marca

#### Métodos:

Se espera encontrar en la clase los métodos Constructores necesarios, los métodos set y get, y aquellos que se consideren oportunos para el trabajo con objetos de tipo Coche. No hace falta validar los atributos.

Se sugiere sobreescibir el método `toString`, para que muestre los atributos del objeto.

#### Clase Taller:

##### Atributo privado:

HashMap que permita almacenar dinámicamente parejas formadas por una matrícula junto a un objeto de tipo Coche.

##### Métodos:

- `anadeElemento`: Incluye la petición de datos (color, marca y matrícula), la creación del objeto de tipo Coche, valida la matrícula, comprueba que no hay ningún contacto ya en la colección con la misma matrícula y añade la pareja matricula-Coche al mapa.
- `eliminaElemento`: pide matrícula del coche a borrar y lo elimina, informando de la no existencia del mismo dado el caso.
- `visualizaMatriculas`: Recorre el mapa mostrando el conjunto de matrículas almacenadas. (Basado en `keySet()`)
- `visualizaCoches`: Recorre el mapa mostrando color, marca de cada coche almacenado. (Basado en `values()`)
- `visualizaTaller`: Recorre el mapa mostrando color, marca y matricula de cada coche almacenado. (Basado en `entrySet().toString()`)

En esta clase se podrán incluir aquellos métodos que se consideren oportunos y estén relacionados con la gestión de elementos del mapa y del mapa en sí.

#### Clase Principal

En la clase Principal se creará un objeto Taller y se interactuará con un usuario al que se le presentará el siguiente menú:

1. Añadir coche
2. Eliminar coche
3. Salir

Al salir se llamará a los métodos visualizaMatriculas, visualizaCoches, visualizaTaller.

El menú se resolverá utilizando los métodos creados. El menú será repetitivo en todos los casos incluidos los casos de opción no válida o dato de formato incorrecto, hasta que el usuario seleccione la opción 3.

## Proyecto Agenda

Se ha de diseñar una aplicación que permita crear una agenda y a través de ella gestionar los contactos que se irán creando. Se distribuirá el código en tres clases:

### Clase Contacto:

#### Atributos privados:

- nombre (válido siempre que comience por mayúsculas y contenga letras)
- teléfono (válido siempre que empiece por 6,7 o 9 y tenga en total 9 dígitos)
- correo (válido siempre que aparezcan en la cadena: (carácter a-z ó dígito ó \_ ó . ó -), (el símbolo @), (caracteres a-z),( punto),( entre dos y cuatro caracteres a-z)).

#### Métodos:

Se espera encontrar en la clase los métodos Constructores necesarios, los métodos set y get, los métodos que permitan validar los diferentes atributos mediante Expresiones regulares, y aquellos que se consideren oportunos para el trabajo con objetos de tipo Contacto.

### Clase Agenda:

#### Atributo privado:

Colección que permita almacenar dinámicamente objetos de tipo Contacto.

#### Métodos:

- anadeContacto: Incluye la petición de datos, la creación del objeto de tipo Contacto, valida los datos recibidos, comprueba que no hay ningún contacto ya en la colección con el mismo nombre y añade el objeto creado a la colección
- buscaContacto: Pide nombre del contacto a buscar. Recorre la colección en busca del contacto y muestra sus datos, informando de la no existencia del mismo dado el caso.
- eliminaContacto: pide nombre del contacto a borrar. Recorre colección en busca del contacto y lo elimina, informando de la no existencia del mismo dado el caso.

- `visualizaAgenda`: Recorre la colección mostrando nombre, teléfono y correo de cada contacto.
- En esta clase se podrán incluir aquellos métodos que se consideren oportunos y estén relacionados con la gestión de objetos de la colección o de la colección en sí.

### Clase Principal

En la clase Principal se creará un objeto Agenda y se interactuará con un usuario al que se le presentará el siguiente menú:

1. Añadir contacto
2. Buscar contacto
3. Eliminar contacto
4. Visualizar agenda
5. Número de contactos de mi agenda
6. Salir

Que se resolverá utilizando la estructura adecuada para dar respuesta a cada opción, utilizando los métodos creados.

El menú será repetitivo en todos los casos incluidos los casos de opción no válida o dato de formato incorrecto, hasta que el usuario seleccione la opción 6.

### Licencia



[CC BY-NC-SA 3.0 ES Reconocimiento - No Comercial - Compartir Igual \(by-nc-sa\)](#)

No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

[NOTA: Esta es una obra derivada de la original realizada por Carlos Cacho y Raquel Torres.](#)