

Detalles de la tarea de esta unidad.

### Enunciado.

Si hasta el momento hemos utilizado en todos los casos estructuras estáticas (arrays) para almacenar datos, llega el momento de aprovechar las bondades de las estructuras dinámicas y la funcionalidad del API **Collections**.

---

### Ejercicio 1

En la tarea de la Unidad de Trabajo 7 utilizamos un array para almacenar las cuentas bancarias. Esta estructura provoca que nuestra aplicación esté limitada a utilizar 100 cuentas. Modifica dicho proyecto para:

1. Utilizar una estructura de datos dinámica. Determina, de las trabajadas en los contenidos, cuál sería la más idónea, justificando tu respuesta.
2. Añade la opción de menú "**Eliminar Cuenta Bancaria**". A través de esta opción se pedirá el CCC de una cuenta bancaria y se eliminará de la estructura siempre que existe y su saldo sea 0. No se podrán eliminar cuentas con saldo superior a 0.

### Ejercicio 2

Nos han sugerido una mejora en la aplicación desarrollada en la unidad de trabajo 6, en la que gestionamos un concesionario cuyos vehículos eran insertados en un array. El objetivo es mantener los **vehículos ordenados por matrícula en la estructura de datos**. El objetivo de este ejercicio es:

- Hacer las modificaciones a la clase **Vehículo** para que sean objetos comparables por matrícula.
- Modificar la clase **Concesionario** para que utilice una estructura de datos dinámica que mantenga los vehículos ordenados. Determina qué estructura es la más apropiada, **justificando tu respuesta** (Puedes hacerlo en la misma declaración de la propiedad).
- Añadir la opción **Eliminar Vehículo**: Dada una matrícula, eliminar el vehículo cuya matrícula coincide si existe.

Se deben entregar los proyectos de Netbeans completos con los nombres PROG08\_EjerX, donde X es el número de ejercicio. Para empaquetar un proyecto en Netbeans, utiliza la opción **File - Export Project** de Netbeans: generarás un fichero .zip con el contenido completo del proyecto.

### Criterios de puntuación. Total 10 puntos.

La tarea tiene una puntuación total de 10 puntos repartidos de la siguiente forma:

- Ejercicio 1: 5 puntos.
  - La aplicación debe ser funcional con la nueva estructura de datos: 3 puntos.
  - Se implementa la eliminación de una Cuenta Bancaria: 2 puntos.
- Ejercicio 2: 5 puntos.
  - Se comparan objetos vehículos: 1 punto.
  - Se modifica el concesionario con una estructura de datos ordenada y sigue siendo

funcional: 3 puntos.

### **Recursos necesarios para realizar la Tarea.**

- Ordenador personal.
- JDK y JRE de Java SE 11 o posterior.
- Entorno de desarrollo NetBeans 11 o posterior.
- Conexión a Internet.

### **Consejos y recomendaciones.**

Te recomendamos, que para llegar a realizar con éxito la tarea, sigas las siguientes indicaciones:

1. Reflexiona sobre la estructura de datos a utilizar en cada caso antes de escribir código.
2. Las nuevas estructuras de datos son más fáciles de utilizar que los arrays, implicando un menor número de líneas de código.
3. Las clases **Concesionario** y la que contiene las cuentas bancarias quizás te interese volver a implementarlas: modificar el código directamente puede ser una mala opción.
4. Para el ejercicio 2, redefine el método **compareTo** de la clase **Vehículo**, eso te permitirá comparar dos objetos instancias de esta clase.

### **Indicaciones de entrega.**

Una vez realizada la tarea elaborarás un único fichero comprimido que contenga los dos proyectos. El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas:

**apellido1\_apellido2\_nombre\_SIGxx\_Tarea**

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna **Begoña Sánchez Mañas para la octava unidad del MP de PROG**, debería nombrar esta tarea como...

**sanchez\_manas\_begona\_PROG08\_Tarea**