



# Estructuras de Datos

Grado en Informática  
Segundo Curso, segundo cuatrimestre  
Escuela Politécnica Superior de Córdoba  
Universidad de Córdoba  
Curso académico 2015-2016



---

## Práctica 2. Listas doblemente enlazadas

- **Objetivo**
  - Implementación de una lista ordenada de donantes de sangre.
- **Primera parte: implementación de la clase Donante**
  - Codifica una clase abstracta, denominada **DonanteInterfaz**, que especifique los métodos “virtuales puros” para acceder y modificar los atributos de un donante de sangre: nombre, apellidos, grupo sanguíneo, factor RH
  - Codifica una clase, denominada **Donante**,
    - que herede de forma pública de la clase **DonanteInterfaz**
    - que defina los atributos de la clase **Donante**:
      - nombre de tipo cadena
      - apellidos de tipo cadena
      - grupo sanguíneo: 0, A, B o AB
      - factor RH: positivo o negativo
    - y que codifique los siguientes métodos de la clase Donante
      - Constructor parametrizado con valores por defecto.
      - Constructor de copia.
      - Funciones de acceso y modificación de los atributos de la clase Donante.
      - Función para leer desde el teclado un Donante.
      - Función para escribir por pantalla un Donante.
      - Sobrecarga del operador “=” para la asignación de Donantes.
      - Sobrecarga del operador “==” para la comparación de Donantes por apellidos y nombre.
      - Sobrecarga del operador “<=” para la comparación de Donantes por apellidos y nombre.
    - Además, también se deben codificar las funciones **amigas** para sobrecargar los operadores “>>” y “<<”
      - **friend** istream &operator>>(istream &stream, Donante &d);
      - **friend** ostream &operator<<(ostream &stream, Donante const &d);
- **Segunda parte: programa de prueba de la clase Donante**
  - Se debe codificar un programa para comprobar el funcionamiento de la clase **Donante**.
  - El programa debe
    - declarar tres donantes
      - Se debe mostrar el uso del constructor parametrizado con valores por defecto y del constructor de copia.

- y permitir el uso de un menú con, al menos, las siguientes operaciones:
    1. Leer un donante desde el teclado
    2. Escribir un donante por pantalla
    3. Modificar los datos de un donante.
    4. Comparar lexicográficamente dos donantes.
    5. Fin del programa
- **Tercera parte: implementación de la clase Donantes**
  - Codifica una clase abstracta, denominada **DonantesInterfaz**, que especifique los métodos “virtuales puros” de acceso y modificación:
    - Observadores:
      - Función para comprobar si hay donantes
      - Función que busque un donante por nombre y apellidos y devuelva el donante.
    - Modificadores:
      - Insertar un donante
      - Borrar un donante
  - Codifica una clase, denominada **Donantes**,
    - que herede de forma pública de la clase **DonantesInterfaz**
    - que tenga un atributo que sea una “Lista de donantes”:
      - La lista deberá estar ordenada por apellidos y nombre.
      - La implementación de dicha lista deberá ser elegida por cada estudiante entre las siguientes opciones: lista simplemente enlazada, lista doblemente enlazada o lista circular enlazada.
      - Cada estudiante deberá decidir si utiliza o no cursor de la lista.
    - y que codifique los siguientes métodos de la clase
      - Constructor parametrizado.
      - Funciones de acceso y modificación
      - Función para leer desde el teclado los donantes.
      - Función para escribir por pantalla los donantes.
- **Cuarta parte: programa de prueba de la clase Donantes**
  - Se debe codificar un programa para comprobar el funcionamiento de la clase **Donantes**
  - El programa debe permitir el uso de un menú que con, al menos, las siguientes operaciones:
    1. Comprobar si hay donantes
    2. Cargar una lista de donantes desde un fichero.
    3. Grabar una lista de donantes en un fichero.
    4. Insertar un nuevo donante en la lista.
    5. Modificar la información de un donante.
    6. Borrar un donante de la lista.
    7. Mostrar por pantalla los donantes de sangre
    8. Fin del programa
- **Observaciones**
  - Se debe usar el espacio de nombres de la asignatura: **ed**
  - Se debe documentar el código con **doxygen**
  - Se debe crear un fichero **makefile**

- **Tiempo de realización:**
  - Comienzo de la práctica: semana del 7 de marzo
  - Entrega de la práctica: semana del 4 de abril.