



## Programación Declarativa

Ingeniería Informática  
Cuarto curso. Primer cuatrimestre.



Escuela Politécnica Superior de Córdoba  
Universidad de Córdoba

Curso académico: 2017 - 2018

---

### Práctica número 5: tipos abstractos de datos y lectura y escritura

1. Utiliza las “listas de asociación” para codificar las funciones de **creación, acceso, consulta y modificación** del tipo abstracto de datos **Estudiante**, que posee los campos *nombre*, *apellidos* y *tipo* (*propio o Erasmus*).
2. Tipo abstracto de datos “**Asignatura**”
  - **Utiliza** las funciones del tipo abstracto **Estudiante** para codificar las siguientes funciones del tipo abstracto “**Asignatura**”, compuesto por estudiantes ordenados alfabéticamente.
    - **crear-asignatura -vacía**: crea una asignatura sin estudiantes.
    - **asignatura-vacía?**: comprueba si una asignatura está vacía.
    - **buscar-estudiante**: busca un estudiante en una asignatura usando su nombre y apellidos.
    - **insertar-estudiante!**: inserta un estudiante en la asignatura de forma ordenada según sus apellidos y nombre.
    - **borrar-estudiante!**: borra un estudiante de la asignatura.
    - **modificar-estudiante!**: modifica un estudiante de asignatura.
    - **consultar-estudiante**: muestra por pantalla los datos de un estudiante.
    - **consultar-estudiantes**: muestra por pantalla los datos de todos los estudiantes.
    - **consultar-estudiantes-tipo**: recibe una asignatura y un tipo y muestra por pantalla los estudiantes de dicho tipo.
    - **ordenar-estudiantes!**: ordenar los estudiantes por apellidos y nombre usando uno de los siguiente métodos de ordenación interna
      - QuickSort
      - MergeSort
      - HeapSort
      - Etc.
    - **grabar-estudiantes**: graba en un fichero los datos de los estudiantes de una asignatura
    - **cargar-fichero-lista-estudiantes**: lee los datos de los estudiantes desde un fichero y los almacena en una **Asignatura**.

- Etc.

### 3. Codifica un programa que permita la gestión de una “Asignatura”

- Crear una asignatura.
- Comprobar si una asignatura está vacía
- Insertar un estudiante.
- Buscar un estudiante.
- Modificar un estudiante.
- Borrar un estudiante.
- Consultar un estudiante.
- Consultar los estudiantes.
- Ordenar los estudiantes por apellidos y nombre
- Consultar los estudiantes de un tipo.
- Grabar los estudiantes en un fichero.
- Cargar los estudiantes de un fichero.
- Etc.

### 4. Tipo abstracto de datos “Polinomio”

$$P(X) = a_0 + a_1 X + a_2 X^2 + \dots + a_n X^n$$

- Se deben codificar las siguientes funciones del tipo abstracto
  - **crear\_polinomio**: crea un polinomio a partir de un número variable de coeficientes y grados.
  - **grado**: indica el grado del polinomio.
  - **coeficiente**: indica el valor del coeficiente del monomio con grado “i” del polinomio.
  - **modificar\_coeficiente!**: modificar el coeficiente del monomio con grado “i” del polinomio.
    - Puede provocar la inclusión o eliminación del monomio.
  - **valor\_polinomio**: calcula el valor del polinomio para un número  $x_0$

$$P(x_0) = a_0 + a_1 x_0 + a_2 x_0^2 + \dots + a_n x_0^n$$

- **Observación**:
  - Se debe elegir la representación interna que se desee: vector o lista.
  - Se valorará la **eficiencia** en la representación.
    - Por ejemplo, ¿cómo se podría representar de forma eficiente el polinomio  $P(X) = X^5 - 1$ ?

### 5. Codifica un programa que permita la gestión de un polinomio

- Crear un polinomio.
- Escribir el polinomio.
- Modificar el coeficiente de un monomio del polinomio.
- Aplicar el polinomio a un número.
- Grabar un polinomio en un fichero.
- Cargar un polinomio desde un fichero.
- Etc.