



# Estructuras de Datos

Grado en Informática  
Segundo Curso, segundo cuatrimestre  
Escuela Politécnica Superior de Córdoba  
Universidad de Córdoba  
Curso académico 2014-2015



---

## Práctica 1. Implementación y uso de tipos abstractos de datos

### Objetivos

El estudiante debe ser capaz de

- implementar los tipos abstractos de datos **Punto2D** y **Recta2D** a partir de sus especificaciones
- y codificar un programa que utilice dichos tipos abstractos para resolver un determinado problema.

### Enunciado

Implementa las clases **Punto2D** y **Recta2D** a partir de las siguientes especificaciones:

- **TAD Punto2D: modela puntos en el plano.**
  - **Atributos**  
*Real x; // Coordenada x de Punto2D*  
*Real y; // Coordenada y de Punto2D*
  - **Constructores**  
*Punto2D(x, y:Real) //Crea un nuevo punto dadas sus coordenadas.*
  - **Observadores**  
*Real getX() //devuelve la coordenada x.*  
*Real getY() //devuelve la coordenada y.*  
*Bool sonIguales(Punto2D p) // ¿Son iguales los puntos?*
  - **Modificadores:**  
*setX(nuevoX: Real) // asigna al atributo x el valor nuevoX*  
*setY(nuevoY: Real) // asigna al atributo y el valor nuevoY*
- **TAD Recta2D (usa Punto2D): modela una recta en el plano.**
  - **Atributos**  
*Real a; // coeficiente a de Recta2D*  
*Real b; // coeficiente b de Recta2D*

*Real c; // coeficiente c de Recta2D*

- **Constructores:**

*Recta2D(a,b,c:Real) //Crea una recta usando la ecuación implícita  $ax+by+c=0$*

*Recta2D(p,q:Punto2D) //Crea una recta que pasa por los punto p y q.*

*precondición: p.sonIguales(q) = false // Los puntos p y q no son iguales*

- **Observadores**

*Real getA() //Devuelve el coeficiente a de la recta*

*Real getB() //Devuelve el coeficiente b de la recta*

*Real getC() //Devuelve el coeficiente c de la recta*

*Real distancia(Punto2D p) // Calcula la mínima distancia del punto a la recta*

- **Modificadores**

*setA(nuevoA: Real) // asigna al atributo a el valor nuevoA*

*setB(nuevoB: Real) // asigna al atributo b el valor nuevoB*

*setC(nuevoC: Real) // asigna al atributo b el valor nuevoC*

- **Programa de uso de las clases**

- Implementa un programa que haga uso de las clases implementadas para calcular el error cometido por la aproximación poligonal de un contorno (curva cerrada).

- El contorno estará definido por una serie de puntos (x,y) que están almacenados en un fichero de texto.

- La aproximación poligonal está compuesta por puntos de ese contorno y está almacenada en un fichero de texto como una secuencia ordenada ascendentemente de números enteros (numerados desde 0), que indican las posiciones de los puntos del contorno que forman la aproximación poligonal.

- El contorno se guardará en memoria interna como un vector de Puntos de la STL y la aproximación poligonal como un vector de enteros de la STL.

- Para cargar el contorno y la aproximación poligonal en memoria, se implementarán las funciones:

*void cargarContorno(vector<Punto2D> &contorno, char \*ficheroContorno);*

*void cargarAproximacionPoligonal (vector<int> &aproximacionPoligonal,*

*char \*ficheroAproximacionPoligonal);*

- El error de la aproximación poligonal se calculará como la suma de los errores de cada uno de los lados que componen la aproximación poligonal.

- El error de un lado ( $P_i, P_j$ ) de la aproximación poligonal se calcula a partir de la siguiente expresión:

$$\text{error}(P_i, P_j) = \sum_{k=i+1}^{k=j-1} d(P_k, \overline{P_i P_j})^2$$

donde

$$d(P_k, \overline{P_i P_j}) = \frac{|a x_k + b y_k + c|}{\sqrt{a^2 + b^2}}$$

es la distancia del punto  $P_k$  a la recta definida por los puntos  $P_i$  y  $P_j$ :

$$a X + b Y + c = 0$$

- Para calcular el error se implementará la función

***double calcularError(const vector<Punto2D> &contorno,  
const vector<int> &aproximacionPoligonal);***

- Usar como ejemplo de prueba el contorno almacenado en el fichero “***abrelatas.txt***” cuya aproximación poligonal está almacenada en el fichero “***aproximacion.txt***”.

### **Desarrollo de la práctica**

- Tiempo de realización: 2 semanas.
- Comienzo de la práctica: semana del 16 al 20 de febrero.
- Entrega de la práctica: semana del 2 al 6 de marzo.