

# **Documentación de Tipos Abstractos de Datos (ADTs)**

Estudiante: Fernando Matamoros Acuña  
Estructuras de Datos I  
Setiembre 2025

# ADT Triángulo

## Especificación informal

Un triángulo es una figura geométrica plana formada por tres vértices no colineales. Este ADT permite representar un triángulo mediante las coordenadas de sus tres vértices en el plano cartesiano, y proporciona operaciones para calcular sus propiedades geométricas y determinar su clasificación.

## Especificación formal

Nombre del ADT	Triangulo
Atributos	x1, y1: Coordenadas del primer vértice (double) x2, y2: Coordenadas del segundo vértice (double) x3, y3: Coordenadas del tercer vértice (double)
Operaciones	Constructor: Triangulo(double x1, double y1, double x2, double y2, double x3, double y3)
	Modificador: void modificarCoordenadas(double x1, double y1, double x2, double y2, double x3, double y3)
	Calcular área: double calcularArea()
	Calcular perímetro: double calcularPerimetro()
	Determinar tipo: string determinarTipo()
	Validar triángulo: bool esTrianguloValido()
	Calcular distancia: double calcularDistancia(double x1, double y1, double x2, double y2)

# ADT Fecha\_hora

## Especificación informal

Una fecha y hora representa un momento específico en el tiempo, incluyendo día, mes, año, hora, minuto y segundo. Este ADT permite manipular fechas y horas, realizar cálculos temporales y validar la corrección de los datos ingresados.

## Especificación formal

Nombre del ADT	FechaHora
Atributos	dia: Día del mes (1-31) mes: Mes del año (1-12) anio: Año (entero positivo) hora: Hora del día (0-23) minuto: Minuto (0-59) segundo: Segundo (0-59)
Operaciones	Constructor: FechaHora(int dia, int mes, int anio, int hora, int minuto, int segundo)
	Modificador: void modificarFechaHora(int dia, int mes, int anio, int hora, int minuto, int segundo)
	Obtener día de la semana: string obtenerDiaSemana()
	Días entre fechas: int diasEntreFechas(FechaHora otraFecha)
	Formatear fecha y hora: string formatearFechaHora()
	Validar fecha: bool esFechaValida()
	Año bisiesto: bool esBisiesto(int anio)
	Días en mes: int diasEnMes(int mes, int anio)

# Análisis de complejidad

## ADT Triangulo

Todas las operaciones del ADT Triangulo tienen complejidad  $O(1)$  porque:

- Los cálculos son directos y no dependen del tamaño de entrada
- Se realizan operaciones matemáticas simples (suma, resta, multiplicación)
- No hay bucles ni recursión

## ADT FechaHora

Todas las operaciones del ADT FechaHora tienen complejidad  $O(1)$  porque:

- Los cálculos son directos y no dependen del tamaño de entrada
- Los bucles en díasEntreFechas son de tamaño constante (máximo 12 meses, años limitados)
- No hay estructuras de datos dinámicas

## Descripción de Archivos

- triangulo.h: Contiene la declaración de la clase Triangulo con todos sus atributos y métodos
- triangulo.cpp: Implementa todas las operaciones del ADT Triangulo
- fecha\_hora.h: Contiene la declaración de la clase FechaHora con todos sus atributos y métodos
- fecha\_hora.cpp: Implementa todas las operaciones del ADT FechaHora
- main.cpp: Programa principal que permite probar ambos ADTs mediante un menú interactivo

## Conclusiones

- Encapsulación: Ambos ADTs implementan correctamente el principio de encapsulación, manteniendo los atributos privados y proporcionando una interfaz pública clara.
- Validación: Se implementaron validaciones robustas para asegurar la integridad de los datos.
- Eficiencia: Todas las operaciones tienen complejidad  $O(1)$ , lo que garantiza un rendimiento óptimo.
- Reutilización: Los métodos auxiliares permiten la reutilización de código y facilitan el mantenimiento.
- Documentación: La especificación formal e informal proporciona una comprensión clara del comportamiento de cada ADT.