

Objetivos

Unidad 3: Estructuras Discretas no recursivas y análisis de algoritmos.

La actividad planteada contribuye al desarrollo del siguiente objetivo específico:

OE3.8. Diseñar e implementar un API para un proyecto de pequeña escala, utilizando un lenguaje de programación orientado a objetos, librerías e incluyendo parametrización y generics.

OE3.9. Diseñar e implementar una solución a un problema utilizando un lenguaje de programación teniendo en cuenta un criterio de eficiencia computacional y estándares de codificación seguros.

Cristian Camilo Cardona Osorio - A00369414

Jesús Garces – A00368408

Defina el TAD para Conjuntos e impleméntela en Java haciendo uso de generics.

1. **[50 pts]** Diligencie la tabla del TAD con todos sus campos: **[1pto]** nombre, **[10 pts]** objeto abstracto, **[15 pts]** invariante, **[14 pts]** operaciones principales, estableciendo entradas, salidas y el tipo de operación (constructora, modificadora o analizadora) y **[10 pts]** especificación detallada de dichas operaciones. No es necesario definir las precondiciones y postcondiciones de las operaciones.
2. **[50 pts]** Utilizando Generics defina la interfaz genérica e implemente la clase genérica Conjunto a partir de su definición de TAD del punto anterior. Para probar el funcionamiento de su implementación reutilice el código de su calculadora de conjuntos pero en esta instancia use su propia clase para ello.

Este seguimiento puede realizarse en parejas.

TAD CONJUNTOS		
Conjuntos = { Nombre = <nombre>, Tipo = <tipo>, Universo = <universo> unión< $A \cup B = (x x \in A \vee x \in B)$ >, intersección=< $A \cap B = (x x \in A \wedge x \in B)$ > diferencia=< $A - B = (x x \in A \wedge x \notin B)$ > }		
{inv : $A = (x_1 (x_1 \in A), A \neq (x_1, x_1))$ }		
Operaciones primitivas:		
Operaciones:	Entradas:	Salidas:
crearConjunto:		→ Conjunto
infoUniverso:	Conjunto	→ Texto
union:	Conjunto x Conjunto	→ Texto
interseccion:	Conjunto x Conjunto	→ Texto
diferencia:	Conjunto x Conjunto	→ Texto

crearConjunto ()

Crea un nuevo conjunto con los datos ingresados

{pre: TRUE}

{post: conjunto = {tipo de dato...}}

Unión ()

Se realiza una unión entre dos conjuntos

{pre: crearDosConjuntos}

{post: unión = {unión." "}}

intersección ()

Se realiza una intersección entre dos conjuntos

{pre: crearDosConjuntos}

{post: interseccion = {intersección." "}}

diferencia ()

Se realiza una diferencia entre dos conjuntos

{pre: crearDosConjuntos}

{post: diferencia = {diferencia." "}}

infoUniverso()

Se realiza una diferencia entre dos conjuntos

{pre: crearUnConjuntos}

{post: infoUniverso = {infoUniverso."tipo,tamaño "}}