# PRESENTACION TAREA 8

Alumnos: Cristóbal Contreras Beltrán

Alfredo Méndez Llaupe

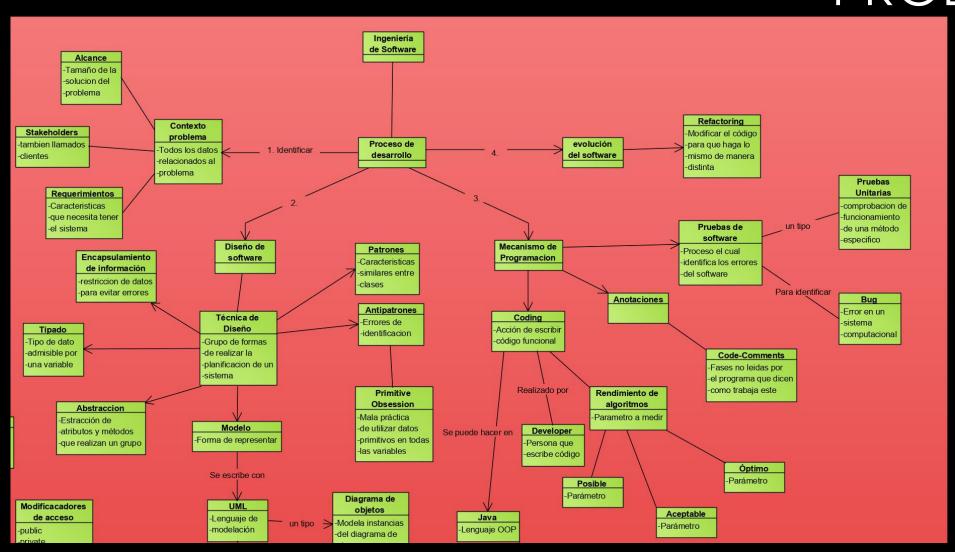
Asignatura: ICC311 - 1: Estructura de Datos

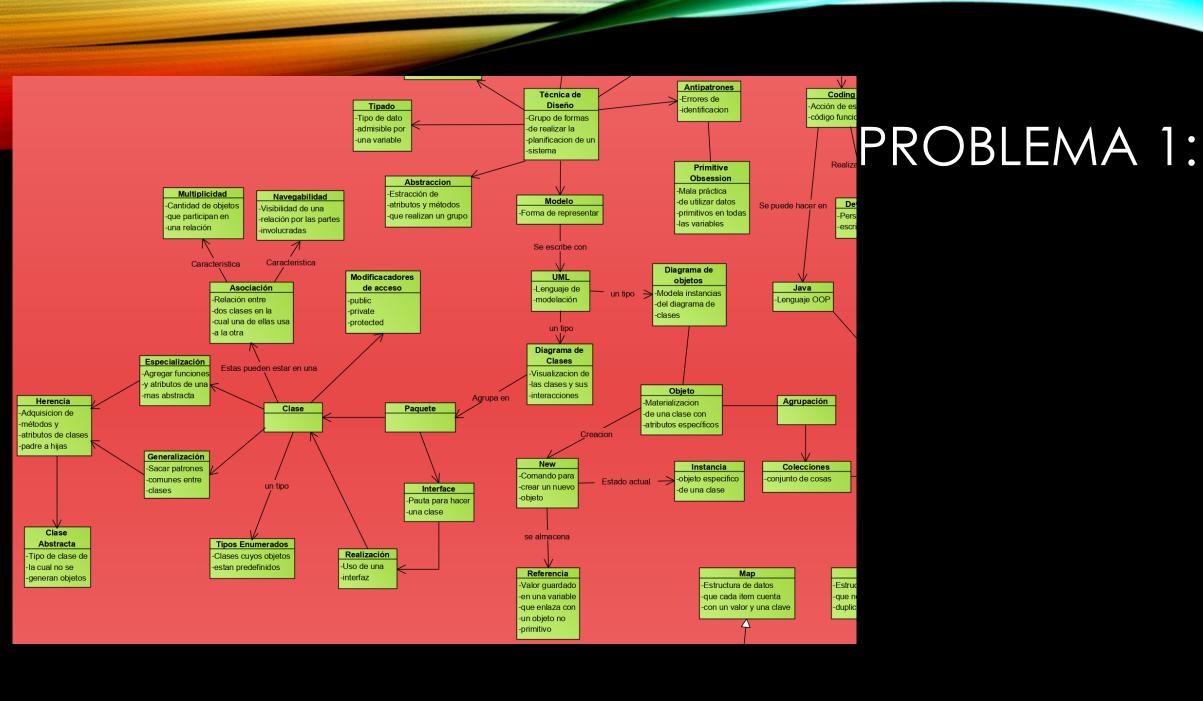
Profesor: Pablo Valenzuela Toledo

### Ingeniería de Software Refactoring -Modificar el código -para que haga lo -mismo de manera -distinta Contexto problema Todos los datos relacionados al problema Proceso de desarrollo evolución del software -funcionamiento -de una método -específico software Mecanismo de Programacion Diseño de software Patrones -Características -similares entre -clases -Proceso el cual -identifica los errore Coding -Acción de escribir -código funcional Grupo de forma de realizar la -Fases no leidas por -el programa que dicen -como trabaja este Primitive Obsession -Mala práctica -de utilizar datos Rendimiento de algoritmos -Parametro a medir -Estracción de -atributos y métodos -que realizan un grupo Modelo -Forma de representar -primitivos en toda -las variables Optimo -Parámetro Posible -Parámetro Java -Lenguaje OOP Especialización -Agregar funciones -y atributos de una -mas abstracta Objeto -Materializacion -de una clase con -atributos especificos Agrupación métodos y atributos de clase padre a hijas Generalización -Sacar patrones -comunes entre -clases Java Collection -Interfaz base Clase Abstracta -Tipo de clase de -la cual no se -generan objetos Tipos Enumerados -Clases cuyos objetos -estan predefinidos Referencia -Vaior guardado -en una variable -que enlaza con -un objeto no -primitivo List -que cada item cuenta -con un valor y una clave -definicion de los -métodos necesarios HashMap -Estructura de -datos que usa -función de hash -y elementos con -clave y valor HashSet -Estructura de -datos basado er -un set con funcio-de hash Fifo -First in -First out ArrayList -Estructura lineal que usa arregios Traverse -recorrer nodo a -nodo -Componente -donde se almacena Hashcode -Valor resultante Head -Primer nodo Buckets -Lugar donde se -almacena uno o -varios items Colisiones -Cuando dos dato Threshold -Valor limite que -puede alcanzar -la division entre

### PROBLEMA 1: DIAGRAMA COMPLETO

## PROBLEMA 1:





### -Parámetro UML objetos Aceptable Lenguaje de Java -Modela instancias Parámetro Lenguaje OOP -modelación -del diagrama de -clases un tipo forEach Diagrama de -Ciclo for que Clases -recorre todos Visualizacion de -elementos que -las clases y sus -usen iterador Objeto -interacciones -Materializacion Agrupación -de una clase con -atributos específicos <<use>>> New Java Instancia Colecciones Collection -Comando para Se represeta -objeto especifico conjunto de cosas -crear un nuevo Interfaz base -objeto se almacena Referencia -Valor guardado Estructura de datos Estructura lineal -estructura lineal estructura lineal Interfaz con la en una variable que cada item cuenta -que permite poner definicion de los -que enlaza con con un valor y una clave duplicidad de items y meter item de v sacar item de una métodos necesarios -un objeto no una sola manera -sola forma para crear estructuras -primitivo lineales Forma forma HashMap HashSet Estructura de datos que usa -Estructura de Lifo Fifo función de hash -datos basado en Last in -First in LinkedList ArrayList y elementos con -un set con función -First out -First out -Estructura lineal -Estructura lineal -de hash -clave y valor que conecta sus que usa arreglos item a adyacentes con largo variable <<use>>> <<use>>> compuesto Se navega Función Hash Función asigna -un valor a cada -item -Componente Traverse -donde se almacenan recorrer nodo a -lo valores nodo Genera Hashcode Valor resultante -Primer nodo -Nodo final almacena Separate chaning Colisiones Buckets Threshold -Sistema para evitar Cuando dos datos Lugar donde se -Valor límite que Si existen -colisiones en un -en tienen el mismo -almacena uno o se redimensionan -puede alcanzar -bucket de un hashset -valor hash -varios items -la division entre -las datos y los -buckets

## PROBLEMA 1:

# PROBLEMA 2:

	COLLECTIONS			
Estructura/Operación	get	add	remove	contains
Array	O(1)	O(n)	O(n)	O(n)
ArrayList	O(1)	O(1)	O(1)	O(n)
LinkedList	O(n)	O(1)	O(1)	O(n)
Queue	O(1)	O(1)	O(1)	O(n)
Stack	O(1)	O(1)	O(1)	O(n)
HashSet	O(1)	O(1)	O(1)	O(1)

# PROBLEMA 2:

	MAP			
Estructura/Operación	get	put	remove	containsKey
HashMap	O(1)	O(1)	O(1)	O(1)