

# Control 2

CIT 2000 - Estructuras de datos

Profesor: Leandro Llanza

Ayudante: Daniel Méndez

Universidad Diego Portales

7 de Abril de 2016

**Problema 1. (60 %)** El ministerio de educación ha adquirido un TDA para desarrollar un software que ayude a enseñar la materia de conjuntos, el **SetTDA**, cuya interface es:

- `void insert(Data x)` //Inserta x en el set.
- `Data delete()` //Saca un dato al azar del set y lo retorna.
- `bool isEmpty()` //Dice si el set está vacío o no.

Como está en un idioma extranjero y no cuenta con las operaciones básicas, el ministerio ha decidido solicitar apoyo a nuestra escuela de ingenieros y le ha pedido a usted que, usando el **setTDA**, desarrolle el TDA **ConjuntoTDA**, con las siguientes interfaces:

- `void agrega(Data x)` //Que agrega el dato x al conjunto.
- `Data saca()` //Que saca un elemento cualquiera del conjunto y lo retorna.
- `bool estaVacio()` //Dice si el conjunto está o no vacío.
- `bool pertenece(Data x)` //Dice si el dato x pertenece o no al conjunto.
- `Conjunto TDA interseccion(Conjunto c)` //Retorna un nuevo conjunto con el resultado de la intersección de ambos conjuntos.
- **BONUS:** `void intersecta(Conjunto c)` //Deja en el conjunto solo los elementos que pertenecen a ambos conjuntos.
- `ConjuntoTDA union(Conjunto c)` //Retorna un nuevo conjunto con el resultado de la unión de ambos conjuntos.
- **BONUS:** `void une(Conjunto c)` //Deja en el conjunto todos los elementos que pertenecen a ambos.

Recuerde que los conjuntos no tienen (y no pueden contener) elementos repetidos.

**Problema 2. (40 %)** Considere la siguiente clase ítem:

```
class Item{
    public:

    string valor;
    int cant_llamados;

}
```

Se le pide implementar una clase lista que almacene elementos de tipo ítem y que tenga como mínimo los siguientes métodos:

- `void agregar(Item i)` //Agrega el Ítem i a la lista.
- `int probabilidad(string valor)` //Devuelve la probabilidad de aparición del ítem con string valor. Considere elementos únicos.
- `string llamar_kesimo(int k)` //Entrega el valor del ítem con la k-ésima mejor probabilidad de aparición. Tenga cuidado de actualizar la cantidad de llamados del ítem.

Se le recomienda, pero no es obligación, mantener la lista ordenada según las probabilidades de aparición. Recuerde implementar la clase lista completa (Incluya al menos el constructor para una lista vacía). Puede definir atributos o clases a su necesidad. Si usa métodos extras, debe programarlos.