

### **PARTE 3: Ejercicio de programación JAVA**

Este ejercicio comprende **2** partes:

1. Desarrollo de la funcionalidad especificada más abajo
2. Desarrollo del programa ejecutable con datos de ejemplo provistos y entrega de todo el proyecto Netbeans en la tarea correspondiente de la webasignatura.

#### **ESCENARIO**

Una institución educativa ofrece distintos cursos, los que suelen tener diferentes reglas de prematriculaciones. El Coordinador desea contar con funcionalidades que le permitan estimar las cantidades de alumnos que se pueden inscribir a un cierto curso.

Por ejemplo, el curso **“INTEGRADOR 101”** puede nutrirse de alumnos que hayan cursado **“BASICO-ING 1”** o **“BASICO-EMP 1”** indistintamente (o ambos).

Otra situación típica es, por ejemplo, en el curso **“EXIGENTE 102”**, que solo puede ser cursado por aquellos alumnos que hayan aprobado **AMBOS** cursos **“BASICO-ING 1”** o **“BASICO-EMP 1”**.

El líder de equipo de desarrollo entiende que estas funcionalidades pueden obtenerse a partir de la abstracción **“CONJUNTO”** y sus correspondientes operaciones típicas: **Unión** e **Intersección**. Te ha encargado a tí que implementes esta abstracción y sus operaciones.

#### **PARTE 1: Funcionalidad a desarrollar (vale 50%):**

**Desarrolla** los algoritmos para implementar las operaciones de **Unión** e **Intersección** sobre el **TDA CONJUNTO**, un derivado del **TDA LISTA** (ó del **TDA LISTA ORDENADA**, ver **PARTE 2**), utilizado para representar un **CONJUNTO**.

Descargar y abrir el proyecto Netbeans desde la webasignatura.

Se requiere implementar la abstracción **Conjunto** y las operaciones asociadas **Union e Interseccion** con la interfaz:

**de iConjunto:**

**public Conjunto Union(Conjunto otroConjunto)**

**public Conjunto Interseccion(Conjunto otroConjunto)**

#### **PARTE 2: PROGRAMA (vale 40% sobre TLista, +10% si realizado sobre TListaOrdenada).**

La clase principal se denomina **“Parcial”**, y tiene su correspondiente método **“main”**. En éste, implementa lo necesario para aplicar los TDA y métodos desarrollados, ejemplificando de la siguiente forma (asumiendo que los **cursos** pueden ser representados como instancias de **Conjunto**):

1. Se provee una clase **“Alumno”** para representar a los alumnos que se inscriben a los cursos
2. Instanciar y cargar los alumnos del curso **“BasicoIng”**, (archivo **“basico-ing.txt”**) (ver la sección **“NOTAS SOBRE ARCHIVOS”** al final de este documento por el formato)

3. Instanciar y cargar los alumnos del curso **“BasicoEmp”**, (archivo **“basico-emp.txt”**) (ver la sección **“NOTAS SOBRE ARCHIVOS”** al final de este documento por el formato)
4. Crear un curso **“integrador101”**, y asignarle los alumnos que hayan cursado **“BasicoIng”** y / o **“BasicoEmp”**, utilizando la operación de **Conjunto** que corresponda. Guardar el listado de este curso en el archivo **“integrador101.txt”**, **IDEALMENTE ORDENADO POR CÓDIGO DE ALUMNO**. (ver la sección **“NOTAS SOBRE ARCHIVOS”** al final de este documento por el formato)
5. Crear un curso **“exigente102”**, y asignarle los alumnos que hayan cursado **“BasicoIng”** Y **“BasicoEmp”**, utilizando la operación de **Conjunto** que corresponda. Guardar el listado de este curso en el archivo **“exigente102.txt”**, **IDEALMENTE ORDENADO POR CÓDIGO DE ALUMNO**. (ver la sección **“NOTAS SOBRE ARCHIVOS”** al final de este documento por el formato)

### **NOTA IMPORTANTE:**

La clase **Conjunto<T>** puede derivarse de **TLista<T>** ó de **TListaOrdenada<T>**. Si se implementa correctamente esta última opción, se podrán emitir los listados solicitados ordenados en forma ascendente.

Se proveen las interfases para ambas listas, la clase **TLista** y también la clase **TListaOrdenada** en la que **se requiere reescribir el método de inserción**.

**ENTREGA:** Debes entregar TODO el proyecto Netbeans y los archivos de salida solicitados, en un archivo comprimido **“Parcial1.zip”** en la tarea **“PARCIAL1-PARTE3”** publicada en la webasignatura, hasta la hora indicada.

### **NOTAS SOBRE ARCHIVOS:**

El departamento de soporte de TI ha provisto las listas de alumnos para **“BASICO-ING1”** y **“BASICO-EMP1”**, en las cuales figuran los alumnos que han aprobado estos cursos, y contienen información de la siguiente manera, en cada línea, **SEPARADAS POR COMA:**

**COD-ALUMNO // entero**

**NOMBRE ALUMNO //string**

**Ejemplo:**

**1,JOSE**

**5,JUAN**

**EL MISMO FORMATO DEBE OBSERVARSE PARA LOS ARCHIVOS DE SALIDA REQUERIDOS**