

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**  
**FACULTAD DE CIENCIAS E INGENIERÍA**

**LENGUAJE DE PROGRAMACIÓN 1**

**10ma práctica (tipo b)**  
**Segundo Semestre 2019**

**Indicaciones Generales:**

- Duración: 110 minutos.
- Se podrá usar como material de consulta solo sus apuntes de clase (**NO** fotocopias, impresos ni hojas sueltas).
- No se pueden emplear variables globales ni estructuras. Tampoco se podrá emplear la clase **string**, **plantillas de clase**, ni las bibliotecas estándar de plantillas (**STL**). Tampoco se podrán emplear las funciones **malloc**, **realloc**, **strdup** o **strtok**, igualmente no se puede emplear cualquier función contenida en las bibliotecas **stdio.h**, **cstdio** o similares y que puedan estar también definidas en otras bibliotecas. Salvo en la sobrecarga de los operadores **>>** y **<<**, no se podrán definir funciones (ni plantillas de funciones) independientes que no estén ligadas como métodos a alguna de las clases planteadas.
- La cláusula **friend** solo se podrá emplear en el caso de clases autoreferenciadas para ligar el nodo con la clase **inmediata** que encapsula la lista, en ningún caso adicional. No se considerará en la nota las clases que violen esto
- Deberá respetar estrictamente el **encapsulamiento de datos** en las clases a todo nivel, no se considerará en la nota aquellos métodos que violen el encapsulamiento de los datos.
- **Deberá guardar los estándares en la definición y uso de todas las clases desarrolladas.**
- Deberá modular correctamente el proyecto en archivos independientes. LAS SOLUCIONES DEBERÁN DESARROLLARSE BAJO UN ERICTO DISEÑO DESCENDENTE. Cada módulo **NO** debe sobrepasar las 20 líneas de código aproximadamente. El archivo **main.cpp** solo podrá contener la función **main** de cada proyecto y el código contenido en él solo podrá estar conformado por tareas implementadas como funciones. En cada archivo que implemente en los proyectos (.h y .cpp) deberá colocar un comentario en el que coloque claramente su nombre y código, de no hacerlo se le descontará 0.5 puntos por archivo.
- El código comentado **NO SE CALIFICARÁ.**
- Los programas que presenten errores de sintaxis o de concepto se calificarán en base al 40% de puntaje de la pregunta. Los que no den resultados coherentes en base al 60%.
- Se tomará en cuenta en la calificación el uso de comentarios relevantes.
- La **presentación, la ortografía y la gramática de los trabajos influirá en la calificación.**

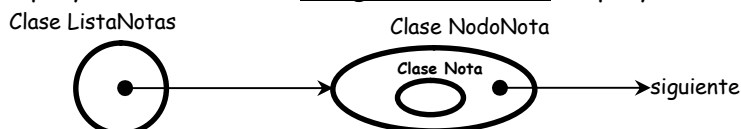
**Puntaje total:** 20 puntos.

**Problema**

Se desea elaborar una aplicación orientada a objetos que permita controlar la información de una institución educativa. Por tal motivo se solicita que se desarrolle los siguientes proyectos:

**Pregunta 1 (6 puntos)**

Elabore un proyecto denominado **"Prog01ListaNotas"**. El proyecto debe contener las siguientes clases:



**"Clase Nota"**, permitirá encapsular la nota de un alumno, tendrá como atributos el código del curso (**char\***), la nota (**int**) y el semestre que lo llevó (**char\***). Esta clase debe contener un constructor por defecto, un destructor, métodos selectores y los métodos que requiera para poder manipularlos.

**"Clase ListaNotas"**, permitirá encapsular la lista ligada de notas.

**"clase NodoNota"**, tendrá como atributos un objeto de la clase nota y el puntero siguiente.

El proyecto deberá ser capaz de armar una lista ligada de notas ordenada por el código del curso. La lista no debe aceptar cursos repetidos, pero si aparece un curso que ya esté en la lista, se le deberá reemplazar sí y solo sí lo llevó después del que esté en la lista. También debe ser capaz de imprimir el contenido de la lista en un archivo de textos, los datos deben aparecer correctamente tabulados,

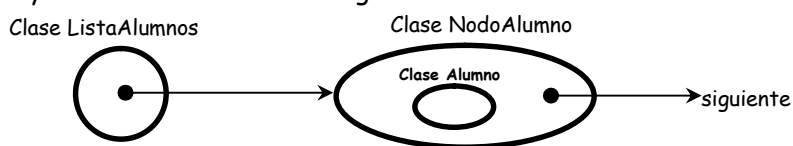
alineados y con un título que explique la naturaleza de los datos (no podrá emplear el carácter '\t'). Adicionalmente se deben contemplar métodos que permitan realizar las acciones que crea conveniente. Las dos últimas clases deben definir obligatoriamente su constructor por defecto y su destructor.

**DEBE TOMAR EN CUENTA QUE NO SE TRATA DE CLASES AISLADAS, LAS CLASES DESARROLLADAS DEBEN ALINEARSE LAS DEMÁS TAREAS REQUERIDAS EN ESTE LABORATORIO.**

**DEBE PROBAR TODOS LOS MÉTODOS DESARROLLADOS.**

### **Pregunta 2 (12 puntos)**

Elabore un proyecto denominado **"Preg02ListaAlumnos"**, copie en él las clases desarrolladas en la primera pregunta e incorpórelas al proyecto. Cualquier modificación que haga en este proyecto a las clases incorporadas no modificará la calificación de la preguntas anterior, NO SE HARÁN EXCEPCIONES. El proyecto debe contener las siguientes clases:



"Clase **ListaAlumnos**", permitirá encapsular la lista ligada de alumnos.

"Clase **NodoAlumno**", tendrá como atributos un objeto de la clase Alumno y el puntero siguiente.

"Clase **Alumno**", permitirá encapsular la información de un alumno. Como en la institución existen dos tipos de alumnos, los cuales pueden ser alumnos regulares y alumnos de intercambio, se requiere que esta información se manipule obligatoriamente con polimorfismo. Para esto se requiere desarrollar una jerarquía de clases como se muestra a continuación:



Los datos miembros o atributos para estas clases serán los siguientes:

"Clase **BaseAlum**": código (int) y nombre (char\*).

"Clase **AlRegular**": especialidad (char\*) y facultad (char\*) a la que pertenece.

"Clase **AlIntercambio**": País de origen (char\*), idiomas que domina (char\*\*) y número de semestres que permanecerá en la institución (int). No podrá modificar, agregar o eliminar los atributos.

La implementación de estas tres últimas clases contemplará la definición de atributos, constructores por defecto, destructores, métodos selectores, un método que permita leer los datos desde un archivo de texto y otro que permita imprimirlo en otro archivo de texto, y un método que devuelva el tipo de miembro ('R' para alumno regular, 'I' para alumno de intercambio). La memoria asignada en todos los casos debe ser óptima.

De acuerdo a esto, la "clase **Alumno**" deberá tener dos atributos: uno será un puntero a la "clase **BaseAlum**" y el otro un objeto a la "clase **ListaNotas**". Deberá definir un constructor por defecto y el destructor para esta clase, además todos los métodos que se requieren para cargar los datos de los alumnos y sus notas.

El proyecto deberá ser capaz de armar una lista ligada de alumnos a partir de los archivos que a continuación se describen, ordenada por el código del alumno, sin tocar la lista de notas. Una vez armada la lista deberá ser capaz de colocar las notas de los alumnos en las listas de notas de cada alumno (sin repetir). También debe ser capaz de imprimir el contenido de toda la lista en un archivo de textos, según el formato que se indica más adelante, los datos deben aparecer correctamente tabulados y alineados (no podrá emplear el carácter '\t').

Los archivos de datos serán similares a los que se muestra a continuación:

**Alumnos.csv**

```
R,20001234,Castro/David,Ing. Informatica,EEGGCC
I,9876,Velarde/Mario,Puerto Rico,Inglés,Francés,Alemán,Italiano,4
I,5371,Caceres/Jose,Brasil,Portugués,Español,1
.....
```

**Notas.csv**

```
20001234,MEC123,10,2017-2
20171122,INF555,17,2019-1
20001234,MEC123,9,2018-1
.....
```

El archivo "**Alumnos.csv**" contiene en cada línea: el tipo de alumno (R: Alumno regular, I: Alumno de intercambio), código y del nombre alumno, y a continuación los datos según su tipo. Si es alumno regular será: especialidad y facultad. Si es alumno de intercambio: país, los idiomas que domina y número de semestres que permanecerá en la institución. El archivo "**Notas.csv**" contiene en cada línea: el código del alumno, el código del curso, la nota y el semestre que lo llevó.

El reporte que se espera será similar al siguiente:

Alumnos de la Institución		
Código alumno: 9876		
Alumno de Intercambio		
Nombre del alumno: Velarde/Mario		
País: Puerto Rico		
Idiomas: Inglés - Francés - Alemán - Italiano		
Permanecerá 4 semestres		
LISTADO DE NOTAS:		
CÓDIGO	NOTA	Semestre
INF159	17	2017-2
ING281	15	2017-2
...		
Código alumno: 20001234		
Alumno Regular		
Nombre del alumno: Castro/David		
Especialidad: Ing. Informática		
Facultad: EEGGCC		
LISTADO DE NOTAS:		
CÓDIGO	NOTA	
MEC123	10	
MAT777	12	
...		

**Pregunta 3 (2 puntos)**

Elabore un proyecto denominado "**Preg03SuperClase**", copie en él las clases desarrolladas en la primera y segunda preguntas e incorpórelas al proyecto. Cualquier modificación que haga en este proyecto a las clases incorporadas no modificará la calificación de las preguntas anteriores, NO SE HARÁN EXCEPCIONES.

El proyecto debe contener una clase denominada "**clase InstituciónEducativa**" la clase debe contener como único atributo un objeto (NO un puntero) de la clase listaAlumnos. La clase se encargará de abrir los archivos e invocar a los métodos que permitan cargar los alumnos y sus notas a la lista de alumnos y luego mostrar el reporte.

**CONSIDERACIONES FINALES:**

**EN NINGUNA PREGUNTA PODRÁ AGREGAR NUEVOS ATRIBUTOS A LAS CLASES SOLICITADAS, TAMPOCO MODIFICAR SUS TIPOS DE DATOS O NOMBRES.**

Al finalizar la práctica, coloque todos los proyectos en una carpeta denominada Laboratorio10, comprima<sup>1</sup> la carpeta Laboratorio10 en un archivo con nombre <código del alumno con 8 dígitos>.<extensión del archivo comprimido y súbalo a la Intranet del curso, en el enlace Documentos, en la carpeta \Laboratorio9\<código del horario>\<aula>.

Profesores del curso: Rony Cueva  
Miguel Guanira

San Miguel, 29 de noviembre del 2019.

<sup>1</sup> Para evitar problemas en la corrección de la prueba, utilice el programa de compresión que viene por defecto en Windows (Zip).