



### Laboratorio No. 3

**Realizar:** Elaborar ordenamientos y búsquedas.

Nota: esta es la continuación del programa para llevar asistencia de una clase (Lab 1).

**Realizarse en grupos de 3 estudiantes.**

**Fecha de entrega:** jueves 12 de febrero.

#### Objetivos:

- Utilizar el polimorfismo en el uso de la interfaz java **Comparable** para implementar ordenamientos y búsquedas.
- Utilizar diagramas **UML de Secuencia** para representar escenarios de Casos de Uso.
- Utilizar archivos para guardar y recuperar información.

#### Primera Parte: Diagramas UML 60 puntos.

- Completar el diagrama de Clases que componen su programa de control de asistencia elaborado en el Laboratorio #1 . Enviar a SAKAI el archivo en formato JPG.
- Completar el diagrama de Casos de Uso, para que incluya la nueva funcionalidad de ordenar y buscar. También debe mostrarse que existirá un almacenamiento en archivos del registro de alumnos y sus asistencias al curso. Usted puede definir los criterios de ordenamiento, ejemplo: ordenar alumnos por apellido/nombre o por número de carnet, o por cantidad de inasistencias. También puede indicar sus criterios de búsqueda: por número de carnet o por apellido/nombre del alumno. Enviar a SAKAI el archivo JPG del diagrama.
- Elaborar la descripción narrativa de los casos de uso que este agregando. Indicar sus precondiciones, postcondiciones, flujo básico, flujos alternos. Enviar a SAKAI el documento Word o PDF de estos casos de uso.
- Elegir dos casos de uso (uno de los que presentó en el Lab #1 y otro de los nuevos que este agregando) y construir su diagrama **UML de Secuencia**. Modele solamente un escenario de cada uno de ellos. Utilice la notación del UML 2 para representarlos. Enviar a SAKAI el archivo JPG del diagrama.

#### Segunda Parte: Programa Java 40 puntos.

- Agregue a su programa la funcionalidad descrita en sus Casos de Uso especialmente para: a) guardar y recuperar de archivos, los datos del control de asistencia; b) ordenar los datos del registro de asistencia; c) buscar uno o más datos en el registro de asistencia.
- Debe implementar la interfaz **Comparable** y su método **compareTo** para sus algoritmos de ordenamiento y búsqueda.
- Recuerde definir sus propias excepciones y utilizar los manejadores de excepciones necesarios para que el programa sea robusto y defensivo.
- Incluya los "print screen" y documentación sobre como utilizar su programa. NOTA: puede grabar la ejecución de su programa, incluyendo audio, en el sitio <http://www.screencast-o-matic.com/>.
- Incluya los archivos java (no los class) y la documentación de uso en un archivo zipeado y súbalo al SAKAI.

#### Tercera parte: OPCIONAL 20 puntos extra.

- Incluya más de un criterio para hacer el ordenamiento y búsqueda de información. Por ejemplo puede dar al usuario la opción de ordenar por apellido/nombre del alumno o por cantidad de inasistencias. Es decir su programa implementa ambos ordenamientos y el usuario los selecciona de un menú.
- HINT: investigue como realizar esto empleando herencia entre interfaces. Puede utilizar cualquier mecanismo que encuentre útil, siempre que implemente la interfaz **Comparable** y su método **compareTo**

**NOTA: les recomendamos que individualmente y en su casa puedan trabajar el ejercicio indicado en el documento "Ordenamiento PreLab3.pdf"**