

Se trata de diseñar una jerarquía de clases (e interfaces) para hacer la gestión de una institución universitaria teniendo en cuenta lo siguiente:

- La institución tiene estudiantes y profesores. A su vez los estudiantes pueden ser de grado o de postgrado; estos últimos de máster o de doctorado.
- Los datos de un profesor son: nombre, DNI, departamento, salario anual y cotización anual.
- Los estudiantes tienen un nombre, un DNI y unas tasas que deben pagar y que están dentro de unos límites mínimo y máximo que dependen del tipo de estudiante. Además, según el tipo de estudiante hay otros datos:
 - Los estudiantes de grado tienen un nombre de título de grado. Las tasas deben estar entre 200 y 1500 euros.
 - Los estudiantes de post grado tienen una beca que les concede la Universidad que no es la misma para todos y un nombre de título de post grado. Además:
 - Las tasas de los estudiantes de Máster están entre 2000 y 4000 euros.
 - Los estudiantes de doctorado tienen una nota del Máster y están inscritos en un Programa de Doctorado. Cada Programa está caracterizado por un nombre y una beca que concede a todos sus estudiantes y que tiene el mismo importe para todos ellos. Esta beca se suma a la que ya tienen por ser alumnos de post grado. Sus tasas están entre 200 y 500 euros.

NOTA: Los métodos set y get “triviales” no hay que programarlos.

En el diseño de la jerarquía, hay que tener en cuenta también que se desean hacer operaciones como las siguientes sobre un vector estático, **vU**, que contiene objetos de todas las clases:

- **saldoTotal (vU)** . Calcula y retorna el saldo total de la institución. Este saldo es la suma de los saldos individuales. El saldo individual de un estudiante de grado o máster es positivo y su importe corresponde a las tasas que paga. El saldo de un profesor es negativo y corresponde a la suma del salario y la cotización. En el caso de los estudiantes de postgrado el saldo es la diferencia entre las tasas que pagan y la beca que reciben.
- **estudiantesMaster (vU, “Ingeniería Informática”)** . Calcula y retorna el número de estudiantes que cursan o han cursado el master con nombre “Ingeniería Informática”.
- **cobroTotal (vU)** . Es la suma de lo que cobran los miembros de la institución en salarios. Actualmente son los profesores pero en un futuro se quiere incluir a otros tipos de empleados aún no contemplados. El método no debería verse modificado ante la aparición de nuevos tipos de asalariados.
- **ordenaPorCobro (vU)** . Ordena el vector de modo que queden primero los que cobran y luego los que no cobran.

Los criterios de valoración son los siguientes:

(20%) Estructura de Clases

- Identificación de las clases apropiadas
- Relaciones de herencia y composición
- Identificación de los datos propios de cada clase
- Identificación de los métodos
- Identificación de constantes en cada clase

(5%) Interfaces

- Identificación de interfaces
- Implementación en las clases adecuadas

(10%) Diagrama UML (Hacerlo en una página independiente)

(30%) Programación de los métodos

- Métodos propios
- Métodos sobreescritos
- Métodos de las interfaces
- Uso de las constantes en los métodos
- Constructores

(20%) Diseño de los algoritmos

- Claridad
- Corrección
- Eficiencia

(15%) Uso del polimorfismo

- Identificación dinámica de clases
- Downcasting