

# Proyecto de Sistema de Gestión Académica de Estudiantes y Asignaturas (SGAEA)

## 1.- Introducción

Se requiere desarrollar un sistema para gestionar la información académica de los estudiantes en una institución educativa. Este sistema debe permitir administrar tanto las calificaciones como las asignaturas de cada estudiante.

## 2.- Requisitos del Proyecto

### 2.1.- Generales

Todas las salidas serán por consola y todas las entradas por teclado. Opcionalmente se puede diseñar un interfaz web sencillo para interactuar con el sistema, lo cual será valorado en la calificación. Es posible que, antes de la fecha de entrega, se añada algo al enunciado si se ha detectado algún error u omisión, siempre que haya suficiente antelación para su desarrollo.

El código debe ser limpio, eficiente y comentado, la nomenclatura coherente y los nombres de variables y métodos intuitivos y bien elegidos.

Todo el sistema debe ser orientado a objetos, usando clases, herencia, encapsulación y sobrecarga y usando objetos y métodos predefinidos del lenguaje, siempre que existan.

Se debe incluir un conjunto de pruebas que demuestren el funcionamiento de la funcionalidad.

### 2.2.- Estudiantes

- Cada estudiante debe tener un id, nombre, edad y dirección, compuesta de calle, número, piso, código postal, provincia y localidad. El nombre debe contener sólo letras y espacios.
- Se debe permitir agregar y eliminar estudiantes de la lista. No debe haber duplicados
- Se debe permitir matricular y desmatricular estudiantes en una o varias asignaturas.
- Se debe registrar la fecha de matrícula o desmatriculación y se debe poder mostrar en formato español.
- Cada estudiante puede recibir varias calificaciones por asignatura. Éstas deben ser números enteros entre 0 y 10.
- Se debe calcular el promedio de todas las calificaciones del estudiante.
- Se debe permitir la búsqueda de asignaturas cuyo nombre coincida parcialmente con un patrón de texto.

## 2.3.- Asignaturas

- Cada asignatura tiene un nombre, que debe contener solo letras, números romanos y espacios, y una lista de calificaciones que habrá ido obteniendo el alumno.
- Se debe poder calcular el promedio de las calificaciones de cada asignatura.
- Se debe permitir agregar y eliminar asignaturas.

## 2.4.- Operaciones sobre la Lista de Estudiantes

- Se debe poder eliminar estudiantes de la lista.
- Se debe calcular el promedio general de todos los estudiantes.
- Se debe generar un reporte con las calificaciones de cada estudiante, sus asignaturas y sus promedios.
- Se debe permitir la búsqueda de estudiantes cuyo nombre coincida parcialmente con un patrón de texto.

## 2.5.- Generación de Listados

- Lista completa de estudiantes.
- Listado de asignaturas de cada estudiante.
- Calificaciones y promedios de cada asignatura.

## 3.- Criterios de evaluación

Los siguientes criterios son obligatorios. Si alguno no se cumple, no se puede obtener una calificación de aprobado

- Código que, evidentemente, resuelva el problema.
- Uso de clases, herencia, encapsulación y sobrecarga.
- Uso de Objetos y métodos predefinidos del lenguaje, siempre que existan.
- Código limpio, nomenclatura coherente, nombres de variables y métodos intuitivos y bien elegidos, eficiente y comentado.
- Inclusión de un conjunto de pruebas que demuestren el funcionamiento de la funcionalidad.
- Defensa del código, sin duda ni vacilación. Posible modificación de código en la defensa para realizar alguna tarea sencilla.

## 4.- Entrega

La fecha de entrega será el 10/12 antes de las 15:00 y la defensa será el 11/12 de 10:15 a 14:45. Habrá que subir el código a un repositorio de github en un fichero .js