



Proyecto: *Sistema de control escolar*^{1.0}

Realiza la implementación de un sistema básico de control escolar para la administración del avance académico de los alumnos de una universidad.

- ▷ Puntos a considerar para el diseño e implementación de las clases:
 - Estarán localizadas en su propio archivo bajo el espacio de nombres **Faculty**.
 - Deberán seguir los patrones de diseño presentados en clase.
 - Se proponen las clases: **Alumno**, **Asignatura**, **Calificacion** y **ControlEscolar**.
 - Para las clases receptoras de la información de los archivos, se deberán elegir apropiadamente los tipos de datos y los atributos de lectura/escritura de sus propiedades, basándose de la información presente en los archivos y del uso de los objetos en la aplicación.
 - Si se considera necesario, se permite la implementación de clases adicionales.
- ▷ La clase **Alumno** implementará lo siguiente:
 - Propiedades **Matricula**, **Nombre** y **Apellido**.
 - Constructor con parámetros.
 - Archivo de origen: **alumnos.txt**.
- ▷ La clase **Asignatura** implementará lo siguiente:
 - Propiedades **Clave**, **Nombre** y **Creditos**.
 - Constructor con parámetros.
 - Archivo de origen: **asignaturas.txt**.
- ▷ La clase **Calificacion** implementará lo siguiente:
 - Propiedades **Matricula** (de alumno), **Clave** (de asignatura) y **Calificacion**; ésta última podrá poseer los siguientes valores: (a) -1, si la asignatura no ha sido cursada, (b) 0 a 69 si la asignatura ha sido cursada y reprobada y (c) 70 a 100 si la asignatura ha sido cursada y aprobada.
 - Constructor con parámetros.
 - Archivo de origen: **calificaciones.txt**.
- ▷ La clase **ControlEscolar** implementará lo siguiente:
 - Listas privadas de alumnos, asignaturas y calificaciones.
 - Constructor sin parámetros; se cargarán las listas desde los archivos correspondientes.



- **Importante:** Esta clase será la encargada de la administración de las listas, así como de la carga y actualización de la información en los archivos. El diseño y la funcionalidad de esta clase se evaluarán mediante la revisión de la funcionalidad de la aplicación.

▷ Menú principal de la aplicación.

- **Mostrar.** Abre un submenú con las siguientes opciones:
 - **Alumnos.** Se muestra un listado de los alumnos, ordenado por apellido.
 - **Asignaturas.** Se muestra un listado de las asignaturas, ordenado por clave.
 - **Volver.**
- **Alta de alumnos.** Se solicitan los datos del nuevo alumno y se genera su registro completo de asignaturas no cursadas. La matrícula es un número consecutivo que debe ser generado por la aplicación.
- **Asignar calificaciones.** Se solicitan la matrícula del alumno, la clave de la asignatura y la calificación obtenida en la misma. Sólo se podrá asignar una calificación a una asignatura que todavía no ha sido aprobada.
- **Reportes.** Abre un submenú con las siguientes opciones:
 - **Promedio total de alumnos.** Se muestra un listado de los alumnos que incluye (a) matrícula y nombre completo del alumno y (b) promedio obtenido en todas sus asignaturas cursadas; el listado se ordena de forma descendente por promedio.
 - **Promedio parcial de alumnos.** Se muestra un listado de los alumnos que incluye (a) matrícula y nombre completo del alumno, (b) promedio obtenido en todas sus asignaturas aprobadas, (c) total de créditos aprobados y (d) porcentaje de créditos aprobados; el listado se ordena de forma descendente por promedio.
 - **Alumnos con asignaturas reprobadas.** Se muestra un listado de los alumnos junto con un sublistado de sus asignaturas reprobadas, ordenados por matrícula y clave respectivamente.
 - **Extraordinarios.** Se muestra un listado de las asignaturas junto con la cantidad de alumnos que se encuentran reprobadas en ellas, ordenado de forma descendente por la cantidad de alumnos reprobados.
 - **Volver.**
- **Salir.**



UNIVERSIDAD AUTÓNOMA
DE YUCATÁN

Facultad de Ingeniería
Ingeniería Mecatrónica

Asignatura: Programación orientada
a objetos (POO)

Grupo: 4M

Valor: N/A

- ▷ **Importante:** Durante la ejecución y prueba de la aplicación se considerará que el usuario no cometerá errores de formato al introducir información desde el teclado, como por ejemplo, escribir letras cuando se esperan números. Sin embargo, el usuario si podría especificar, por ejemplo, un valor fuera de rango o un identificador inválido, por lo cual la aplicación deberá considerar estos posibles eventos y protegerse contra la ocurrencia de excepciones o de una posible corrupción de los datos en la memoria o en los archivos.