

TRABAJO REQUERIDO NÚMERO 1

EJERCICIO 1:

Nos solicitan crear un programa que maneje una lista de alumnos. El alumno estará representado por la clase **Alumno**.

Detalle de la clase alumno.

Propiedades:

Legajo	int	lectura/escritura
Nombre	string	lectura/escritura
Apellido	string	lectura/escritura
Fecha_Nacimiento	date	solo escritura
Fecha_Ingreso	date	solo escritura
Edad	int	solo lectura
Activo	boolean	lectura escritura
Cant_Materia_Aprobadas	int	solo escritura

Métodos:

Antigüedad	int	Retorna la cantidad de años/meses/días que hace que el alumno asiste a la institución. El método posee un parámetro que indica en que unidad (años, meses, días) se desea el retorno. Para el cálculo se considera solo año/mes cumplido, que hace que el alumno acude a la institución. Par determinar el número de años se considera la fecha actual y la fecha de ingreso.
Materias_No_Aprobadas	int	Retorna la cantidad de materias no aprobadas a la fecha sabiendo que la carrera posee 36 Materias y considerando la cantidad de materias aprobadas.
Edad_De_Ingreso	int	Retorna la edad a la que el alumno ingresó a la institución considerando la fecha de nacimiento y la fecha de ingreso.

Constructores:

1. Constructor sin parámetros
2. Constructor con todos los parámetros que permiten inicializar las propiedades.

Finalizadores:

1. Que cuando el objeto queda liberado muestre una leyenda indicando el Legajo, Nombre y Apellido del Alumno.

Nos solicitan que la GUI (interfaz gráfica del usuario) permita visualizar la lista de los alumnos ingresados en una lista del tipo DataGridView.

La GUI debe tener botones para:

Agregar un alumno a la lista.
Borrar el alumno seleccionado de la lista.
Modificar el alumno seleccionado de la lista.

La GUI debe tener cajas de texto para ver del alumno seleccionado en la grilla:

Antigüedad
Materias_No_Aprobadas
Edad_De_Ingreso

Recursos a considerar para la resolución:

1. **Tema** Listas de programación 1. **Sugerencia:** Usar **List(of...)**
2. **Tema** Clases, Instancias, Propiedades, Métodos, Constructores, Finalizadores de la Unidad I de Programación Orientada a Objetos. **Sugerencia:** Ver el Orientador y la bibliografía recomendada.
3. **Tema** DataGridView. Grilla vista el programación 1. **Sugerencia:** <https://docs.microsoft.com/es-es/dotnet/framework/winforms/controls/datagridview-control-windows-forms>

EJERCICIO 2:

NOTA: RESPETAR LAS BUENAS PRÁCTICAS QUE PROPONE LA ORIENTACIÓN A OBJETOS.

Dado un escenario bancario nos manifiestan la necesidad de desarrollar un sistema para administrar las cuentas de los clientes.

Nos manifiestan que poseen dos tipos de cuentas: las cuentas corrientes y las cajas de ahorro.

Para el banco sus clientes son los titulares de cuentas.

Una cuenta puede poseer más de un titular y estos pueden tener más de una cuenta.

Las cuentas se identifican por un número y los titulares por su tipo y número de documento, de los titulares también se desea saber el nombre y apellido.

En las cuentas del banco se pueden realizar depósitos, extracciones y transferencias de una cuenta a otra. Estas operaciones afectan su saldo.

Un aspecto significativo es que las cuentas corrientes poseen una característica que las habilita para poder girar dinero en descubierto. El límite de descubierto es una característica de cada cuenta corriente.

Cuando en una cuenta se depositan o transfieren importes superior a 1000 pesos de debe desencadenar un evento que en su parámetro personalizado (EventArg) lleve el importe y en caso de ser una transferencia los datos de la cuenta de origen.

Si se intenta extraer más dinero del posible se debe producir una excepción personalizada que pueda ser atrapada por un try catch que indique esta condición.

Los datos deben mostrarse en grillas. Se sugieren 4:

1. Todos los titulares.
2. Todas las cuentas.
3. Las cuentas del titular seleccionado en la grilla 1.
4. Los titulares de la cuenta seleccionada en la grilla 2.

Genere además de las operaciones mencionadas, la posibilidad de realizar altas, bajas y modificaciones sobre los titulares y las cuentas.