**Programación Lógica**

Segundo Parcial

**Ejercicio Nro.1**

Realizar una aplicación consola que permita el ingreso por pantalla de nombre, apellido, dni, fecha de nacimiento de un alumno, luego desarrolle una función que calcule la edad del alumno y si es mayor de 18 devuelva al main una cadena con la expresión “El alumno tiene aprobado el ingreso”, “El alumno no tiene aprobado el ingreso”.

Consideraciones:

* La aplicación nunca debe lanzar ninguna excepción.
* Las fechas se deben mostrar como fecha corta
* Las leyendas por pantalla deben ser claras y deben guiar al usuario sobre los datos que se deben ingresar.
* Utilizar validaciones y try/catch.
* No es necesario que la aplicación itere.

**Ejercicio Nro.2**

Desarrolle una aplicación consola para un sistema de ventas de productos de una cadena de electrodomésticos, la misma contara con la siguiente estructura:

* Producto🡪 Lista tipo string
* Descripcion🡪 Lista tipo string.
* Cantidad🡪 Lista tipo entero.
* Precio🡪Lista tipo doble.

La aplicación deberá tener como mínimo las siguientes funciones:

* Crear nuevos productos: se deberán generar los nuevos productos a vender. (las listas también deben tener datos pre-cargados para no demorar la ejecución/defensa del parcial. (Los nuevos productos se deberán ingresar solo con un usuario y pass aprobados).
* Incrementar los precios: Se deberá incrementar los precios a partir de una variable por producto ingresado por pantalla.
* Realizar ventas: se deberá realizar la venta correspondiente, si hay stock suficiente, además del producto a vender se deberá solicitar también la cantidad y se deberá mostrar por pantalla la fecha de venta, el producto, la cantidad, el precio unitario y el precio final. Todas las ventas se deberán volcar a un archivo físico en el disco rígido (C:\>ventas)
* Actualizar stock: se deberá ingresar nuevo stock por producto ya existente.
* Mostrar Productos existentes.

Consideraciones:

* Las leyendas por pantalla deben ser claras y deben guiar al usuario sobre los datos que se deben ingresar.
* Se deberá revisar el código para evitar utilizar variables innecesarias.
* El código deberá estar bien tabulado y con comentarios.
* La aplicación no deberá lanzar excepción en ningún momento. Se deberán utilizar validaciones y Try Catch.
* Se deberá utilizar funciones para la codificación más eficiente.
* Debe contar con un menú e itere x cantidad de veces.