



Tecnicatura Universitaria
en Programación

PROGRAMACIÓN I

Unidad Temática N°4:
Diseño de Formularios con C#

Guía de Estudio
1° Año – 1° Cuatrimestre



Índice

DISEÑO DE FORMULARIOS CON C#	2
Problema 4.1:	2
Problema 4.2:	2
Problema 4.3:	2
Problema 4.4:	2
Problema 4.5:	3
Problema 4.6:	3
Problema 4.7:	4
Problema 4.8:	4
Problema 4.9:	4
Problema 4.10:	5
Problema 4.11:	5
Problema 4.12:	5
EJERCICIO MODELO ABM	6
Problema 4.13:	6

DISEÑO DE FORMULARIOS CON C#

Basado en el Paradigma Orientado a Objetos, diseñe una solución escrita en lenguaje de programación C# para cada uno de los problemas que se plantean a continuación.

En cada caso defina:

- Clases que incluyan atributos, propiedades, constructor sin parámetros, constructor con parámetros, métodos de control y método ToString().
- Relaciones que implementen el modelo planteado en cada caso.
- Desarrollar una Interfaz Gráfica de Usuario (mediante WinForms) que permita implementar el modelo de objetos diseñado. Tener en cuenta que al momento de crear los objetos se deberán hacer las validaciones de datos correspondientes y utilizar los componentes visuales adecuados para cada caso.

Problema 4.1:

Rectángulo. Calcular el perímetro y la superficie de un rectángulo de base y alto ingresados como datos por el usuario.

Problema 4.2:

Silo. El dueño de un campo almacena sus cosechas de granos en silos cilíndricos. Le pide que desarrolle un programa que le permita conocer el volumen máximo de un silo.

Problema 4.3:

Punto. nos han solicitado desarrollar una clase que permita modelar un punto en el plano. Cada punto se representa mediante un par ordenado de coordenadas cartesianas y es posible conocer:

- cuál es su distancia al eje de coordenadas
- Adicionalmente se necesita mostrar un punto como una cadena con formato "(x;y)".

Problema 4.4:

Persona. Se nos pide modelar una clase Persona con las siguientes condiciones:

- Sus atributos son: nombre, edad, sexo (H hombre, M mujer), peso y altura.

- Los métodos que se implementarán son:
 - `calcularIMC()`: calculará si la persona está en su peso ideal (peso en $\text{kg}/(\text{altura}^2 \text{ en m})$), si esta fórmula devuelve un valor menor que 20, la función devuelve un -1, si devuelve un número entre 20 y 25 (incluidos), significa que está por debajo de su peso ideal la función devuelve un 0 y si devuelve un valor mayor que 25 significa que tiene sobrepeso, la función devuelve un 1. Se recomienda usar constantes para devolver estos valores.
 - `esMayorDeEdad()`: indica si es mayor de edad(21 años), devuelve un booleano.

Problema 4.5:

Password. Modelar una clase llamada Password que tenga los atributos longitud y valor. Por defecto, la longitud será de 8.

Se pide:

- `esFuerte()`: devuelve un booleano si es fuerte o no, para que sea fuerte debe tener mas de 2 mayúsculas, mas de 1 minúscula y mas de 5 números.
- `generarPassword()`: genera la contraseña del objeto con la longitud que tenga.

Problema 4.6

Auto. Diseñar una clase que permita modelar el funcionamiento básico de un automóvil, tal como se indica a continuación:

- **conducir**: esta opción debe recibir la cantidad de kilómetros a recorrer e informar si pueden ser recorridos y en tal caso descontar el combustible utilizado del tanque (se conoce que, por especificaciones técnicas del vehículo, con un litro recorre 11 km). Caso contrario informar que no hay combustible suficiente para el recorrido indicado.
- **cargar combustible**: está opción debe reponer combustible en el tanque tantos litros como se reciban como argumento pero teniendo en cuenta que el tanque tiene una capacidad de 49 lts más una reserva de 5 lts., en el caso de que los litros informados superen la capacidad del tanque, devolver la cantidad de lts de combustible derramados.
- **chequear nivel de combustible**: informar si el tanque está al 25%, 50%, 75% o lleno y dar una alerta en el caso que el tanque se encuentre en

reserva, indicando que hay que reponer combustible de inmediato. El mensaje debe devolver una cadena con 25% ... o el alerta.

Ayuda: para una resolución más sencilla, pensar en la existencia de dos clases: Automóvil y Tanque.

Problema 4.7:

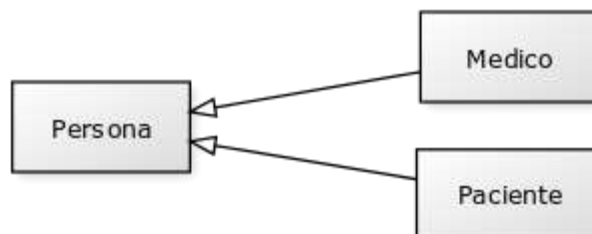
Alumnos. Dados el nombre, legajo, DNI, teléfono y tres notas de alumnos, calcular su promedio y mostrar todos sus datos.



Además, indicar la cantidad y porcentaje de alumnos promocionados (promedio \geq 8), regulares (promedio \geq 6) y libres (promedio $<$ 6) y el promedio general de todos los alumnos registrados.

Problema 4.8:

Clínica. Desarrollar un programa que permita registrar el nombre, DNI, teléfono y sexo de médicos y pacientes de una clínica. Para médicos, cargar también su especialidad (1-Traumatología, 2-Pediatría, 3-Cardiología) y costo de la consulta y para pacientes la obra social (1-Apross, 2-Met, 3-Osde, 4-Otras) y datos de su historia clínica.



Además, indicar la cantidad de médicos y pacientes registrados, el costo promedio de las consultas médicas y el porcentaje de pacientes que tienen Osde como obra Social.

Problema 4.9:

Banco. Dados el nombre, DNI, sexo, código y límite de crédito de un cliente y el número, tipo (1-caja de ahorro, 2-cuenta corriente) y saldo de una cuenta, desarrollar un programa que permita registrar las cuentas de los clientes de un banco de acuerdo al siguiente diagrama.



Además, calcular la cantidad de cajas de ahorro y de cuentas corriente, saldo promedio para caja de ahorro, para cuenta corriente y general, porcentaje de clientes femeninos y el cliente con mayor límite de crédito.

Problema 4.10:

Inmobiliaria. Dados los datos de un propietario: tipo y número de DNI, nombre y sexo; y de un inmueble: metros, costo por metro y tipo de inmueble (1-Casa, 2-Dpto, 3-Lote), desarrollar un programa que permita registrar los propietarios con sus inmuebles de acuerdo al siguiente diagrama.



Además, determinar la cantidad de casas, departamentos y lotes, su valuación promedio, el valor promedio de todos los inmuebles, el porcentaje de casas, la cantidad de mujeres con departamento, el propietario con inmueble más valioso y el propietario con lote más pequeño.

NOTA: tener en cuenta que la valuación de los inmuebles resulta de la cantidad de metros por el costo por metro en cada caso.

Problema 4.11:

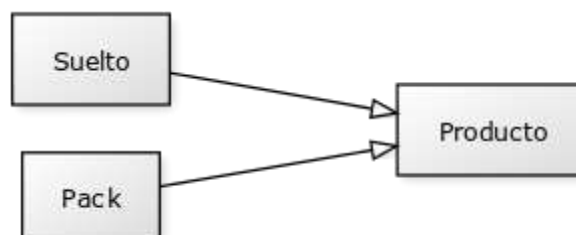
Veterinaria. Dados los datos del dueño (nombre, sexo, código) y de su mascota (edad, nombre y tipo: 1-perro, 2-gato, 3-araña, 4-iguana), desarrollar un programa que permita registrar los dueños con sus mascotas de acuerdo al siguiente diagrama.



Además, determinar la cantidad, porcentaje y edad promedio de cada tipo de mascota, la cantidad de mujeres con iguanas, la cantidad de hombres con gatos, el dueño con perro más viejo y la mascota más joven.

Problema 4.12:

Productos. Dados el código, marca, precio unitario y tipo (suelto o pack) de productos, calcular la cantidad de productos sueltos, la cantidad de productos en



packs, el precio promedio de productos sueltos, el porcentaje de packs y el código y marca del pack más costoso. Los productos sueltos tienen medida y los packs cantidad. Implementar las clases Producto, Pack y Suelto y el método abstracto `getPrecio()`. En cada caso, el método devuelve $(cantidad * precio)$ para packs y $(medida * precio)$ para productos sueltos. Con ello crear un arreglo de productos. Mostrar dicho arreglo en una lista y mostrar el precio total.

EJERCICIO MODELO ABM

Problema 4.13:

Personas. Desarrollar una GUI utilizando WinForms de C# .NET Framework de acuerdo al siguiente diseño.

The image shows a Windows Forms application window titled "A B M Personas". The window has a standard Windows title bar with minimize, maximize, and close buttons. The main area of the window is divided into two sections. On the left, there is a form with several input fields and controls: "Apellido" with the text "Perez", "Nombres" with "Juan", "Tipo Documento" with a dropdown menu showing "DNI", "Documento" with "123456", "Estado Civil" with a dropdown menu showing "SOLTERO", "Sexo" with two radio buttons labeled "Femenino" and "Masculino" (the "Masculino" button is selected), and "Fallecido" with an unchecked checkbox. On the right side of the window, there is a list box containing the text "Perez, Juan". At the bottom of the window, there is a row of six buttons: "Nuevo", "Editar", "Borrar", "Grabar", "Cancelar", and "Salir". The "Salir" button is highlighted with a blue border.

Imagen 1: Elaboración propia

Luego implementar la programación necesaria para gestionar la funcionalidad que se sugiere a continuación:

- En todo momento la **navegación** de los componentes dentro del formulario debe tener un sentido lógico de arriba hacia abajo y de izquierda a derecha.
- Al **iniciar el formulario** deben estar desactivados los componentes de carga de datos y los botones Grabar y Cancelar. Los botones Nuevo, Editar, Borrar, Salir junto con la Lista deben estar activados y el foco en la Lista.

- Al presionar el botón **Nuevo** se deben limpiar los componentes de carga de datos y activar junto a los botones Grabar y Cancelar. Los botones Nuevo, Editar, Borrar, Salir junto con la Lista se deben desactivar. El foco debe quedar en el primer componente a cargar.
- Al presionar el botón **Editar** se deben activar los componentes de carga de datos junto a los botones Grabar y Cancelar. Los botones Nuevo, Editar, Borrar, Salir junto con la Lista se deben desactivar. El foco debe quedar en el primer componente a cargar.
- Al presionar el botón **Borrar** se debe mostrar un cuadro de diálogo configurado de modo tal que permita al usuario confirmar o cancelar el borrado.
- Al presionar el botón **Grabar** se debe mostrar un cuadro de diálogo configurado de modo tal que notifique al usuario de la grabación exitosa y luego debe quedar el formulario en su estado inicial. En este punto se debe crear un objeto Persona y agregarlo a un arreglo llamado personas actualizando la Lista correspondiente.
- Al presionar el botón **Cancelar** se deben limpiar los componentes de carga de datos y luego debe quedar el formulario en su estado inicial.
- Al presionar el botón **Salir** se debe mostrar un cuadro de diálogo configurado de modo tal que permita al usuario confirmar, o no, cerrar el formulario.



Atribución-No Comercial-Sin Derivadas

Se permite descargar esta obra y compartirla, siempre y cuando no sea modificado y/o alterado su contenido, ni se comercialice. Referenciarlo de la siguiente manera:
Universidad Tecnológica Nacional Facultad Regional Córdoba (S/D). Material para la Tecnicatura Universitaria en Programación, modalidad virtual, Córdoba, Argentina.