



Tecnicatura Universitaria
en Programación

PROGRAMACIÓN II

Unidad Temática N°1:

Programación Orientada a Objetos
Avanzada

Guía de Estudio
1° Año – 2° Cuatrimestre



Índice

INTERFACES	2
Problema 1.1:	2
Problema 1.2:	2
Problema 1.3:	2
PATRONES Y CAPAS	3
Problemas modelo	3
Problema 1.4:	3
Problema 1.5:	4
Problema 1.6:	4
Problema 1.7:	5
Problema 1.8:	5
Problema 1.9	6

INTERFACES

Problema 1.1:

Pilas. Escribe una interfaz, llamada `ICollection` que declare los siguientes métodos:

`estaVacia()`: devuelve true si la colección está vacía y false en caso contrario.

`extraer()`: devuelve y elimina el primer elemento de la colección.

`primero()`: devuelve el primer elemento de la colección.

`añadir()`: añade un objeto por el extremo que corresponda, y devuelve true si se ha añadido y false en caso contrario.

A continuación, escribe una clase `Pila`, que implemente esta interfaz, utilizando para ello un array de `Object` y un contador de objetos.

Problema 1.2:

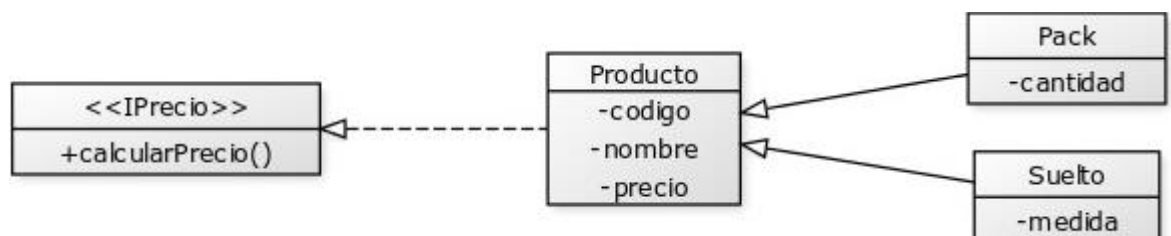
Colas. Desarrollar una clase `Cola` que implemente la interfaz definida en el problema anterior pero esta vez utilizando un objeto `List`. Tener en cuenta que una Cola es una estructura FIFO (Primero en entrar primero en salir).

Problema 1.3:

Productos. En un comercio se tienen dos tipos de productos: sueltos y en pack.

- De cada producto se conocen: código, nombre y precio.
- Los productos sueltos tienen medida
- Los packs cantidad por pack.

Implementar el siguiente modelo de clases:



Considerar que el precio de los packs se calcula como $\text{cantidad} * \text{precio}$, mientras que el de los sueltos es $\text{medida} * \text{precio}$.

Crear un arreglo de productos. Mostrar dicho arreglo junto con el precio total de los productos registrados.

PATRONES Y CAPAS

Para cada uno de los problemas indicados en **Problemas modelo** generar las soluciones C# incluyendo:

- Capas de presentación, servicios y acceso a datos (Se sugiere implementar proyectos independientes en la misma solución)
- Implementar los objetos Repository necesarios para modelar el acceso a datos utilizando los patrones de diseño estudiados en la unidad.
- Considerar el uso de interfaces y enumeraciones en las soluciones propuestas.

Problemas modelo

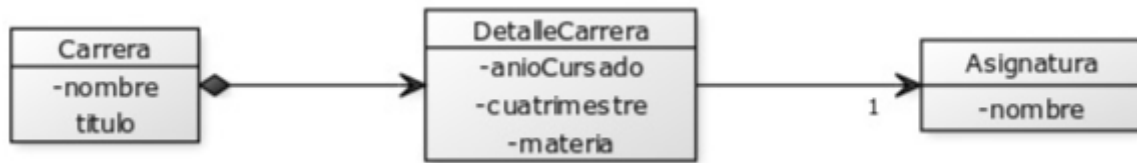
Basado en el Paradigma Orientado a Objetos, diseñe una solución escrita en lenguaje de programación C# para cada uno de los problemas que se plantean a continuación.

En cada caso defina:

- Generar un Modelo E-R que permita crear la base de datos
- 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.
- Deberá diseñar una capa de negocio que exponga los servicios necesarios para:
 - ✓ Recuperar todas las entidades
 - ✓ Filtrar y recuperar entidades según criterios de búsqueda
 - ✓ Crear, modificar y registrar baja lógica de las entidades

Problema 1.4:

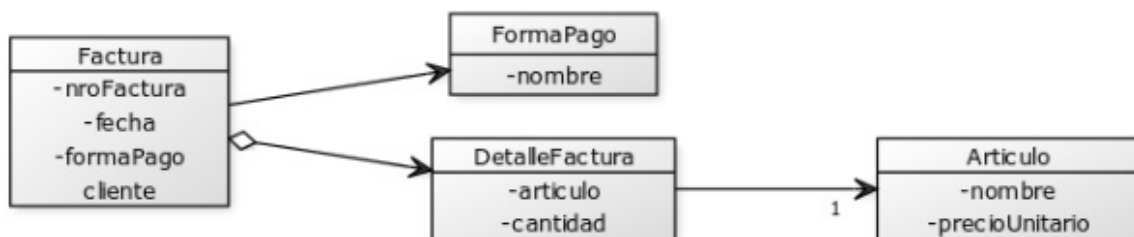
Carrera. Desarrollar un programa que permita registrar los planes de estudio asociados a las carreras dictadas por una universidad. El modelo de objetos sugerido se muestra a continuación:



- Solo es necesario gestionar carreras con sus detalles de asignaturas (planes), las asignaturas deberían estar disponibles al lanzar la aplicación.
- Deberá controlar que una misma materia no puede agregarse más de una vez como detalle de carrera.

Problema 1.5:

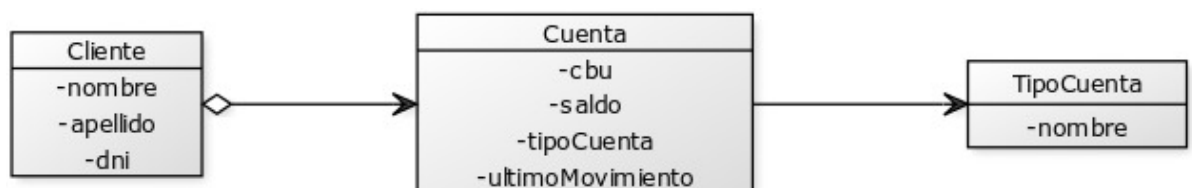
Facturación. Desarrollar un programa que permita registrar facturas de venta para un pequeño comercio del interior de la provincia. El modelo sugerido es:



- Deberá controlar que, si un mismo artículo se agrega más de una vez, se deberá incrementar las cantidades del mismo detalle.

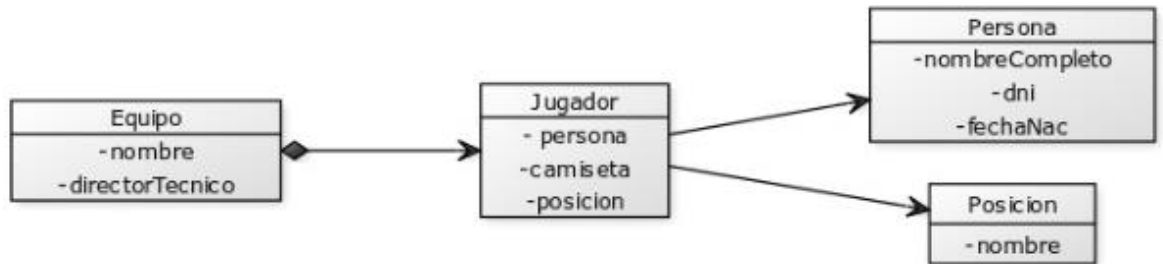
Problema 1.6:

Banco. Desarrollar un programa que permita registrar las cuentas de un cliente del banco. Se deberán registrar tanto los datos de un cliente (que puede existir previamente) junto con sus cuentas según el siguiente modelo:



Problema 1.7:

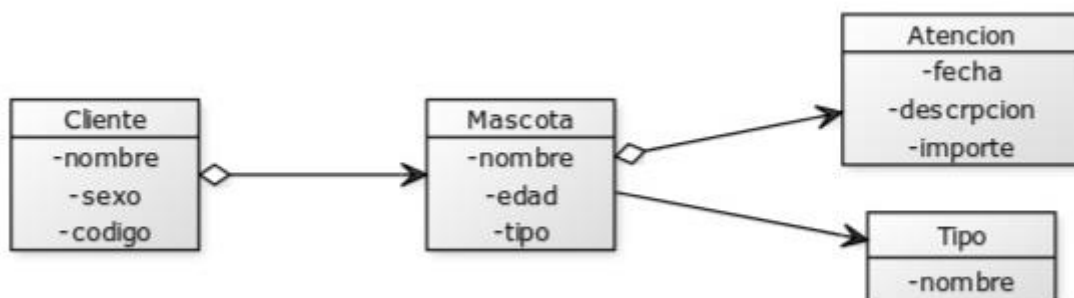
Equipo. Desarrolla una aplicación que permita cargar los datos de un equipo de futbol de una liga cordobesa. Se deberá registrar además los datos de las fechas con los compromisos pendientes.



- Las personas deberían estar disponibles al lanzar la aplicación.
- Una misma persona puede jugar en más de un equipo.
- Dentro de un equipo una persona no puede cubrir dos posiciones diferentes.

Problema 1.8:

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 las atenciones de una mascota indicando para cada una la descripción de los tratamientos y/o vacunas aplicadas.



- Deberá registrar los datos de las mascotas junto con sus atenciones
- Los clientes están disponibles al comienzo de la aplicación

Problema 1.9

Transporte de cargas. Una empresa de transporte de cargas necesita un software que la ayude a organizarse con la carga de los camiones que maneja. La empresa puede distintos tipos de cargas: 1-Packing | 2-Cajas | 3-Bidones.

Cada camión puede llevar hasta una carga máxima medida en kg. Además, cada camión puede: estar disponible para la carga (en cuyo caso ya puede tener cosas cargadas), estar en reparación, o estar de viaje.

Algunos requerimientos:

- Subir cargas al camión, donde la carga puede ser un packing, una caja suelta, o un bidón. Considerar que no se puede saturar un camión con más peso de lo que su carga máxima permite.
- Bajar una carga del camión, siempre que el camión se encuentre disponible con cargas y que a su vez la carga se encuentre presente dentro de él.
- Conocer el total de cargas de un camión en todo momento y su peso.
- Saber si un camión está listo para partir, que es: si está disponible para la carga, y el peso total de lo que tiene cargado es de al menos 75% de su carga máxima.



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.