

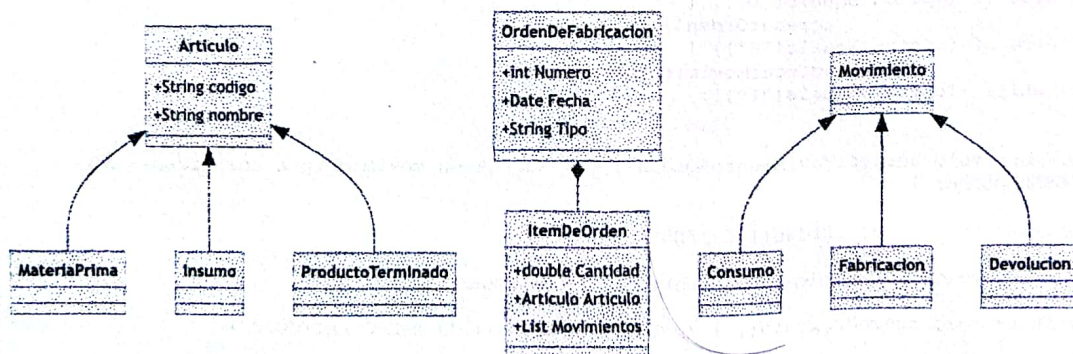
**UNIVERSIDAD DE BUENOS AIRES – FACULTAD DE CIENCIAS ECONOMICAS
TEORIA DE LOS LENGUAJES Y LOS ALGORITMOS – 2022 PRIMER CUATRIMESTRE
SEGUNDO PARCIAL**

Nombre y Apellido: Nro. de registro:

Debemos construir la funcionalidad de control de fabricación y consumos de una empresa comercial e industrial dedicada a la fabricación de artículos textiles. El sistema tiene la siguiente funcionalidad:

1. Presenta el menú con las siguientes opciones:
 - a. Listar artículos
 - b. Nuevo artículo
 - c. Listar órdenes de fabricación pendientes
 - d. Nueva orden de fabricación
 - e. Ingresar movimientos sobre la orden
2. **[NO DESARROLLAR]** Si el usuario ingresa a la opción A, el sistema lista los artículos ingresados, indicando si son productos terminados, materias primas o insumos
3. **[NO DESARROLLAR]** Si el usuario ingresa a la opción B, el sistema permite cargar un nuevo artículo, indicando si es materia prima, insumo o producto terminado
4. Si el usuario ingresa a la opción C, el sistema lista las órdenes de fabricación pendientes. Se considera a una orden de fabricación como pendiente si la cantidad fabricada de algún producto a fabricar es menor a la cantidad a fabricar
5. Si el usuario ingresa a la opción D, el sistema permite cargar una nueva orden de fabricación. El proceso es el siguiente:
 - a. El usuario ingresa los datos de cabecera: número, fecha (la del día), un texto indicando si es una orden sobre pedido o para stock
 - b. Si la orden existe (el número de orden no está cargado) el sistema informa de esta situación y vuelve al menú principal
 - c. Si la orden no existe, el usuario ingresa un artículo a fabricar o a consumir, indicando la cantidad. El sistema valida que el artículo exista. Si es correcto, agrega el artículo
 - d. El sistema pregunta si desea agregar más artículos
 - e. Si el usuario dice que quiere ingresar, retorna al punto 5.b
 - f. Si el usuario dice que no quiere ingresar, agrega la orden, que debe tener al menos un artículo a fabricar y un artículo a consumir y vuelve al punto 1
6. **[NO DESARROLLAR]** Si el usuario ingresa a la opción E:
 - a. El sistema muestra el listado de órdenes pendientes y solicita el ingreso de una orden
 - b. Si la orden existe y está pendiente, solicita al usuario el ítem (artículo) a mover. Si el artículo es incorrecto, solicita el dato nuevamente
 - c. Si es un artículo a fabricar, se pueden ingresar movimientos de fabricación (signo entrada) o de devolución de la fabricación (signo salida). El sistema solicita la clase de movimiento y luego la cantidad. La cantidad total de movimientos del artículo no puede superar a la cantidad planificada para fabricar
 - d. Si es un artículo que se consume (materia prima o insumo) se pueden ingresar movimientos de consumo (signo salida). La cantidad en movimientos de consumo para un ítem no puede superar a la cantidad a consumir de dicho ítem
 - e. El sistema intenta ingresar el movimiento, considerando las restricciones de 6.c y 6.d, y vuelve al menú principal

El siguiente es el detalle de datos por entidad:



IMPORTANTE

- Todos los ingresos de datos deberán ser validados
- El no uso de alguna funcionalidad ya desarrollada será considerada un error
- Se valorará la correcta definición de clases e interfaces necesarias

SE SOLICITA:

1. Desarrollar los puntos 4 y 5 de la funcionalidad, agregando los métodos necesarios a OrdenDeFabricacion e ItemDeOrden
2. **No desarrollar** la jerarquía de clases de Movimiento. Sólo clarificar si Movimiento es clase o interfaz, y cuáles son sus métodos públicos

```
using System;
using System.Collections.Generic;

namespace _2022_1C_2P_Resolver
{
    class Program
    {
        static void Main(string[] args)
        {
            AppFabricacion app = new AppFabricacion();
            app.ejecutar();
        }
        internal class AppFabricacion
        {
            Validador validador;
            private List<Articulo> articulos;
            private List<OrdenDeFabricacion> ordenes;

            public AppFabricacion() {
                validador = new Validador();
                articulos = new List<Articulo>();
                ordenes = new List<OrdenDeFabricacion>();
            }
            internal void ejecutar() {
                string opcion = "";
                do
                {
                    opcion = validador.pedirStringEnColeccion("Ingrese la opcion de menu deseada: \n" +
                        "A - Listar articulos\nB - Nuevo articulo\nC - Listar ordenes de\n" +
                        "fabricacion pendientes\n" +
                        "D - Nueva orden de fabricacion\nE - Ingresar movimientos sobre la\n" +
                        "orden\nF - Salir", new List<String> { "A", "B", "C", "D", "E" }).ToString();
                    if (opcion.Equals("A")) {
                        listarArticulos();
                    } else if (opcion.Equals("B")) {
                        nuevoArticulo();
                    } else if (opcion.Equals("C")) {
                        listarOrdenesPendientes();
                    } else if (opcion.Equals("D")) {
                        agregarOrden();
                    } else if (opcion.Equals("E")) {
                        agregarMovimientoAOrden();
                    }
                } while (!opcion.Equals("F"));
            }

            private void agregarMovimientoAOrden() { //Agrega un movimiento a una orden - NO
                DESARROLLAR; }

            private void agregarOrden() { //DESARROLLAR }

            private void listarOrdenesPendientes() { //DESARROLLAR }

            private void nuevoArticulo() { //PERMITE INGRESAR UN NUEVO ARTICULO }

            private void listarArticulos() { //LISTA LOS ARTICULOS CARGADOS }
        }
        internal class OrdenDeFabricacion
```