Proyecto: PedidosApp

# 1. Introducción

PedidosApp es una aplicación de escritorio desarrollada como parte de un proyecto de programación en C#. El objetivo principal de esta aplicación es permitir registrar pedidos y asignarles un método de entrega, utilizando el patrón de diseño Factory. El proyecto fue desarrollado usando Windows Forms y está orientado a reforzar conceptos de programación orientada a objetos como interfaces, clases abstractas y patrones de diseño.

# 2. Objetivos

- Aplicar el patrón de diseño Factory Method.  
- Desarrollar una interfaz gráfica con Windows Forms.  
- Gestionar diferentes tipos de entregas de manera polimórfica.  
- Mostrar los pedidos registrados en una lista o historial dentro del formulario.

# 3. Estructura del Proyecto

El proyecto se compone de las siguientes partes principales:  
a) Interfaz IMetodoEntrega: Define el método EntregarPedido().  
b) Clases concretas: Camion, Moto, Dron. Cada una implementa IMetodoEntrega.  
c) Clase MetodoEntregaFactory: Se encarga de crear instancias del método de entrega según el tipo seleccionado.  
d) Formulario principal (Form1): Contiene controles como TextBox, ComboBox y ListBox para ingresar pedidos, seleccionar entrega y visualizar resultados.

# 4. Funcionamiento General

Cuando el usuario ingresa un pedido y selecciona un método de entrega desde el ComboBox, se crea una instancia correspondiente mediante la factory. Luego se ejecuta el método EntregarPedido() y el mensaje correspondiente se muestra en un ListBox junto con el nombre del pedido. Esto permite cambiar dinámicamente el comportamiento del sistema sin modificar el código principal del formulario.

# 5. Patrón de Diseño: Factory Method

Se utiliza el patrón Factory Method para instanciar clases concretas que implementan la interfaz IMetodoEntrega. Esto proporciona flexibilidad y escalabilidad, ya que se pueden agregar nuevos métodos de entrega sin modificar la lógica del formulario.

# 6. Conclusión

Este proyecto fue una excelente oportunidad para aplicar conceptos de POO y patrones de diseño en un entorno gráfico. Permitió entender cómo separar la lógica de negocio de la interfaz y cómo implementar el patrón Factory para generar objetos dinámicamente según una entrada del usuario.