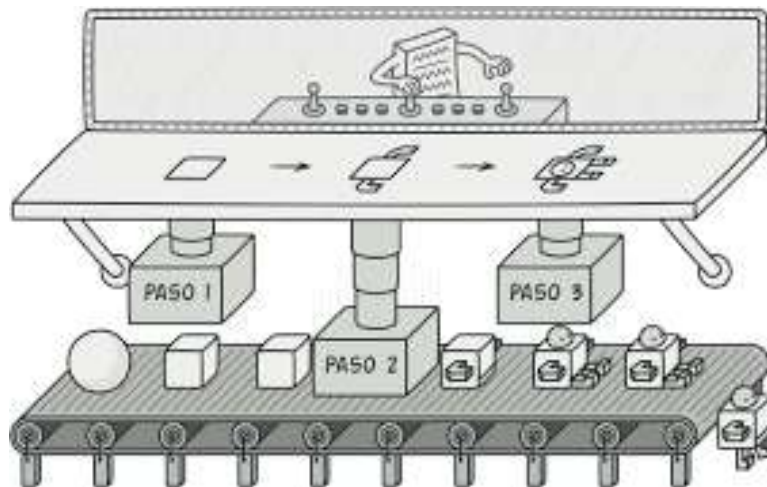


# PATRÓN BUILDER



MODELADO Y DISEÑO

VIZCARRA MORENO JESUS DANIEL

GUSTAVO IBARRA

CESUN UNIVERSIDAD

## PASO 1

```
Patron Adapter.cs  Patron builder.cs X
index.html > Patron builder.cs > Program > Main
1  using System;
2  using System.Collections.Generic;
3
4  // Paso 1: Definición de la clase Pizza
5  public class Pizza
6  {
7      public string TipoMasa { get; set; }
8      public string Salsa { get; set; }
9      public string Queso { get; set; }
10     public List<string> Ingredientes { get; set; }
11
12     public Pizza()
13     {
14         Ingredientes = new List<string>();
15     }
16 }
17
```

## PASO 2

```
// Paso 2: Definición de la interfaz IPizzaBuilder
2 references
public interface IPizzaBuilder
{
    void BuildTipoMasa();
    void BuildSalsa();
    void BuildQueso();
    void BuildIngredientes();
    Pizza GetPizza();
}
```

### PASO 3

```
28 // Paso 3: Implementación del PizzaBuilder
29 1 reference
30 public class PizzaBuilder : IPizzaBuilder
31 {
32     7 references
33     private Pizza _pizza = new Pizza();
34
35     2 references
36     public void BuildTipoMasa()
37     {
38         _pizza.TipoMasa = "Delgada";
39     }
40
41     2 references
42     public void BuildSalsa()
43     {
44         _pizza.Salsa = "Tomate";
45     }
46
47     2 references
48     public void BuildQueso()
49     {
50         _pizza.Queso = "Mozzarella";
51     }
52
53     2 references
54     public void BuildIngredientes()
55     {
56         _pizza.Ingredientes.Add("Jamón");
57         _pizza.Ingredientes.Add("Champiñones");
58         _pizza.Ingredientes.Add("Aceitunas");
59     }
60 }
```

### PASO 4 Y 5

```
0 references
class Program
{
    // Paso 4: Uso del PizzaBuilder para construir una pizza
    0 references
    static void Main(string[] args)
    {
        IPizzaBuilder pizzaBuilder = new PizzaBuilder();
        pizzaBuilder.BuildTipoMasa();
        pizzaBuilder.BuildSalsa();
        pizzaBuilder.BuildQueso();
        pizzaBuilder.BuildIngredientes();

        Pizza pizza = pizzaBuilder.GetPizza();

        // Paso 5: Resultado
        Console.WriteLine("Pizza construida:");
        Console.WriteLine($"Tipo de masa: {pizza.TipoMasa}");
        Console.WriteLine($"Salsa: {pizza.Salsa}");
        Console.WriteLine($"Queso: {pizza.Queso}");
        Console.WriteLine("Ingredientes:");
        foreach (var ingrediente in pizza.Ingredientes)
        {
            Console.WriteLine(ingrediente);
        }
    }
}
```

## Conclusión:

Este código implementa el patrón de diseño Builder para crear objetos de tipo Pizza de manera más flexible y controlada. El patrón Builder permite construir objetos paso a paso, separando el proceso de construcción del objeto de su representación final. Esto facilita la creación de objetos con diferentes configuraciones sin tener que modificar el código que los utiliza. En este caso, el PizzaBuilder permite construir pizzas con diferentes tipos de masa, salsas, quesos e ingredientes de manera fácil y clara.