**ABSTRACT FACTORY**

La principal diferencia es que en el **abstract Factory** la creación de objetos es a través de la composición mientras que en el **Factory Method** lo era a través de la herencia.

Asi mismo en este patrón **Abstract Factory** lo que se crea es una familia de objetos

Vamos a realizar este patrón de diseño con el mismo ejercicio; lo que se va a implementar es que en cada local de pizzería puedan realizar sus propios ingredientes para eso se creara una familia de ingredientes:

Lima

Arequipa

Salsa de Tomate

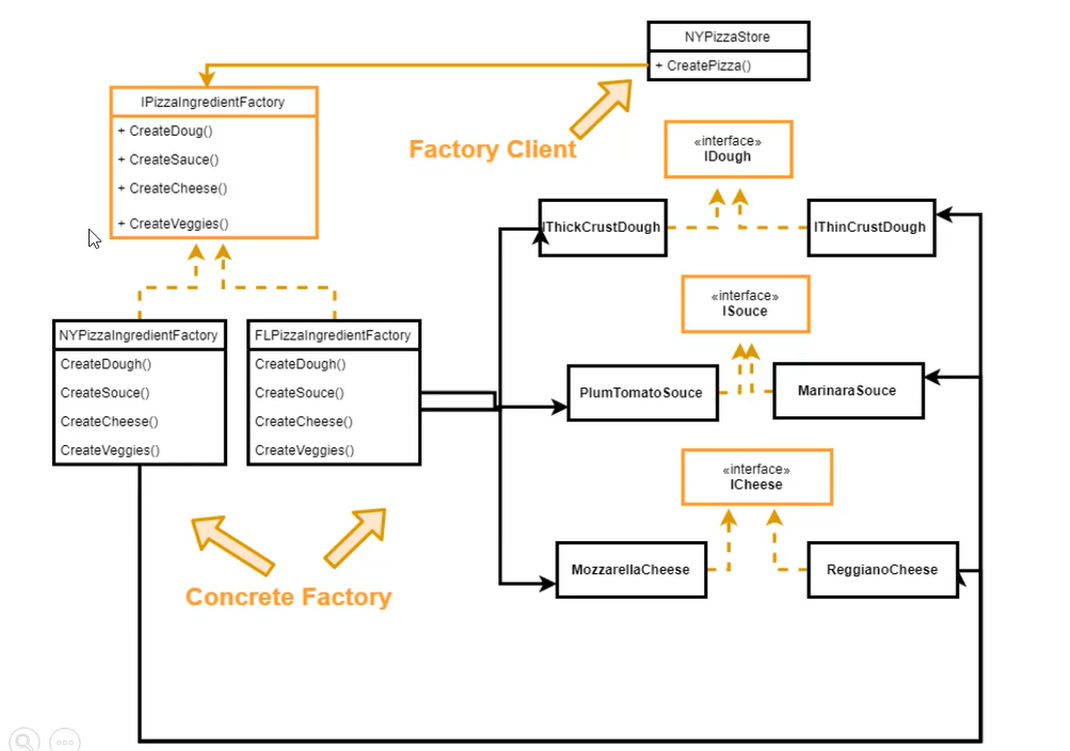
Masa Gruesa

Salsa Marinada

Masa Delgada

Queso Mozarela

Queso Parmesano



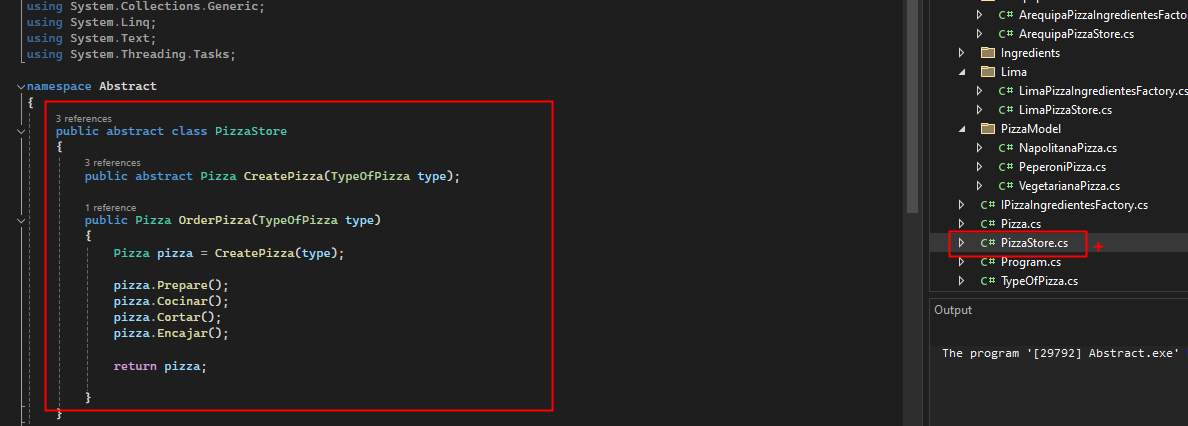
**Ejercicio1:**

Utilizamos el mismo ejercicio anterior, a manera de repaso lo volvemos a crear, pero ahora organizado en carpetas.

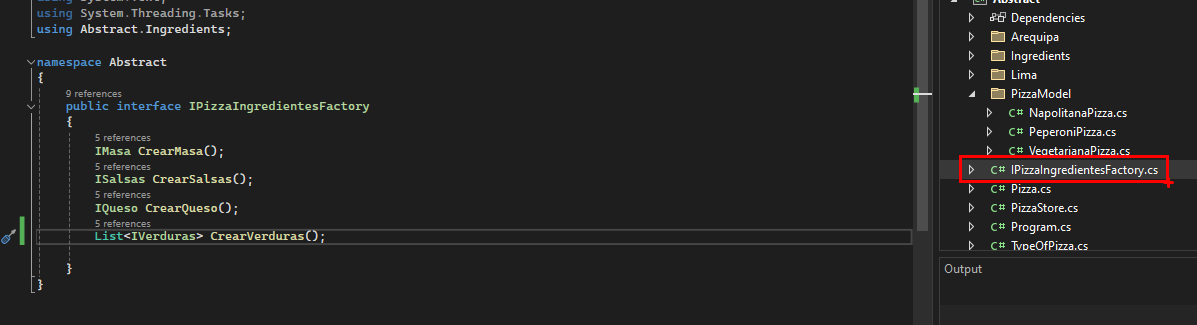
**REFACTORIZACION**

***Abrimos nuestro archivo Exercise1AfterA:***

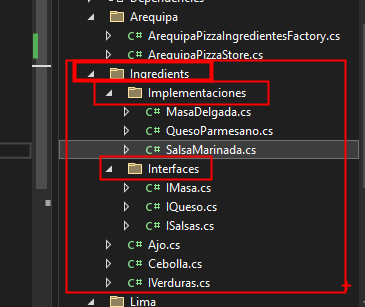
Implementamos la clase **PizzaStore**

****

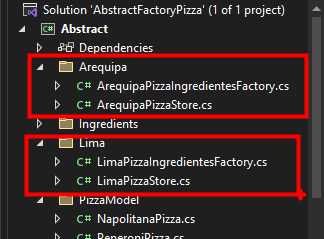
Seguidamente se crea **IPizzaIngredientesFactory**

****

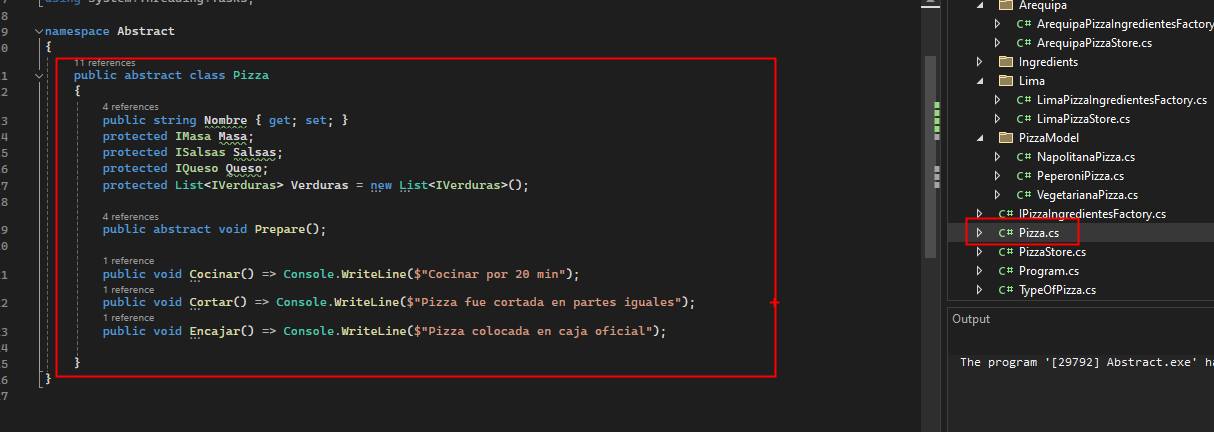
Seguidamente implementamos las interfaces y las clases de la carpeta **Ingredientes:**

****

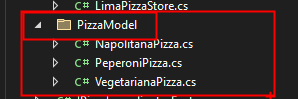
Ahora implementamos las dos clases de la carpeta Lima y Arequipa:



Seguidamente implementamos la clase **Pizza:**

****

Ahora implementamos las clases que se encuentran dentro de la carpeta **PizzaModel**



Por ultimo implementamos el **Program.cs**