Interfaz

El interfaz indica que se va hacer; además declara métodos sin describir la lógica del comportamiento de los métodos; ya que habrá una clase que implemente la Interfaz usando todos los métodos de la interface esto simula a la herencia de clase, pero la ventaja de implementar es que se puede implementar varias interfaces en una sola clase ya que en herencia solo se heredar una clase.

- En java se declara la interfaz con la palabra **interface;** ejemplo;

  public **interface** NombreDAO{

  }

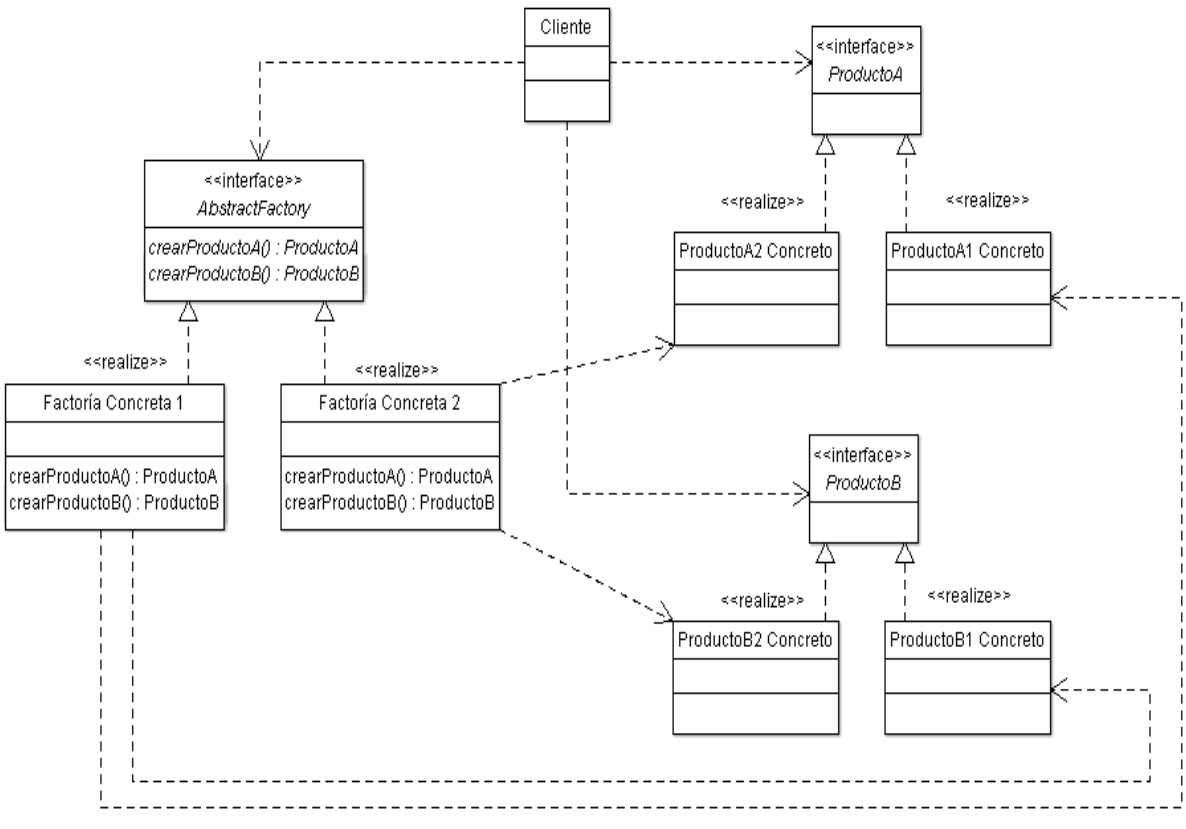
- En java se realiza la implementación de una interfaz con la palabra **implements,** ejemplo:
  public class NombreDaoImpl **implements** NombreDAO{
  }

Patrón de Diseño

Los patrones de diseño tienen como objetivo solucionar problemas que se dan en el desarrollo de software.

- **Abstract Factory:**
  Permite crear mediante una interfaz, conjuntos o familias de objetos (productos) que dependen mutuamente y todo esto sin especificar sus clases concretas; además el objeto fabrica proporciona servicios de creación para toda la familia

La estructura típica del patrón **Abstract Factory** :



Cliente: es la clase que quiere obtener una instancia de los productos; Producto A, Producto B; para crear uno de los objetos que declaro la factoría.

AbstractFactory: su objetivo es proporcionar métodos para la obtención de cada objeto que cree el método “crearProduvtoA()” y “crearProductoB()”.

Factorias Concretas: son las diefertes familias de los productos.

Producto abstracto: define las interfaces para la familia de productos genéricos.

Producto concreto: implementa la interfaz de los diferentes productos.