Sistema de ventas - Visitor

Se requiere diseñar un sistema de ventas para “Contoso” una tienda minorista que maneje dos tipos de entidades: **productos** (por ejemplo, laptops, calzado o comestibles) y **paquetes promocionales** (ofertas que agrupan varios productos). Se debe poder añadir operaciones como calcular el **total de ventas**, aplicar los **impuestos** correspondientes según la región o generar **etiquetas promocionales**, **sin** modificar las clases de producto ni de paquete cada vez que surja una nueva necesidad. Para ello, se deberá utilizar el **Patrón Visitor**: para hacer que los productos y los paquetes “acepten” un visitante, y crea visitantes concretos (por ejemplo, CalculadorDeVentasVisitor, AplicadorDeImpuestosVisitor, GeneradorDeEtiquetasVisitor) que recorran la estructura y realicen cada operación.

Solucion :

BE :

En la BE declare las siguientes entidades :

1. IVisitado :

* Declaro el método Aceptar(IVistante visitante)
* Cada entidad que puede “ser visitada” implementa esta interfaz
* Es el primer paso del doble despacho: el objeto acepta al visitante y le pasa el control a este

1. IVisitante:

* Declara un método Visitar(..) por cada tipo concreto de elemento (Producto, PaquetePromocional)
* Cada operación que querramos añadir (calculo de ventas, aplicación de impuestos, generación de etiquetas etc), se implementarán comoi una clase que implementa IVisitante

1. Producto

* Implementa IVisitado
* En su método Aceptar(..) invoca visitante.Visitar(this) delegando al visitante la lógica concreta para manejar el producto

1. PaquetePromocional

* Implementa IVisitado
* Su Aceptar(...) invoca visitante.Visitar(this), delegando al visitante la lógica concreta para manejar un paquete.
* Además, cuando más adelante recorramos sus Elementos (una lista de IVisitado), volveremos a invocar Aceptar en cada uno, propagando así el visitante por toda la jerarquía.