# Documentación del Código.

Paquete com.irudigital.concurrency.activity.sn Thread

1. **Clase Cajera Producto Main  
   Descripción:** Esta es la clase  
   **Atributos:**
   * **productos** : Lista de productos que están disponibles para la compra.
   * **cliente1, cliente2, cliente3** : Clientes que compran productos.
   * **cajera 1, cajera 2, cajeras** : Cajeras que procesan las compras de los clientes.
2. **Métodos:**
   * main(String[] args): Método principal que inicia el proceso de compra
   * set Productos(List<Producto> productos): Método que agrega tres productos(Zapatos, Blue Jeans y Camisas).
3. **Clase Cajera Producto  
   Descripción:** Esta clase representa a una cajera que procesa las compras de los clientes de forma secuencial. Para cada cliente, la cajera procesa cada uno de sus productos e imprime información sobre.  
   **Atributos:**
   * **Nombre Cajera** : Nombre de la cajera que procesa la compra.
4. **Métodos:**
   * procesar Compra(Cliente Producto cliente, long timeStamp):Este
   * esperar Segundos(): Método que simula una pausa de 1 segundo entre el procesamiento de cada producto.
5. Clase Producto  
   Descripción:  
    La clase Producto Representa un producto con información  
   Atributos:
   * **Nombre** : Nombre del producto.
   * **precio** : Precio del producto.
   * **cantidad** : Cantidad del producto disponible para la venta.
6. **Métodos:**
   * getNombre():Retorna el nombre del producto
   * Nombre(String nombre): Establezca el nombre del producto.
   * get Precio(): Retorna el precio del producto.
   * set Precio(float precio): Establece el precio del producto
   * get Cantidad(): Retorna la cantidad del producto disponible.
   * set Cantidad(int cantidad):Establecer el canto
7. **Clase Cliente Producto  
   Descripción:** Esta clase representa a un cliente que tiene una lista de productos que desea comprar. Contiene el nombre  
   **Atributos:**
   * **Nombre** del cliente.
   * **productos** : Lista de productos que
8. Métodos:
   * get Productos(): Retorna la lista de productos que el cliente comprará.
   * getNombre(): Retorna el nombre del cliente.

Paquetecom.iudigital.concurrency.activity.whitThread

1. Clase Cajera Product  
   Descripción:  
    Esta clase implementa la interfaz Runnable, lo que permite q  
   Atributos:
   * **Nombre Cajera** : Nombre de la cajera que procesa la compra.
   * **cliente** : Cliente
   * **inicial Time** : El tiempo inicial (marca de tiempo) que marca el inicio del proceso de compra.
2. **Métodos:**
   * run(): Implementa el método run()de la interfaz Runnable. Este método inicia el procesamiento de la compra del cliente en el hilo correspondiente.
   * procesar Compra(Cliente Producto cliente, long timeStamp): Este método simula el procesamiento de la compra, más
   * esperar Segundos(): Este método simula una espera de 1 segundo entre el procesamiento de cada producto.
3. **Clase Cajera Product Main  
   Descripción:** La clase principal que simula el procesamiento concurrente de las compras de varios clientes. Para cada cliente, se crea un hilo.Cajera Product procesa la compra de manera concurrente.  
   **Atributos:**
   * **productos** : Lista de productos disponibles para la compra.
   * cliente1, cliente2, cliente3 : Instancias de Cliente Producto que representan a los clientes
   * **hilo cajera, cajera, cajera 3** : Hilos que ejecutan las cajeras para procesar las compras de los clientes.
4. **Métodos:**
   * main(String[] args): Método principal donde se inicializan los productos,
   * set Productos(List<Producto> productos):Método de productos. Los productos son similares a los que se usan en la clase Cajera Producto Main.

# Resumen general del proyecto

Este proyecto simula un sistema de compras en una tienda. Hay dos enfoques de procesamiento:

1. Enfoque secuencial ( Cajera Producto) :  
    Las compras de los clientes son procesadas por las cajas de forma secuencial, una a la vez.
2. Enfoque concurrente ( Cajera Product) :  
    Las compras de los clientes son procesadas de forma concurrente, utilizando hilos independientes

El código cubre el procesamiento de productos, cálculo de costos, y muestra el tiempo que se tarda en procesar la compra de cada cliente. Utiliza la programación orientada a objetos y la concurrencia para gestionar el proceso de compra de manera eficiente.