**Propuesta de Arquitectura**

**Nombre: Paul Plaza**

**1. Modularidad y Arquitectura Hexagonal**

* La **Arquitectura Hexagonal** es ideal para facilitar la extensibilidad y el mantenimiento del sistema. Dividiremos nuestra aplicación en tres capas principales:
  + **Dominio**: Contendrá la lógica de negocio central (entidades como Usuario, Producto, Orden).
  + **Aplicación**: Representa la lógica de los casos de uso, es decir, cómo los usuarios interactúan con la lógica de negocio.
  + **Adaptadores**: Puertos de entrada y salida. En esta capa ubicaremos los controladores REST (entrada) y servicios de infraestructura como bases de datos, conectores de mensajería con Kafka, entre otros (salida).

**2. Contenedorización y Orquestación**

* **Contenedorización con Docker**:
  + Cada servicio (Usuarios, Ordenes, Productos) estará contenedorizado con Docker.
  + Se generan imágenes de cada microservicio que luego se despliegan y gestionan mediante Kubernetes.
* **Orquestación con Kubernetes**:
  + Se utilizará **Kubernetes** para orquestar y escalar los microservicios.
  + Cada microservicio se despliega en un **Pod**, y estos se agrupan en **Deployments**.
  + Se utiliza un **Load Balancer** que distribuye las solicitudes hacia los diferentes Pods.

**3. Componentes de Integración y Comunicación**

* **Apache Kafka**: Se utilizará como **bus de mensajería** entre los microservicios para eventos que requieran sincronización, por ejemplo:
  + Cuando se actualiza un producto, se emite un evento de "actualización de inventario" para asegurar que todos los servicios tengan la misma información.
  + Kafka facilita la **comunicación asíncrona**, asegurando una comunicación eficiente y desacoplada.
* **MongoDB**: Como base de datos principal, se utilizará **MongoDB** por su flexibilidad en manejar documentos de productos y ordenes con atributos variados.
  + Utilizaremos **MongoDB Atlas** como solución gestionada en la nube para mayor disponibilidad y facilidad de escalado.