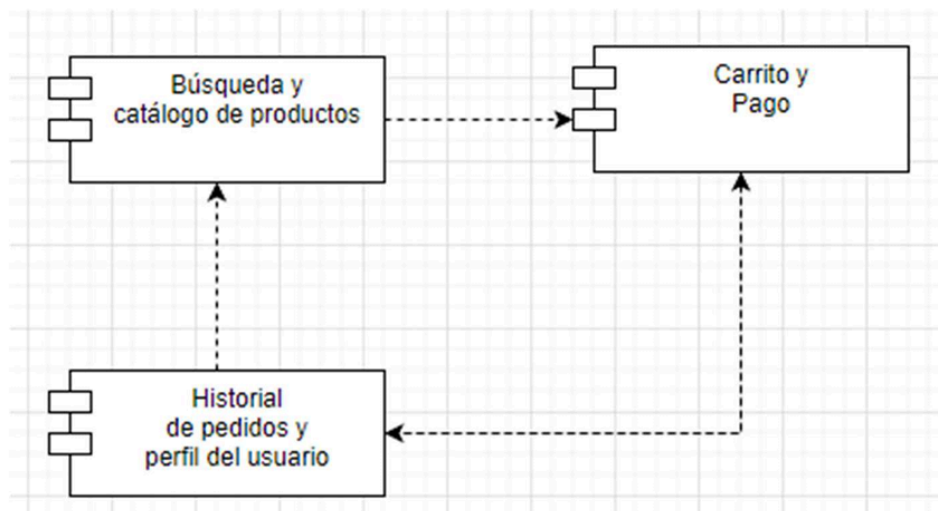


# Diseño y UML

## Ejercicio 1

Supongamos que estamos desarrollando una aplicación de comercio electrónico. La aplicación permite a los usuarios buscar productos, añadirlos al carrito, realizar el pago y ver el historial de pedidos. Desde el área de arquitectura nos proponen esta distribución de módulos:

1. Módulo de búsqueda y catálogo de productos
2. Módulo de carrito y pago
3. Módulo de historial de pedidos y perfil del usuario

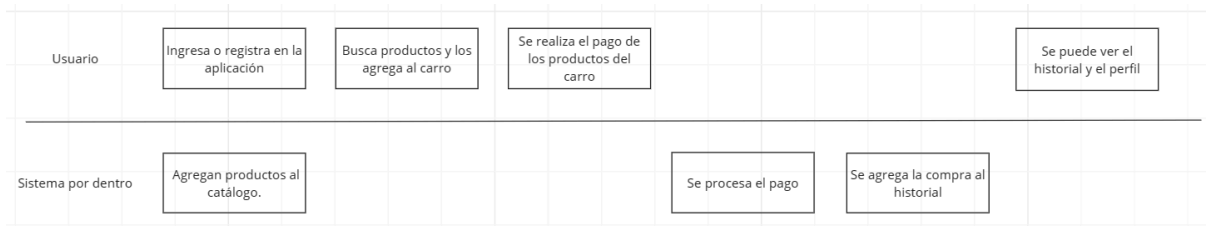


## Parte 1

a. Discuta y defina las responsabilidades y que hace cada módulo.

1. Muestra los productos, filtra los productos
2. Gestionar el historial de pedidos. Gestionar el perfil del usuario.
3. Almacenar productos para la compra. Gestionar los pagos de los productos.

b. Haga un boceto (Journey Map) de cómo sería el proceso de compra utilizando este modelo.



## Parte 2

Analicen si este diseño inicial cumple con los principios de bajo acoplamiento y alta cohesión. Si no es el caso, deben proponer una solución que cumpla con estos principios.

Para el análisis, considere estas preguntas:

- ¿Los módulos tienen responsabilidades claramente definidas y relacionadas entre sí?
- ¿Existen dependencias innecesarias entre los módulos?
- ¿Se puede modificar un módulo sin afectar significativamente a los otros?

Para que se tenga alta cohesión y bajo acoplamiento se deberían separar algunas funcionalidades en diferentes módulos.

Por ejemplo el carrito y pago no debería ser un único módulo sino que deberían estar separados, de esta manera cada módulo por separado tendría responsabilidades bien definidas y no tengan muchas (lo ideal sería una sola responsabilidad). Esto puede dificultar la modificación o expansión de parte del sistema además tiene múltiples responsabilidades (almacenar productos y gestionar pagos), lo que disminuye la cohesión dentro del módulo.

Otro punto raro del esquema es que tiene un ciclo, lo que en ciertos casos podría ser un problema.

