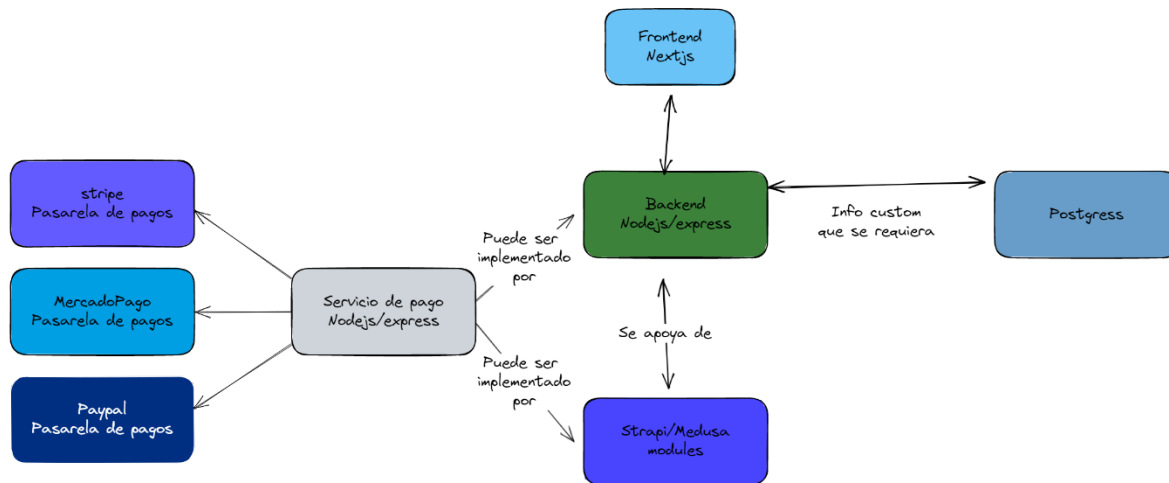


Aplicación de comercio electrónico



Como primer factor a tener en cuenta es conocer un estimado de las solicitudes o del número de clientes. Ya que este comercio puede ser uno ya existente, el cual se desea renovar o re construir. Esto puede pasar con algún cliente que tenga su ecommerce hecho en Shopify, Mercadolibre o incluso Wordpress/Woocommerce. y ya tiene un tráfico base de entrada.

Si es el caso y es una tienda con un tráfico considerable (10k+ solicitudes por minuto). Un buen punto de partida sería estructurar toda la app en base a microservicios, de lo contrario, empezar con un monolito la forma más sencilla, rápida, fiable y sobre todo, económica de desarrollar y de mantener. El desarrollo debería estar guiado en lo posible por DDD (Domain Driven design) y una arquitectura hexagonal, esto ayudará a implementar más fácilmente las diferentes etapas de testing y al momento de que la aplicación crezca se pueden ir migrando a microservicios los componentes que vayan siendo necesarios

Para no reinventar la rueda, tomaríamos como base librerías opensource que nos brinden herramientas para implementar fácilmente la lógica de negocio requerida. Entre estas está Strapi, y Medusa.

Strapi (<https://strapi.io/>) es un Headless CMS completamente personalizable y que provee integraciones con múltiples servicios a través de plugins. Entre sus ventajas está su madurez y su comunidad detrás.

Medusa, es una librería que se conforma por módulos que se van incluyendo. Ya tiene integrado todo un sistema de inventario, sistema de pagos, sistemas de gestión de vida de un pedido entre otros. está pensado para crearse directamente con microservicios y crear tu propia lógica de negocio en base a los módulos ya existentes. Medusa es altamente personalizable y cuenta con integraciones a través de plugins. Como desventaja, Medusa apenas está en su versión 1.10 lo cual puede no ser lo suficientemente maduro para implementar en una tienda.

Como lenguaje base se usaría TypeScript, Jest como librería de Testing, de base de datos se puede usar una BD relacional como postgres o una no relacional como MongoDB, todo según lo que se pueda llegar a necesitar que no proporcione Strapi o Medusa. Se puede llegar a hacer uso de Redis

para cachear las solicitudes y mejorar el performance.

El sistema de carpetas sería el siguiente:

- src
 - subdominios
 - app
 - domain
 - infraestructura
 - utils
 - utils
- test (La carpeta test replicaría la misma estructura de src)

Si hablamos de Patrones de Diseño, se estarían usando el patrón Adaptador, Inyección de dependencias, Factories, Repository, entre otros que se puedan llegar a requerir para implementar o personalizar la lógica de negocio.