Planteamiento

“Emazon” es una tienda virtual que nos permite llegar a millones de personas, ofreciendo artículos a nivel general de mucho valor para los clientes. Están interesados en contratarte para desarrollar su sitio web, el cual servirá como plataforma de interacción entre los clientes y los artículos de Emazon.

El propósito principal del sitio web es facilitar la compra de los articulos por parte de los clientes, y la administración de la entrada de articulos por parte de las personas auxiliares de bodega. los articulos pueden tener categorias y tienen marcas, eso nos ayuda a identificar y filtrar de manera más fácil.

Un articulo que tiene, un nombre, una cantidad de existencias, tiene un precio, tiene una marca a la que pertenece, y tiene muchas categorias a la cual puede pertenecer. La merca esta compuesta por nombre y descripcion, y las categorias, tambien tienen un nombre y descripcion cada una.

Una vez que tengamos el stock, que es todo lo relacionado con articulo y existencias del articulo, podemos seguir con el suministro que es como aumentar la cantidad de existencias de los articulos por medio del rol auxiliar de bodega, una vez tengamos todo lo de suministros, seguimos con el carrito, que es un cliente va a poder ver los articulos y seleccionar para agregar al carrito los que mas le gusten, este carrito servira para que el usuario pueda hacer una compra de varios articulos a la vez, teniendo en cuenta que cada vez que compra, el carrito debera limpiarse.

Una de las funcionalidades centrales, es la compra, y es poder facilitarle al cliente que haga la compra por medio de la plataforma, se puede comprar desde el carrito o directamente desde un articulo, la compra debe disminuir la cantidad de existencias de los articulos.

El administrador debera tener informes que le permitan tomar decisiones sobre la operaciones del negocio, cual es el articulo favorito de los clientes y cual es el articulo mas vendido, esta informacion solo podra ser obtenida por el administrador.

Reglas:

|  | Front | Back |
| --- | --- | --- |
| * Cada microservicio es un repositorio aparte. |  | x |
| * Cada microservicio debe persistir únicamente su base de datos. |  | x |
| * Cada regla de negocio debe tener sus respectivos tests unitarios. | x | x |
| * Cada microservicio debe tener su respectiva documentación OpenApi. |  | x |
| * Cada microservicio debe estar basado en arquitectura hexagonal. |  | x |
| * Cada hu debe estar implementada en su rama única (usar de gitflow). | x | x |
| * Cada hu debe estar basada en atomic design | x |  |
| * Cada hu debe estar pensado en mobile first | x |  |
| * Cada hu debe ser validada por sonarLint | x | x |
| * Las funcionalidades que se generen deben tener buenas practicas (manejo de constantes, correcto nombramiento de las variables, metodos y clases, el metodo no puede ser muy grande) | x | x |

consideraciones a tener en cuenta en front:

* version de angular 14
* no utilizar librerias de estilos tipo bootstrap o angular material
* utilizar libreria de iconos [link](https://fontawesome.com/)
* pruebas unitarias con jest
* utilizar modules
* utilizar routing
* utilizar guards
* utilizar interceptors