

ASIGNACION: DISEÑO DE MODELO DE DATOS PARA UN NUEVO PROYECTO

1. Elección del Tema:

Una aplicación de supermercado, donde el usuario pueda crear su carrito en línea y pueda hacer sus encargos

2. Identificación de Entidades y Atributos:

❖ **User:**

User_id: ID para cada usuario de acuerdo su registro

Name: Nombre del Usuario

Last name: apellido del usuario

birthday: para llevar un control de la edad del cliente

direction: lugar en el cual vive

email: Correo electrónico

type: saber si es un user o un admin

❖ **Product:**

Product_id: ID del producto

Name: nombre del producto

price: precio unitario

Category_id: categoría a la que pertenece

❖ **Category:**

Category_id: ID asignado según la categoría que se usara

Name: Nombre de la categoría

❖ **Inventory:**

inventory_id: el producto que tenemos en existencia

product_id: producto del que estamos llevando la existencia

Stock: Cuanto producto existe en el almacén

Date_in: nos dice si el producto entro

Date_out: nos dice si el producto salió

❖ **Catálogo:**

catalog_id: ID asignado para el producto

product_id: referencia a los productos

Name: Nombre del producto

availability: productos con o sin stock dentro de la tienda

❖ **order:**

order_id: ID que será única para poder reconocer el pedido

User_id: persona que hizo la orden

inventory_id: lista de productos con sus datos para la venta

Total: total a pagar

3. Análisis de Cardinalidades y Relaciones:

User puede hacer muchos **order**, pero todos estos solo le van a pertenecer a un user **(1: N)**

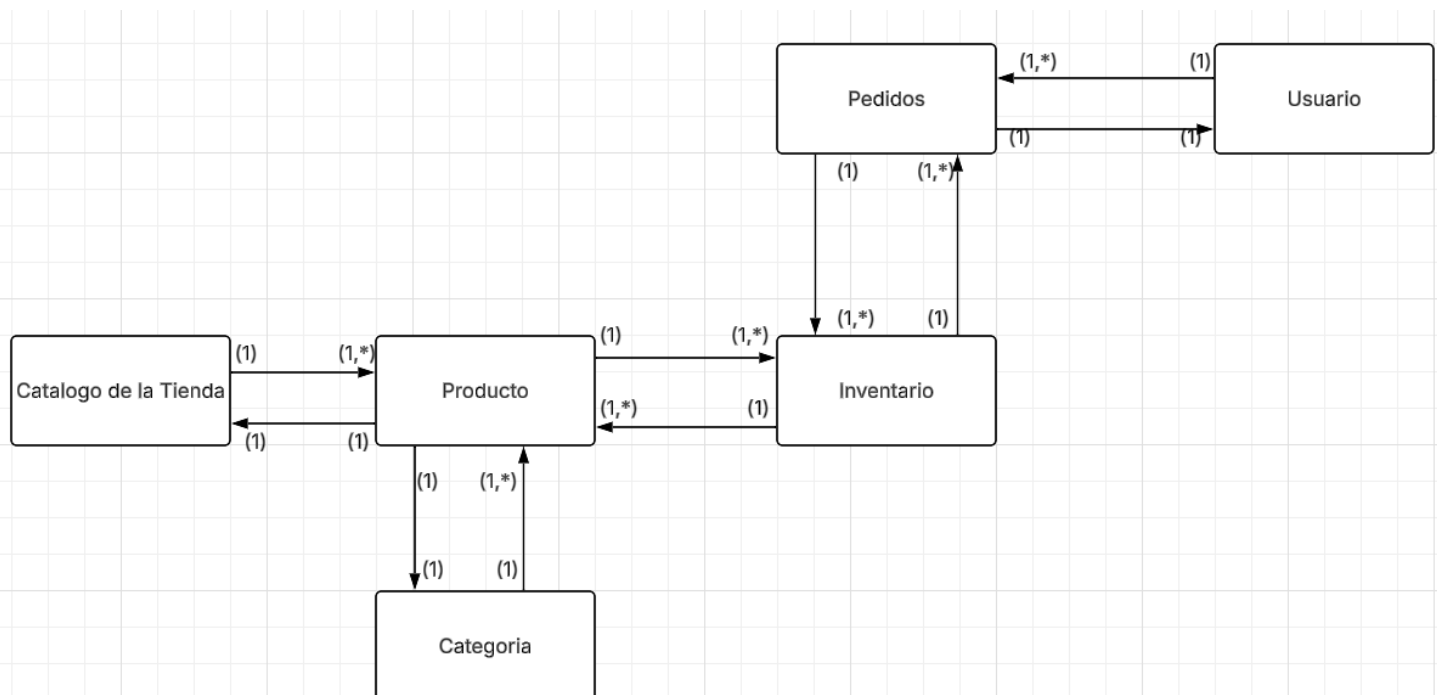
Category puede tener muchos **Product**, pero un producto solo puede pertenecer a una categoría **(1: N)**

Product puede tener muchas existencias en el **Inventory** y en el inventario puede haber mucha existencia de ese **Product** **(N: M)**

Product tiene un solo registro en el **Catálogo** y el catalogo, tiene el registro de muchos productos **(1: N)**

Los productos del **Inventory** pueden incluirse muchas veces en un **Order** y en el order puede existir varios productos de este inventario **(N: M)**

4. Definición Detallada de Colecciones:



- Traducción

user		
nombre	tipo	observacion
_id	objectid	
name	str	
last name	str	
birthday	date	
direction	str	
email	str	

category		
nombre	tipo	observacion
_id	objectid	
name	str	

inventory		
nombre	tipo	observacion
_id	objectid	
product_id	objectid	
stock	int	
date_in	int	
date_out	int	

catalog		
nombre	tipo	observacion
_id	objectid	
product_id	objectid	
name	str	
availability	bol	

product		
nombre	tipo	observacion
_id	objectid	
name	str	
price	float	
category_id	objectid	

order		
nombre	tipo	observacion
_id	objectid	
user_id	objectid	
inventory_id	objectid	
total	float	

order_detail		
nombre	tipo	observacion
_id	objectid	
inventory_id	objectid	
order_id	objectid	
quantity	int	
status	str	recibido, en proceso, en camino, entregado

inventory_detail		
nombre	tipo	observacion
_id	objectid	
inventory_id	objectid	
product_id	objectid	
quantity	int	

user_type		
nombre	tipo	observacion
_id	objectid	
descripton	str	admin o common user

5. Deficion de Endpoints RESTful

User

URI	HTTPMethod	Header	Payload(Body)	Response
/user	GET	H-admin H-user		{ } → admin: lista de los usuarios { } → user: informacion del user
/user/{id}	GET	H-admin		{ } → informacion de el usuario al que le pertenece el ID
/user	POST		{"name":"Lionel Andres","last name":"Messi ", "Birtday": "1987-04-24", "direction":"Buenos Aires", "email" : "lm10@example.com"}	{ } → documtno creado
/user/{id}	DELETE	H-admin		{ } -> el admin elimino/desactivo el usuario
/user	DELETE	H-user		{ } → el usuario elimino/desactivo su cuenta
/user/{id}/order	GET	H-admin		{ } → el admin puede ver las ordenes que tiene en su cuenta el usuario
/user/{id}	PUT	H-admin H-user	{"name":"Lionel Andres","last name":"Messi ", "Birtday": "1987-04-24", "direction":"Av. Campeones del mundo,Buenos Aires,Arg", "email" : "lm10@example.com"}	

Catalog

URI	HTTPMethod	Header	Payload(Body)	Response
/catalog	POST	H-admin	{"Name":"Coca Cola 3 litros", "Price":"72.00","Category":"Refrescos"}	{ } → el documento esta creado
/catalog	GET	H-admin H-user		{ } → admin y user: mirar la lista de productos en el catalogo
/catalog/{id}	DELETE	H-admin		{ } → el admin elimino un objeto del catalogo
/catalog/{id}	PUT	H-admin	{"Name":"Coca Cola 3 litros", "Price":"75.00","Category":"Refrescos"}	{ } → el documento fue actualizado
/Catalog/{id}/availability	PUT	H-admin	{"estado":"no disponible"}	{ } → el documento fue actualizado

ORDER

URI	HTTPMethod	Header	Payload(Body)	Response
/order	GET	H-admin H-user		<code>{}</code> → admin: todos los pedidos realizados por los users autenticados <code>{}</code> → user: los pedidos realizados por el usuario
/order/{id}	GET	H-admin		<code>{}</code> → el administrador ve las ordenes realizadas por un usuario con el id ingresada
/order/{id}	POST	H-user	<code>{}</code>	<code>{}</code> → crea el documento en order detail, en el campus status

Inventory

URI	HTTPMethod	Header	Payload(Body)	Response
/inventory	GET	H-admin		<code>{}</code> → el admin revisa la existencia que hay en el inventario
/inventory/{id}	GET	H-admin		<code>{}</code> → el admin revisa la existencia del producto con esta id
/inventory/	POST	H-admin	<code>{"product": "leyde leche", "stock": "25", "input date": "2025-06-29", "outpout date": "2025-07-20"}</code>	<code>{}</code> → el documento esta creado

Link: <https://github.com/jxsus-1/miAPI>

