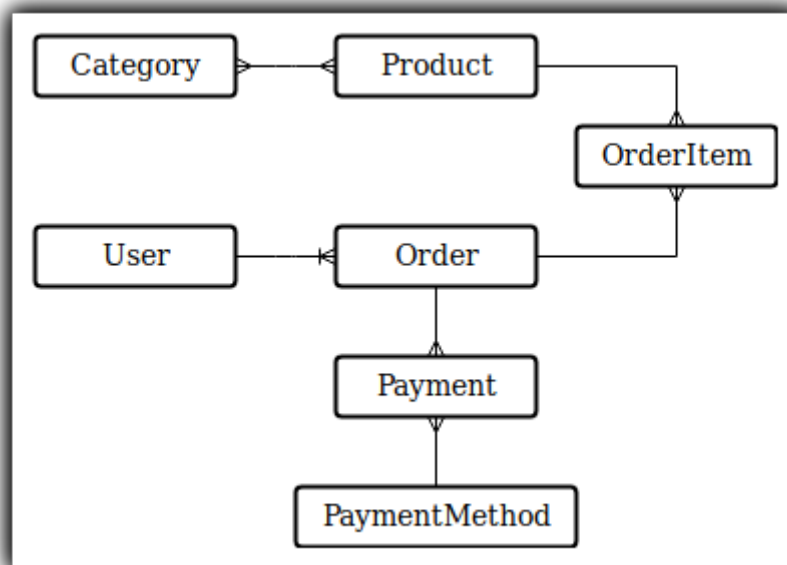


Desafío - Sistema e-Commerce

- Para realizar este desafío debes haber estudiado previamente todo el material disponibilizado correspondiente a la unidad.
- Una vez terminado el desafío, comprime la carpeta y sube el `.zip`

Descripción

Tenemos un sistema de e-commerce donde el diagrama de modelos se representa en la siguiente imagen:



Uno de nuestros clientes nos ha solicitado una serie de mejoras, ya que nuestro actual sistema no se ajusta a sus necesidades.

Nos pide que extendamos algunas funcionalidades de tal manera que los productos puedan tener muchas variaciones, por ejemplo, una polera puede tener talla X, M y L a la vez que cada una de ellas puede tener diferentes colores. El cliente es claro en indicarnos que cada una de estas variaciones maneja un stock independiente al resto.

Otro de los problemas que presenta, es que sus productos no pueden clasificarse sólo con un nivel de categorías, ya que la categoría "Electrodomésticos" puede tener otras contenidas como línea blanca o refrigeradores. El cliente nos indica que no hay límite de cuantas categorías anidadas pueden haber, sólo nos asegura que una categoría sólo pertenece a una, osea no puede haber una categoría con dos padres.

Finalmente, nos indica que deben haber dos funcionalidades específicas: en el catálogo sólo se muestra un producto de cada tipo, esto es, si existe un modelo de zapatilla roja y otra negra, en el catálogo debe aparecer sólo una de ellas, esto implica que debe haber una página de descripción del producto en donde ahí podamos ver y elegir los colores y la talla, si una variación no tiene stock, no se puede comprar. Esto es importante, ya que si todas las variaciones no tienen stock, no debe aparecer en el catálogo principal.

Por problemas de implementación, el modelo *OrderItem* no podemos modificarlo y tendrá que guardar los mismos datos: `order_id`, `product_id`, `quantity` y `price`.

Por fortuna, el cliente ha encargado el desarrollo del front-end a otra empresa, por lo que solo debemos asegurar calidad en la estructuración de los datos.

Instrucciones

1. Hacer un diagrama de modelos para cubrir las necesidades del cliente.
2. Crear la o las relaciones y modelos para manejar los productos y sus variaciones.
3. Crear la o las relaciones y modelos para manejar las tallas y colores de las variaciones.
4. Implementar la solución para manejar distintos niveles de categorías y asegurarse que no tenga dos padres. (hint: usar test unitario).
5. Generar un método o scope en las categorías, de tal forma que entregue una lista de cada categoría padre y otro de sus hijos, y los hijos de sus hijos, en todos los niveles.
6. Según su diseño, explicar al cliente cómo implementar la lista de productos del catálogo. (de un ejemplo en código). Si un modelo necesita código, debes entregarlo al cliente como parte de la implementación.
7. Implementar o explicar las modificaciones (si las hay) al modelo *OrderItem* para que siga funcionando sin que afecte el resto del sistema.
8. Nuestro cliente, a último minuto nos solicita que nuestro sistema soporte cupones de dos tipos: uno para distribuir en redes sociales (1 cupón lo pueden ocupar muchas personas) y otro para clientes específicos (1 cupón solo lo puede ocupar un cliente específico). Los cupones pueden descontar un porcentaje de la compra o un monto específico. En el caso de un monto específico, si el cupón es mayor que la compra, el cupón no puede usarse en otra. Por temas de tiempo, sólo podrás entregarle una prueba de conceptos en la que debes incluir el modelado de los datos extendiendo el resultado de la pregunta 1 y además algunas implementaciones de código con los conceptos mas importantes.