

UNIVERSIDAD DE ANTIOQUIA
PROGRAMA INGENIERÍA DE SISTEMAS
TÉCNICA DE PROGRAMACIÓN Y LABORATORIO G1. (P.O.O)
TALLER SOBRE PROGRAMACIÓN FUNCIONAL
PROF: ING GEOVANNY MENDOZA GONZALEZ

El ejercicio se basa en un modelo de datos: **Cliente**, **Pedido** y **Producto**. Como se puede observar en el diagrama de relación de entidades que se muestra en la figura # 1, los clientes pueden realizar varios pedidos, por lo que se trata de una relación de uno a muchos, mientras que la relación entre los productos y los pedidos es de muchos a muchos.

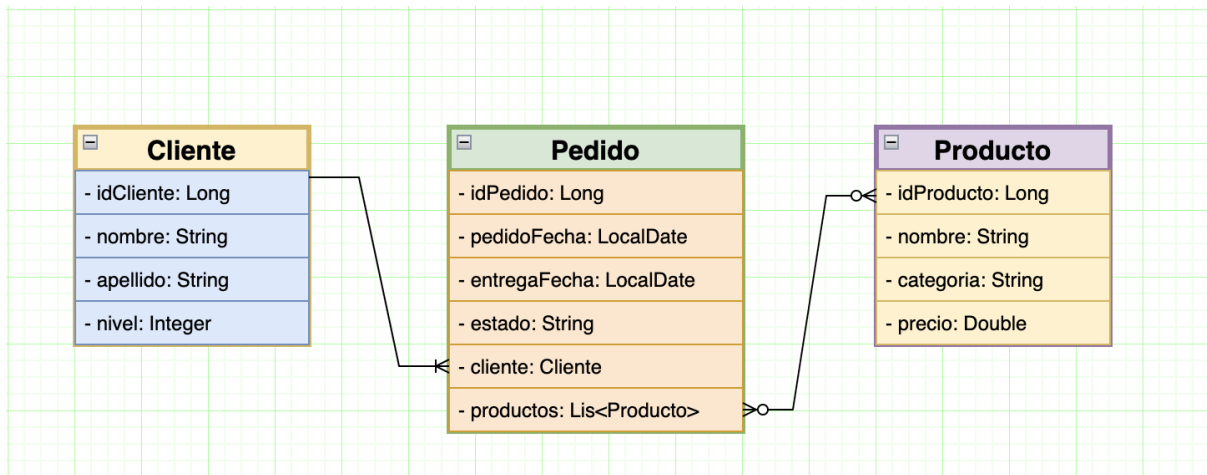


Figura # 1

Se agrega el proyecto creado con sus respectivas clases, repositorios y los datos de muestra cargados en la base de datos H2 en memoria.

Exploramos las siguientes operaciones.

Operaciones no terminales

- filter()
- map()

Operaciones de terminal

- anyMatch()
- collect()

En el proyecto encontrará un ejemplo en la siguiente ruta: <test/java/com.udea.EjercicioFuncional> en el archivo **EjercicioFuncionalApplicationTests**, en ese archivo realizarán la solución de los ejercicios propuesto basándose del ejemplo.

Ejercicio # 1 - Obtener los 5 pedidos más recientes (Valor del Punto (1.25))

Tips: En este ejercicio se pretende ordenar de forma descendente los registros de pedidos por el campo de fecha de pedido. Pueden utilizar el **Comparator.reversed()**.

Ejercicio # 2 - Obtener una lista de productos con categoría = "Juguetes" y luego aplicar el 15% de descuento (Valor del Punto (1.25))

Tips: En este ejercicio, transformarás los datos utilizando la API de flujo. Después de obtener una lista de productos con una categoría que pertenece a **"Juguetes"** utilizando **filter()**, puede aplicar un descuento del **15%** al precio del producto utilizando **map()**.

Ejercicio # 3 - Obtener una lista de pedidos con productos pertenecientes a la categoría = "Bebé" (Valor del Punto (1.25))

Tips: Hay que partir del flujo de datos de las entidades del pedido y luego comprobar si los productos del pedido pertenecen a la categoría **"Bebé"**. Para ello, la lógica del **filter()** busca en el flujo de productos de cada registro de pedido, se debe utilizar **anyMatch()** para determinar si algún producto cumple con los criterios.

Ejercicio # 4 - Obtener una lista de productos pertenecientes a la categoría "Libros" con precio > 100 (Valor del Punto (1.25))

Tips: Se pueden aplicar 2 operaciones **filter()** para obtener el resultado.