

UNIVERSIDAD DE ANTIOQUIA
PROGRAMA INGENIERÍA DE SISTEMAS
TÉCNICA DE PROGRAMACIÓN Y LABORATORIO G1. (P.O.O)
TALLER SOBRE COLECCIONES
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.

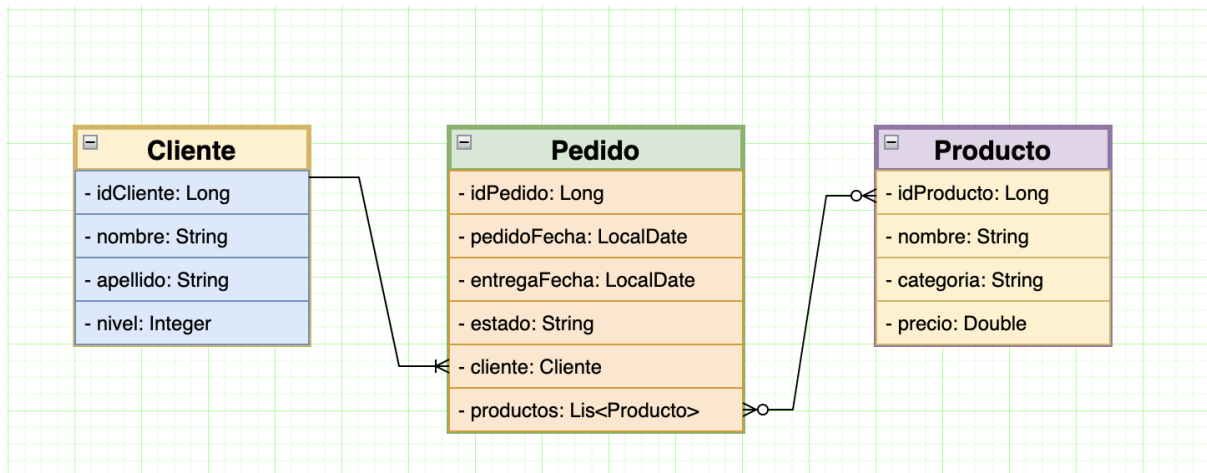


Figura # 1

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 un mapa (map) de datos con el id del pedido y el número de productos del mismo (Valor del Punto (2.5))

Tips: El ejercicio es crear un mapa (map) de datos con el id del pedido como clave, mientras que el valor asociado es el recuento de productos. La operación terminal **Collectors.toMap()** que acepta dos argumentos para que especifiques la clave y el valor respectivamente.

Ejercicio # 2 - Generar un mapa (map) de datos con registros de pedidos agrupados por cliente. (Valor del Punto (2.5))

Tips: Este ejercicio consiste en consolidar una lista de pedidos por cliente. Utilizar el **Collectors.groupingBy()**