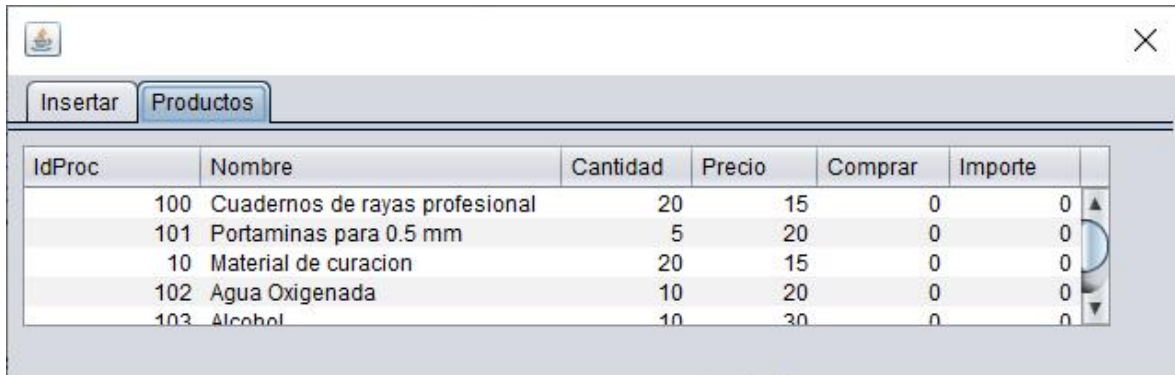


Practica 2 de Conexión a BD y manipulación de datos

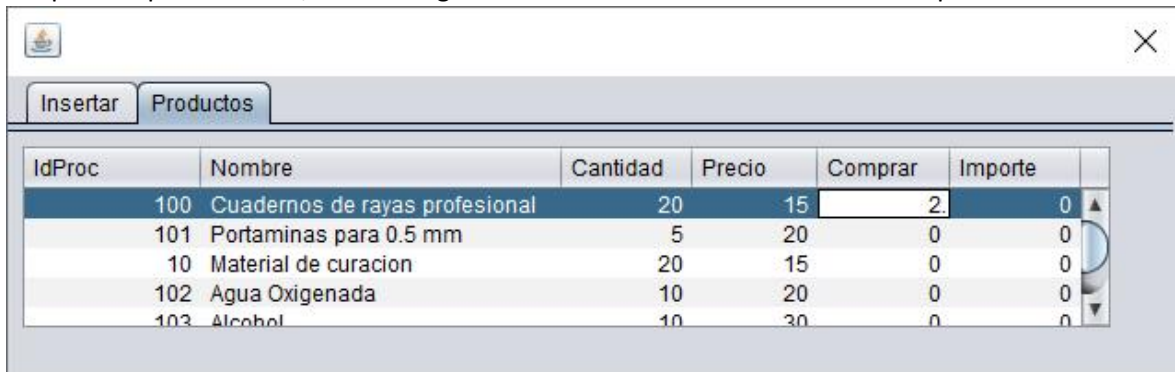
De tu tabla (vista) de productos ahora agregaras otras dos columnas: comprar e importe. La primera servirá para poder especificar la cantidad de un producto que se desee comprar y la segunda reflejará el importe a pagar, que corresponde al precio del producto por la cantidad especificada a comprar, tal como se muestra en el siguiente esquema:



IdProc	Nombre	Cantidad	Precio	Comprar	Importe
100	Cuadernos de rayas profesional	20	15	0	0
101	Portaminas para 0.5 mm	5	20	0	0
10	Material de curacion	20	15	0	0
102	Agua Oxigenada	10	20	0	0
103	Alcohol	10	30	0	0

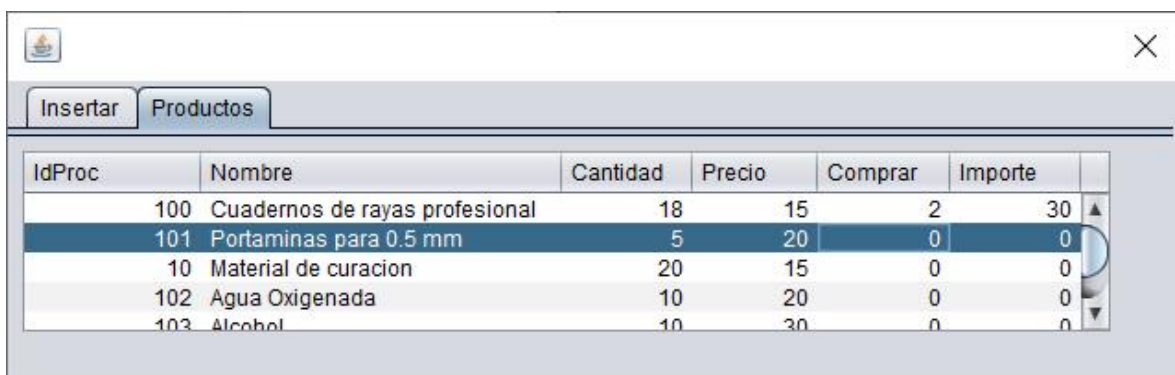
Las dos columnas en un principio aparecen en cero, porque no haya valores aun

Después se podrá editar , es decir ingresar valores solo en la columna de 'Comprar'



IdProc	Nombre	Cantidad	Precio	Comprar	Importe
100	Cuadernos de rayas profesional	20	15	2	0
101	Portaminas para 0.5 mm	5	20	0	0
10	Material de curacion	20	15	0	0
102	Agua Oxigenada	10	20	0	0
103	Alcohol	10	30	0	0

Al dar [enter] el valor del importe debe actualizarse, tal como se muestra en la siguiente figura:



IdProc	Nombre	Cantidad	Precio	Comprar	Importe
100	Cuadernos de rayas profesional	18	15	2	30
101	Portaminas para 0.5 mm	5	20	0	0
10	Material de curacion	20	15	0	0
102	Agua Oxigenada	10	20	0	0
103	Alcohol	10	30	0	0

Nota que también cambió el valor la cantidad de existencia que se tenía de 20 a 18

Lo que tienes que hacer es:

1 Modifica tu modelo de tabla para que incorpore las dos nuevas columnas, tanto el nombre de ellas como su tipo

2. En la clase AccesoDatos , modifica el tamaño del arreglo de ArrayList para que incluya las dos nuevas columnas asígnale el valor de cero a cada una

3. en el mismo modelo, has que se permita editar la columna de comprar (es decir digitalizar un valor numérico), las demás columnas no deben poder editarse

Esto se consigue agregando en el modelo el método:

@Override

```
public boolean isCellEditable(int r, int c)
{
    return c== numero de columna que puede editarse;
}
```

4. Has que la columna a editar tenga el valor que se digitaliza, esto se logra con el método :

```
public void setValueAt(Object v,int r,int c)
{
    La columna a editar debe tomar el valor que se digitalice
}
```

5. Has que el valor de la columna correspondiente a la cantidad actualice su valor: valor actual de cantidad menos la cantidad a comprar

6. Has que el valor del importe se le asigne el resultado de la compra por el precio, esto se hace el método getValueAt(r,c), en el modelo de la tabla