**Actividad 3.** Comparación de Queries relacionales y orientadas a objetos:

1. Listar todos los productos de la empresa con todos sus datos incluyendo los datos del Tipo de Producto para cada producto.

**Query en SQL:**

**SELECT** \* **FROM** productos **LEFT JOIN** tipos\_productos

**ON** (productos.tipo\_producto\_id = tipos\_productos.id)

**Query en JPQL:**

**public void consulta1(){**

**EntityManagerFactory emf = Persistence.createEntityManagerFactory("Actividad\_1.7PU");**

**EntityManager em = emf.createEntityManager();**

**em.getTransaction().begin();**

**Query query = em.createQuery("SELECT p.id, p.nombre, p.cantidad, p.precioVenta, p.unidad, p.tipoProducto, tp.id FROM Producto p , TipoProducto tp WHERE p.tipoProducto.id = tp.id");**

**List<Object[]> l = query.getResultList();**

**if (l.isEmpty()){**

**System.out.println("Lista vacía");**

**}**

**for (Object[] u : l){**

**System.out.println("Id Producto: " + u[0]**

**+ "\n Producto: " + u[1]**

**+ "\n Cantidad: " + u[2]**

**+ "\n Precio Venta: " + u[3]**

**+ "\n Unidad: " + u[4]**

**+ "\n Tipo Producto: " + u[5]**

**+ "\n Id Tipo Producto: " + u[6]);**

**}**

**}**

1. Listar los datos de las facturas de un cliente específico usando como parámetro el Documento del cliente. No mostrar los productos.

**Query en SQL:**

**SELECT** \* **FROM** facturas **WHERE** facturas.cliente\_id = 258064

**Query en JPQL:**

"SELECT f.id, f.fecha, f.cliente.id" +  
" FROM Factura f " +  
" WHERE f.cliente.id=:"+documento

1. Listar todos los productos vendidos, incluyendo los datos del producto, la cantidad vendida, los datos de la factura y los datos del cliente.

**Query en SQL:**

**SELECT** productos.id **AS** productos\_id,

productos.nombre **AS** nombre\_producto,

productos.cantidad,

productos.precio\_venta,

productos.unidad,

productos.tipo\_producto\_id,

facturas\_productos.cantidad,

facturas.id **AS** facturas\_id,

facturas.fecha,

facturas.cliente\_id,

clientes.nombre\_completo,

clientes.documento,

clientes.telefono,

clientes.email,

clientes.tipo

**FROM** ((facturas\_productos **LEFT JOIN** productos

**ON** (facturas\_productos.producto\_ID = productos.id)) **LEFT JOIN** facturas

**ON** (facturas\_productos.factura\_id = facturas.id)) **LEFT JOIN** clientes

**ON** (facturas.cliente\_id = clientes.id)

**Query en JPQL:**

"SELECT p.id, p.nombre, p.cantidad, p.precioVenta, p.unidad, p.tipoProducto.id, " +  
" fp.cantidad, " +  
" f.id, f.fecha, f.cliente.id, " +  
" c.nombreCompleto, c.documento, c.telefono, c.email, c.tipo" +  
" FROM Factura\_Producto fp LEFT JOIN fp.producto p" +  
" LEFT JOIN fp.factura f" +  
" LEFT JOIN f.cliente c"

1. Utilizando como parámetro el ID de una materia prima, listar los proveedores que la venden, indicando información de la materia prima el precio de venta e información del proveedor.

**Query en SQL:**

**SELECT** A1.id **AS** materia\_prima\_id,

A1.nombre **AS** nombre\_materia\_prima,

A1.cantidad **AS** cantidad\_materia\_prima,

A2.precio,

proveedores.id **AS** id\_proveedor,

proveedores.nombre **AS** nombre\_proveedor,

proveedores.telefono **AS** telefono\_proveedor,

proveedores.direccion **AS** direccion\_proveedor,

proveedores.email **AS** email\_proveedor

**FROM** ((materias\_primas AS A1 **INNER JOIN** proveedor\_materia\_prima **AS** A2

**ON** (A1.id = A2.proveedor\_materia\_prima\_id)) **LEFT JOIN** proveedores

**ON** (A2.proveedor\_id=proveedores.id)) **WHERE** A1.id = 31;

**Query en JPQL:**

**public void consulta4(int id){**

**EntityManagerFactory emf = Persistence.createEntityManagerFactory("Actividad\_1.7PU");**

**EntityManager em = emf.createEntityManager();**

**em.getTransaction().begin();**

**Query query = em.createQuery("SELECT p FROM ProveedorMateriaPrima p WHERE p.materiaPrima.id = ?1");**

**query.setParameter(1, id);**

**List<ProveedorMateriaPrima> prov = query.getResultList();**

**if (prov.isEmpty()){**

**System.out.println("esta sola");**

**}else{**

**for (ProveedorMateriaPrima u : prov){**

**System.out.println("Nombre Proveedor: "+ u.getProveedor().getNombre() +**

**"\n Teléfono: " + u.getProveedor().getTelefono()+**

**"\n Dirección: " + u.getProveedor().getDireccion()+**

**"\n Email: " + u.getProveedor().getEmail());**

**System.out.println("Materia Prima: "+ u.getMateriaPrima().getNombre());**

**}**

**}**

**}**

1. Listar los productos que coincidan con los siguientes parámetros de búsqueda: El precio del producto debe ser mayor a un número ingresado y que el producto esté compuesto por una materia o varias materias primas que contengan en el nombre una cadena de texto ingresada.

**Query en SQL:**

**SELECT** productos.nombre, productos.precio\_venta **FROM** (productos **RIGHT JOIN** productos\_materias\_primas **AS** A1

**ON** (productos.id = A1.producto\_id)) **LEFT JOIN** materias\_primas A2

**ON** (A1.materia\_prima\_id = A2.id)

**WHERE** A2.nombre **LIKE** '%nt%' **AND** productos.precio\_venta > 5000