

---

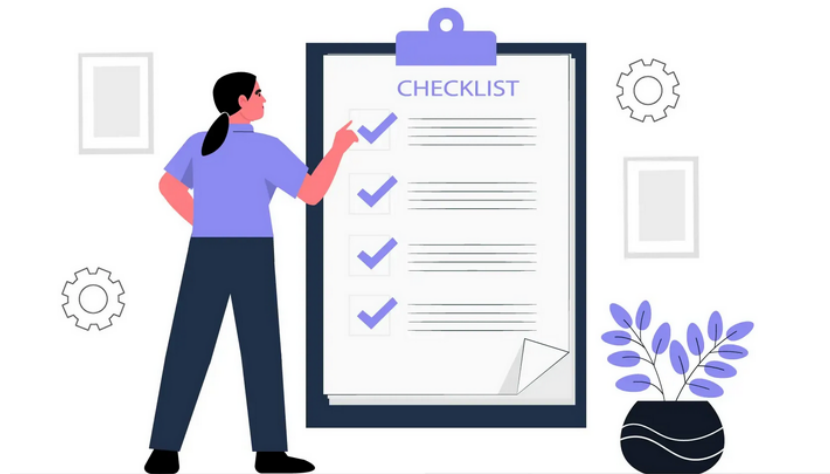
# CRITERIOS PROYECTO JDBC

*by*

Jonathan Carrero

---

**Abstract.** — El objetivo de este documento es entender y dar algunas pinceladas sobre lo que se pide para desarrollar el proyecto de la asignatura de Acceso a Datos. Al mismo tiempo, se exponen los puntos que son de obligado cumplimiento y todos los requerimientos que este proyecto debe tener para su entrega y correcta evaluación.



## 1. Enunciado principal

Para este proyecto se pide hacer una gestión de productos en una tienda. En otras palabras; sois los dueños de una tienda física que vende productos y hay que organizar y llevar un control de dichos productos. Para que sea algo más creativo, se deja a elección el tipo de tienda y productos que vais a vender, pero para explicar el enunciado se va a dar por hecho que, esta vez, estamos vendiendo productos de comida japonesa.

La siguiente tabla muestra algunas de las características que podría tener nuestra tienda. Nótese que no se trata de considerar todos estos detalles, sino que se trata de un ejemplo para dar ideas. La lista de requerimientos obligatorios se verá al final del documento. Como decía, existe una gestión referente a los productos que estamos vendiendo, una gestión con los proveedores que nos facilitan dichos productos, gestión de las ventas a nuestros clientes, algunos reportes estadísticos interesantes como por ejemplo los productos más vendidos, ganancias, productos más rentables en relación coste/precio de venta, etc. También tenemos una gestión de usuarios que, tal vez en este modelo no tendría mucho sentido, ya que nuestros clientes son simplemente personas a pie de calle.

| Característica          | Descripción  |
|-------------------------|--|
| Gestión de Productos    | Registro de productos con detalles como nombre, categoría (frescos, secos, congelados, bebidas), precio unitario, cantidad en stock y fecha de caducidad. Podría incluir alertas para productos próximos a caducar o con bajo inventario en nuestros almacenes.                |
| Gestión de Proveedores  | Registro de proveedores con nombre, información de contacto y productos suministrados. Permite consultar el historial de pedidos realizados y gestionar entregas.  |
| Gestión de Ventas       | Registro de ventas con información detallada sobre productos vendidos, cantidad, precio total y fecha. Generación de tickets de compra y aplicación de descuentos.   |
| Reportes y Estadísticas | Generación de reportes sobre productos más vendidos, ganancias diarias/semanales/mensuales, y productos con bajo rendimiento en ventas. La verdad, desconozco si desde MySQL se podrían exportar reportes (salidas de queries) en formato PDF o Excel. Podría ser interesante. |
| Gestión de Usuarios     | Creación de cuentas para empleados y administrador, con control de accesos. Los empleados gestionan ventas y consultas; el administrador tiene control total.  |
| Interfaz Amigable       | Interfaz de consola con mensajes intuitivos en todo momento cada vez que haya algún tipo de interacción con la base de datos. Mostrar todo tipo de queries en un formato amigable y formal, sobre todo las queries de tipo SELECT.   |

La siguiente tabla pretende identificar algunas de las funciones que tendría nuestro sistema. Por ejemplo, está claro que habría funciones de registro para añadir nuevos productos, productos ya existentes, cambiar precios, añadir descuentos, etc. También tendríamos consultas de selección y filtrado, así como todo lo referente a la gestión de proveedores y ventas de productos.

| Función                     | Descripción  |
|-----------------------------|--|
| Registro de Productos       | Permite ingresar productos nuevos al inventario, asociarlos con un proveedor, establecer precios y cantidades iniciales. |
| Actualización de Inventario | Actualiza el stock automáticamente tras cada venta o manualmente después de recibir pedidos de proveedores.              |
| Consulta de Productos       | Permite filtrar productos por categorías, estado (bajo inventario, próximo a caducar) o buscar por nombre o código.      |
| Gestión de Proveedores      | Registra y actualiza la información de los proveedores. Permite consultar productos suministrados y realizar pedidos.    |
| Venta de Productos          | Registra las ventas ingresando productos y cantidades. Calcula automáticamente el precio total con impuestos.            |
| Generación de Reportes      | Genera reportes de ventas realizadas en un rango de fechas, productos más vendidos y aquellos con baja rotación.         |

Por último, encontramos una posible distribución de tablas y campos. Échale un ojo, también puede darte ideas a la hora de diseñar tu base de datos.

| Tabla          | Descripción  |
|----------------|--|
| Productos      | <p>Contiene información sobre los productos disponibles en el inventario:</p> <ul style="list-style-type: none"> <li>▪ <code>id_producto</code> (PK, AUTO_INCREMENT)</li> <li>▪ <code>nombre</code> (VARCHAR)</li> <li>▪ <code>categoria</code> (VARCHAR)</li> <li>▪ <code>precio</code> (DECIMAL)</li> <li>▪ <code>stock</code> (INT)</li> <li>▪ <code>fecha_caducidad</code> (DATE)</li> </ul> |
| Proveedores    | <p>Registra información sobre los proveedores:</p> <ul style="list-style-type: none"> <li>▪ <code>id_proveedor</code> (PK, AUTO_INCREMENT)</li> <li>▪ <code>nombre</code> (VARCHAR)</li> <li>▪ <code>contacto</code> (VARCHAR)</li> <li>▪ <code>direccion</code> (VARCHAR)</li> </ul>  |
| Pedidos        | <p>Registra los pedidos realizados a proveedores:</p> <ul style="list-style-type: none"> <li>▪ <code>id_pedido</code> (PK, AUTO_INCREMENT)</li> <li>▪ <code>id_proveedor</code> (FK)</li> <li>▪ <code>fecha_pedido</code> (DATE)</li> <li>▪ <code>cantidad_total</code> (INT)</li> </ul>   |
| Ventas         | <p>Almacena los datos de las ventas realizadas:</p> <ul style="list-style-type: none"> <li>▪ <code>id_venta</code> (PK, AUTO_INCREMENT)</li> <li>▪ <code>fecha_venta</code> (DATETIME)</li> <li>▪ <code>monto_total</code> (DECIMAL)</li> </ul>  |
| Detalle_Ventas | <p>Contiene los detalles de cada venta:</p> <ul style="list-style-type: none"> <li>▪ <code>id_detalle</code> (PK, AUTO_INCREMENT)</li> <li>▪ <code>id_venta</code> (FK)</li> <li>▪ <code>id_producto</code> (FK)</li> <li>▪ <code>cantidad</code> (INT)</li> <li>▪ <code>subtotal</code> (DECIMAL)</li> </ul>  |
| Usuarios       | <p>Registra los datos de los usuarios del sistema:</p> <ul style="list-style-type: none"> <li>▪ <code>id_usuario</code> (PK, AUTO_INCREMENT)</li> <li>▪ <code>nombre_usuario</code> (VARCHAR)</li> <li>▪ <code>contraseña</code> (VARCHAR)</li> <li>▪ <code>rol</code> (ENUM: 'administrador', 'empleado')</li> </ul>  |

## 2. Requisitos obligatorios

Esta es la lista de cosas que debe tener vuestro proyecto para poder ser evaluado:

- I. Javadoc en el programa Java. No es necesario generar el Javadoc, pero sí comentar el programa de manera correcta, tal y como si fuese a ser generado.
- II. Repositorio donde se vaya actualizando el programa Java que tenéis que desarrollar.
- III. Las tablas deben estar normalizadas. Vosotros pensáis en un diseño inicial de base de datos, y antes de crear las tablas y sus correspondientes campos, debéis asegurar que dichas tablas están normalizadas.
- IV. Entrega de un documento técnico donde se explique en qué consiste vuestro proyecto. Repito, documento técnico, por lo que podéis usar tecnicismos. La idea en vuestro proyecto de Multimedia es hacer un manual de usuario, algo que no es técnico para nada. Pues bien, aquí todo lo contrario; un documento que explique y recoja aquellos aspectos técnicos del desarrollo, tecnologías utilizadas, y ciertos esquemas que se nombran en los siguientes puntos. Extensión del manual entre 3.000 y 4.000 palabras.
- V. No tiene sentido hacer mil esquemas, pero sí que hay tres que resultan especialmente interesantes, así que esos son los obligatorios: Modelo ER, Modelo Relacional (mirar apuntes de Bases de Datos) y Diagrama de Clases (mirar apuntes de Entornos de Desarrollo).
- VI. El manejo de la base de datos debe hacerse totalmente desde el programa en Java, incluidas aquellas queries para la creación de las tablas (o sea, lenguaje DDL). Pensad la mejor manera de estructurar vuestro programa Java antes de empezar a programar. Debéis incluir operaciones CRUD tanto de DDL como de DML. Una buena idea sería crear menús por consola, de manera que, por ejemplo, selecciones la opción de insertar un nuevo producto. Entonces se entra en un túnel de preguntas en relación al nuevo producto (nombre, precio, cantidad inicial, tipo, etc.). Al finalizar esa ronda de pregunta, se procede a hacer el INSERT correspondiente. Pues bien, lo mismo para cuando se quiera hacer un UPDATE o DELETE.
- VII. En todo momento, deben ir apareciendo mensajes por consola cuando se vendan productos (lo cual habrá que simular con algún método), se actualicen, se borren, se muestren con SELECT y cosas así.
- VIII. Se deben realizar queries de tipo JOIN.

## 3. Requisitos opcionales

No hay una lista como tal de funciones opcionales, ya que cada cual implementará una tienda diferente. Claro está, cuantas más funcionalidades extra haya y cuanto más originales sean, mejor. Tampoco se trata de volverse loco y añadir quince o veinte cosas extra, pero sí que sería interesante pensar en unas cuentas. Vamos, algo así como lo que estáis haciendo con la calculadora, sólo que un poco más ampliado y currado, ya que esta vez sois dos personas. También se valorará positivamente que el programa disponga de una gran cantidad de consultas DDL y DML, de forma que se pueda manejar la base de datos con total flexibilidad desde el propio programa en Java.

---

*January 19, 2025*

J.C., Colegio Litterator • *E-mail* : [jonathan.carrero@colegiolitterator.com](mailto:jonathan.carrero@colegiolitterator.com)