

Ejercicios.

Para los ejercicios siguientes utilizaremos la base de datos **classicModels**.

1. Diseña un proyecto Java que obtenga:
 - a. un listado de clientes de una localidad que será pasada como parámetro. Debes mostrar las columnas: número de cliente, nombre, teléfono y ciudad.
 - b. listado de todas las categorías existentes (categoría y descripción).
 - c. listado de todos los productos de una categoría: código, nombre, precio de compra, de venta y categoría.
 - d. listado de todos los productos con un precio de compra entre dos valores.
2. Diseña un proyecto Java que permita insertar filas en la tabla “categoriasproductos”. Los datos se solicitarán al usuario.
 - a. Mediante Statement.
 - b. Mediante PreparedStatement.
3. Diseña un proyecto Java que permita realizar estas tareas sobre la base de datos classicModels (yo he utilizado una copia de la tabla productos).
 - a. Actualizar precios de venta de productos. Incrementar todos un 5%
 - b. Actualizar el precio de venta de un producto. Incrementar un % el precio de venta de un producto identificado por su código.
 - c. Eliminar producto a partir de su código
 - d. Insertar nuevo producto.
4. Diseña un programa Java que muestre al usuario una lista de los clientes de la base de datos “classicModels” (número de cliente y nombre), el usuario elegirá uno y el programa mostrará pantalla todos los pedidos de ese cliente con los datos: número de pedido, fecha de pedido y estado.
5. Obtener en PDF un listado de la tabla productos cuyo precio de venta esté entre dos valores dados. Ordena los datos de forma ascendente por el precio de venta. Crea una clase con un método que reciba como parámetros el nombre del archivo que debe crearse y los precios. Observa la imagen.
Para realizar este ejercicio necesitas cargar la dependencia de la API iTextPdf.

Puedes encontrar ayuda en ...

- a. <https://www.baeldung.com/java-pdf-creation>
- b. http://chuwiki.chuidiang.org/index.php?title=Ejemplo_sencillo_de_creaci%C3%B3n_de_un_pdf_con_iText

Listado de productos con precio entre 10.0 y 56.0			
Fecha: 2022-05-03			
Número producto	Nombre producto	Vendedor	Precio de venta
S24_1937	1939 Chevrolet Deluxe Coupe	Motor City Art Classics	33,19
S24_2840	1958 Chevy Corvette Limited Edition	Carousel DieCast Legends	35,36
S24_2972	1982 Lamborghini Diablo	Second Gear Diecast	37,76
S32_2206	1982 Ducati 996 R	Gearbox Collectibles	40,23
S24_3969	1936 Mercedes Benz 500k Roadster	Red Start Diecast	41,03

6. Diseña una aplicación Java que permita crear un PDF con los clientes de una localidad concreta. Las columnas que deben aparecer en el archivo son: número de cliente, nombre, teléfono, ciudad y país. El archivo debe llamarse ClientesDeXXXX.pdf, siendo XXXX la localidad.
7. Diseña una aplicación Java que genere un archivo de texto con los productos de una categoría concreta de la base de datos "classicModels". Cada registro de la tabla debe corresponderse con una línea del archivo. Utiliza el punto y coma (;) como separador de columnas en el archivo. Usa solo los campos código, nombre, categoría, precio de venta y de compra. El archivo tendrá el nombre Productos_Categoria_XXXX.txt, donde XXXX será la categoría elegida.
8. Envío de correo electrónico básico desde Java utilizando una cuenta de Google.
 - a. Incluye en el proyecto la dependencia Java Mail Api de com.sun.mail
 - b. Debes configurar tu cuenta para que permita el envío de correo mediante aplicaciones de terceros (Acceso de aplicaciones poco seguras).
9. Visto el envío de correo electrónico, vamos a aplicarlo a las bases de datos. La idea es enviar un PDF que contenga un listado de los productos de nuestra base de datos a los clientes de la empresa que tienen correo electrónico. Mediante PhpMyadmin, crea una columna nueva para el correo y asigna correo a dos o tres clientes. No es necesario mostrar todas las columnas.
 - a. Enviar PDF de productos al cliente con número XXX.
 - b. Enviar PDF de productos a todos los clientes con correo
10. Diseña un programa Java que muestre en pantalla las tablas de la base de datos classicModels consultando los metadatos.
11. Modifica el programa anterior para que muestre también las columnas de cada tabla.
12. Diseña una aplicación Java como se indica:
 - a. Base de datos: classicmodels.

- b. Crea una tabla nueva llamada usuarios e inserta datos en ella.
 - i. Campos: id, nombre, contraseña, tipousuario (normal y administrador). Guarda varios usuarios en dicha tabla.
 - ii. Esta tabla almacena los usuarios que pueden trabajar con la aplicación que estamos diseñando.
 - c. La aplicación pedirá nombre de usuario y password.
 - i. Si el usuario es normal se le mostrarán las tablas de la base de datos (menos la tabla de usuarios), de ellas elegirá una y se visualizarán sus datos.
 - ii. Si el usuario es administrador se le dará la opción anterior y además se le permitirá crear nuevos usuarios.
 - iii. Si el usuario no existe como tal no se le permitirá acceso a la aplicación.
13. Diseña una aplicación Java que contenga una clase que a su vez contenga un método estático llamada pedidosClientes. Este método recibe una lista de números de cliente y con ellos debe:
- a. Si el cliente tiene pedidos, elaborar un PDF con sus pedidos (nº de pedido, fecha de pedido, estado e **importe total del pedido**) .
 - b. Para los clientes inexistentes o sin pedidos genera una línea en un archivo de texto plano llamado "incidenciasPedidos.txt" en la que se refleje el nº del cliente, nombre (si es cliente) y el comentario "No existe como cliente" o "No tienen pedidos" según corresponda.