

Acceso a Datos.

Examen 2ª Evaluación.

Consideraciones

iniciales:

Crea una carpeta con tu nombre y apellidos con el siguiente formato:

- **apellido1_apellido2_nombre_EV2.**

Dentro de ella deberás crear los tres proyectos que debes entregar al finalizar el examen con los ejercicios resueltos (llamados **Ejercicio1**, **Ejercicio2** y **Ejercicio3**).

Puedes organizar el código como creas más conveniente, pero dentro de cada proyecto solo puede haber un único método "main" que será el encargado de guiar la ejecución de todas las partes de cada ejercicio.

Genera un código lo más estructurado y limpio posible, que facilite su lectura y comprensión (por favor), comenta aquellas partes que consideres que necesitan una aclaración adicional, y sobre todo si por algún motivo tomas alguna decisión que se salga de lo pedido justifícala y explícala lo mejor posible mediante un comentario en el código.

En la máquina virtual que con el nombre de "**Windows10 21H2**", dispones de todo lo necesario para realizar este examen (librerías, BD, servidores y trozos de código).

Si tienes cualquier duda consulta con la profesora que estará presente en el aula.

Ejercicio 1.

Crea un nuevo proyecto (llamado **"Ejercicio1"**) que nos permita manejar el archivo llamado **"IESBDDDB4O"** (que encontrarás en la carpeta **"BD"** del escritorio de tu máquina virtual) con **DB4O**.

En dicho archivo encontrarás la información que viene en las siguientes tablas:

Empleado					
nombre	apellidos	dni	telefono	numDepartamentos	ArrayList<Departamento> departamentos
Angela	del Barrio Fernandez	11111111-A	111111111	4	(1 2 3 4 5 6 7 8)
Valentin	Rebollada	22222222-B	887654321	4	(2 3)
Gema	Portilla	33333333-C	787654321	1	(8)
Miguel	Llano Rios	44444444-D	687654321	4	(4 5 6 7)
Angel Damaso	Gomez	55555555-E	587654321	3	(1 2)

Departamento	
id	nombre
1	Servicio Tecnico
2	Informatica
3	RRHH
4	Gestion
5	Administrativo
6	Ventas
7	Clientes
8	Proyectos Innovacion

Deberás crear una clase para almacenar los datos de cada tabla, teniendo en cuenta lo siguiente:

- Las clases deberán llamarse **"Empleado"** y **"Departamento"**.
- Los atributos se deberán llamar como los encabezados de las columnas.

- El atributo "departamentos" de la clase "Empleado" debe ser un **ArrayList** que contenga objetos tipo "Departamento". Los números que vienen en la tabla solo indican los datos de que asignatura se guarda en el array, pero lo que está realmente almacenado en el archivo son todos los datos de cada una de esas asignaturas.
- Elige el tipo de dato correcto para cada uno de los demás atributos.
- Ten en cuenta que los datos se han introducido sin incluir las tildes.

Apartado 1. Crea un método que reciba como parámetro el objeto que establece la conexión con la BD de BD4O y muestre por pantalla la información de todos los empleados cuyo nombre no empiece por "V" ni acabe en "a".

Por cada empleado deben aparecer todos sus datos, seguidos del nombre y el **nombre** de los departamentos a los que pertenezca.

Apartado 2. Crea un método que reciba como parámetro el objeto que establece la conexión con la BD de BD4O y modifique los datos del empleado con dni "11111111-A", cambiando su nombre y sus apellidos por los tuyos.

Muestra la información de ese empleado antes y después de realizar la modificación.

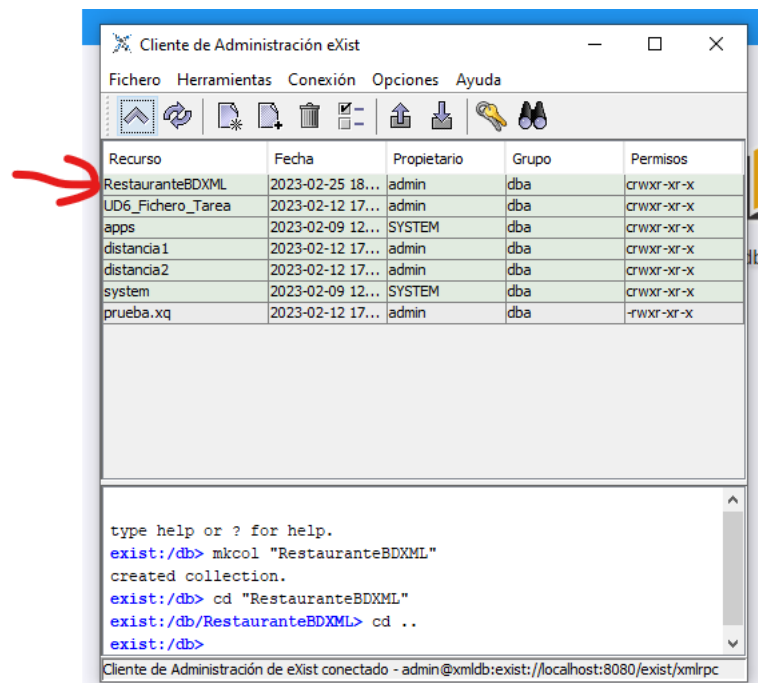
Apartado 3. Crea un método que reciba como parámetro el objeto que establece la conexión con la BD de BD4O y borre la información de todos los empleados que estén en más de tres departamentos (tener en cuenta numDepartamentos). Muestra la información de todos los empleados que quedan en la BD una vez efectuado el borrado.

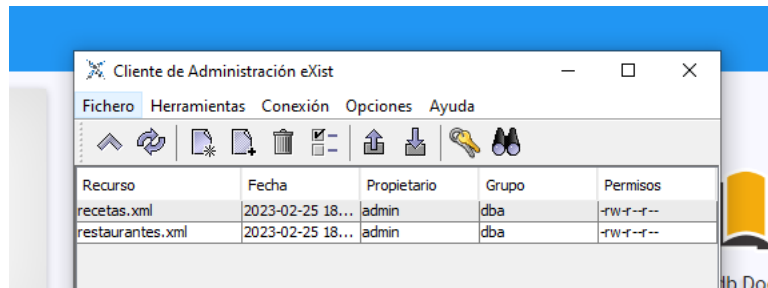
Por cada empleado deben aparecer todos sus datos, seguidos del nombre de todos los departamentos a los que pertenezca.

Ejercicio 2.

Crea un nuevo proyecto (llamado **"Ejercicio2TuNombre"**) que nos permita manejar la colección llamada **"RestauranteBDXML"** (que encontrarás en el servidor de eXist de tu máquina virtual) con **eXist**.

En dicha colección encontrarás dos archivos XML (**"recetas.xml"** y **"restaurantes.xml"**) que contienen información de recetas y restaurantes que elaboran dichas recetas.





Si necesitas ver el código de los documentos puedes acceder a ellos mediante el IDE de eXist.

Los datos de la conexión con eXist son los siguientes:

- **Nombre del servidor:** localhost.
- **Puerto:** 8080.
- **Usuario:** admin.
- **Contraseña:** admin.

Deberás crear un menú en java que nos permita ir realizando una serie de consultas, cada apartado irá relacionado con una consulta y por lo tanto con una opción del menú, además la última opción deberá permitir abandonar la ejecución del programa.

En la carpeta "código" que encontrarás en el escritorio de tu máquina virtual, tienes el código necesario para crear el menú y pintarlo por pantalla. El primer bucle lo deberás copiar en el "main" (y deberás añadir lo que sea necesario para que sea 100% operativo), y los otros dos métodos que vienen en el archivo deberán estar en la misma clase del "main" pero obviamente fuera de él. Si lo prefieres puedes crear tú el menú desde cero.

Realiza los siguientes apartados mediante XPath.

Apartado 1. Muestra todos los datos de la receta con id=r005.

Apartado 2. Los nombres de todas las recetas que tengan entre 300 y 450 calorías (ambos valores incluidos).

Apartado 3. Muestra los nombres de todas las recetas de la categoría verduras.

Apartado 4. Muestra todos los nombres de la colección (no importa de a que archivo pertenezcan).

Apartado 5. Muestra los nombres de todos los restaurantes que preparan 2 recetas o menos.

Realiza los siguientes apartados mediante xQuery en JAVA.

Apartado 6. Muestra los nombres de los restaurantes que preparan alguna receta de pasta..

Apartado 7. Muestra el id de todos los restaurantes que preparan "Tiramisú". Saca solo los datos.

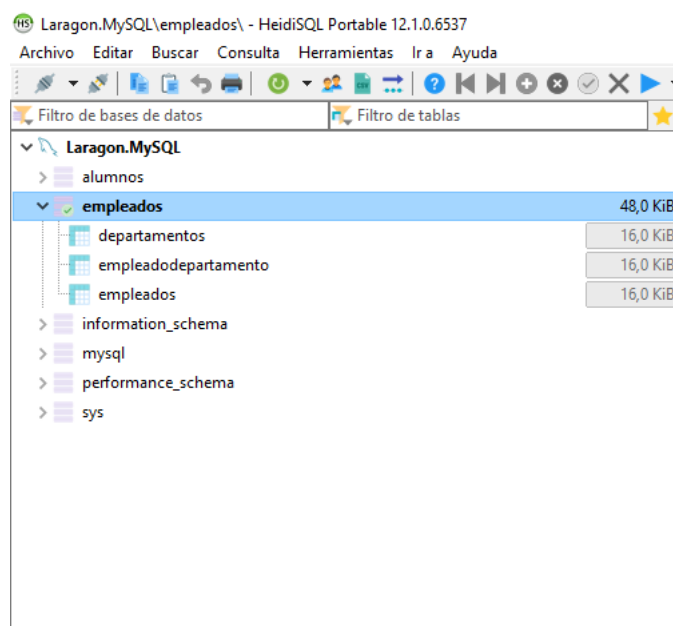
Apartado 8. Muestra el nombre de todas las recetas que prepara el restaurante "Restaurante Santander". Agrupa todo dentro de una etiqueta.

Apartado 9. Muestra el nombre y la categoría de las recetas cuya categoría tenga una "u". Solo debes mostrar los datos sin etiquetas, es decir debería salir algo como: "Ensalada de pollo verduras" para cada receta que cumpla la condición. Inserta un espacio entre el nombre y la categoría

Apartado 10. Muestra el nombre de cada restaurante y el total de calorías de sus recetas. Agrupa todo dentro de una etiqueta restaurante.

Ejercicio 3.

Crea un nuevo proyecto (llamado "**Ejercicio3TuNombre**") que nos permita manejar la base de datos llamada "**empleados**" (que tienes disponible en el servidor de MySQL de tu máquina virtual), mediante el uso de componentes de acceso a datos, los componentes están ya creados en el proyecto llamado "**EmpresaBean**" (tienes el proyecto creado y listo para usarse en la carpeta de librerías que encontrarás en el escritorio de tu máquina virtual).



Una vez creado deberás **desarrollar dos clases** dentro de tu proyecto, una que nos permita acceder a la base de datos utilizando los componentes que tienes disponibles y otra que nos permita manejar la ejecución de todo el programa.

En caso de ser necesario deberás modificar ligeramente los componentes JavaBeans.

Apartado 1. En la clase que nos permite acceder a los datos deberás crear 3 métodos que muestren por pantalla todos los datos de la base de datos. Crea un método para cada una de las tablas, haciendo uso de los métodos creados en los componentes asociados a cada una de ellas.

Apartado 2. En la clase que nos permite acceder a los datos deberás crear 1 método que nos permita introducir un nuevo Departamento. El método debe recibir todos los datos a insertar por parámetros.

Los datos de este nuevo registro son:

- **ID.** 10
- **Nombre.** "Relaciones Públicas TuNombre"

Apartado 3. Deberás añadir una funcionalidad nueva a los componentes de forma que se permita MODIFICAR un NOMBRE de un EMPLEADO determinado. (Recuerda recargar el jar en caso de ser necesario).

En la clase que nos permite acceder a los datos deberás crear 1 método que nos permita seleccionar los datos del empleado con DNI = 44444444-D y modifica su nombre por el tuyo.

Apartado 4. En la clase que nos permite acceder a los datos deberás crear 1 método que nos permita añadir una fila en la tabla empleadoDepartamento con los siguientes parámetros:

modificar el profesor asignado a una matrícula. Este método deberá recibir los siguientes parámetros:

- **DNI del profesor actual.** 55555555-E
- **ID del nuevo departamento.** 5

Apartado 5. En la clase que nos permite acceder a los datos deberás crear 1 método que nos permita borrar una relación empleadoDepartamento. Este método recibe como parámetros los siguientes:

- **DNI del profesor.** 22222222-B
- **ID del departamento.** 3

En la clase principal.

En la clase que va a controlar la ejecución el orden que debes seguir a la hora de invocar a los métodos de la clase de acceso a los datos es el siguiente:

1. Llama a los 3 métodos del apartado 1.
2. Llama al método del apartado 2.
3. Llama al método del apartado 3.
4. Llama al método del apartado 4.
5. Llama al método del apartado 5.
6. Llama a los 3 métodos del apartado 1.

Criterios de puntuación. Total 10 puntos.

Ejercicio 1

(3 puntos).

- **Apartado 1.**
 - Establecer que no empiece por "D" (0.25 puntos).
 - Establecer que no acabe por "a" (0.25 puntos).
 - Mostrar correctamente la salida con el nombre (no el número) del departamento. (0.5 puntos)
- **Apartado 2.**
 - Modificar los datos correctamente (0.5 puntos).
 - Mostrar correctamente los datos cuando se pide. (0.5 puntos).
- **Apartado 3.**
 - Borrar correctamente los datos pedidos (0.75 puntos).
 - Mostrar correctamente la salida. (0.25 puntos).

Ejercicio 2 (4 puntos).

- **Apartados del 1 al 10.**
 - Realizar la consulta correctamente (0.3 puntos/cada apartado).
 - Mostrar los datos correctamente por pantalla (0.1 puntos/cada apartado).

Ejercicio 3 (3 puntos).

- **Apartados del 1 al 5.**
 - Crear el método correctamente (0.5 puntos/cada apartado).
 - Correcta llamada al método desde la clase principal (0.1 puntos/cada apartado).