

# Tarea para AD05.

Detalles de la tarea de esta unidad.

## Enunciado.

### EJERCICIO 1

Se trata de hacer una pequeña base de datos en Db4o para guardar información de películas de cine y sus directores.

Del director hay que almacenar: nombre y nacionalidad.

De las películas hay que almacenar: título, director y duración.

Almacena al menos 6 películas y 3 directores, de manera que haya directores con con más de una película y varias películas con duración en un rango, por ejemplo 100 y 125 minutos.

Realiza las siguientes operaciones en este orden: (utiliza al menos dos tipos diferentes de consulta soportados por la base de objetos):

- 1. Consulta todas las películas.
- 2. Consulta las películas de un director por nombre.
- 3. Consulta las películas con duración entre 2 valores.
- 4. Elimina una película por título.
- 5. Modifica la duración de una película.

Se trata de utilizar los tipos estructurados de PostgreSQL desde una interfaz gráfica basada en Swing, para gestionar los datos de una oferta inmobiliaria de viviendas nuevas. Estos datos se almacenarán en la base de datos de PostgreSQL, anaconda, creada al principio del tema. La interfaz gráfica donde tendrás que integrar la estructura de datos que te pedimos, la puedes encontrar aquí.

[Interfaz de la estructura. \(link: AD05 Recurso Tarea.zip\)](#) (0.50 MB)

Descomprime el fichero, y abre el proyecto PostgreTarea con el NetBeans. Dentro del PaquetePrincipal encontrarás un JFrame denominado jfMain (el formulario principal de la aplicación). Ábrelo con doble clic para mostrar los controles swing que componen su interfaz gráfica:

A continuación pulsa en el botón Fuente para ver el código, y entonces:

- ✓ Introduce tu contraseña para el usuario postgres en la variable passwd.

Echa una ojeada al constructor del formulario jfMain(). Debajo tienes una llamada al método drop\_Ejemplo(), que lo único que hace es borrar todo lo que va a crearse (si es que existe) para evitar conflictos.

Continúa hasta el método crear\_Tipos(). Es este método tienes que escribir una consulta SQL que realice las siguientes acciones:

- ✓ Crear el tipo estructurado vivienda, a partir de un entero llamado planta, un entero llamado metros\_2, un entero llamado num\_habitaciones, un entero llamado num\_bannios, y un varchar(25) denominado promotor.
- ✓ Crear el tipo estructurado direccion, a partir de un varchar(40) llamado calle, un varchar(25) llamado ciudad, y un varchar(5) llamado codigo\_postal.
- ✓ Crear la función publicidad sobre el tipo estructurado vivienda, para que devuelva un varchar con el mensaje publicitario compuesto 'Se vende piso de ... metros cuadrados en planta ...', con ... habitaciones y ... baños'.

Luego pasa al método crear\_Tabla(), y escribe una consulta SQL para:

- ✓ Crear la tabla viviendas\_nuevas, con un campo serial denominado vivienda\_id como clave principal, un campo vivienda denominado características, y un campo direccion denominado ubicacion.

A continuación, introduce una consulta SQL en el método insertar\_Registros() para:

- ✓ Insertar al menos 4 registros en la tabla 'viviendas\_nuevas', sin dejar ningún valor vacío en una columna (excepto naturalmente, la columna auto-numérica vivienda\_id) o tipo estructurado. Emplea la función ROW() para construir los valores de los tipos estructurados.

En el método rellenar\_JComboBox() que rellena la lista del cuadro combinado, escribe una consulta SQL para:

- ✓ Obtener todas las ciudades insertadas en la columna ubicación como parte del tipo dirección, ordenadas por nombre.

Después pasa al método rellenar\_JTable() que rellena la cuadrícula a partir de la conexión con la base de datos y la provincia selecciona en el cuadro combinado, y completa una consulta SQL para:

- ✓ Obtener todos las columnas de las viviendas nuevas ubicadas en la provincia seleccionada.

Finalmente completa la sentencia SQL del método `mensajePublicitario()`, para:

- ✓ Consultar el valor de la función `publicidad` sobre la columna `características`, de la vivienda nueva correspondiente a la clave principal especificada como parámetro del método.

#### Criterios de puntuación. Total 10 puntos.

Total 10 puntos.

- ✓ **Ejercicio 1.** 5 puntos.
- ✓ **Ejercicio 2.** 5 puntos.

Se tendrá en cuenta que:

- ✓ El funcionamiento correcto de los programas que se piden.
- ✓ El uso adecuado del API del sistema gestor y/o conectores.
- ✓ Tratamiento adecuado de posibles excepciones.

#### Recursos necesarios para realizar la Tarea.

**Ejercicio 1.**- Los contenidos y ejemplos realizados en la unidad.

**Ejercicio 2.**- Además de los contenidos y ejemplos de la unidad, te proporcionamos el archivo `AD05_Recurso_Tarea.zip` con la interfaz gráfica basada en Swing que debes utilizar.

#### Consejos y recomendaciones.

Para el **Ejercicio 1.**- Ninguno en particular.

Para el **Ejercicio 2.**- Sigue los pasos indicados en el enunciado. Solo debes ir completando las sentencias SQL de los comandos `Statment` indicados.

#### Indicaciones de entrega.

Una vez realizada la tarea elaborarás un único documento donde figuren las respuestas correspondientes. El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas:

**apellido1\_apellido2\_nombre\_SIGxx\_Tarea**

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna **Begoña Sánchez Mañas para la quinta unidad del MP de AD**, debería nombrar esta tarea como...

**sanchez\_manas\_begona\_AD05\_Tarea**