



# Linkia FP

Formación Profesional Oficial a Distancia

DAM – M06 – Clase 07

# Acceso a datos

BDOO y BDOR

CLASE

# Contenido del módulo

- UF2: Manejo de conectores.
  - JDBC (mySQL).
  - ORM (Hibernate).
  - BDOO.
  - BDOR.

Temas 2, 3 y 4

6 Clases

2 Actividades

# Plan de trabajo

## Plan de trabajo - Curso 2021/2022

### DAM - M06: Acceso a datos

UF	FECHAS	TEMAS	Recursos complementarios: Video clases	EVALUACIONES				EXAMEN	
				ACTIVIDADES DE EVALUACIÓN CONTÍNUA					
				Entregas o participaciones 70%	Clases 10%*	Test 20%			
UF1: Persistencia en ficheros	02/02/2022 - 14/02/2022	Tema 1. Persistencia en ficheros	VC01	Actividad 1	C01 C02	Test UF1	Examen UF1		
	14/02/2022 - 28/02/2022	Tema 2. Persistencia en BDR con JDBC	VC02		C03 C04	Test UF2	Examen UF2		
	28/02/2022 - 14/03/2022	Tema 3. Persistencia BDR con ORM	VC03	Actividad 2	C05 C06				
	14/03/2022 - 28/03/2022	Tema 4. Persistencia en BDOO – BDOR	VC04	Actividad 3	C07 C08				
UF3: Persistencia en BD Nativas XML	28/03/2022 - 19/04/2022	Tema 5. Persistencia en BBDD nativas XML	VC05	Actividad 4	C09 C10 C11	Test UF3	Examen UF3		
UF4: Componentes de acceso a datos	19/04/2022 - 02/05/2022	Tema 6. Componentes da acceso a datos	VC06	Actividad 5	C12 C13	Test UF4	Examen UF4		
	02/05/2022 - 16/05/2022				C14				

# Horario de las clases

NÚMERO CLASE	DÍA DE LA SEMANA	FECHA	HORA INICIO
C01	MARTES	08/02/2022	15:15
C02	VIERNES	11/02/2022	14:30
C03	VIERNES	18/02/2022	14:30
C04	VIERNES	25/02/2022	14:30
C05	VIERNES	04/03/2022	14:30
C06	VIERNES	11/03/2022	14:30
C07	VIERNES	18/03/2022	14:30
C08	VIERNES	25/03/2022	14:30
C09	VIERNES	01/04/2022	14:30
C10	MARTES	05/04/2022	15:15
C11	VIERNES	08/04/2022	14:30
C12	VIERNES	22/04/2022	14:30
C13	VIERNES	29/04/2022	14:30
C14	VIERNES	06/05/2022	14:30

# Contenido clase

## UF2: Manejo de conectores.

- Características de las BBDD OO.
- Sistemas gestores de BBDD OO.
- Interfaz de programación de aplicaciones de la BBDD.

CLASE

# Características de las BD00



# El modelo de datos ODMG

- Entre muchas otras especificaciones el estándar ODMG define el Modelo de Objetos que debe ser soportador por el SGBD-OO.
- El lenguaje de BBDD es especificado mediante:
  - Un lenguaje de Definición de Objetos (ODL) que se corresponde con el DDL de los SGBD relacionales.
  - Un Lenguaje de Manipulación de Objetos (OML).
  - Un Lenguaje de Consulta (OQL) que equivale a SQL.

# El modelo de datos ODMG

- Un objeto es una instancia de una entidad de interés del mundo real.
- Los objetos necesitan un identificador único (Identificador de Objeto [OID]).
- Un literal es un valor específico. Los literales no tienen identificadores.

CLASE

# El modelo de datos ODMG

- Un literal no tiene que ser necesariamente un solo valor, puede ser una estructura o un conjunto de valores relacionados que se guardan bajo un solo nombre (por ejemplo, enumeraciones).

CLASE

# El modelo de datos ODMG

- El estado de los objetos se define por los valores que tienen para un conjunto de propiedades.
- Las propiedades pueden ser:
  - Atributos. Los atributos toman literales por valores y son accedidos por operaciones del tipo `get_value` y `set_value` (como exige la OO pura, y nunca accedemos a ellos directamente).

CLASE

# El modelo de datos ODMG

- Las propiedades pueden ser:
  - Relaciones entre el objeto y uno o más objetos. Son propiedades que se definen entre tipos de objetos, no entre instancias. Las relaciones pueden ser uno-a-uno, uno-a-muchos o muchos-a-muchos.

CLASE

# ODL

- ODL (Lenguaje de Definición de Objetos) es un lenguaje para definir la especificación de los tipos de objetos en sistemas compatibles con ODMG.
- Una característica importante que debe cumplir un ODL, es ofrecer al diseñador de BBDD un sistema de tipos semejante a los de programación OO.

CLASE

# Ejemplo ODL

```
CREATE CLASS Autor  
(  
    nombre String,  
    apellidos String,  
    edad Integer  
)
```

# OML

- ODMG no define ningún lenguaje de manipulación de objetos (OML), deja descansar esta tarea en los propios lenguajes de programación.

# OQL

- OQL – Lenguaje de Consultas de Objetos.  
Es un lenguaje declarativo de tipo SQL que permite realizar consultas sobre las BBDD OO, incluyendo primitivas de alto nivel para conjuntos de objetos y estructuras.
- Ejemplos:

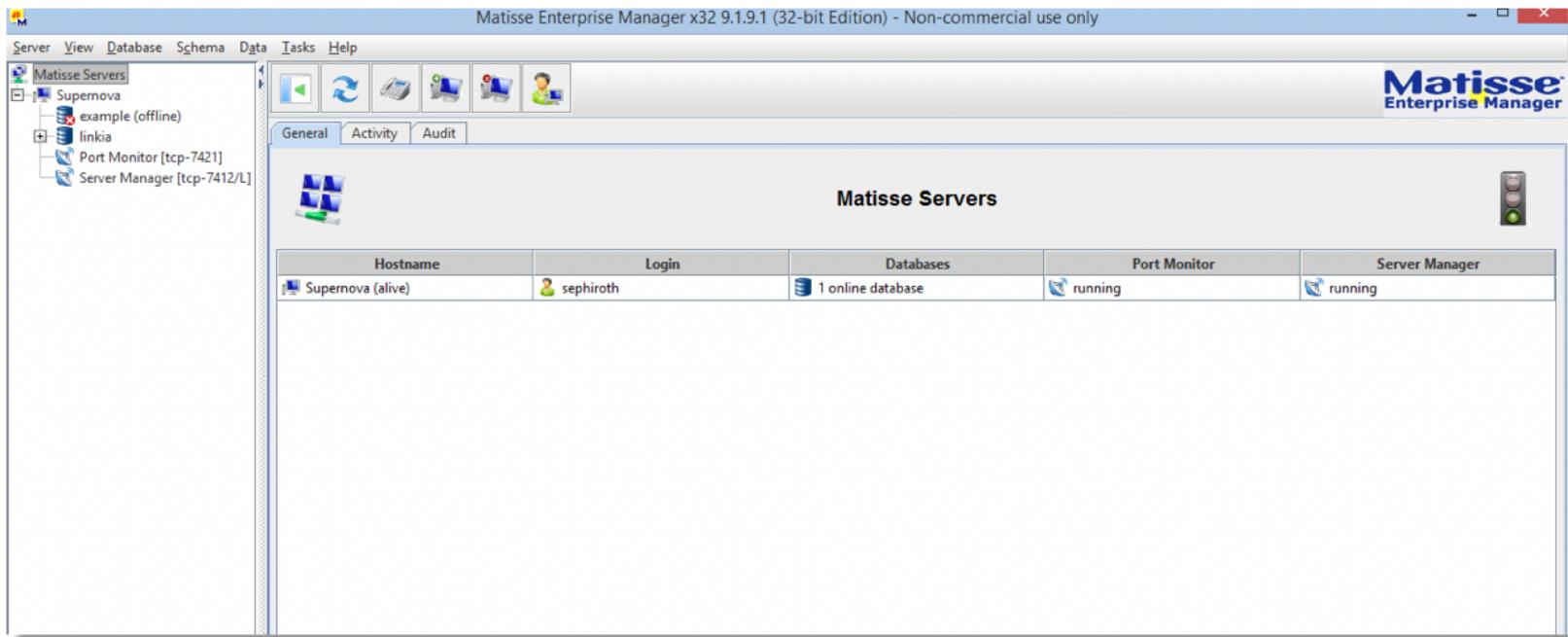
```
select a.titulo from Articulo a where a.paginas>30;  
select distinct a.nombre, a.apellidos from Autor a  
where apellidos like 'Mura%';
```

# SGBO-OO

- Matisse es una alternativa que respeta en gran medida el estándar ODMG.
- Matisse tiene las ventajas para la gestión propias de la OO:
  - Técnicas para fragmentar objetos grandes.
  - Una ubicación optimizada de los objetos.
  - Mecanismo automático de duplicación.
  - Mecanismo de versiones de objetos incorporado.
  - Soporte para una arquitectura cliente-servidor.
  - Optimización de acceso a objetos relacionados.

# Matisse

- Matisse se puede descargar desde [www.fresher.com](http://www.fresher.com)



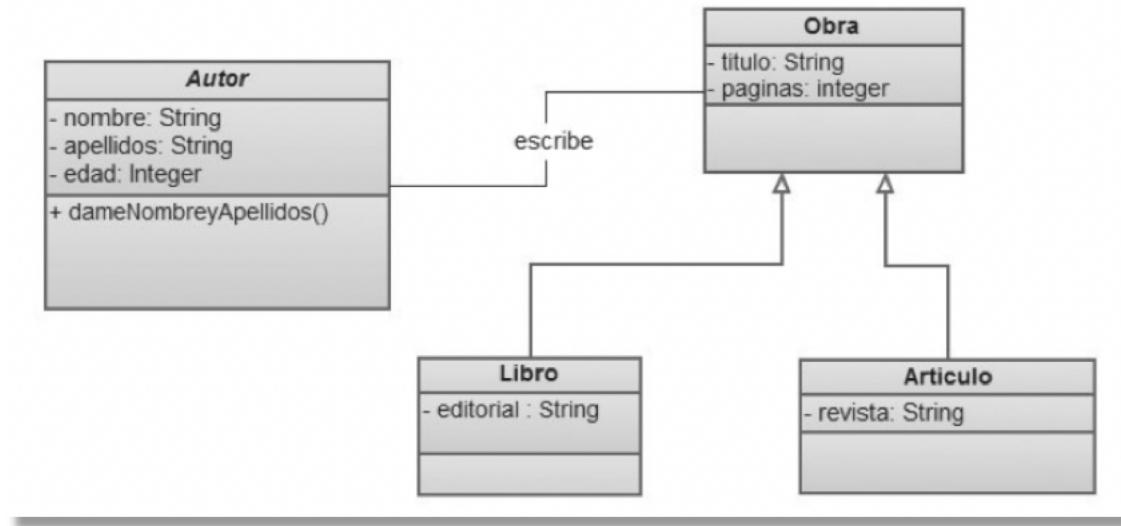
# Matisse

- El árbol de la izquierda muestra las BBDD creadas. Desplegando, podemos acceder a los elementos.
- La parte derecha sirve para ejecutar los lenguajes ODL y OQL sobre las BBDD y mostrar los resultados.

CLASE

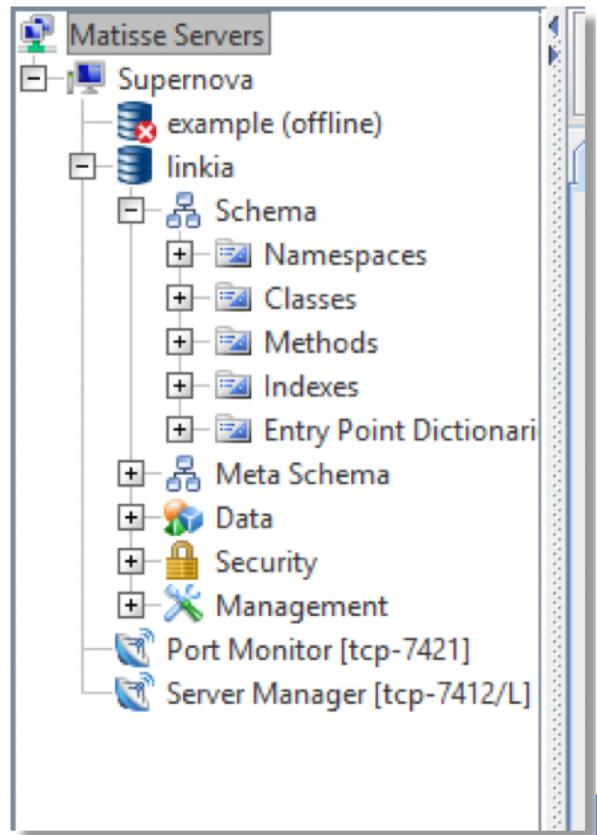
# Matisse

- Considerar un modelo con las siguientes clases:
  - Autor
  - Obra
  - Libro
  - Articulo



# Matisse

- Para crear un Esquema:
  1. Crear la BBDD.
    - Opción Server → New Database
  2. Iniciar la BBDD.
    - Botón derecho → Start



# Matisse

- Para crear un espacio de nombres:
  1. Namespaces → New Namespace.
  2. En la parte derecha, aparecerá una plantilla de ayuda para escribir la sentencia ODL  
*'create namespace'*
  3. Tenemos que ejecutar la sentencia.



# Matisse

- Para crear una clase:
  1. Classes → New Class.
  2. En la parte derecha, aparecerá una plantilla de ayuda para escribir la sentencia ODL.
  3. Tenemos que seleccionar el namespace donde queremos crear la clase en el desplegable '*In namespace*'. Si no lo hacemos, la clase se creará en root.

CLASE

# Matisse

Query Objects

In snapshot Latest Version In namespace biblioteca

SQL statement

```
-- Create Class Specifying inheritance, relationships and constraints
-- Right click in the editor to select the possible attribute types
-- When your class has been successfully created,
-- Click Refresh from the Classes Node to see it in the tree.
--
```

```
CREATE CLASS Autor
(
    nombre String,
    apellidos String,
    edad String
)
```

# Matisse

- Para crear relaciones entre clases:
  1. Alter class → Add Relationship.
  2. En la parte derecha, aparecerá una plantilla de ayuda para escribir la sentencia ODL.
  3. Tenemos que crear una relación entre dos clases y luego la relación inversa entre esas dos mismas clases.

CLASE

# Matisse

```
--  
-- Update Class: adding relationships  
--  
-- When your class has been successfully updated,  
-- Click Refresh from the Class Node to see your newly created  
-- relationships.  
--
```

```
ALTER CLASS Autor  
ADD RELATIONSHIP escribe  
RELATIONSHIP SET (Obra)  
INVERSE Obra.escrito_por;
```

```
--  
-- Update Class: adding relationships  
--  
-- When your class has been successfully updated,  
-- Click Refresh from the Class Node to see your newly created  
-- relationships.  
--  
ALTER CLASS Obra  
ADD RELATIONSHIP escrito_por  
RELATIONSHIP SET (Autor)  
INVERSE Autor.escribe;
```

# Matisse

- Para crear métodos:
  1. Alter class → Add Method.

```
CREATE METHOD darNombreyApellidos ()  
RETURNS String  
FOR Autor  
--  
-- Describe your method here  
--  
BEGIN  
    RETURN concat (nombre, apellidos);  
END;
```

# Interfaz de programación de aplicaciones de la BBDD

- Una vez hemos creado el esquema de la BBDD, el siguiente paso es preparar el sistema para que pueda ser accedido desde código Java.
- La única manera de añadir, eliminar, modificar y consultar objetos en el esquema creado es usando código fuente.

CLASE

# Preparando el código Java

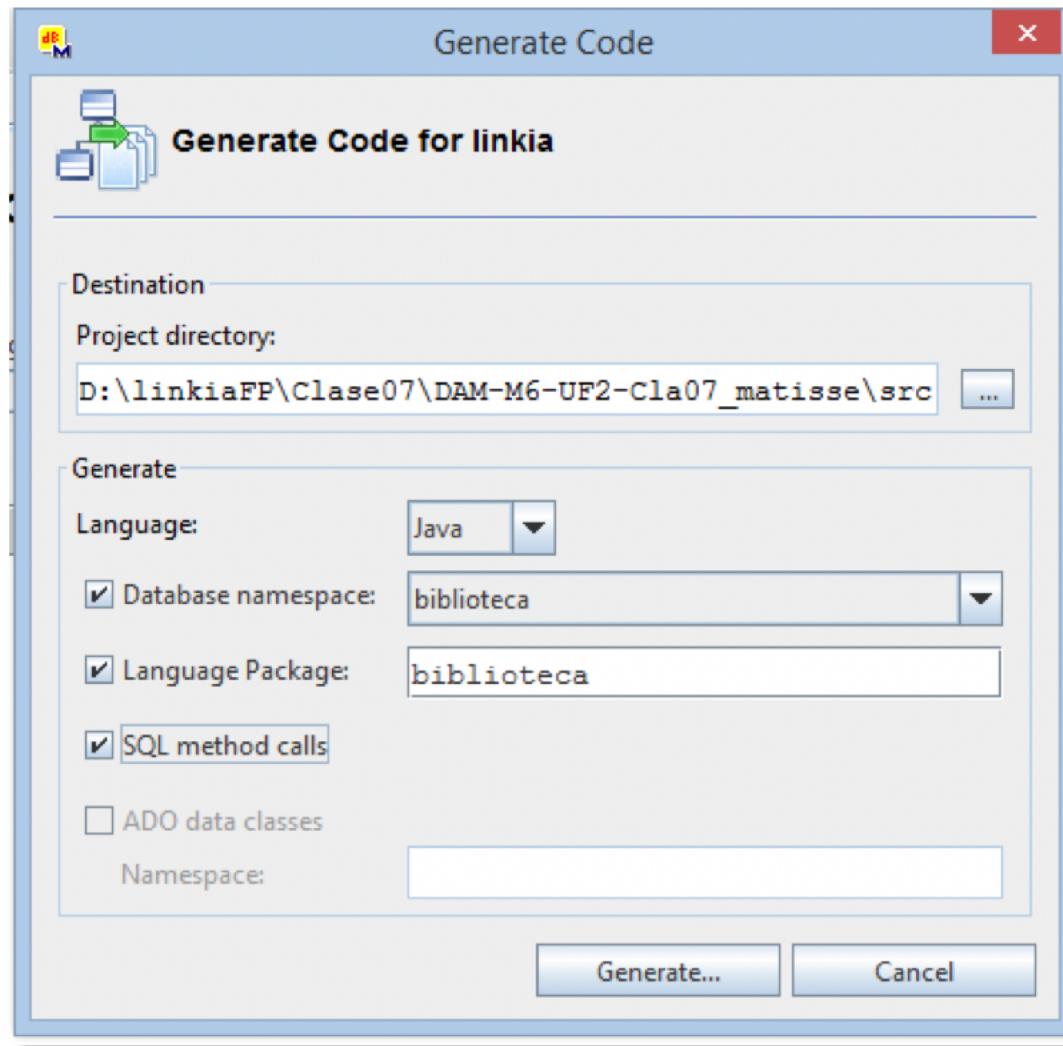
- Para poder entender el proceso para acceder a los objetos almacenados en la BBDD OO, tenemos que tener siempre presente que lo que buscamos es tener la sensación de trabajar sólo con objetos.
- El mecanismo por el cual el contenido de un objeto se almacena en la BBDD es totalmente transparente al programador.

CLASE

# Preparando el código Java

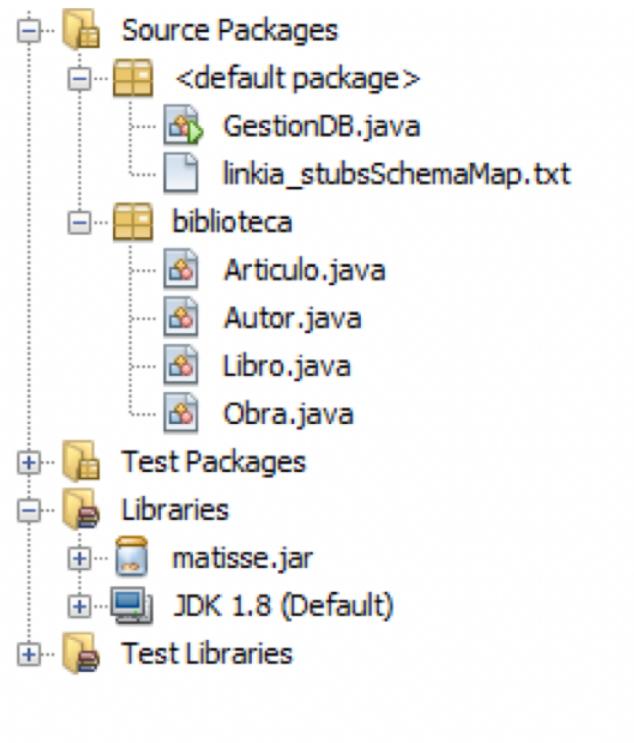
- Para crear la estructura de clases que representan el esquema creado en Matisse:
  1. Schema → Generate Code.
  2. Tenemos que llenar la ventana con los datos para la generación de clases.
  3. Utilizamos Project directory para indicar dónde está ubicada la carpeta del proyecto Java en el que integraremos el código generado.

# Preparando el código Java



# Preparando el código Java

- Una vez generadas todas las clases, todas se ubicarán en el paquete seleccionado.



# Preparando el código Java

- Para que el proyecto reconozca las clases y métodos creados por Matisse, es necesario incluir en el proyecto la librería matisse.jar.
  - En Windows (C:\Products\Matisse\lib)
- Además, tenemos que añadir los siguientes archivos a la carpeta del proyecto:
  - matisseJAVA.dll
  - matisseJAVA.lib

CLASE

# Añadiendo objetos

- Una vez creadas las clases, la forma de añadir objetos a la BBDD es sencilla, sólo tenemos que crear en Java objetos y luego llamarlos para almacenarlos.
- El objeto necesario para la persistencia es:
  - MtDatabase()
    - open()
    - startTransaction()
    - commit()
    - close()
  - MtException

CLASE

## Eliminando objetos

- El borrado de objetos lo hacemos utilizando el método `deepRemove()` definido en todos los objetos de Matisse.

# Consultando objetos con OQL

- La última de las operaciones básicas sobre una BBDD es la ejecución de consultas.
- Las consultas OQL también las podemos ejecutar directamente sobre el propio entorno de Matisse.

CLASE

# Matisse

- Una vez hemos ejecutado el código Java, en Matisse podemos ver el resultado de la ejecución del código.
- Sobre la clase de la que queremos ver los objetos → View data.



OID	titulo	paginas	editorial	escrito_por
0x10a6 Libro	Baila Baila Baila	512	TusQuests	0x10a5
0x10a7 Libro	Tokio Blues	498	TusQuests	0x10a5
0x10a9 Libro	Baila Baila Baila	512	TusQuests	0x10a8
0x10aa Libro	Tokio Blues	498	TusQuests	0x10a8

## Actividad 03

Base de datos con BDOR-BDOO.

- Fecha de entrega: 28/03/2022
- Proyecto creado en NetBeans  
comprimido en .zip

CLASE



# Linkia FP

Formación Profesional Oficial a Distancia