

**Urbano de Jesús Flores Zaragoza**

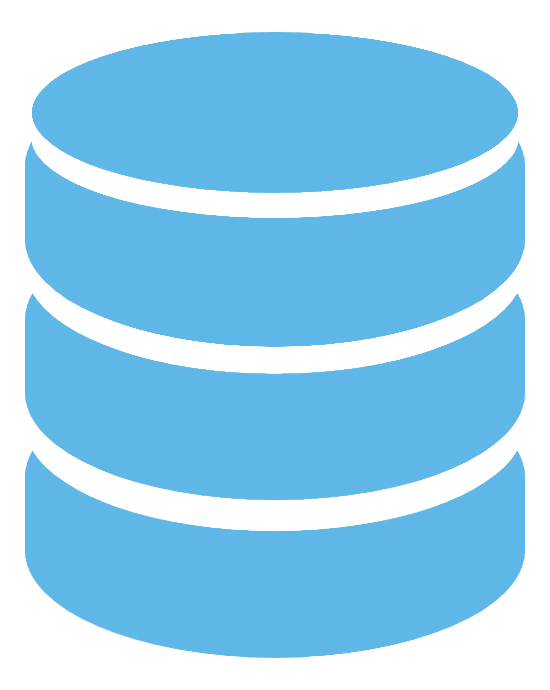


TABLA DE CONTENIDO

Objetivos: 3

Antecedentes. 4

Alternaticas. 5

Desarrollo del trabajo. 6

Conclusiones. 7

# Objetivos:

* **El objetivo principal es ver la diferencia entre el modelo relacional y el modelo orientado a objetos, verlo en una comparación y saber como se comportan**
* **Hacer una investigación sobre las bases de datos orientadas a objetos**
* **Saber como se implementan estas bases y como funcionan sus ventajas y desventajas**
* **Incorporar la base de datos con un lenguaje de programación y a través de el interactuar con el mediante consultas SQL y con los objetos de la base de datos**

# Antecedentes.

**¿Qué es una Base de datos?**

Una base de datos es una colección organizada de información estructurada, o datos, típicamente almacenados electrónicamente en un sistema de computadora. Una base de datos es usualmente controlada por un sistema de gestión de base de datos (DBMS). En conjunto, los datos y el DBMS, junto con las aplicaciones que están asociados con ellos, se conocen como un sistema de base de datos, que a menudo se reducen a solo base de datos.

Los datos dentro de los tipos más comunes de bases de datos en funcionamiento hoy en día se modelan típicamente en filas y columnas en una serie de tablas para que el procesamiento y la consulta de datos sean eficientes. Luego se puede acceder, administrar, modificar, actualizar, controlar y organizar fácilmente los datos. La mayoría de las bases de datos utilizan lenguaje de consulta estructurado (SQL) para escribir y consultar datos.

SQL es un lenguaje de programación usado por casi todas las bases de datos relacionales para consultar, manipular y definir datos, y para proporcionar control de acceso. SQL se desarrolló por primera vez en IBM en la década de 1970 con Oracle como uno de los principales contribuyentes, lo que llevó a la implementación del estándar ANSI de SQL. SQL ha generado muchas extensiones por parte de compañías como IBM, Oracle y Microsoft.

Hay muchos tipos diferentes de bases de datos. La mejor base de datos para una organización específica depende de cómo la organización pretende utilizar los datos.

**Bases de datos relacionales**. Las bases de datos relacionales se popularizaron en los años ochenta. Los elementos de una base de datos relacional se organizan como un conjunto de tablas con columnas y filas. La tecnología de base de datos relacional proporciona la manera más eficiente y flexible de acceder a información estructurada.

**Bases de datos orientadas a objetos**. La información en una base de datos orientada a objetos se representa en forma de objetos, como en la programación orientada a objetos.

**¿Qué son las bases de datos orientadas a objetos?**

El modelo de base de datos orientada a objetos agrupa la información en paquetes relacionados entre sí: los datos de cada registro se combinan en un solo objeto, con todos sus atributos. De esta manera, toda la información está disponible en el objeto, ya que sus datos quedan agrupados en lugar de distribuidos en diferentes tablas. En los objetos no solo pueden guardarse los atributos, sino también los métodos, lo que refleja la afinidad de estas bases de datos con los lenguajes de programación orientados a objetos: al igual que en estos, cada objeto presenta un conjunto de acciones que pueden llevarse a cabo.

**Historia**

Las bases de datos orientadas a objetos, fue un tema que se pensó, que revolucionaría la manera de hacer persistente la información en los sistemas software durante los años 90. En la actualidad es evidente que esto no fue así. Sin embargo, un resurgimiento de este concepto, gracias a las comunidades de software libre, y la identificación de aplicaciones idóneas para el mismo, motivan la revisión de las características de esta alternativa a las omnipresentes bases de datos relacionales. Las bases de datos orientadas a objetos se crearon para tratar de satisfacer las necesidades de estas nuevas aplicaciones.

La orientación a objetos ofrece flexibilidad para manejar algunos o de estos requisitos y no están limitadas por los tipos de datos y los lenguajes de consulta de los sistemas de bases de datos tradicionales. Una característica clave de las bases de datos orientadas a objetos es la potencia que proporcionan al diseñador al permitirle especificar tanto la estructura de objetos complejos, como las operaciones que se pueden aplicar sobre dichos objetos. Otro motivo para la creación de las bases de datos orientadas a objetos es el creciente uso de los lenguajes orientados a objetos para desarrollar aplicaciones.

**Características**

Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

· **Encapsulación** - Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.

· **Herencia** - Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.

· **Polimorfismo** - Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

En una base de datos objeto relacional los dominios de dicha base de datos ya no son sólo atómicos por esta razón no cumplen la 1FN debido a que las tuplas también pueden ser una relación, que llevará a la creación de una relación de relaciones es así que no se puede aplicar el concepto de normalización. Esto porque ni siquiera se puede aplicar la primera forma normal y como consecuencia ni la segunda, ni tercera etc.

De este modo, se genera la posibilidad de guardar objetos más complejos en una sola tabla con referencias a otras relaciones, con lo que se acerca más al paradigma de programación orientada a objetos.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamenteLa orientación a objetos representa el mundo real y resuelve problemas a través de objetos, ya sean tangibles o digitales. Este paradigma tecnológico considera un sistema como una entidad dinámica formada de componentes. Un sistema sólo se define por sus componentes y la manera en que éstos interactúan.

Esto implica almacenar los valores de atributos de un objeto con la transparencia necesaria para que el desarrollador de aplicaciones no tenga que implementar ningún mecanismo distinto al mismo lenguaje de programación orientado a objetos.

Estas bases de datos mezclan las potencialidades de una base de datos de programación orientada a objetos con diferentes lenguajes de programación. Estas se coordinan muy bien con Delphi, Phyton, Ruby, JavaScript, Java, Visual BASIC, NET, C++, entre otros.

**Un tipo de objetos representa a una entidad del mundo real. Un tipo de objetos se compone de los siguientes elementos:**

* Un **nombre** que sirve para identificar el tipo de los objetos.
* Unos **atributos** que modelan la estructura y los valores de los datos de ese tipo. Cada atributo puede ser de un tipo de datos básico o de un tipo de usuario.
* Unos **métodos** que son procedimientos o funciones escritos en el lenguaje PL/SQL (almacenados en la base de datos).

**Diagrama

Descripción generada automáticamente**

**Uso actual de la base de datos orientada a objetos**

La base de datos orientada a objetos ha encontrado su nicho de aplicación en áreas tales como la ingeniería, las bases de datos espaciales, las telecomunicaciones y en áreas científicas como la física de alta energía y la biología molecular.

Otro conjunto de bases de datos orientadas a objetos se centra en el uso integrado en dispositivos, paquetes de software y sistemas en tiempo real.

# Desarrollo del trabajo.

Primero elegí las entidades cliente, ciudad y estado del modelo relacional que ya teníamos hecho y para empezar a trabajar primero cree los tipos de objeto

Tabla

Descripción generada automáticamente

Como observamos los tipos objetos en ellos pueden tener otro objeto dentro de el

Como el tipo ciudad tiene en el un atributo de tipo estado\_typ.

Una vez creados los tipos creamos las tablas con esos objetos

Texto

Descripción generada automáticamente

Las tablas son creadas a partir de los objetos definidos anteriormente, en la creación de las tablas ponemos la llave primaria respectivamente a cada tabla y para la relación se establece una llave foránea pero que esta contenida dentro de un objeto

Después de crear las tablas ya podemos ir insertando datos dentro de ellas para poder insertar datos en una tabla que tiene objetos se utiliza la siguiente sentencia SQL.

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

*Este el modelo que me quedo*

Una vez que tengamos las tablas con datos podemos hacer una conexión con un lenguaje de programación orientado a objetos en mi caso yo trabaje con **Java.**

Para poder hacer una conexión con java necesitamos la librería JDBC la 6 o 7, yo utilice la 6 esta la podemos descargar desde la página oficial de Java.

Imagen que contiene Texto

Descripción generada automáticamente

En la clase main importamos las librarías necesarias y agregamos el JDBC6 declaramos variables para establecer la conexión donde JUNO es mi usuario, bca la clave y le damos la ruta de la base de datos.

Creamos unas clases para nuestro proyecto para poder hacer un formulario con el cual haremos un CRUD (Create, Read, Update, Delete) de los registros de nuestra base de datos en la tabla **cliente\_tb.**

Creamos también un JFrame Form y en el ponemos todos los controles, textbox y botones que necesitaremos

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Para el botón agregar puse el siguiente código

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteUna cosa muy importante es que al montar la sentencia INSERT no definimos los valores de la sentencia. En su lugar ponemos interrogaciones (?) que serán sustituidas posteriormente por valores. Le daremos los valores por medio de la variable psta junto con los jtxt.getText() que yo los enumere de acuerdo con el orden que tengo en la base de datos.



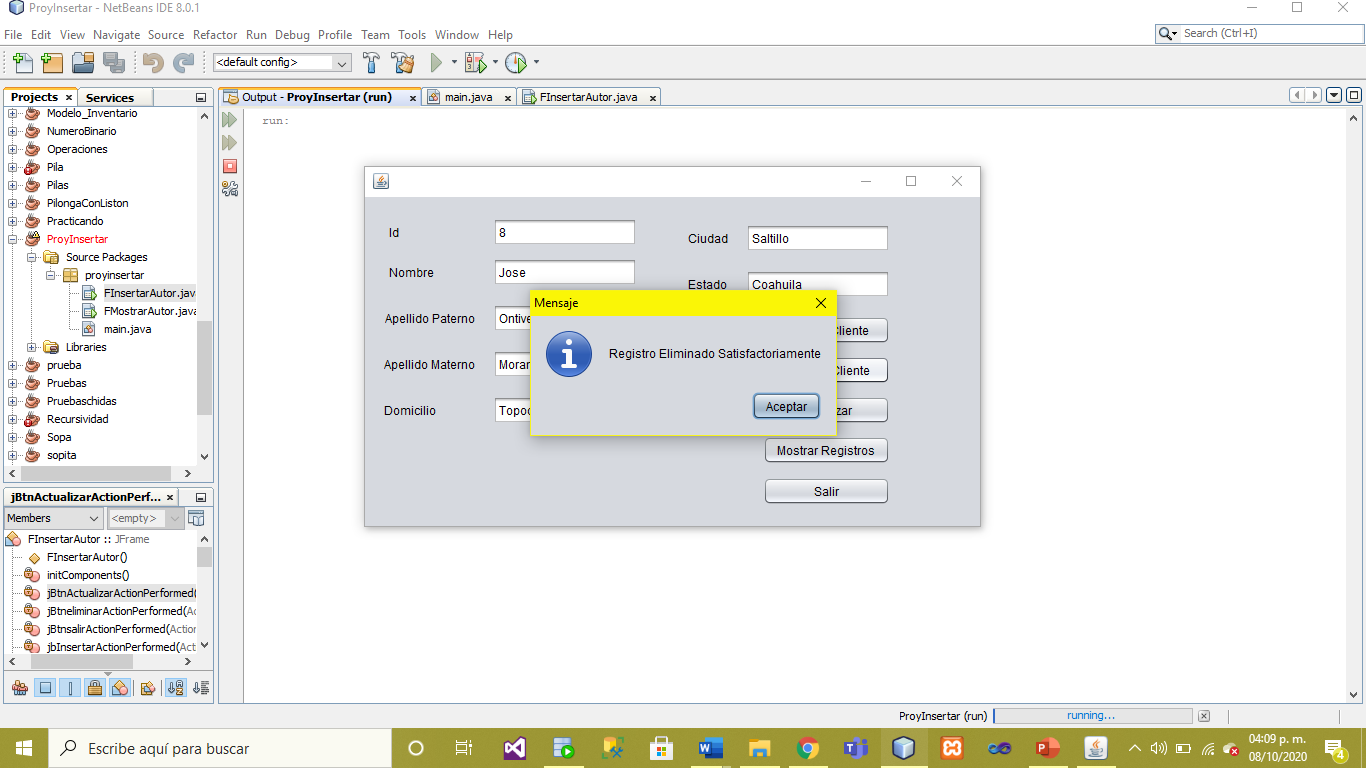
Para el botón Eliminar

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto

Descripción generada automáticamente

El id que contenga el jtxt1 será el cliente que se eliminara



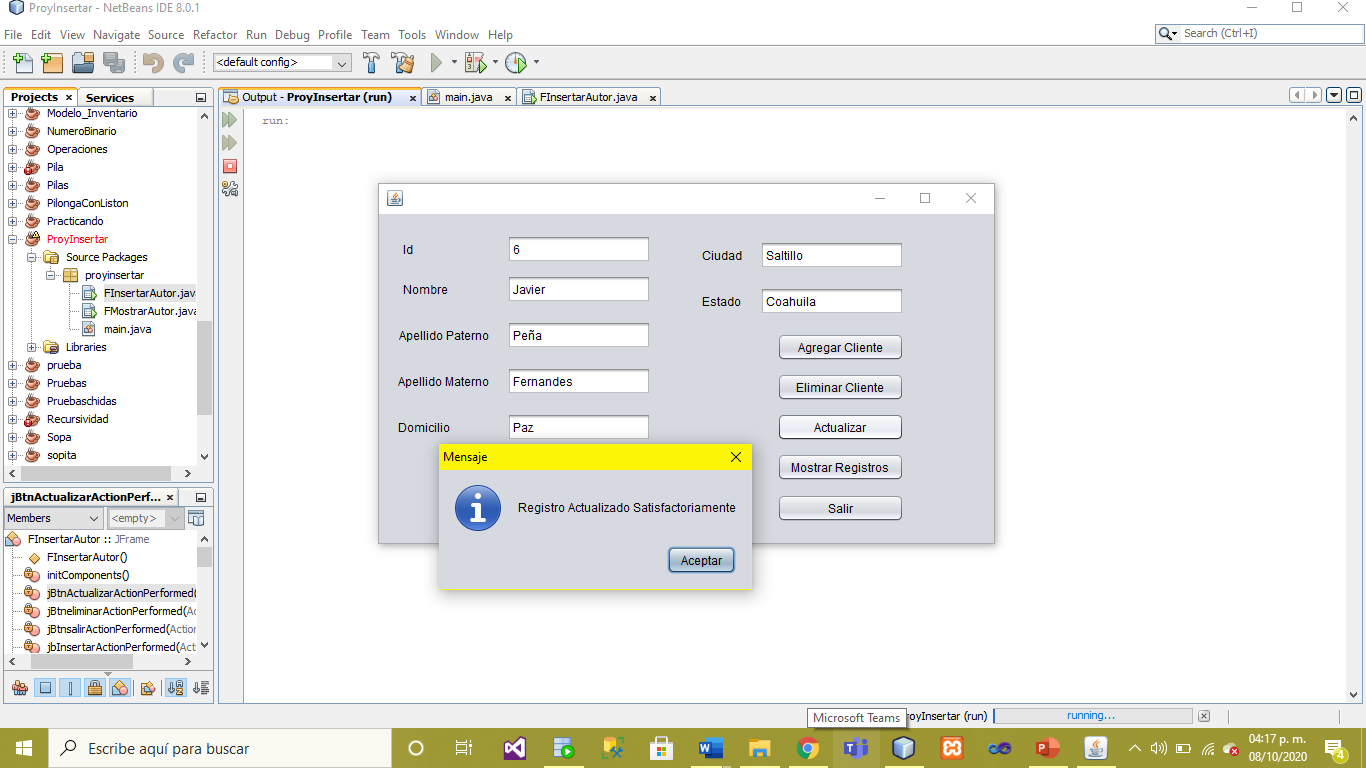
Para el botón de actualizar

En el set de los atributos que son objetos se debe especificar el nombre del objeto y dentro de ellos se debe poner un (?).

Actualizaremos el registro 6

Tabla

Descripción generada automáticamente





Para el botón mostrar

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Para poder mostrar los datos creamos otro JFrame Form y agrumaos una tabla que contendrá los datos de la tabla cliente\_tb.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

También dentro de los objetos podemos crear métodos dentro del cliente\_typ hacemos un método para que nos devuelva el id\_cliente para poder usarlo en la tabla cliente\_tb

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

# Comparativa con el modelo ER.

**Orientada a objetos**

* Encapsulamiento
* Soporte de objetos complejos
* Herencia
* Soporte de un lenguaje de programación orientado a objetos
* Polimorfismo
* Identificador de objeto

**Modelo relacional**

Una base de datos relacional es un conjunto finito de relaciones junto con una serie de restricciones o reglas de integridad.

* Independencia de Datos
* Coherencia de Resultados: Acciones únicas , se evita inconsistencia
* Cumplimiento de ciertas normas
* Restricciones de seguridad
* Compartir datos: Se comparte con todos los usuarios que estén autorizados.
* Mejora en la seguridad y en la integridad de datos
* Mejora en la productividad : Proporciona muchas de las funciones estándar

**Ventajas de las Bases de datos orientadas a objetos**

* Admite cualquier tipo de datos.
* Oculta información entre un objeto y el resto para evitar conflictos o accesos incorrectos.
* Le permite al diseñador especificar tanto la estructura de objetos complejos, como las operaciones que se pueden aplicar sobre dichos objetos.

**Lenguajes de Consulta**

Entre los aspectos más positivos que poseen dichas bases de datos se encuentra su lenguaje de consulta. Estas implementan un sistema declarativo de permiten encontrar y recuperar los objetos de la base de datos. Como ya mencionamos, estas bases de datos son capaces de funcionar con varios tipos de lenguajes de programación conocidos.

**Persistencia transparente**

Esta es una característica muy importante. Se refiere a la posibilidad de la base de datos de utilizar uno o varios objetos a través del lenguaje de programación para obtener análisis o manipulación de la data.

**Transacciones ACID**

Estas bases de datos tienen un alto nivel de confiabilidad en la ejecución de transacciones ACID (atomicidad, consistencia, aislamiento y durabilidad). Esto significa que las transacciones se cumplen o terminan únicamente cuando no existen conflictos con los datos. También los usuarios pueden abortar las acciones en caso de que se presente algún error en el sistema de la base de datos.

**Almacenamiento en cache**

Las bases de datos orientadas a objetos tienen un mecanismo de almacenamiento en cache que crea replicas parciales de la base de datos. El almacenamiento en caché es un método de almacenamiento y resguardo de datos que permite minimizar el acceso al disco o el flujo de tráfico de la red. Esto permite que las aplicaciones accedan a objetos desde la memoria del programa en lugar del utilizar el almacenamiento en disco.

También es importante mencionar que estas bases de datos tienen amplias capacidades para recuperación de datos, manejo complejo de altos volúmenes de datos. Sus características le proporcionan gran rendimiento y un manejo mínimo de código.

**Desventajas de las bases de datos orientadas a objetos**

* La optimización de consultas suele comprometer la encapsulación.
* La mayoría de los modelos carecen de una base teórica.
* No hay ningún modelo universalmente aceptado por los SGBD orientados a objetos.

Al igual que diversas bases de datos NoSQL, carecen de madurez en el desarrollo. Esto lleva a un mayor riesgo de sostenibilidad de la base de datos en el largo plazo. No hay criterios claros de estandarización, ya que estas apenas están reemergiendo en el mercado digital.

Al tener relativamente poco tiempo en el mercado, existe poca experiencia y documentación sobre proyectos que las implementen. Esto pudiese extender los tiempos de desarrollo de un proyecto con este modelo de bases de datos.

**Desventajas del modelo relacional**

* Los datos abstractos o no estructurados como los del big data no son admitidos.
* El mantenimiento es muy costoso y complicado cuando la base de datos crece a un gran tamaño.
* Los tiempos de respuesta suelen ralentizarse a medida que la base de datos crece.

**Dificultades de crecimiento**: Cuando estas bases de datos comienzan a crecer en volumen, el almacenamiento y el costo de mantenimiento se convierten en un problema de alto costo.

**Cambios en la estructura**: el entorno empresarial es altamente dinámico. Esto exige que se realicen cambios de forma eventual en los registros de datos. Si ejecutamos cambios, la Base de Datos debe ser modificada en su estructura para admitir las modificaciones. Si las modificaciones no se realizan esta se verá afectada y sus procesos interrumpidos.

**Complejidad en la instalación**: Algunas bases de datos SQL se ven condicionadas por el sistema operativo en el cual van a funcionar y los requisitos mínimos de funcionamiento de los servidores u ordenadores.

**Dificultad en la interfaz:** La interfaz de una base de datos SQL son más complejas que agregar algunas líneas de código.

**Más características implementadas de forma patentada**: Aunque las bases de datos SQL se ajustan a los estándares ANSI e ISO, algunas bases de datos implementan extensiones propietarias al SQL estándar para garantizar el bloqueo del proveedor.

**Marcas reconocidas de bases de datos orientadas a objetos**

En la familia de las bases orientadas a objetos existen diversas marcas. Todas ellas poseen características especiales. Conozcamos a continuación cuales son las más reconocidas y las preferidas por los desarrolladores.

**Object Database ++**

Esta base de datos integrada pensada para alojar aplicaciones en servidores remotos. Es una de las más actualizadas y forma parte de las favoritas de los desarrolladores. Tiene una serie de índices incorporados que permiten un acceso rápido y como a los datos. Permite variedad en las búsquedas básicas B+Tree y en el modo de texto completo. Es ideal para los proyectos en los que debe ejercerse un control exhaustivo sobre las búsquedas.

**ObjectDB**

La base de datos orientada a objetos Object DB es un sistema muy potente y compacto. Brinda la confiabilidad que se requiere en cualquier proyecto y es extremadamente rápido.

Ella proporciona todos los servicios estándar de administración de bases de datos, garantizando las facilidades que necesitamos para implementar aplicaciones. En ObjectDB existen diversas funciones compatibles con las API de JPA y JDO.

Forma

Descripción generada automáticamente

# Conclusiones.

En conclusión con este trabajo pienso que las bases orientadas a objetos son más difíciles de implementar que las relacionales, y también pienso que aun no son muy usadas ya que no hay suficiente documentación sobre ellas. Para mí el modelo relacional es mucho mejor y más fácil de comprender además de que es más usado y por ende ya tiene estándares establecidos, aunque las bases orientadas a objetos pueden ser muy útiles para almacenar datos complejos y para nuevas tecnologías que las van requiriendo.

También con este trabajo entendí un poco sobre el manejo de objetos dentro de la base de datos junto con el lenguaje de programación, con solo hacer unas cuantas sentencias SQL pude notar la complejidad de estas y que no es fácil, al menos para mí comprenderlas del todo.

# Referencias.

<http://lineadecodigo.com/java/insertar-datos-con-jdbc/>

<https://www.ionos.mx/digitalguide/hosting/cuestiones-tecnicas/base-de-datos-orientada-a-objetos/>

<https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/782/mod_resource/content/8/contenido/index.html>

<https://sites.google.com/a/espe.edu.ec/bases-de-datos-ii/introduccion/bdd-objeto-relacional>

<https://www.grapheverywhere.com/bases-de-datos-orientadas-objetos/>

<https://slideplayer.es/slide/1837524/>