



18 DE DICIEMBRE DE 2023


GEMSTONE/S

GemTalk Systems

JOSE MIGUEL GARCÍA NAVARRO

ACCESO A DATIS

2º DAM



Contenido

1.	Presentación	2
2.	Casos de uso común:.....	2
2.1.	Aplicaciones Empresariales:	2
2.2.	Sistemas de Misión Crítica:.....	2
2.3.	Entornos de Desarrollo Multilenguaje:	2
2.4.	Gestión de Datos Complejos:.....	2
2.5.	Aplicaciones que Requieren Persistencia Transparente:	2
3.	Características:	3
3.1.	Modelo Orientado a Objetos:	3
3.2.	Persistencia Transparente:	3
3.3.	Escalabilidad Horizontal:	3
3.4.	Transacciones ACID:.....	3
3.5.	Desarrollo Multilenguaje:	3
3.6.	Gestión de Versiones:	3
4.	Ventajas	3
4.1.	Modelo Orientado a Objetos:	3
4.2.	Persistencia Transparente:	3
4.3.	Escalabilidad y Rendimiento:	3
4.4.	Gestión de Versiones:	3
4.5.	Durabilidad y confiabilidad:	3
5.	Desventajas	4
5.1.	Curva de Aprendizaje:	4
5.2.	Adopción Limitada:	4
5.3.	Dependencia de la Experiencia:	4
5.4.	Compatibilidad de Lenguajes:	4
6.	Integración con lenguajes de programación.....	5
6.1.	Código de ejemplo:.....	5
6.2.	Capturas del software descargado:	8

1. Presentación

GemStone/S es un sistema de gestión de bases de datos orientado a objetos (ODBMS, por sus siglas en inglés) que se utiliza para almacenar y gestionar datos en entornos empresariales.

Es una solución que adopta un enfoque orientado a objetos para el almacenamiento y recuperación de datos. En lugar de utilizar el modelo tradicional de bases de datos relacionales, donde la información se organiza en tablas y filas, almacena y gestiona datos en forma de objetos.

Cada objeto puede contener datos y lógica de negocio relacionada, proporcionando una representación más natural de las estructuras de datos en el ámbito de la programación orientada a objetos.

Se utiliza en entornos empresariales que requieren persistencia eficiente de objetos y gestionan datos complejos.

2. Casos de uso común:

2.1. Aplicaciones Empresariales:

GemStone/S es adecuado para aplicaciones empresariales que involucran una gran cantidad de datos interrelacionados y que se benefician de un modelo orientado a objetos.

2.2. Sistemas de Misión Crítica:

Debido a su capacidad para garantizar la integridad de los datos y ofrecer transacciones ACID, GemStone/S se utiliza en sistemas donde la fiabilidad y la consistencia de los datos son críticas.

2.3. Entornos de Desarrollo Multilenguaje:

GemStone/S es compatible con varios lenguajes de programación, lo que permite a los desarrolladores trabajar en el entorno que mejor se adapte a sus necesidades, ya sea Java, Smalltalk, o incluso otros.

2.4. Gestión de Datos Complejos:

En situaciones donde la estructura de los datos es compleja y está en constante evolución, GemStone/S brinda soluciones para la gestión de versiones y la migración de esquemas.

2.5. Aplicaciones que Requieren Persistencia Transparente:

GemStone/S proporciona persistencia transparente, facilitando la gestión de objetos y eliminando la necesidad de que los desarrolladores se ocupen directamente de la lógica de almacenamiento.

3. Características:

3.1. Modelo Orientado a Objetos:

GemStone/S utiliza un modelo orientado a objetos para representar y gestionar datos. Esto facilita la representación natural de la lógica de negocio en las aplicaciones.

3.2. Persistencia Transparente:

Ofrece persistencia transparente para objetos, lo que significa que los objetos pueden ser almacenados en la base de datos de manera automática sin la intervención del desarrollador.

3.3. Escalabilidad Horizontal:

Es altamente escalable y puede distribuirse horizontalmente, permitiendo manejar grandes volúmenes de datos y proporcionar un rendimiento eficiente.

3.4. Transacciones ACID:

Proporciona soporte para transacciones ACID, garantizando la integridad de los datos incluso en situaciones de error.

3.5. Desarrollo Multilenguaje:

Facilita el desarrollo en varios lenguajes de programación, incluyendo Java, Smalltalk y otros, proporcionando flexibilidad a los desarrolladores.

3.6. Gestión de Versiones:

Destaca por su capacidad para gestionar eficientemente la evolución de esquemas de objetos y la migración de datos, lo que es crucial en entornos empresariales en constante cambio.

4. Ventajas

4.1. Modelo Orientado a Objetos:

Facilita el desarrollo de aplicaciones alineado con el paradigma de programación orientada a objetos.

4.2. Persistencia Transparente:

Simplifica la gestión de la persistencia al almacenar objetos de manera automática.

4.3. Escalabilidad y Rendimiento:

Ofrece escalabilidad horizontal y rendimiento eficiente, adaptándose a las necesidades de entornos empresariales.

4.4. Gestión de Versiones:

Proporciona una solución robusta para la gestión de versiones y la evolución de esquemas de datos.

4.5. Durabilidad y confiabilidad:

Utiliza técnicas de replicación y distribución para garantizar la durabilidad y confiabilidad de los datos almacenados.

5. Desventajas

5.1. Curva de Aprendizaje:

Puede haber una curva de aprendizaje para los desarrolladores, especialmente aquellos acostumbrados a modelos de bases de datos relacionales.

5.2. Adopción Limitada:

En comparación con algunas otras soluciones, GemStone/S puede tener una adopción más limitada en la industria.

5.3. Dependencia de la Experiencia:

La efectividad de GemStone/S puede depender en parte de la experiencia y comprensión del equipo de desarrollo en el uso de bases de datos orientadas a objetos.

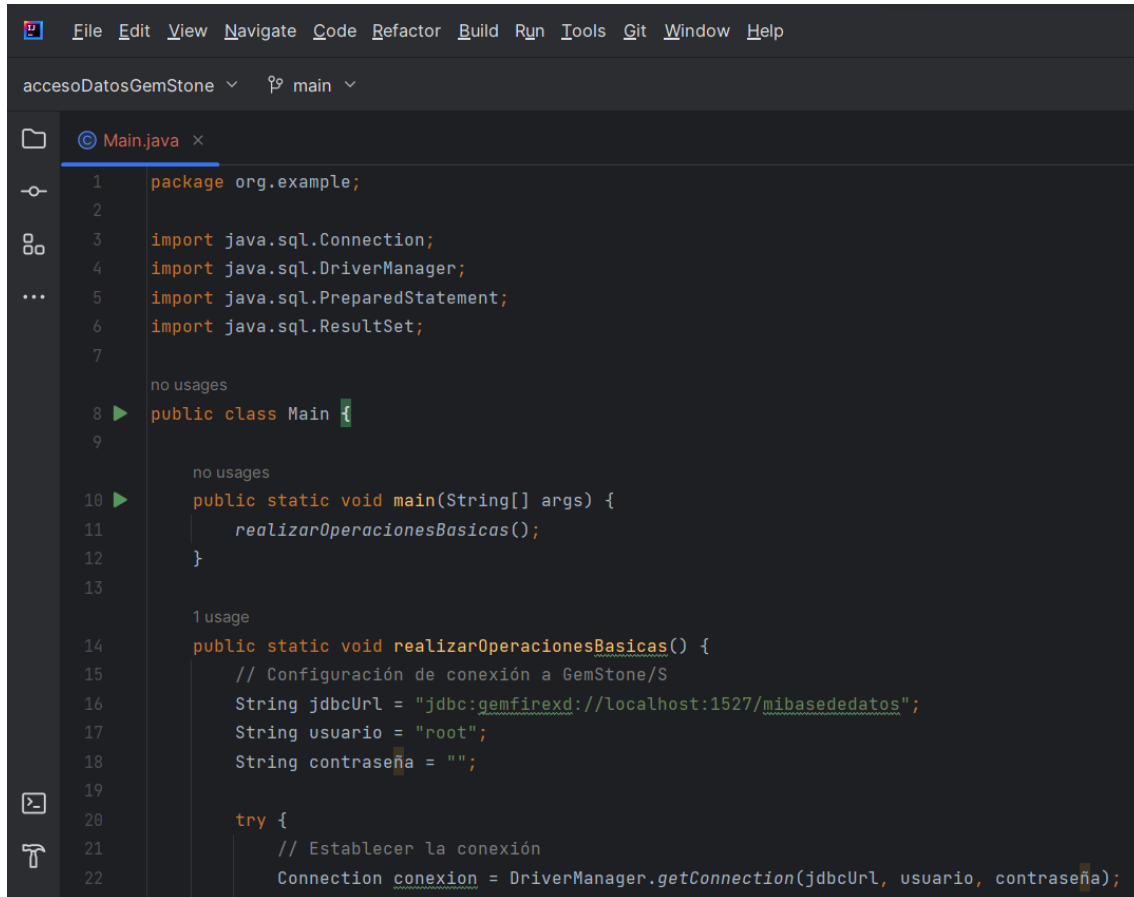
5.4. Compatibilidad de Lenguajes:

Aunque admite múltiples lenguajes, algunas características específicas pueden ser más pronunciadas en ciertos entornos de programación.

6. Integración con lenguajes de programación.

La integración de GemStone/S con Java implica el uso de controladores JDBC (Java Database Connectivity) para establecer conexiones y realizar operaciones en la base de datos.

6.1. Código de ejemplo:

A screenshot of an IDE window showing a Java file named Main.java. The code is for a package org.example and includes imports for java.sql.Connection, DriverManager, PreparedStatement, and ResultSet. It defines a public class Main with a main method that calls realizarOperacionesBasicas(). This method sets up a JDBC connection to GemStone/S using jdbc:gemfirexd://localhost:1527/mibasededatos, with user 'root' and an empty password. The code is as follows:

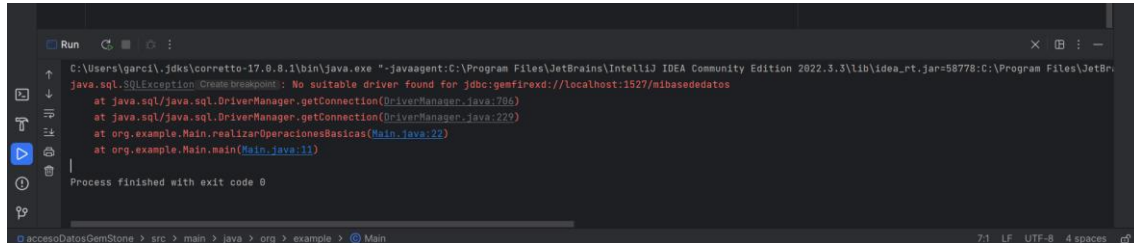
```
1 package org.example;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7
8 no usages
9 public class Main {
10
11     no usages
12     public static void main(String[] args) {
13         realizarOperacionesBasicas();
14     }
15
16     1 usage
17     public static void realizarOperacionesBasicas() {
18         // Configuración de conexión a GemStone/S
19         String jdbcUrl = "jdbc:gemfirexd://localhost:1527/mibasededatos";
20         String usuario = "root";
21         String contraseña = "";
22
23         try {
24             // Establecer la conexión
25             Connection conexion = DriverManager.getConnection(jdbcUrl, usuario, contraseña);
```

```
File Edit View Navigate Code Refactor Build Run Tools Git Window Help
accesoDatosGemStone main
Main.java x
23
24 // Crear una tabla y realizar operaciones básicas
25 crearTabla(conexion);
26 insertarDatos(conexion);
27 recuperarDatos(conexion);
28
29 // Cerrar la conexión
30 conexion.close();
31
32 } catch (Exception e) {
33     e.printStackTrace();
34 }
35 }
36
1 usage
37 @ private static void crearTabla(Connection conexion) throws Exception {
38     // Crear una tabla
39     String sqlCrearTabla = "CREATE TABLE mitabla (id INT, nombre VARCHAR(255))";
40     try (PreparedStatement declaracion = conexion.prepareStatement(sqlCrearTabla)) {
41         declaracion.execute();
42     }
43 }
44
```

```
File Edit View Navigate Code Refactor Build Run Tools Git Window Help
accesoDatosGemStone main
Main.java x
45 @ private static void insertarDatos(Connection conexion) throws Exception {
46     // Insertar datos
47     String sqlInsertarDatos = "INSERT INTO mitabla VALUES (?, ?)";
48     try (PreparedStatement declaracion = conexion.prepareStatement(sqlInsertarDatos)) {
49         declaracion.setInt( parameterIndex: 1, x: 1);
50         declaracion.setString( parameterIndex: 2, x: "Ejemplo");
51         declaracion.execute();
52     }
53 }
54
1 usage
55 private static void recuperarDatos(Connection conexion) throws Exception {
56     // Recuperar datos
57     String sqlRecuperarDatos = "SELECT * FROM mitabla";
58     try (PreparedStatement declaracion = conexion.prepareStatement(sqlRecuperarDatos);
59         ResultSet conjuntoResultados = declaracion.executeQuery()) {
60
61         while (conjuntoResultados.next()) {
62             int id = conjuntoResultados.getInt( columnLabel: "id");
63             String nombre = conjuntoResultados.getString( columnLabel: "nombre");
64             System.out.println("ID: " + id + ", Nombre: " + nombre);
65         }
66     }
67 }
68 }
```

La librería necesaria para hacer este código funcionar sería “Gemfirexd” proporcionada por la web de GemTalk Systems para GemStone/s la cual no ha sido encontrada y no se ha podido hacer el programa funcional.

Al ejecutar el código nos dice que no se encuentra el driver necesario.



```
C:\Users\garcia\jdk\corretto-17.0.8.1\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.3.3\lib\idea_rt.jar=58778:C:\Program Files\JetBr...
java.sql.SQLException: No suitable driver found for jdbc:gemfirexd://localhost:1527/mibasededatos
    at java.sql/java.sql.DriverManager.getConnection(DriverManager.java:706)
    at java.sql/java.sql.DriverManager.getConnection(DriverManager.java:229)
    at org.example.Main.realizarOperacionesBasicas(Main.java:22)
    at org.example.Main.main(Main.java:11)
Process finished with exit code 0
```

El código proporcionado es un ejemplo de cómo podríamos insertar datos en las tablas de la y realizar una búsqueda básica con una consulta simple SQL.











Aparte la base de datos no ha sido creada puesto que el software necesario para la creación de la base de datos proporcionado por GemTalk Systems es complejo de usar y no he encontrado mucha documentación ni ningún tutorial.

Download

*Before downloading, you must agree to the
[Community Edition License Terms](#), or the terms of
your GemTalk Systems contract license agreement.*

product download	checksums		
3.7 Linux/x86_64	SHA256	SHA1	MD5
3.7 Linux/arm64	SHA256	SHA1	MD5
3.7 Linux RPM/x86_64	SHA256	SHA1	MD5
3.7 AIX	SHA256	SHA1	MD5
3.7 Mac/x86_64	SHA256	SHA1	MD5
3.7 Mac/silicon	SHA256	SHA1	MD5
3.7 Windows Client	SHA256	SHA1	MD5

6.2. Capturas del software descargado:

Nombre	Fecha de modificación	Tipo	Tamaño
 lz4.exe	25/08/2023 2:17	Aplicación	163 KB
 openssl.exe	25/08/2023 2:17	Aplicación	705 KB
 topaz.exe	25/08/2023 2:17	Aplicación	10 KB
 vsd.exe	28/06/2023 1:53	Aplicación	3.505 KB
 lz4.statmon.dict.bin	25/08/2023 2:17	Archivo BIN	64 KB
 topaz.ico	25/08/2023 2:17	Archivo ICO	1 KB
 libgcirpc-3.7.0-64.pdb	25/08/2023 2:17	Archivo PDB	5.668 KB
 libgcits-3.7.0-64.pdb	25/08/2023 2:17	Archivo PDB	4.436 KB
 libssl-3.7.0-64.pdb	25/08/2023 2:17	Archivo PDB	17.204 KB
 openssl.pdb	25/08/2023 2:17	Archivo PDB	3.004 KB

```
C:\Users\lgarci\OneDrive\Docu > + ~
-----
GemStone/S64 Object-Oriented Data Management System
Copyright (C) GemTalk Systems 1986-2023
All rights reserved.
-----
PROGRAM: topaz, Linear GemStone Interface (Remote Session)
VERSION: 3.7.0, Thu Aug 24 14:24:31 2023 (branch 3.7.0)
COMMIT: 2023-08-24T14:12:58-07:00 87d4894f48fb88b6589e392e200ea1a60e1a143b
BUILT FOR: Microsoft Windows x86-64
RUNNING ON: 16-CPU jmgm: x86_64, Windows NT 10.0 build 22621
PROCESSOR: 16-core AMD Ryzen 7 5700U with Radeon Graphics (K8)
MEMORY: 12288 MB
PROCESS ID: 10404    DATE: 01/09/2024 12:53:50 Hora estándar romance
-----
neither topazini.tpz nor $HOME\topazini.tpz were found
topaz> |
```

