Título: Benchmark para Sistemas Gestores de Bases de Datos Relacionales

Director: Antonio César Gómez Lora

Alumno: Jose Antonio Jamilena Daza

# Introducción

Los sistemas gestores de bases de datos relacionales son los sistemas de almacenamiento de datos más usados en la actualidad para grandes volúmenes de datos, ya que la complejidad de acceso y escritura bastante liviana en comparación con los tradicionales archivos secuenciales, tanto binarios como texto.

Estos sistemas software están orientados a permanecer en equipos servidores durante periodos de tiempo muy largos, meses o años, sin que la maquina se apague. Esto hace que el software albergado en el equipo servidor se vayan degradando en su funcionamiento, en especial el sistema operativo y el sistema gestor de bases de datos.

Este proyecto es desarrollar un software que elabore estadísticas sobre la salud del sistema gestor de bases de datos, las muestre al usuario y las compare con los comportamientos para dichos lotes de sentencias.

# Objetivos

El objetivo del proyecto es elaborar básicamente un conjunto de lotes de scripts de procesado que se lancen sobre el sistema gestor de bases de datos de forma asidua y que obtenga estadísticas de comportamiento normal del sistema gestor de bases de datos. De forma que el usuario a partir de las muestras obtenidas conozco la salud del susodicho sistema y pueda apreciar el grado del deterioro del mismo para, que si fuera necesario, actué con las herramientas propias de dicho sistema gestor de bases de datos para suavizar dicha degradación.

Para ello se desarrollará además del conjunto de scripts mencionados, un servicio de sistema para que con una temporización adecuada lance dichos scripts según la programación dada.

También una aplicación de escritorio que reciba los datos del servicio indicado y muestre gráficas sobre dichos datos comparados con los resultados normales y los actuales en el momento de solicitud de análisis.

# Metodologías

Para el desarrollo del proyecto se empleará el paradigma de programación orientado a objetos y el Lenguaje Unificado de Modelado (UML,  Unified Modeling Language)  para modelar el software.

El modelo de desarrollo será el Desarrollo Iterativo Incremental (1).

# Fases de Trabajo

1. Análisis de requisitos.

En esta fase se realiza una prospección inicial del sistema y se marcan objetivos y métodos de desarrollo.

1. Especificación.

Elección de las herramientas de desarrollo. Esbozar los componentes principales del sistema software.

1. Arquitectura.

Diseño de los distintos componentes del sistema software. Diseño de la jerarquía de clases de los distintos componentes del sistema.

1. Implementación, pruebas y documentación
   1. Implementación. Desarrollo de los objetos que forman la arquitectura.
   2. Pruebas. Pruebas de funcionamiento.
   3. Documentación. Documentación del sistema software.
2. Implantación y mantenimiento.
   1. Implantación. Implantar el software en el cliente.
   2. Mantenimiento. Mejorar la solución periódicamente según un proceso de incidencias.

# Herramientas de desarrollo

El software se desarrollará con las siguientes herramientas:

* Java. JDK 1.6.0\_14-b08: Lenguaje de programación orientado a objetos en su última versión (2).
* Netbeans 6.5: Entorno integrado de desarrollo para Java (3).
* Subversion: Sistema de control de versiones. Se emplea para mantener históricos de ficheros durante los desarrollos software (4).
* TortoiseSVN: Cliente de Subversión (5).
* Enterprise Architect y MySQL: Herramienta de desarrollo UML que almacenará los diagramas en una base de datos MySQL (6) (7).
* SQLite: Sistema gestor de bases de datos integrado en el proyecto (8).
* SQLiteJDBC: Biblioteca de acceso a ficheros SQLite para Java (9).
* SQLite Management Studio. Herramienta para gestión gráfica de ficheros SQLite (10).
* Apache Log4J. Sistema de diario de ejecución para aplicaciones Java (11).
* JFreeChart. Sistema de generación de graficas para el entorno grafico Swing de Java (12).

# Bibliografía

1. **Wikipedia.** Iterative and incremental development. [En línea] http://en.wikipedia.org/wiki/Iterative\_development.

2. **Sun Microsystems.** Java™ Platform, Standard Edition 6. [En línea] http://java.sun.com/javase/6/docs/api/.

3. **Sun Microsystem.** Netbeans. [En línea] http://www.netbeans.org/.

4. **Visualsvn.** Visualsvn. [En línea] http://www.visualsvn.com/server/.

5. **TortoiseSVN.** TortoiseSVN The coolest Interface to (Sub)Version Control. [En línea] http://tortoisesvn.net/.

6. **Sparx Systems.** Enterprise Architect - UML for Business, Software and Systems. [En línea] http://www.sparxsystems.com.au/.

7. **Sun Microsystems.** MySQL - The world's most popular open source database. [En línea] http://www.mysql.com/.

8. **SQLite.** SQLite - Small. Fast. Reliable. [En línea] http://www.sqlite.org/.

9. **SQLiteJDBC.** SQLiteJDBC. [En línea] www.zentus.com/sqlitejdbc/.

10. **Osenxpsuite.** SQLite2009 Pro Enterprise Manager. [En línea] http://link.osenxpsuite.net/?uid=homepage&id=sqlite2009pro.zip.

11. **Apache Software Foundation.** Logging Services. [En línea] 1999-2007. http://logging.apache.org/log4j/1.2/index.html.

12. **JFree.org.** JFreeChart. [En línea] http://www.jfree.org/jfreechart/.

13. **Dr. Bert Scalzo, Claudia Fernandez, Donald K. Burleson, Mike Ault, Kevin Kline.** *Database Benchmarking: Practical Methods for Oracle & SQL Server.* s.l. : Rampant TechPress, 2007. ISBN 0977671534, 9780977671533.

14. **Darwin, Ian F.** *Java cookbook.* s.l. : O'Reilly, 2001. ISBN 0596001703, 9780596001704.