Título: Informador sobre la Salud de Sistemas Gestores de Bases de Datos Relacionales

Director: Antonio César Gómez Lora

Alumno: Jose Antonio Jamilena Daza

# Introducción

--- CAMBIAR ---

# Objetivos

Los objetivos de este proyecto final de carrera serian los que siguen:

* Elaborar un software que bajo una planificación dada, ejecute sobre un equipo con un sistema gestor de bases de datos relacionales, unos scripts de prueba y a partir de ellos generar datos sobre el comportamiento, en tiempo, de los mismos.
* Construir una interfaz gráfica que sea capaz de recoger los datos de la herramienta descrita anteriormente, y a partir de las mismas, generar unas estadísticas que han de ser mostradas de forma visual. Para que el usuario las interpreté. Dichas estadísticas se agruparan según los diferentes tipos de scripts lanzados.
* Generar un conjunto de scripts para realizar las evaluaciones. Dicho conjunto de scripts, han de ser factibles de aglutinar en conjuntos no disjuntos de categorías o propiedades. Con esto se consiguen estadísticas según las posibles características evaluar. Además de un sistema de almacenamiento para los mismos de forma dinámica y ágil.

# Metodologías y fases de trabajo

Para el desarrollo del proyecto se empleará el paradigma de programación orientado a objetos y el Lenguaje Unificado de Modelado modelar el software, UML iniciales que corresponden al acrónimo inglés Unified Modeling Language.

El modelo de desarrollo será el Desarrollo Iterativo Incremental [1].

En cuanto a las fases en las que se dividirá el desarrollo software será una prospección inicial del desarrollo en el que barajaremos posibles lenguajes para desarrollar el proyecto, tecnologías a emplear, etc. Así también como la obtención de los objetivos que ha de cumplir el desarrollo software que formará el proyecto. Tras la cual habremos obtenido el análisis de requisitos.

Se pasará a hacer un esbozo de componentes para después pasar a una especificación de los distintos componentes que formarán parte del desarrollo software.

Una vez que tenemos los componentes que debemos implementar pasaremos a una fase de diseño de la arquitectura de la solución. En esta fase del desarrollo se realizará el diseño de los distintos componentes del sistema software, así como el diseño de la jerarquía de clases de los distintos componentes del desarrollo software.

Una vez finalizado esto, pasaremos a la implementación del diseño. En esta fase se puede llegar a tener que hacer cambios en el diseño, dado que la naturaleza del desarrollo así lo exija.

Durante todo el proceso de desarrollo y siguiendo el modelo de desarrollo será el Desarrollo Iterativo Incremental [1], se irá obteniendo prototipos de desarrollo sobre los que se irá haciendo pruebas de funcionamiento y de completitud de requisitos.

Durante todas estas fases se llevará un amplio proceso de documentación, tanto para la escritura de la memoria final como para guardar el conocimiento adquirido durante el desarrollo.

# Herramientas de desarrollo

El desarrollo se hará sobre el lenguaje de programación Java. JDK 1.6.0\_14-b08, la última versión de este leguaje de programación orientado a objetos [2].

El interfaz elegido para el desarrollo será Netbeans 6.5, un entorno integrado de desarrollo especializado en Java [3].

Todo el proyecto se guardará con un sistema de control de versiones llamado Subversion. Este se encarga de mantener históricos de ficheros durante los desarrollos software [4]. Subversión se instala como un software servidor al que se accede mediante un cliente. El cliente elegido es TortoiseSVN, un cliente que se integra tanto en la shell del sistema operativo como en el propio Netbeans [5].

Se evaluará el diseño de la aplicación sobre Enterprise Architect y sobre un complemento específico de Netbeans para estos menesteres. Enterprise Architect es una herramienta CASE, Computer Aided Software Engineering o Ingeniería de Software Asistida por Ordenador, y en el diseño de este proyecto en cuestión se hará que se guarden los datos en un sistema de bases de datos MySQL [6][7]. En cuanto al componente de Netbeans lo almacenará con estructura de archivos de proyecto de Netbeans.

El proyecto tendrá integrada su propia base de datos de estadísticas. Para dicho cometido se empleará SQLite, un sistema gestor de bases de datos integrado en el proyecto, rápido y de poco peso en memoria [8]. Para comunicarse con Java se empleará la librería SQLiteJDBC [9]. Para facilitar el diseño en dicho sistema gestor de bases de datos se empleará la herramienta SQLite Management Studio, una herramienta para gestión gráfica de ficheros SQLite [10].

Para velar por la integridad y conocer inequívocamente los fallos que pueden acaecer, tanto la ejecución normal como en las sesiones de pruebas, se empleará Apache Log4J. Apache Log4J es un sistema de diario de ejecución para aplicaciones Java [11]. Dicho sistema nos da la capacidad de generar ficheros de diario de ejecución con las trazas de la misma e incluso la capacidad de que llegado el caso, si el sistema al que estamos evaluando puede llegar a colapsarse, enviar un email advirtiendo este hecho.

El componente encargado de mostrar las estadísticas de las pruebas se empleara un interfaz gráfica Java implementada con Swing. Como sistema visualizador de los dato emplearemos la biblioteca JFreeChart. JFreeChart es un sistema de generación de graficas para el entorno grafico Swing de Java [12].

Las conexiones entre el proyecto y los diferentes gestores de bases de datos se harán mediante la tecnología JDBC (Java Data Base Connectivity) es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

Toda la planificación temporal del proyecto será llevada acabó con la ayuda de OpenProj [13]. Y la documentación se generará a partir de Enterprise Architect y se editará el texto con OpenOffice [14] y con Microsoft Visio [15] los diagramas que fueran necesarios.

# Bibliografía

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

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

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

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

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

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

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

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

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

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

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

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

[] Serena. OpenProj is a free, open source project management solution. [En línea] http://openproj.org/openproj.

[] Sun Microsystems. OpenOffice - the free and open productivity suite. [En línea] http://www.openoffice.org/.

[] Microsoft Corporation. Microsoft Office Visio 2007. [En línea] http://office.microsoft.com/es-es/visio/FX100487863082.aspx.

[] 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.

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