



# PROYECTO DADO JUSTO

**Grupo Fila 3+1**

Equipo:

Manolo Kabezabolo

El Chivi

Manolo el del Bomo

Mariano Rajau

# ÍNDICE

## [Introducción](#)

### [Contextualización](#)

#### [Situación inicial](#)

#### [Situación de conflicto](#)

#### [Resolución del conflicto](#)

## [Especificación de requisitos](#)

### [Propósito](#)

### [Ámbito](#)

## [Descripción general](#)

### [Perspectiva del proyecto](#)

### [Funciones del producto](#)

## [Requisitos específicos](#)

### [TABLAS](#)

### [CASOS DE USO](#)

### [DIAGRAMA DE CLASES INICIAL](#)

### [CONSTRUCCIÓN DE LA SQL](#)

### [ANÁLISIS FUNCIONAL](#)

### [ESTIMACIÓN DE TIEMPOS](#)

## [Diagrama de clases final](#)

### [Empaquetados](#)

### [Clases](#)

## [Manual de instalación](#)

### [Herramientas necesarias](#)

### [Requisitos](#)

#### [MySQL y MySQL WorkBench](#)

#### [Plataforma JAVA](#)

#### [Eclipse](#)

#### [Subversion o SVN](#)

#### [Google Code](#)

### [Configuración del entorno](#)

#### [MySQL](#)

#### [MySQL WorkBench](#)

#### [Plataforma JAVA](#)

#### [Eclipse](#)

#### [Subversion o SVN](#)

## [Ejecución de la aplicación](#)

## [Bibliografía](#)



# 1. Introducción

Mediante el presente documento se pretende describir el proyecto a desarrollar por el equipo de trabajo espartano “*Fila 3+1*”. El objetivo de este proyecto es dar solución al problema que en el curso de Java JSE/JEE surgió a la hora de realizar las lecturas por parte de los alumnos y/o el profesor (Gran Kahuna).

## 1.1 Contextualización

### *Situación inicial*

Los alumnos han de ser elegidos para la lectura y/o presentación de documentos y tareas varias durante el curso señalado. Hasta el momento, dicha selección se realiza mediante una aplicación básica desarrollada en java al inicio del curso. En caso de que el alumno se repitiera, el profesor decidía qué hacer (repetía alumno, volvía a tirar o seleccionaba bajo criterio propio uno nuevo).

### *Situación de conflicto*

La selección del alumno, que es una tarea totalmente secundaria para el desarrollo del curso, conlleva una sustanciosa pérdida de tiempo. El problema principal radica en que las selecciones se repiten, no son justas y excluye a alumnos.

### *Resolución del conflicto*

Para la correcta y justa selección del alumno, se pretende desarrollar una aplicación en JAVA que dé solución al problema. La idea es crear un sistema basado en niveles, con lo que por cada lectura que realiza cada alumno, se le sube de nivel. Un alumno no será seleccionado nuevamente mientras aún existan alumnos en el nivel anterior.

La salvaguarda de los datos se hará uso de una BBDD MySQL en la cual quedará un fiel reflejo de las selecciones realizadas, con lo que podrá saberse cuántas veces, a que hora y día fue elegido.

## 2. Especificación de requisitos

### 2.1 Propósito

El propósito de la especificación de requisitos es definir cuales son los requerimientos que debe tener la aplicación que se va a desarrollar y describir la funcionalidad del usuario a lo largo de ella.

### 2.2 Ámbito

El desarrollo de la aplicación Java está orientado a ofrecer una funcionalidad que ayude a poder obtener el nombre de la persona que deberá leer en clase. Por tanto tendrá una única funcionalidad, que será la de tirar dado, para que el proceso calcule de la forma más justa quien debe ser.

## 3. Descripción general

### 3.1 Perspectiva del proyecto

La aplicación desarrollada pretende calcular de la forma más justa, el siguiente alumno que deberá leer, responder o corregir el ejercicio en clase, en función de las veces que ya le haya tocado.

### 3.2 Funciones del producto

A continuación se muestra la función principal del producto.

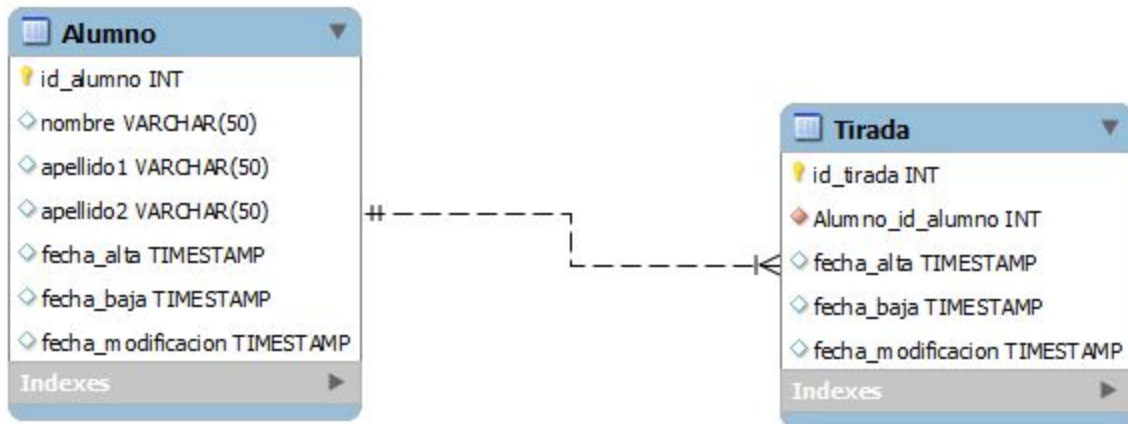
*Profesor o alumno*

- Lanzar dado: Se mostrará el nombre de la persona elegida de forma justa que deberá leer, responder o corregir el ejercicio en clase.

## 4. Requisitos específicos

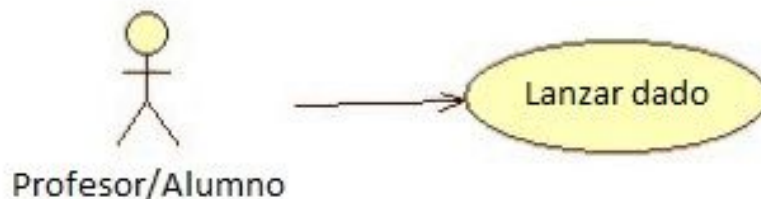
### TABLAS

<u>Alumnos</u>	<u>Tiradas</u>
<ul style="list-style-type: none"> <li>• id_alumno Int(2) : (Primary Key)</li> <li>• nombre : varChar(50)</li> <li>• apellido1 : varChar(50)</li> <li>• apellido2 : varChar(50)</li> <li>• fecha_alta : DateTime</li> <li>• fecha_modificacion : DateTime</li> <li>• fecha_baja : DateTime</li> </ul>	<ul style="list-style-type: none"> <li>• id_tirada Int(4) : (Primary Key)</li> <li>• id_alumno Int(2) : (Foreign Key)</li> <li>• fecha_tirada : DateTime</li> </ul>

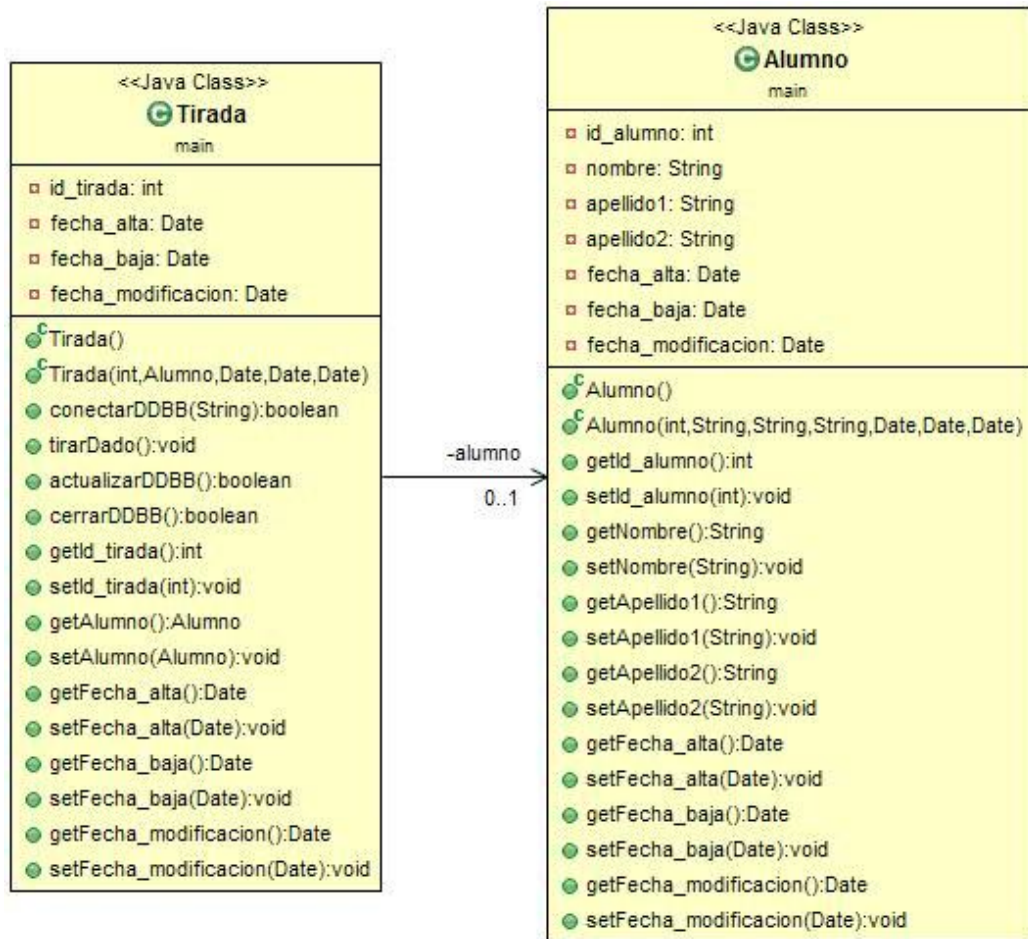


### CASOS DE USO

#### Casos de uso del Profesor/Alumno



## DIAGRAMA DE CLASES INICIAL



## CONSTRUCCIÓN DE LA SQL

- Tirada por niveles: Obtenemos un random de alumnos con todos los que no tengan una cantidad de tiradas igual a la máxima cantidad de tiradas por alumno.

*max(count(id\_tirada)) GROUP BY id\_alumno*: Obtenemos el listado de id de alumno. válidos.

- Comprobar fecha de baja del alumno: Evita obtener alumnos ausentes en el curso (a partir de ahora, caídos en combate).

→ *fecha\_baja NOT null*: Obtenemos el listado de alumnos a excluir.

- Empezar nivel: Si la cantidad de resultados válidos obtenidos en la tirada por niveles es igual a 0 habrá que volver a realizar una tirada con todos los alumnos a excepción de los que tengan la baja.
- Evitar redundancias: En caso de que se complete un nivel, el mismo alumno podría resultar seleccionado dos veces seguidas. Por ello, se buscará quién ha sido el último en ser seleccionado para excluirlo de la tirada (amén de todos lo que tienen la baja).
- El truco: Si no perteneces al equipo 3+1, mejor no leas estas líneas. *El truco consta de una pequeña criba que evitará que el azar elija, por error, a un miembro de integrante de nuestro equipo.*

## ANÁLISIS FUNCIONAL

- **FrontEnd**: El usuario lanzará la aplicación ya implementada, y obtendrá como resultado un alumno de los restantes por leer. Para ello, esta herramienta se organizará por niveles, es decir, todos los alumnos comenzarán con el nivel inicial (0), e irán aumentando de nivel conforme realicen lecturas. Una vez que termine el turno del alumno en cuestión, su nivel aumentará del 0 al 1, o del nivel en el que en ese momento esté X, al siguiente, X+1. Esto hará que en la siguiente tirada, al haber pasado de nivel, no se tenga en cuenta y solo se contemplen los del nivel anterior, nivel X. Una vez todos estén en el mismo nivel, se volvería a comenzar nuevamente desde el principio.
- **BackEnd**: Tenemos que hacer una selección de los alumnos aptos para la lectura, corrección o respuesta de las exigencias del profesor. Para ello tendremos que tener en cuenta los siguientes casos en este orden usando el siguiente algoritmo:

- IF alumno.fecha\_baja == null ⇒ Apto
- IF alumno.fecha\_baja != null ⇒ No Apto
- SELECT TOP 1 id\_alumno FROM tirada ORDER BY fecha\_alta ⇒ Ultimo
- max(count(id\_tirada)) AS lvl\_max GROUP BY id\_alumno ⇒ Obtenemos el nivel más alto
- IF count(id\_tirada) GROUP BY id\_alumno < lvl\_max
  - ◆ Si no hay resultados
    - RANDOM (Apto-Ultimo)
  - ◆ Si hay resultados
    - Los que ha devuelto la consulta



## ESTIMACIÓN DE TIEMPOS

Vamos a realizar una estimación previa de tiempos sin basarnos en métodos de cálculo existentes. Separamos el proyecto en varias fases y a cada una le daremos un “peso” que será representada mediante una variable para luego obtener el costo en tiempo de cada fase.

- **Análisis de requisitos:** Necesidades que el proyecto presenta ante la posibilidad de su uso con las distintas tecnologías elegidas para su desarrollo.
  - Tiempo estimado: 4 horas
- **Definición del proyecto:** Presentación del proyecto por parte del cliente, donde se nos muestran los requerimientos que cree necesitar.
  - Tiempo estimado: 2 horas
- **Reunión:** Se lleva a cabo una presentación en el aula del problema y las posibles implementaciones para diseñar el mismo.
  - Tiempo estimado: 2 horas
- **Captura de requisitos:** Definición de los principales requisitos a priori.
  - Tiempo estimado: 4 horas
- **Estrategia inicial:** Definición de la estrategia a tomar en la implementación del proyecto.
  - Tiempo estimado: 3 horas.
- **Implementación:** Desarrollo de la aplicación del proyecto.
  - Tiempo estimado: 10 horas
- **Documentación:** El presente documento donde se realizará una presentación por escrito y de forma detallada de todo lo acontecido durante el desarrollo del proyecto.
  - Tiempo estimado: Tiempo total del proyecto.
- **Implantación:** Fase en la cual el cliente podrá hacer pruebas.
  - Tiempo estimado: 2 horas.

Tiempo total estimado: 5 días.

Fase 2: Desarrollo del software

Reparto de tareas inicial el 25/11/2013

<i>25/11/2013</i>	Tarea	Tiempos estimados <sup>1</sup>
<b>Ulises</b>	<ul style="list-style-type: none"> <li>• Alumno</li> <li>• Tirada</li> </ul>	1 día
<b>Beñat</b>	<ul style="list-style-type: none"> <li>• Menu</li> </ul>	1 día
<b>Iker</b>	<ul style="list-style-type: none"> <li>• Factory</li> <li>• MySqlFactory</li> <li>• MySqlTiradaDAO</li> </ul>	1 día
<b>Jonan</b>	<ul style="list-style-type: none"> <li>• Interfaces</li> <li>• MySqlAlumnoDAO</li> </ul>	1 día
<b>Julen</b>	<ul style="list-style-type: none"> <li>• Dado</li> <li>• Documentación</li> <li>• Reparto y supervisión de tareas</li> </ul>	4 días

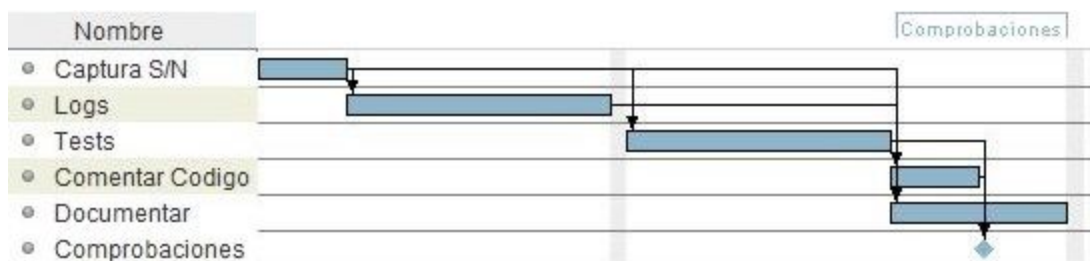
Situación a fecha de 03/12/2013

<i>03/12/2013</i>	Tarea	Situación
<b>Ulises</b>	<ul style="list-style-type: none"> <li>• Alumno</li> <li>• Tirada</li> </ul>	Terminado
<b>Beñat</b>	<ul style="list-style-type: none"> <li>• Menu</li> </ul>	En proceso
<b>Iker</b>	<ul style="list-style-type: none"> <li>• Factroy</li> <li>• MySqlFactory</li> <li>• MySqlTiradaDAO</li> </ul>	Terminado a falta de prueba final
<b>Jonan</b>	<ul style="list-style-type: none"> <li>• Interfaces</li> <li>• MySqlAlumnoDAO</li> </ul>	Terminado a falta de prueba final
<b>Julen</b>	<ul style="list-style-type: none"> <li>• Dado</li> <li>• Documentación</li> <li>• Reparto y supervisión de tareas</li> </ul>	En proceso

- Leyenda: Terminado, A falta de prueba y En proceso

<sup>1</sup> Todos los tiempos estimados pueden variar en función de los requisitos que el proyecto exija en cada situación.

## GANTT del reparto de tareas actual



Situación a fecha de 04/12/2013

04/12/2013	Tarea	Situación
Ulises	<ul style="list-style-type: none"> <li>• Pruebas unitarias</li> <li>• Pruebas funcionales</li> </ul>	<ul style="list-style-type: none"> <li>• En proceso</li> </ul>
Beñat	<ul style="list-style-type: none"> <li>• Menu</li> <li>• Añadir funcionalidades extras</li> </ul>	<ul style="list-style-type: none"> <li>• Terminado</li> <li>• En proceso</li> </ul>
Iker	<ul style="list-style-type: none"> <li>• Factroy</li> <li>• MySqlFactory</li> <li>• MySqlTiradaDAO</li> </ul>	<ul style="list-style-type: none"> <li>• Terminado</li> </ul>
Jonan	<ul style="list-style-type: none"> <li>• logs</li> </ul>	<ul style="list-style-type: none"> <li>• En proceso</li> </ul>
Julen	<ul style="list-style-type: none"> <li>• Documentación</li> <li>• Reparto y supervisión de tareas</li> </ul>	<ul style="list-style-type: none"> <li>• En proceso</li> </ul>

Actualmente el proyecto ha alcanzado la fase de ALPHA, logrando un TAG y una rama de PRE-PRODUCCIÓN fiable y estable. Esto conlleva, debido a la buena planificación temporal calculada al inicio, a un repaso general y a la consideración de poder añadir nuevas funcionalidades extras así como la reconsideración de mejoras en la implementación al proyecto tal como se muestra en la tabla superior.

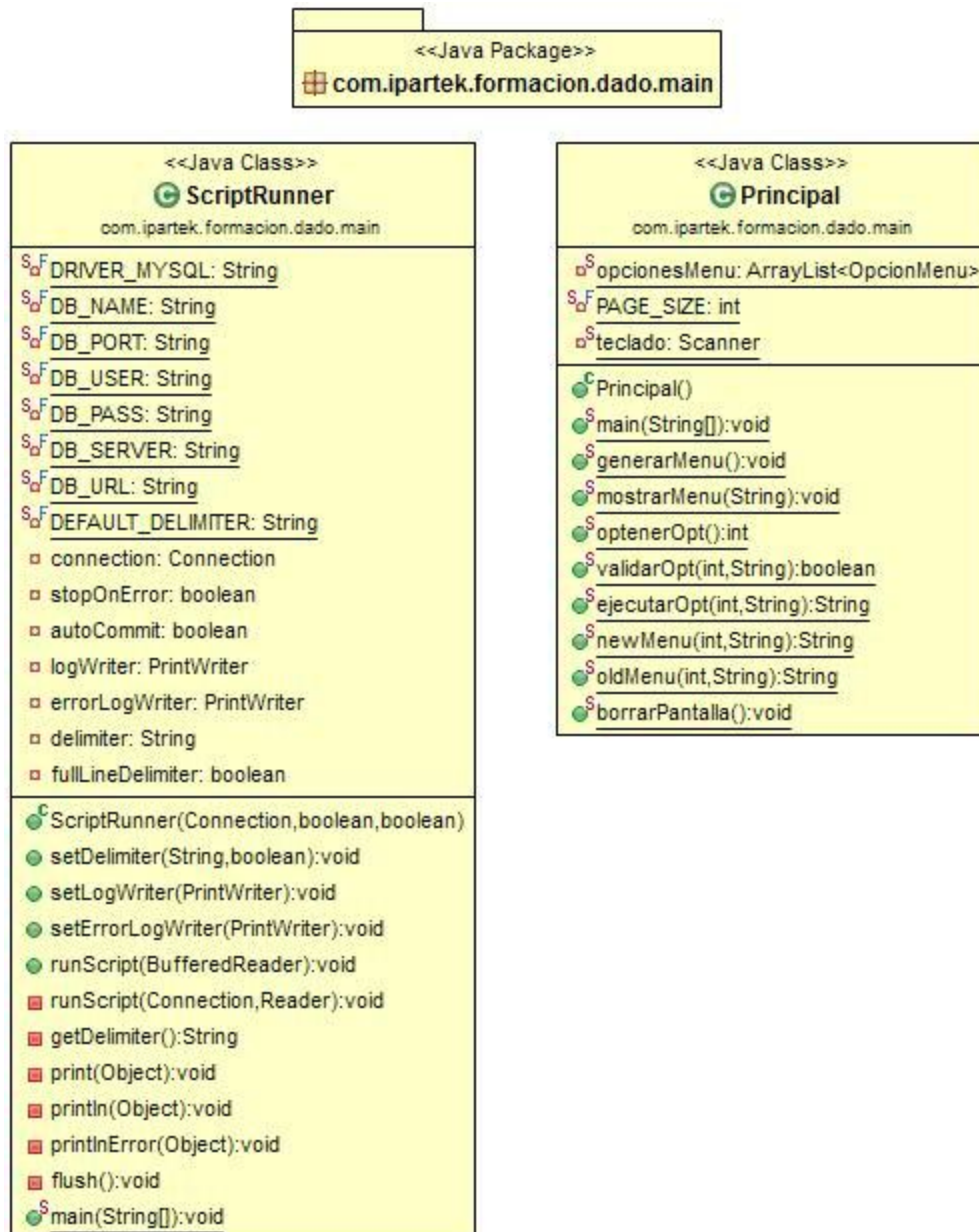
Situación a fecha de 05/12/2013

05/12/2013	Tarea	Situación
Ulises	<ul style="list-style-type: none"> <li>• Pruebas unitarias</li> <li>• Pruebas funcionales</li> </ul>	<ul style="list-style-type: none"> <li>• En proceso</li> </ul>
Beñat	<ul style="list-style-type: none"> <li>• Terminado</li> </ul>	<ul style="list-style-type: none"> <li>• Terminado</li> </ul>
Iker	<ul style="list-style-type: none"> <li>• Terminado</li> </ul>	<ul style="list-style-type: none"> <li>• Terminado</li> </ul>
Jonan	<ul style="list-style-type: none"> <li>• Logs</li> <li>• ANT</li> </ul>	<ul style="list-style-type: none"> <li>• Terminado</li> <li>• En proceso</li> </ul>
Julen	<ul style="list-style-type: none"> <li>• Documentación</li> <li>• Unificación y repaso de las tareas</li> </ul>	<ul style="list-style-type: none"> <li>• En proceso</li> <li>• En proceso</li> </ul>

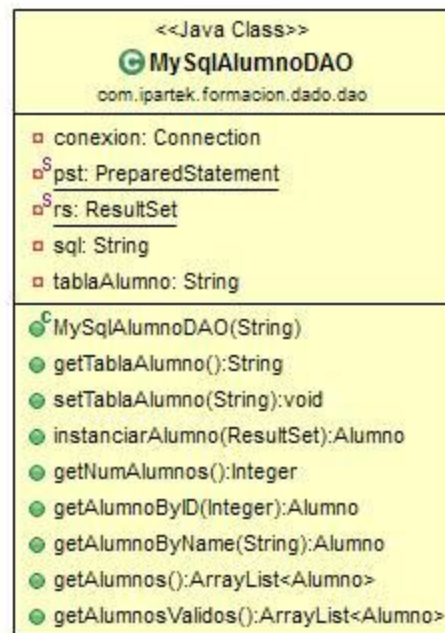
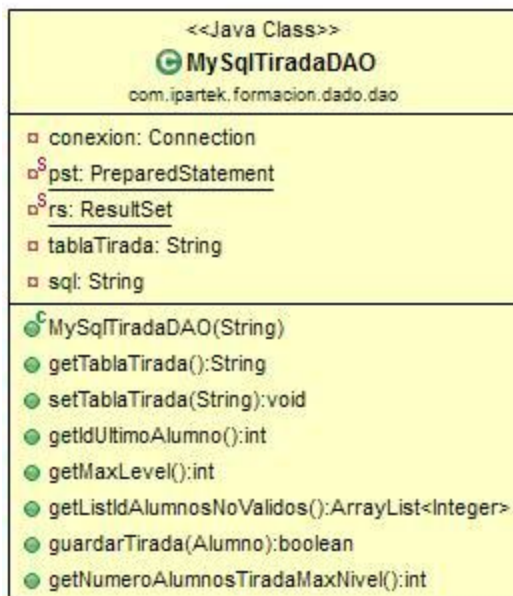
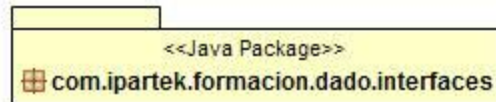
En esta fecha, el proyecto está llegando a su fin a falta tan sólo de las pruebas JUNIT y el ANT que automatiza las tareas.

## 6. Diagrama de clases final

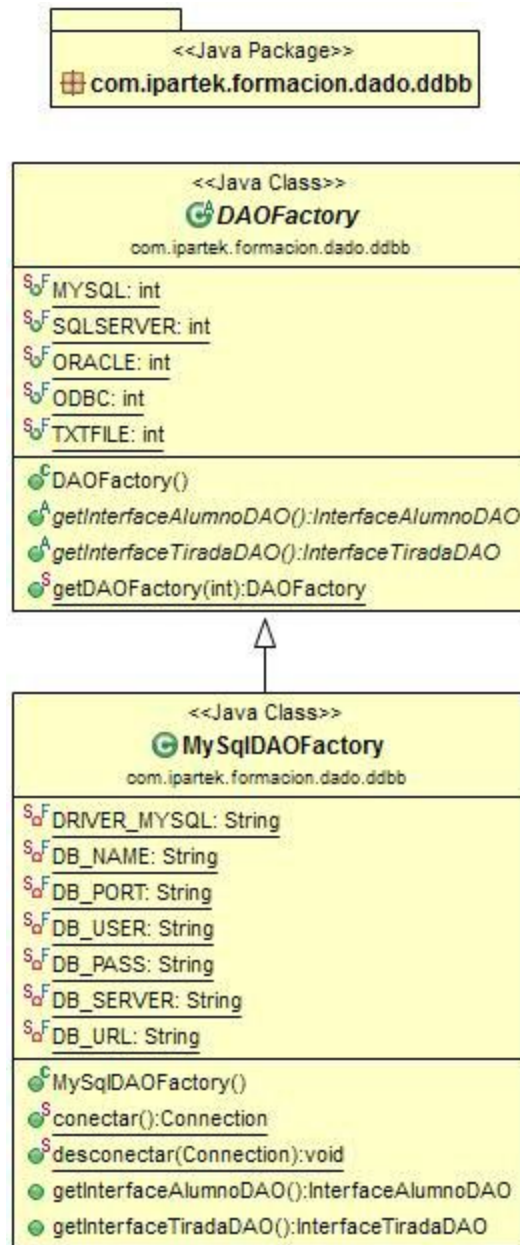
### *Empaquetados*



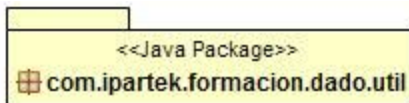




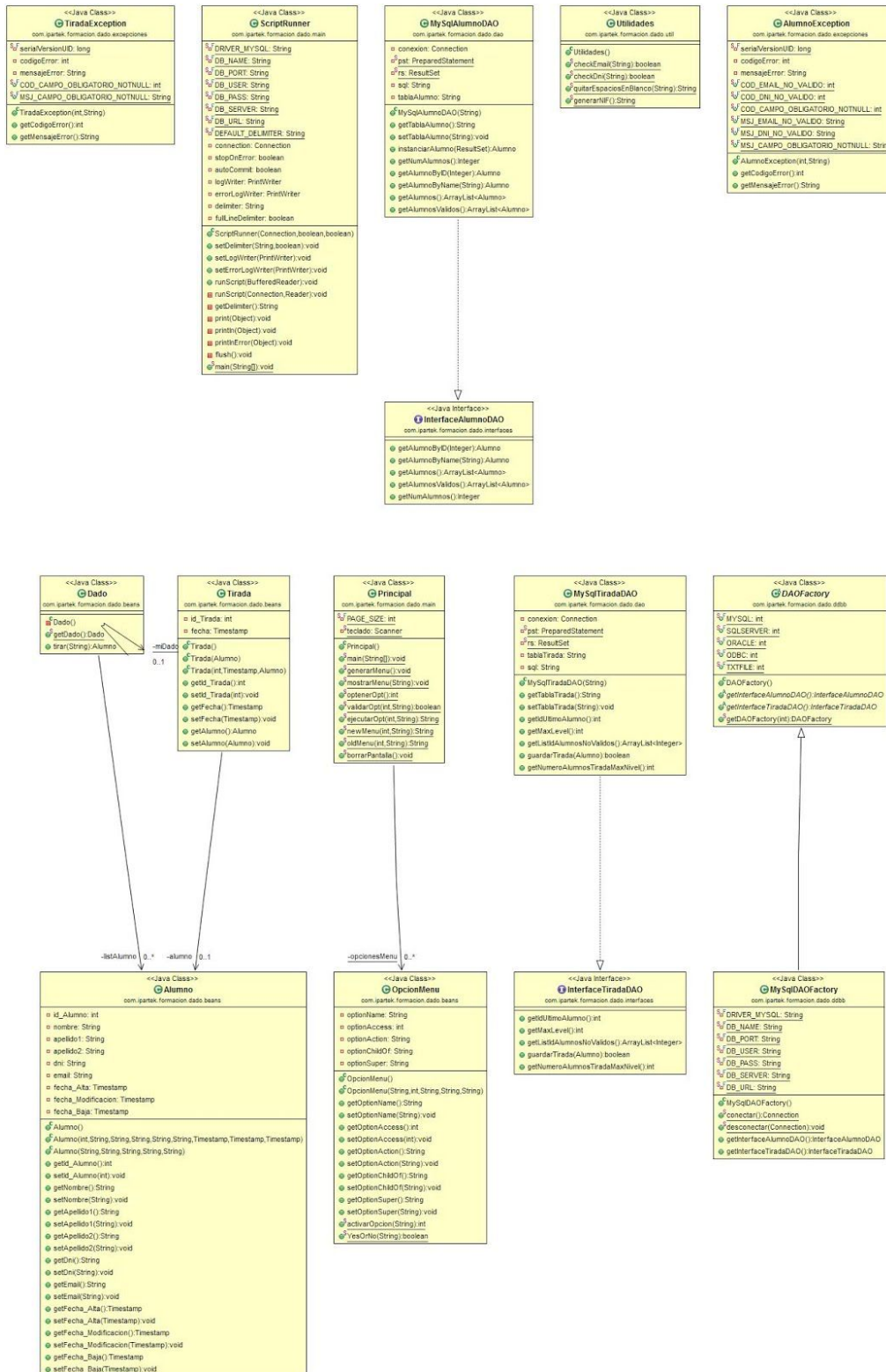








# Clases



## 7. Manual de instalación

### Herramientas necesarias

Antes de poder hacer uso de la aplicación desarrollada, se ha de configurar el entorno dónde se utilizará dicha aplicación. Para ello, se ha creado este manual de instalación y uso. Siendo conscientes de que hay múltiples herramientas para el correcto funcionamiento de este proyecto, en este manual recomendamos las que han sido usadas y testeadas para él.

1. **MySQL:** Sistema gestor de base de datos relacionales multihilo y multiusuario que será necesario para poder hacer uso de la base de datos que usaremos.
  - a. Descarga: <http://dev.mysql.com/downloads/windows/installer/>
2. **MySQL Workbench:** Herramienta visual de diseño de bases de datos que integra desarrollo de software, Administración de bases de datos, diseño de bases de datos, creación y mantenimiento para el sistema de base de datos MySQL.
  - a. Descarga: <http://dev.mysql.com/downloads/tools/workbench/>
3. **Plataforma JAVA:** Entorno o plataforma de desarrollo originaria de Sun Microsystems, capaz de ejecutar aplicaciones desarrolladas usando el lenguaje de programación Java.
  - a. Descarga: <http://www.java.com/es/download/>
4. **Eclipse:** Conjunto de herramientas de programación de código abierto multiplataforma para desarrollar proyectos en diferentes lenguajes.
  - a. Descarga: <http://www.eclipse.org/downloads/>
5. **Subversion o SVN:** Aplicación informática que implementa un sistema de control de versiones, manteniendo el registro de todo el trabajo y los cambios en los ficheros (código fuente principalmente) que forman un proyecto (de programa) y permite que distintos desarrolladores (potencialmente situados a gran distancia) colaboren.
  - a. Descarga: <http://subversion.apache.org/download/>
6. **Acceso a Google Code:** Sitio de Google para desarrolladores interesados en el desarrollo Google-related/open-source. El sitio contiene códigos fuente abiertos, una lista de sus servicios de apoyo público y APIs. Para poder acceder a este lugar es necesaria la creación de una cuenta de gmail.
  - a. Para la creación de cuenta: <https://accounts.google.com/SignUp?hl=es>
  - b. Acceso a Google Code: <https://code.google.com/intl/es/>

## Requisitos

### MySQL y MySQL WorkBench

- Un sistema operativo Windows de 32 bits, tal como 9x, Me, NT, 2000, XP, o Windows Server 2003.
- Se recomienda fuertemente el uso de un sistema operativo Windows basado en NT (NT, 2000, XP, 2003) puesto que éstos permiten ejecutar el servidor MySQL como un servicio. Consulte [Sección 2.3.12, “Arrancar MySQL como un servicio de Windows”](#).
- Soporte para protocolo TCP/IP.
- Una copia de la distribución binaria de MySQL para Windows, que se puede descargar de <http://dev.mysql.com/downloads/>. Consulte [Sección 2.1.3, “Cómo obtener MySQL”](#).
- Nota: Si se descarga la distribución a través de FTP, se recomienda el uso de un cliente FTP adecuado que posea la característica de reanudación (resume) para evitar la corrupción de ficheros durante el proceso de descarga.
- Una herramienta capaz de leer ficheros .zip, para descomprimir el fichero de distribución.
- Suficiente espacio en disco rígido para descomprimir, instalar, y crear las bases de datos de acuerdo a sus requisitos. Generalmente se recomienda un mínimo de 200 megabytes.

También se podría necesitar los siguientes ítems opcionales:

- Si se planea conectarse al servidor MySQL a través de ODBC, se deberá contar con un driver Connector/ODBC. Consulte [Sección 25.1, “MySQL Connector/ODBC”](#).
- Si se necesitan tablas con un tamaño superior a 4GB, debe instalarse MySQL en un sistema de ficheros NTFS o posterior. Al crear las tablas no debe olvidarse el uso de MAX\_ROWS y AVG\_ROW\_LENGTH. Consulte [Sección 13.1.5, “Sintaxis de CREATE TABLE”](#).

## Plataforma JAVA

- **Windows**

- Windows 8 (escritorio), Windows 7, Windows Vista SP2, Windows XP SP3 (32 bits), Windows XP SP2 (64 bits), Windows Server 2008, Windows Server 2012 (64 bits)
- RAM: 128 MB; 64 MB para Windows XP (32 bits)
- Espacio en disco: 124 MB
- Exploradores: Internet Explorer 7.0 y superior, Firefox 3.6 y superior, Chrome

- **Mac OS X**

- Mac basado en Intel que ejecuta Mac OS X 10.7.3 (Lion) o posterior.
- Privilegios de administrador para la instalación
- Explorador de 64 bits
- Es necesario un explorador de 64 bits (Safari o Firefox, por ejemplo) para ejecutar Java 7 en Mac OS X. Java 7 no está soportado en exploradores de 32 bits como, por ejemplo, Chrome, en la plataforma Mac.

- **Linux**

- Oracle Linux 5.5+
- Oracle Linux 6.x (32 bits)\*, 6.x (64 bits)\*\*
- Red Hat Enterprise Linux 5.5+, 6.x (32 bits)\*, 6.x (64 bits)\*\*
- Ubuntu Linux\* 10.04 y superior
- SUSE Linux Enterprise Server\* 10 SP2, 11.x
- 
- RAM: 64 MB
- Espacio en disco: 58 MB
- Exploradores: todos los sistemas operativos que soportan Firefox 3.6 y versiones superiores

## Eclipse

- Procesador: 350 MHz
- Memoria RAM: 256 MB

## Subversion o SVN

- Entorno de programación (Eclipse, NetBeans, etc...) con soporte para SVN.
- Repositorio público o privado. En nuestro caso google Code.

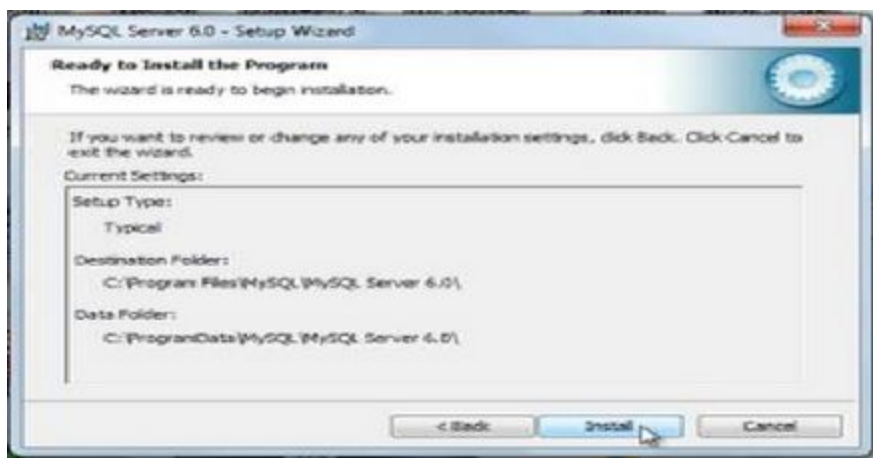
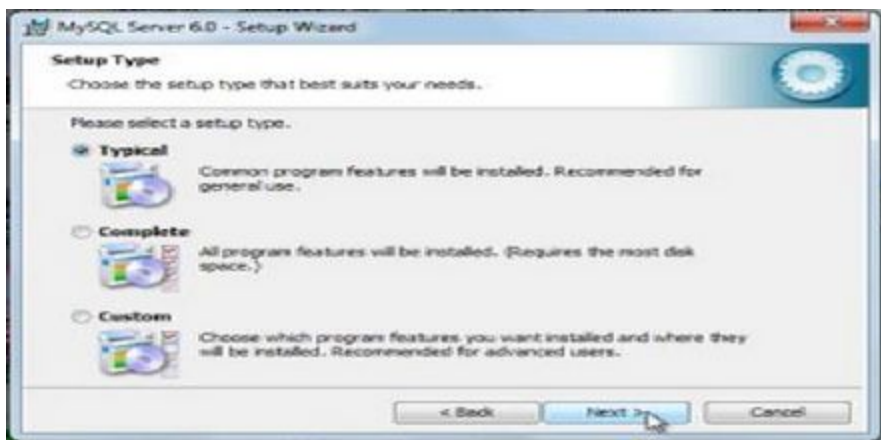
## Google Code

- Cuenta GMail

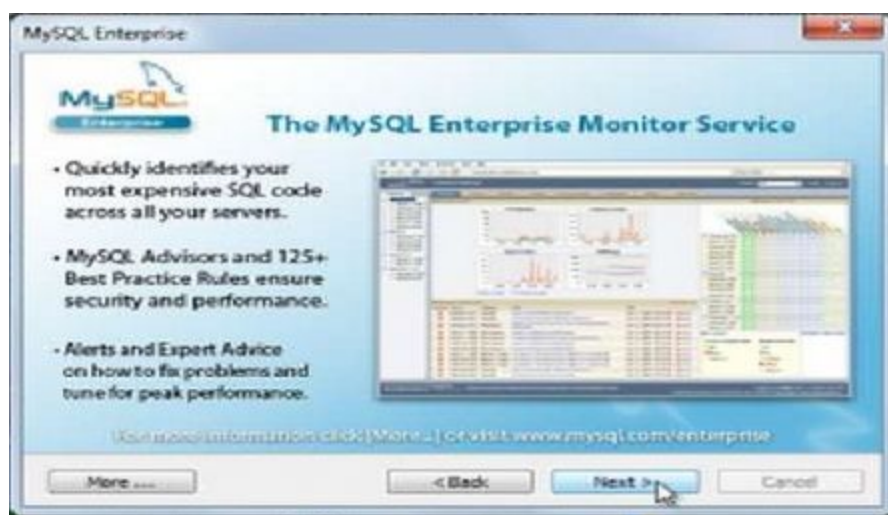
## Configuración del entorno

Una vez descargadas las herramientas a instalar, procederemos a la instalación de los mismo de la siguiente manera:

### MySQL





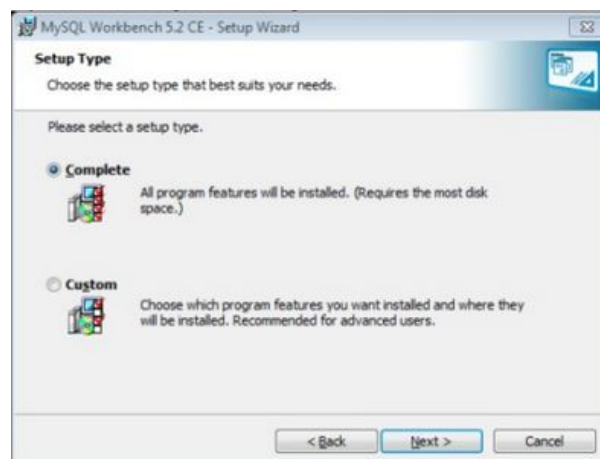
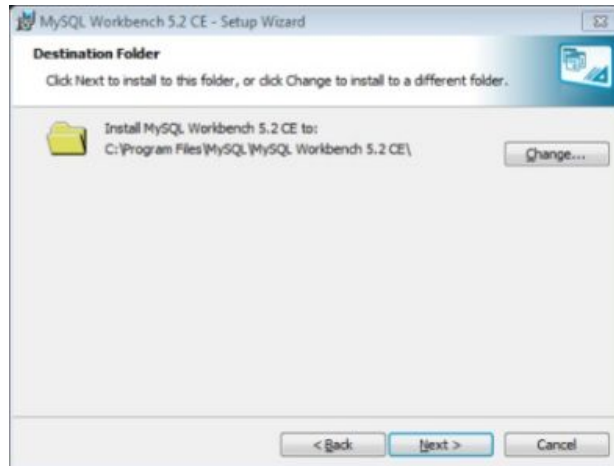
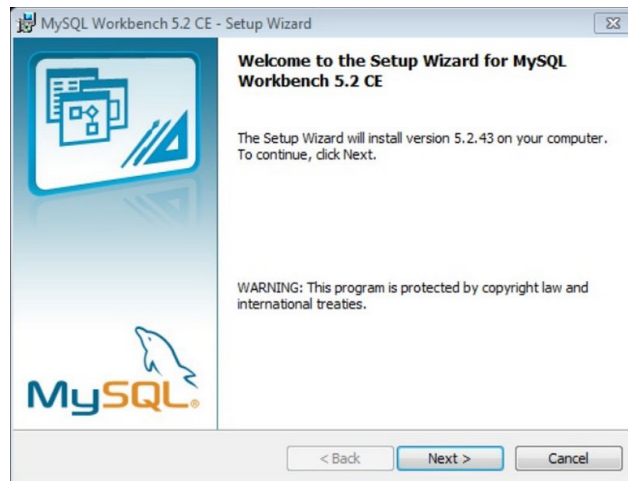


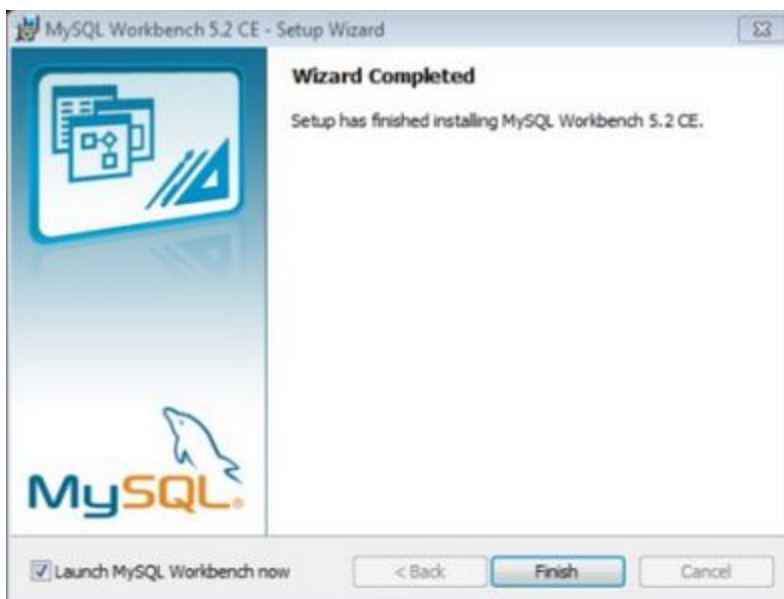
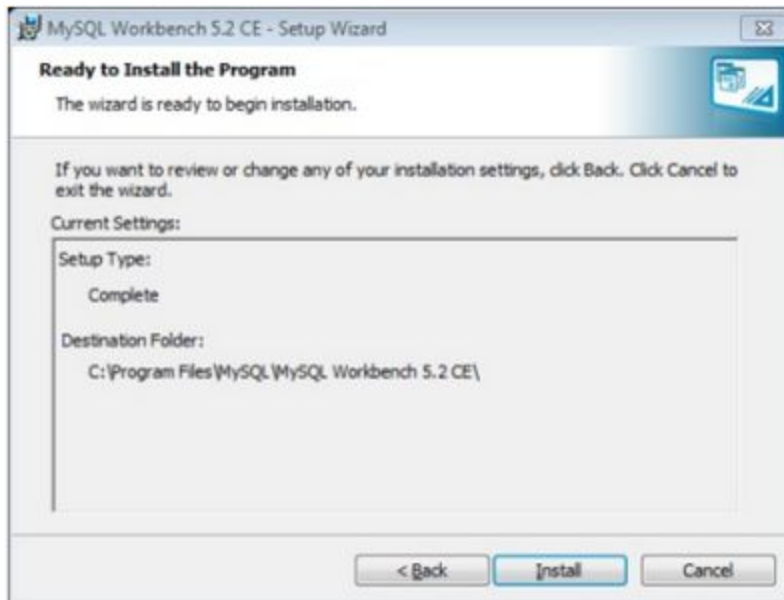


Para más información: <http://dev.mysql.com/doc/refman/5.0/es/installing.html>



## MySQL WorkBench

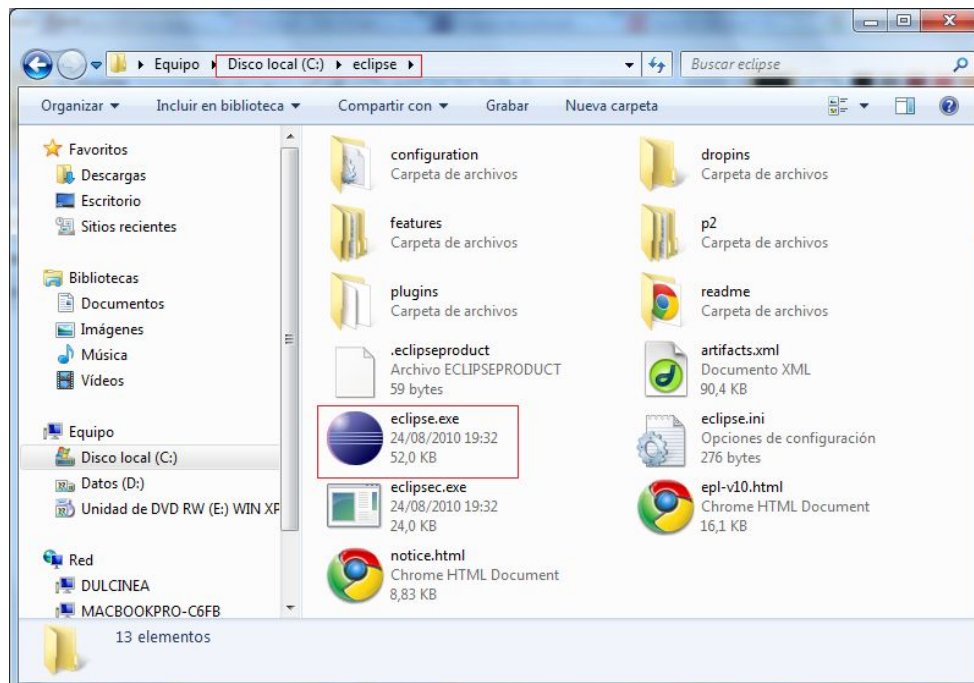
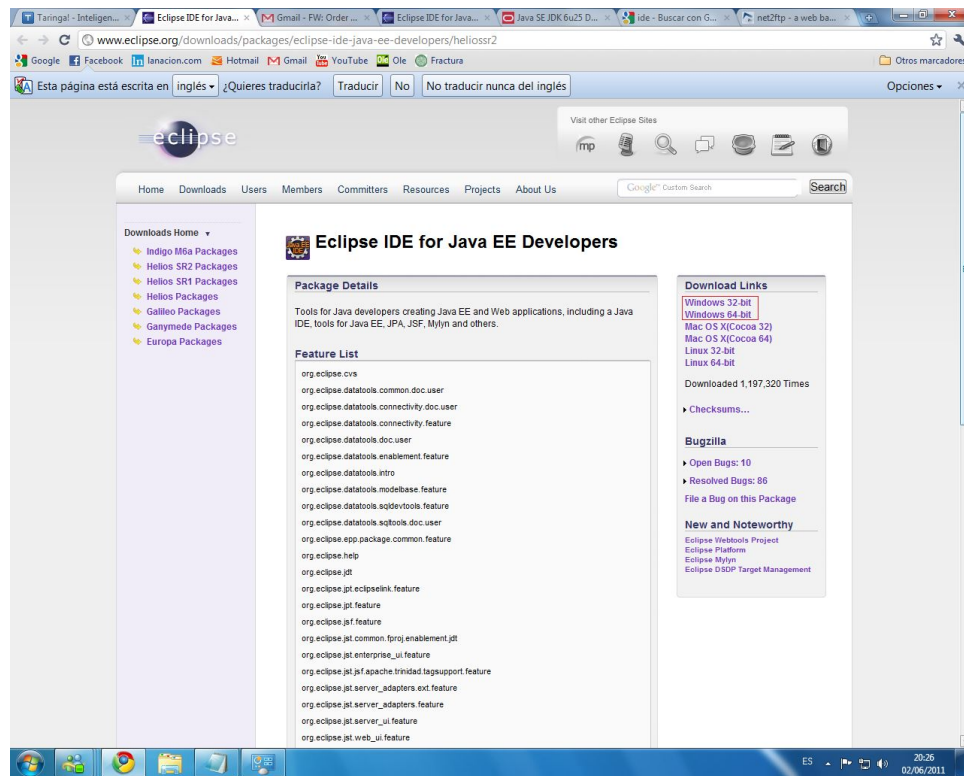


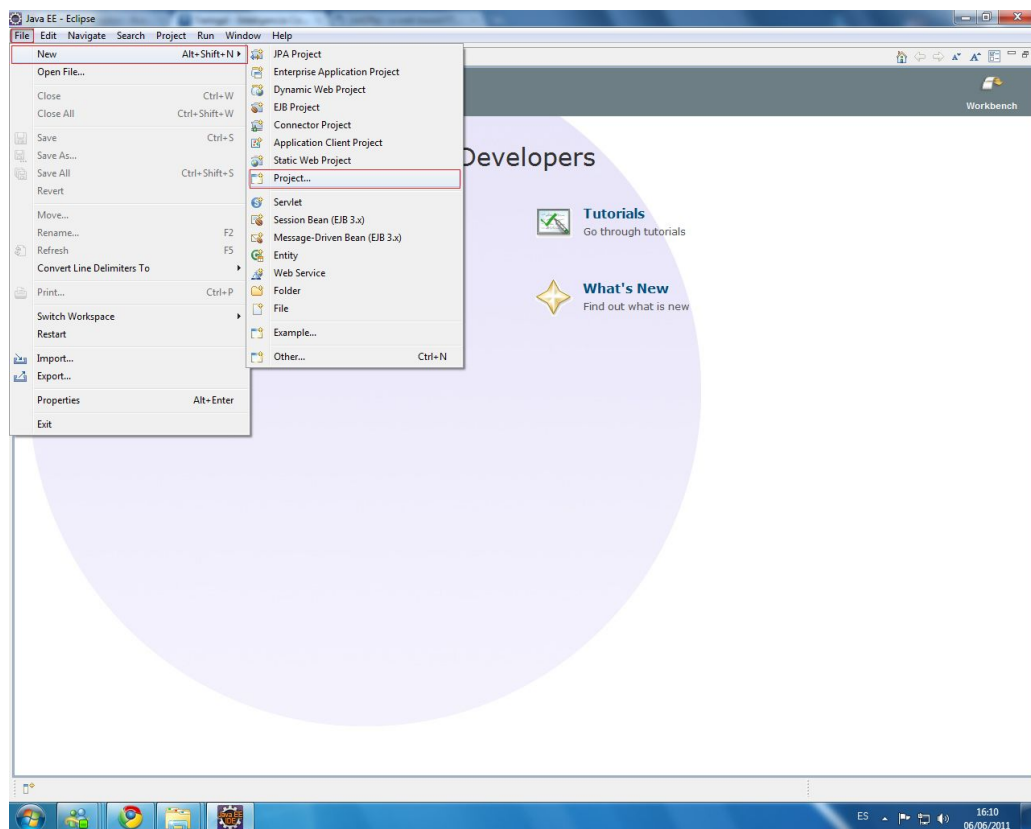
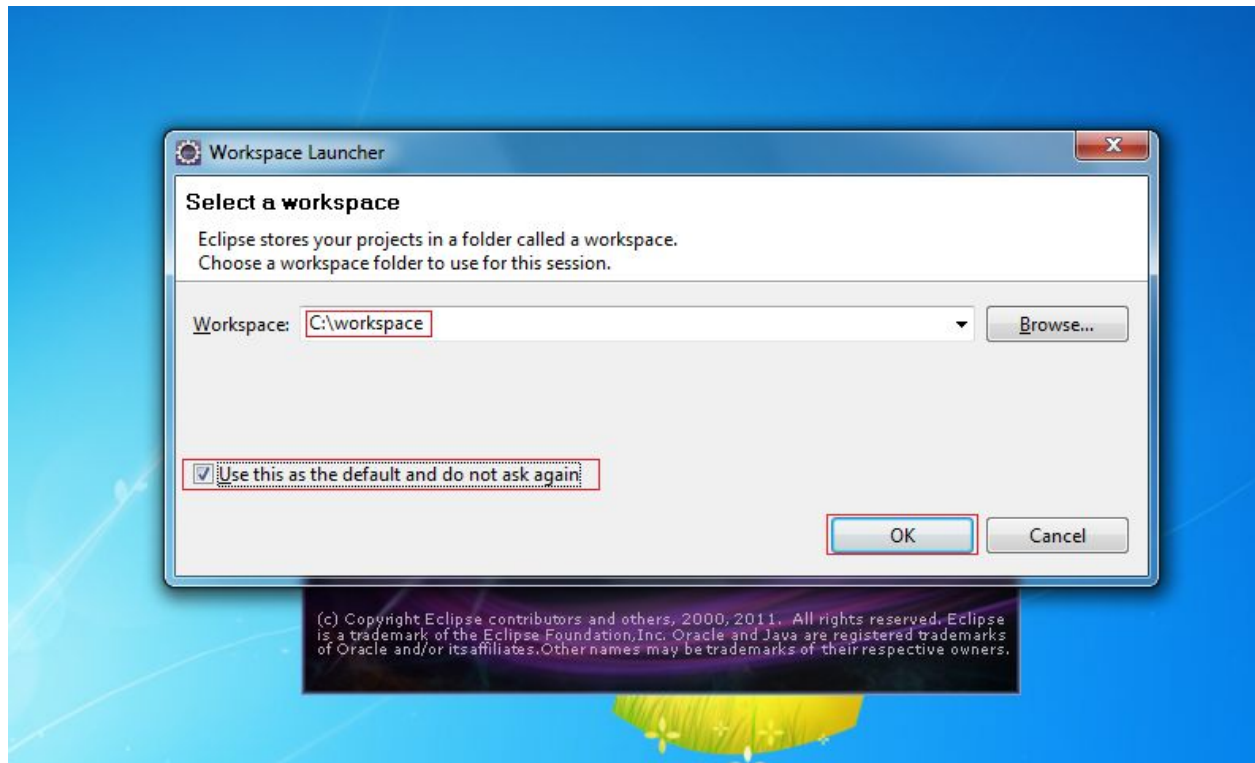


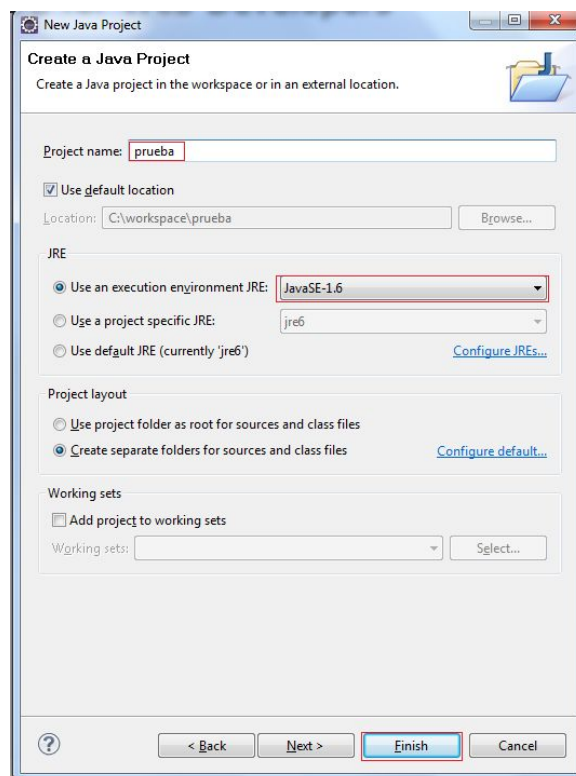
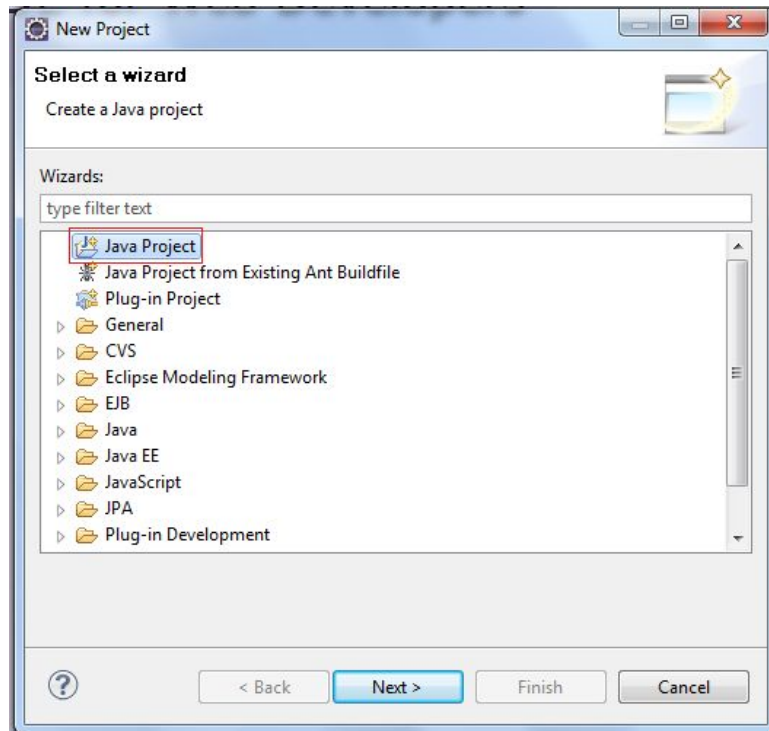
## Plataforma JAVA



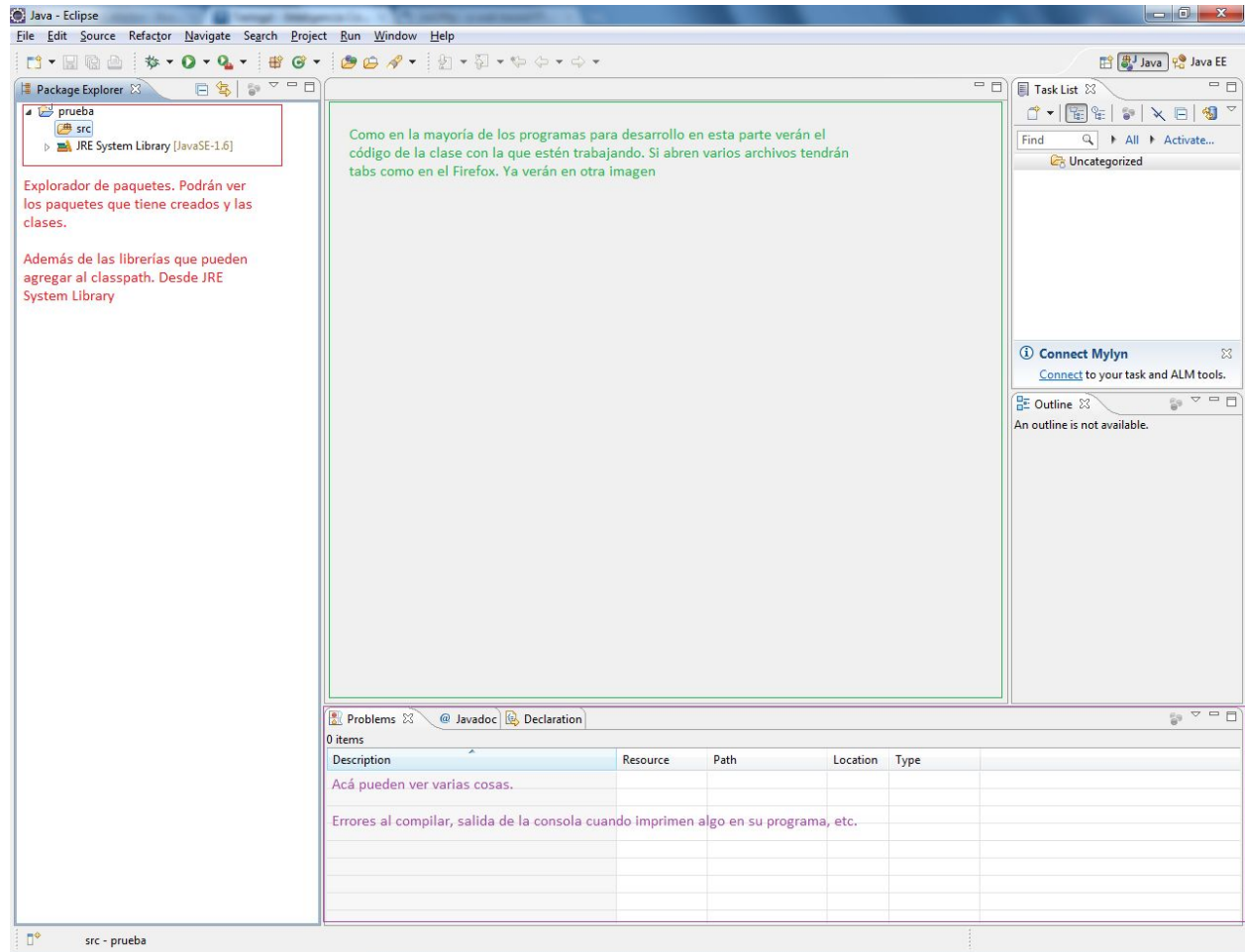
# Eclipse











## Subversion o SVN

subversion

Documents & files: Windows Apache 2.2.x

- subversion (0, 1206)
- Documentation (1, 7)
- Presentations (6, 6)
- Releases (0, 1193)

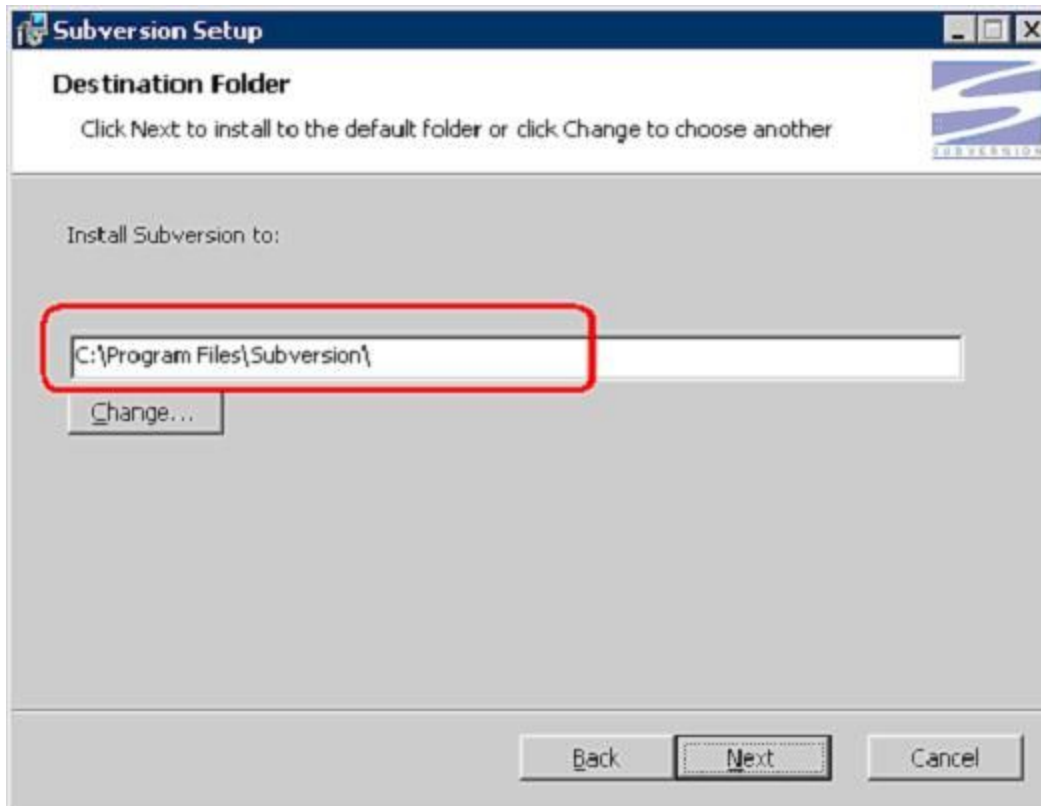
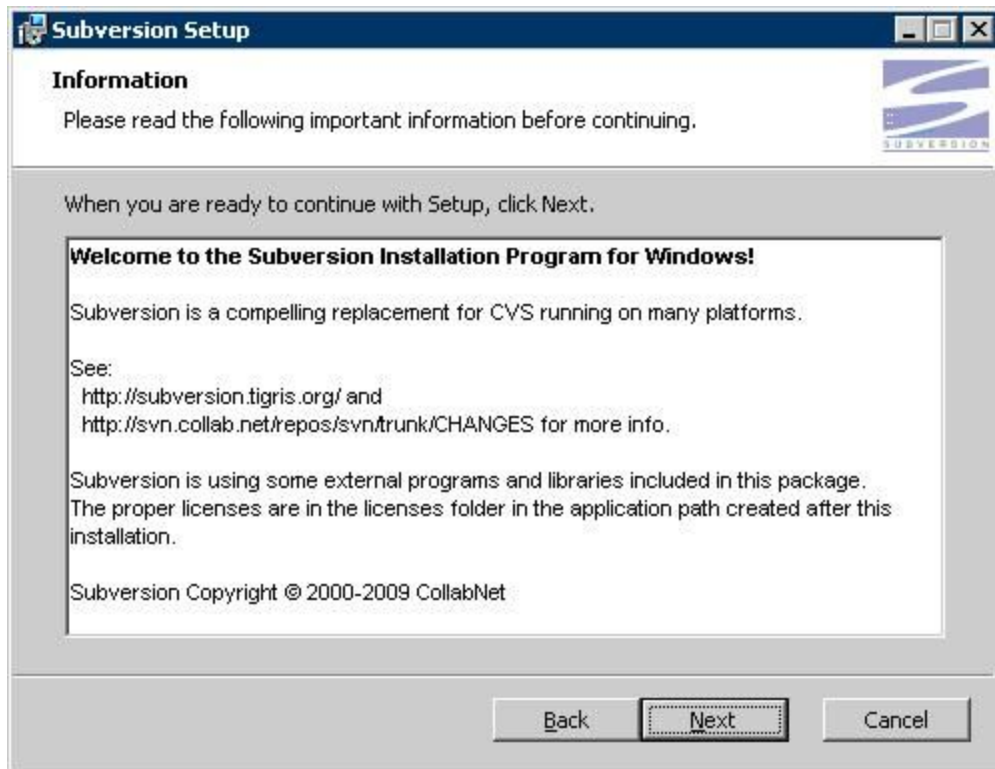
### Windows Apache 2.2.x

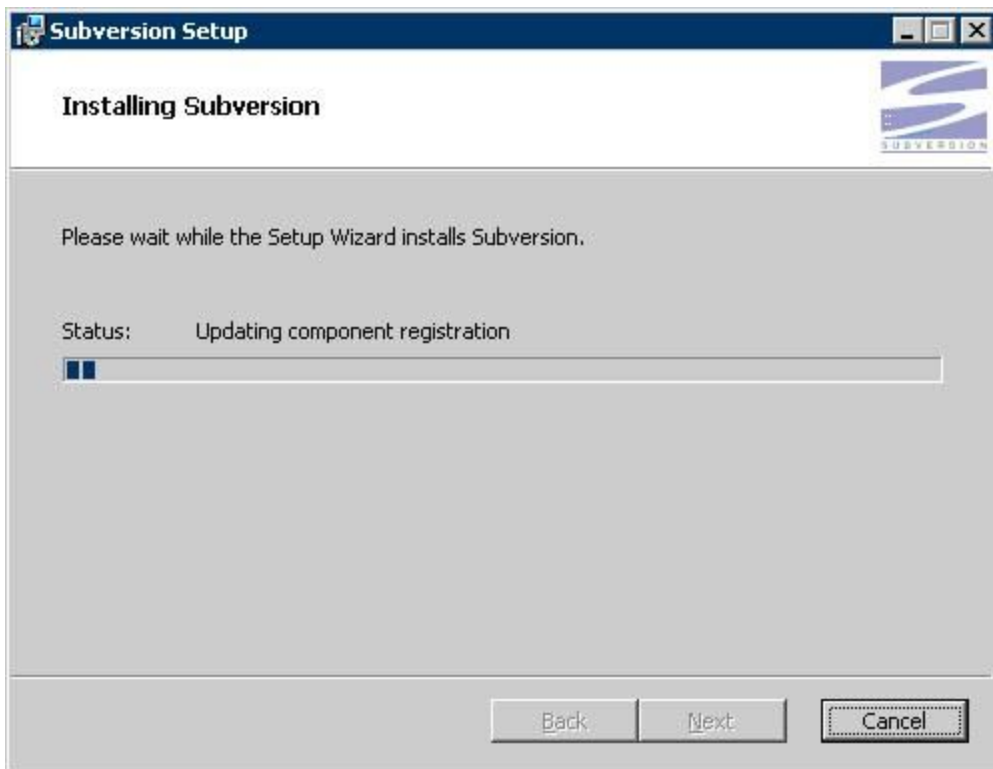
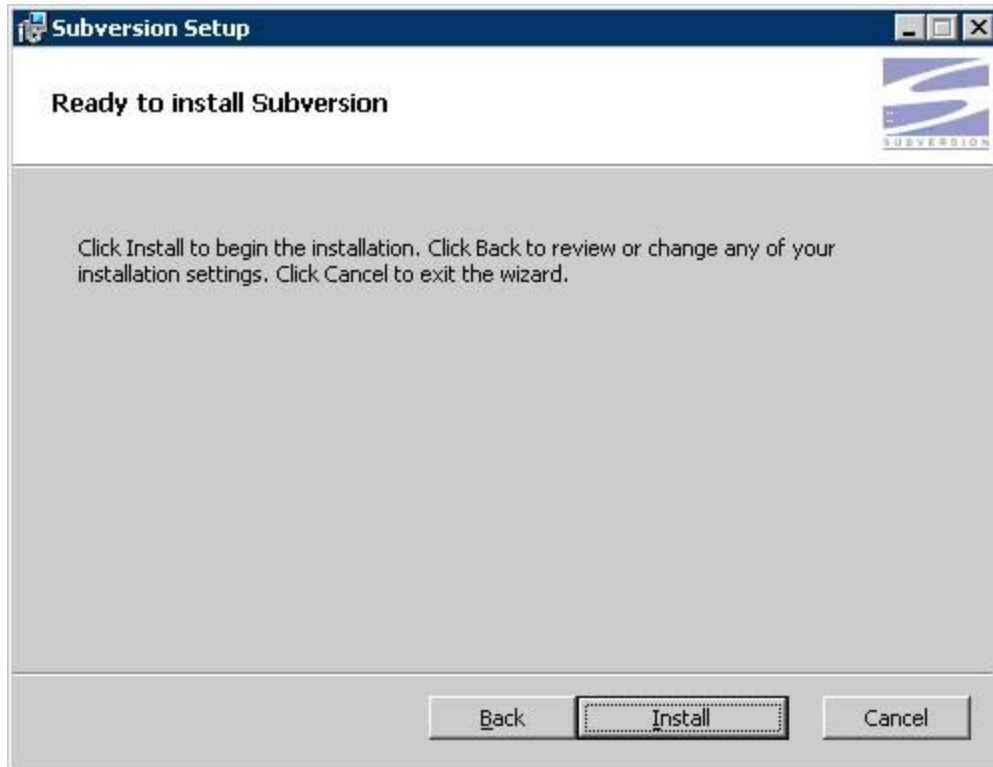
Windows binaries built against Apache 2.2.x

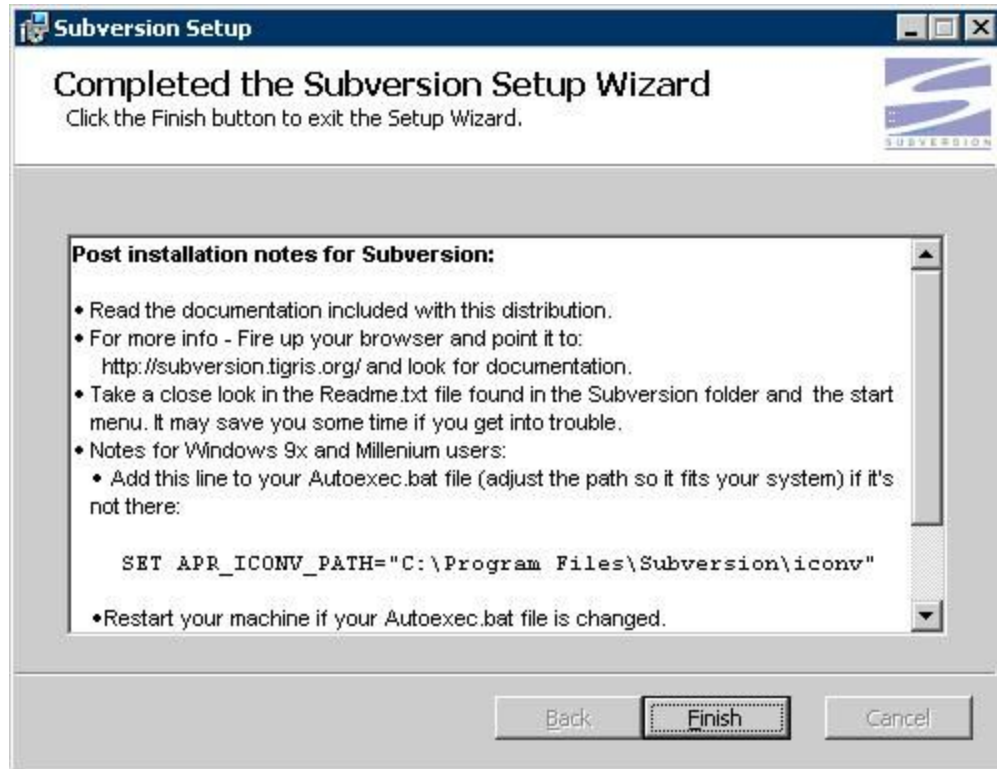
Filter this list		Filter			
Name	Status	Modified by	Size	Reservations	Description
 Setup-Subversion-1.6.4.msi	Stable	ebswift on Monday, August 10, 2009 at 12:09:07 PM	7.26 MB		Windows MSI installer with the basic win32 binaries - includes binaries for Apache 2.2.x (2.2.9 or higher is required within the 2.2.x series)
 Setup-Subversion-1.6.5.msi	Stable	ebswift on Wednesday, August 26, 2009 at 1:44:21 PM	7.27 MB		Windows MSI installer with the basic win32 binaries - includes binaries for Apache 2.2.x (2.2.9 or higher is required within the 2.2.x series)
 Setup-Subversion-1.6.6.msi	Stable	ebswift on Thursday, November 5, 2009 at 1:42:41 PM	4.61 MB		Windows MSI installer with the basic win32 binaries - includes binaries for Apache 2.2.x (2.2.9 or higher is required within the 2.2.x series)
 svn-python-1.6.5.win32-py2.5.exe	Stable	djh on Saturday, August 22, 2009 at 9:27:09 AM	2.57 MB		Python 2.5 bindings installer for Win32.
 svn-python-1.6.5.win32-py2.6.exe	Stable	djh on Saturday, August 22, 2009 at 9:26:38 AM	2.57 MB		Python 2.6 bindings installer for Win32.
 svn-python-1.6.6.win32-py2.5.exe	Stable	djh on Saturday, October 31, 2009 at 10:16:01 AM	2.58 MB		Python 2.5 bindings installer for Win32.
 svn-python-1.6.6.win32-py2.6.exe	Stable	djh on Saturday, October 31, 2009 at 10:15:33 AM	2.58 MB		Python 2.6 bindings installer for Win32.
 svn-win32-1.6.5_dev.zip	Stable	djh on Saturday,	6.42		Subversion Win32 development libraries











#### Google Code

Para la descarga de este proyecto antes de nada se ha de estar en posesión de una cuenta de **GMAIL** para podernos conectar a los repositorios de **google code**. Una vez tenemos esta característica nos conectamos al proyecto dado-fila3-1 y nos descargamos el BRANCH, TAG o TRUNK que estimemos oportuno.

## 8. Ejecución de la aplicación

Antes de ejecutar la aplicación, han de configurarse los siguientes campos en la clase **MySqlDAOFactory**, ubicada en el paquete **com.ipartek.formacion.dado.ddbb** con la configuración propia de cada usuario para su MySQL instalado en cada maquina:

```
private static final String DRIVER_MYSQL = "com.mysql.jdbc.Driver";
private static final String DB_NAME = "dado";
private static final String DB_PORT = "3306";
private static final String DB_USER = "root";
private static final String DB_PASS = "curso";
private static final String DB_SERVER = "localhost";
private static final String DB_URL = "jdbc:mysql://" + DB_SERVER + ":"
    + DB_PORT + "/" + DB_NAME;
```

Una vez configurado el entorno, y ya listos para ejecutar la aplicación, este proyecto consta de un main donde entraremos en la opción Inicializar / Resetear Base de Datos, despues podemos ejecutar el build.xml que compila la aplicación creando un *.jar* para usarla, pasa los test, y por último crea los *javadocs*.

## 9. Bibliografía

<http://riunet.upv.es/bitstream/handle/10251/10273/Memoria.pdf>

<https://code.google.com/p/dado-fila3-1/>