



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Diseño e implementación de una aplicación
para realizar análisis estadístico de
evaluaciones en las asignaturas de Grado de
ETSINF

Proyecto Final de Carrera

ITIG

Código: DISCA-310

Autores: Vicente Manuel Vila Figuerola
Andrea Gisele Siscar Cordero

Director: Dr. Lenin G. Lemus Zúñiga

Septiembre, 2013

Tabla de contenidos

1.	Introducción y objetivos	4
	1.1.- Introducción.	4
	1.2.- Objetivos.	4
	1.3.- Reparto del trabajo entre los integrantes del grupo.	4
1.	Contraste de medias	5
2.	Herramientas utilizadas	10
	3.1.- Eclipse	10
	3.2.- IBM SPSS Statistics	10
	3.3.- Plug-in de Integración con Java	10
	3.4.- JDBC: Acceso a Bases de Datos	13
	3.5.- JFreeChart	15
	3.6.- HTML.....	15
	3.7.-PHP	16
3.	Casos de uso.....	17
	4.1.-Comparar medias.	17
	4.2.-Elegir tipo de gráfico.....	17
4.	Estructura de la aplicación.....	18
5.	Pruebas	20
6.	Manual de usuario.....	23
7.	Conclusiones	25
8.	Bibliografía	26

1. Introducción y objetivos

En este primer capítulo realizaremos una pequeña introducción a la herramienta EvalTICS de la que ya disponemos y expondremos los objetivos de este Proyecto Final de Carrera.

1.1.- Introducción.

EvalTICS es una herramienta diseñada para mejorar la calidad de la enseñanza mediante evaluaciones periódicas vía web. Con ella el profesor puede evaluar a sus alumnos al finalizar una clase para comprobar si éstos han interiorizado los conocimientos de dicha clase, al finalizar un tema o, básicamente, en cualquier momento que estime oportuno, y con los resultados de dichas evaluaciones puede guiarse y hacer hincapié en unos contenidos u otros.

1.2.- Objetivos.

El objetivo de este proyecto es el de incorporar a EvalTICS la posibilidad de comparar y validar la eficacia de los métodos de enseñanza utilizados para lograr que los alumnos adquieran determinadas objetivos de aprendizaje, para ello vamos a hacer uso de varios conceptos estadísticos con el fin de determinar esa validez.

La herramienta ahora generará una comparación de los grupos para poder detectar las diferencias estadísticamente significativas y las mostrará en gráficos para una visualización más directa de los resultados, lo que nos servirá para evaluar la calidad de la enseñanza, así como que grupo o grupos han mejorado desde la última evaluación.

1.3.- Reparto del trabajo entre los integrantes del grupo.

Para mejorar las capacidades de EvalTICS, se quiere agregar:

1. Facilidad para realizar los análisis estadísticos utilizando el test de Bonferroni.
2. Mejorar la presentación de los resultados.

D. Vicente Manuel Vila Figuerola se encargará de mejorar las facilidades para realizar los análisis estadísticos.

Doña Andrea Gisele Siscar Cordero se encargará de mejorar la presentación de resultados.

1. Contraste de medias

El objetivo primario de EvalTICS es el de validar la eficacia de los métodos de enseñanza utilizados para lograr que los alumnos adquieran determinados objetivos de aprendizaje, para ello vamos a hacer uso del *contraste de medias con el fin de determinar esa validez*. Antes de ver el procedimiento conviene tener una idea general de la utilidad de estos contrastes.

Muchos *diseños experimentales, y planteamientos de evaluación* en general, desembocan en una comparación de las medias de dos grupos: un grupo *experimental* (el que ha seguido nuestra metodología, el que ha tenido una determinada experiencia o formación) lo comparamos con otro grupo de *control* o de *contraste* (un grupo comparable al grupo experimental pero que no ha pasado por esta experiencia o método). Son varios los diseños experimentales que se pueden proponer y cuyo análisis básico es un simple contraste de medias.

En definitiva esperamos que si la experiencia o método ha sido provechosa, los sujetos del grupo *experimental* tendrán una *media mayor* que la del otro grupo. Por una media mayor entendemos mayor de lo que cabría esperar por azar, o mayor que lo que se puede esperar de la variación natural de una muestra a otra.

La comparación de las medias de los subgrupos con puntuación total más alta y más baja (en un examen o en un test) tiene aplicaciones específicas:

1) Aporta información de interés como en qué se parecen y en qué se diferencian los *altos* y los *bajos resultados* en cualquier variable.

2) Es una manera sencilla de analizar los ítems de una escala o test, para quedarnos con los más *diferenciadores*: con estos ítems tenemos una fiabilidad mayor o podemos reducir la longitud del instrumento con el que medimos los datos.

Lo que vamos a comprobar es cuándo una diferencia es mayor de lo que se podría esperar por azar si entre los dos grupos no hubiera más diferencias que las puramente casuales.

Nuestro modelo teórico es la *distribución muestral de las diferencias entre medias de muestras que proceden de la misma población*: entre estas medias no hay más diferencias que las puramente aleatorias.

En este modelo se supone lo siguiente:

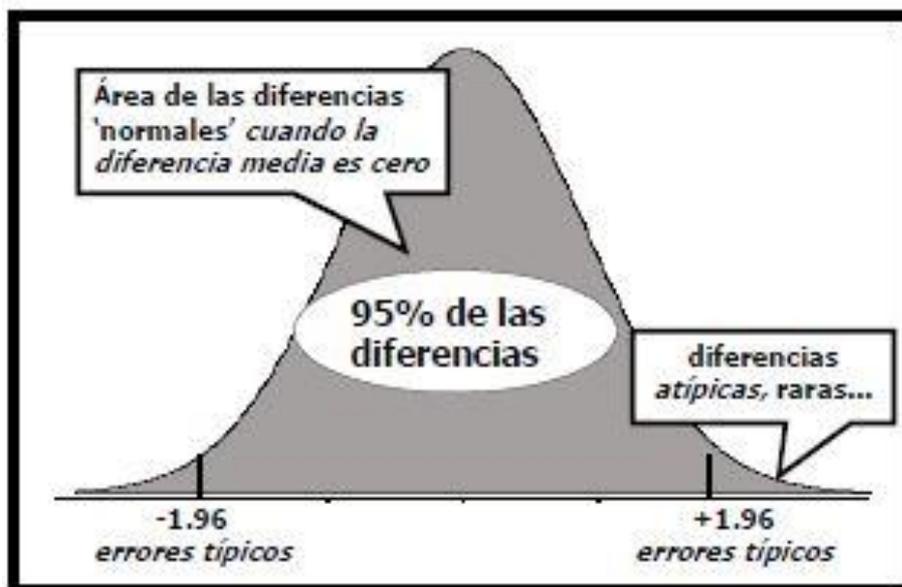
1º) De una misma población extraemos un número muy grande de *pares de muestras* y calculamos la *diferencia entre sus medias*: la media de la primera muestra menos la media de la segunda muestra.

Unas veces la diferencia será positiva (la media de la primera muestra es mayor que la de la segunda muestra), otras negativa y otras veces la diferencia será cero.

2º Al tener muchas diferencias entre medias por hipótesis tendremos que:

- Estas diferencias tendrán *una distribución normal*.
- La *media de esta distribución será cero* porque las diferencias positivas anulan a las negativas, si es que todas proceden de la misma población y no hay más diferencias que las puramente *casuales*; en términos más propios *debidas al error muestral*.
- La *desviación típica* de esta distribución que como se trata de una distribución muestral se denomina *error típico* podemos *estimarla* a partir de los valores del *tamaño* y de las *desviaciones típicas* de las muestras.
- La *mayoría* de estas diferencias (el 95%) estará entre -1.96 *errores típicos* y $+1.96$ *errores típicos*; convencionalmente situamos la *normalidad* entre estos límites.

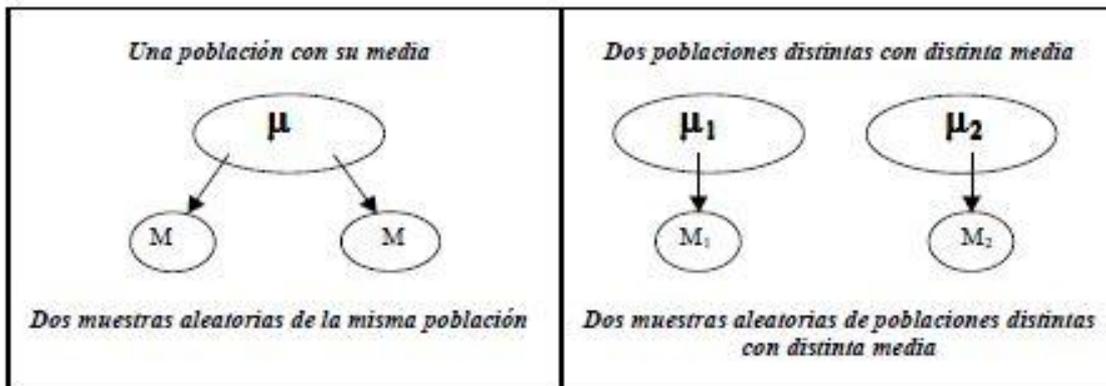
Lo que vamos a hacer es comprobar si *nuestra diferencia está dentro de lo normal*, si pertenece a esa población de diferencias cuya media es cero. En ese caso concluiremos que la diferencia está dentro de lo normal y aleatorio.



En la figura 1 tenemos representada la distribución de las diferencias entre medias *cuando la media de las diferencias es cero* porque no hay más diferencias que las puramente casuales.

Entre -1.96 *errores típicos* (o desviaciones típicas) y $+1.96$ *errores típicos* tendremos el 95% de las diferencias. A estas diferencias las consideramos *normales* porque están dentro de los límites que convencionalmente hemos determinado como criterio de lo que suele suceder cuando no hay más diferencias que las casuales.

Al comparar dos medias podemos encontrar una de las dos posibilidades representadas en la figura 2, cada posibilidad nos llevará a una conclusión distinta



¿Cuándo podemos afirmar que las dos muestras pertenecen a la *misma población*?



Afirmamos que las dos muestras pertenecen a la *misma población* cuando la diferencia está *dentro de lo normal o frecuente*;



Es decir, cuando se trata de una diferencia *probable* en el caso de que las dos muestras procedan de la misma población.

¿Cuándo podemos afirmar que las dos muestras proceden de *poblaciones distintas*?



Afirmamos que las dos muestras proceden de *poblaciones distintas* cuando la diferencia es *atípica, poco frecuente, se sale de lo normal*



Es decir, cuando se trata de una diferencia *poco probable* en el caso de que las dos muestras procedan de la misma población. En ese caso podremos afirmar que las medias pertenecen a muestras de poblaciones distintas, con distinta media.

Diseño e implementación de una aplicación para realizar análisis estadístico de evaluaciones en las asignaturas de Grado de ETSINF

Las fórmulas que vamos a utilizar para comprobar la validez de los métodos usados son el F-test de Fisher y el test de Bonferroni que nos servirán para comprobar si el cambio en las medias y en las varianzas de dos o más muestras obtenidas denotan una mejora significativa.

El Test F de Fisher es un test de comparaciones múltiples. Permite comparar las medias de los t niveles de un factor después de haber rechazado la hipótesis nula de igualdad de medias mediante la técnica ANOVA. Todos los test de comparaciones múltiples son test que tratan de perfilar, especificar y concretar, una hipótesis alternativa genérica como la de cualquiera de los Test ANOVA.

El Test se basa en la creación de un valor común, un umbral, basado en un test de la t de Student. Se realizan todas las diferencias entre medias de los t niveles. Las diferencias que estén por encima de este umbral indicarán una diferencia de medias significativa y las diferencias que estén por debajo indicarán una diferencia no significativa.

El Test de Bonferroni permite comparar las medias de los t niveles de un factor después de haber rechazado la hipótesis nula de igualdad de medias mediante la técnica ANOVA.

Todos los test de comparaciones múltiples son test que tratan de concretar una hipótesis alternativa genérica como la de cualquiera de los Test ANOVA.

El Test de Bonferroni hay que entenderlo en relación con el F-test de Fisher. Se basa en la creación de un umbral por encima del cual la diferencia entre las dos medias será significativa y por debajo del cual esa diferencia no lo será de estadísticamente significativa.

Si se comparan ambos test se verá que el cambio está en el nivel de significación elegido. En el Test de Bonferroni el nivel de significación se modifica en función del número de comparaciones a hacer. Esto elimina el problema de hacer comparaciones múltiples. Reduce el nivel de significación en tal medida que elimine el error de aplicar el test tantas veces al mismo tiempo.

Ejemplo:

Dados los siguientes datos (extraídos de realizar el mismo examen a tres grupos distintos):

Nota Media	2,4	3	3	3	5,5	5,9	6,6	6,9	7,5	9,6	3	5	5,8	6,3	6,4	6,9	7,4	8,4	2,8	3	3	4	4	4,4	4,4	5	5
Grupo	1									2									3								

Vamos a proceder a aplicar los métodos estadísticos antes descritos, con lo que obtenemos la siguiente tabla:



Comparaciones múltiples

Variable dependiente: Nota_media

Bonferroni

(I) Grupo	(J) Grupo	Diferencia de medias (I-J)	Error típico	Sig.	Intervalo de confianza al 95%	
					Límite inferior	Límite superior
1	2	-1,66667	,78788	,135	-3,6944	,3611
	3	,91111	,78788	,777	-1,1166	2,9388
2	1	1,66667	,78788	,135	-,3611	3,6944
	3	2,57778*	,78788	,010	,5500	4,6055
3	1	-,91111	,78788	,777	-2,9388	1,1166
	2	-2,57778*	,78788	,010	-4,6055	-,5500

*. La diferencia de medias es significativa al nivel 0.05.

Los valores resaltados en color rojo nos indican que existe una diferencia estadísticamente significativa entre los grupos 2 y 3, mientras que el grupo 1 estadísticamente no presentaría ninguna diferencia significativa con los grupos 2 y 3.



2. Herramientas utilizadas

3.1.- Eclipse

Eclipse es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido". Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse.

Eclipse es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas. Un ejemplo es el recientemente creado Eclipse Modeling Project, cubriendo casi todas las áreas de Model Driven Engineering.

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

3.2.- IBM SPSS Statistics

IBM SPSS (Statistical Product and Service Solutions) Statistics ofrece los procedimientos estadísticos principales que los gestores y los analistas necesitan para tratar las cuestiones empresariales y de investigación básicas. Este software proporciona herramientas que permiten a los usuarios consultar datos y formular hipótesis para pruebas adicionales de forma rápida, así como ejecutar procedimientos para ayudar a aclarar las relaciones entre variables, crear clústeres, identificar tendencias y realizar predicciones.

El programa SPSS es un conjunto de potentes herramientas de tratamiento de datos y análisis estadístico, en el que a partir de la versión 21 incluye un plug-in de integración con Java.

3.3.- Plug-in de Integración con Java

El Plug-in de integración con Java permite a los desarrolladores de aplicaciones crear aplicaciones Java que pueden invocar y controlar el procesador del IBM SPSS Statistics, o para implementar comandos de extensión en Java que se puede ejecutar desde dentro de SPSS Statistics.

Los comandos de extensión de SPSS Statistics que se ejecutan en un lenguaje externo y permiten a los usuarios que son competentes en ese lenguaje para compartir funciones externas con los usuarios de la sintaxis de comandos estándar de SPSS Statistics.

Con este plug-in para Java, podemos hacer lo siguiente:



- Ejecutar sintaxis de comandos de SPSS Statistics.
- Lee datos de casos del conjunto de datos activo.
- Obtener información acerca de los datos del conjunto de datos activo.
- Agregar nuevas variables y añadir los casos al conjunto de datos activo.
- Obtener resultados de la salida de los comandos de sintaxis.
- Crear una salida personalizada en forma de tablas dinámicas y bloques de texto.

El Plug-in se instala con SPSS Statistics y no requiere instalación ni configuración separada. Para la versión 21 del SPSS Statistics, el Plug-in es compatible con la versión 6 de Java.

Invocación desde una aplicación Java

La interfaz para invocar el IBM SPSS Statistics y el control del procesador de SPSS Statistics se proporciona en el archivo JAR **spssjavaplugin.jar**, que se instala junto al SPSS Statistics. El archivo JAR contiene el paquete **com.ibm.statistics.plugin**, que contiene las clases Java disponibles en el Plug-in de Integración con Java. Asegúrese de añadir **spssjavaplugin.jar** a la ruta de la clase Java.

- En Windows, **spssjavaplugin.jar** se encuentra en el directorio de instalación de SPSS Statistics.
- En los sistemas UNIX y Linux Server, **spssjavaplugin.jar** se encuentra en el directorio bin bajo el directorio de instalación de SPSS Statistics.
- En Mac OS, **spssjavaplugin.jar** se encuentra en el directorio bin bajo el directorio de contenido en el paquete de aplicaciones de SPSS Statistics.

Cuando se invoca desde una aplicación Java externa, el procesador de SPSS Statistics se ejecuta sin una instancia asociada del cliente de SPSS Statistics. En este modo, la salida generada puede ser manejada con los parámetros especificados en el método con el que se inicia el procesador o a través del uso del Sistema de gestión de resultados de SPSS Statistics, que se invoca con el comando **OMS**.

El siguiente es un ejemplo sencillo del uso del Plug-in de Java para crear un conjunto de datos SPSS Statistics, calcular los estadísticos descriptivos y generar salida. Esto ilustra las características básicas para invocar SPSS Statistics desde una aplicación Java externa.

```
import com.ibm.statistics.plugin.*;
public class demo {public static void main(String[] args) {
    try {
```



Diseño e implementación de una aplicación para realizar análisis estadístico de evaluaciones en las asignaturas de Grado de ETSINF

```
StatsUtil.start();
String[] command={"OMS",
                 "/DESTINATION FORMAT=HTML
                 OUTFILE='/output/demo.html'.",
                 "DATA LIST FREE /salary (F).",
                 "BEGIN DATA",
                 "21450",
                 "30000",
                 "57000",
                 "END DATA.",
                 "DESCRIPTIVES salary.",
                 "OMSEND."};
StatsUtil.submit(command);
StatsUtil.stop();
} catch (StatsException e) {
    e.printStackTrace();
}
}
```

- La sentencia **import com.ibm.statistics.plugin*** importa todas las clases contenidas en el paquete **com.ibm.statistics.plugin**.
- El método **StatsUtil.start()** inicia el procesador de SPSS Statistics.
- Un array de String que especifica la sintaxis de comandos de SPSS crea un conjunto de datos y ejecuta los procedimientos descriptivos. La sintaxis del comando es ejecutado con SPSS Statistics utilizando el método **StatsUtil.submit()**. La salida del procedimiento descriptivo es encaminada a un Archivo HTML con el comando OMS.
- El método **StatsUtil.stop()** detiene el procesador de SPSS Statistics y debe ejecutarse para finalizar correctamente una sesión de SPSS Statistics.
- La clase **StatsException** es una subclase de la clase **Exception** Java nativa, y se ocupa de las excepciones que son específicas del Plug-in de Java.

Ahora procederemos a analizar la cadena de comandos que vamos a usar para nuestro proyecto.

```
String[] command = { "OMS /SELECT TABLES ", "/IF COMMANDS = ['Oneway'] SUBTYPES = ['Multiple Comparisons']", "/DESTINATION FORMAT = TABTEXT ", "IMAGES = NO OUTFILE = '"+TXTPath+".txt'.", "ONEWAY NotaFinal BY GRUPO_NUM /MISSING ANALYSIS /POSTHOC=BONFERRONI ALPHA(0.05).", "OMSEND."};
```

- "OMS /SELECT TABLES", indica el inicio de la serie de comandos y especifica el tipo de los objetos de salida.



- `"/IF COMMANDS = ['Oneway'] SUBTYPES = ['Multiple Comparisons']"`, con este comando conseguimos filtrar los datos para obtener en el archivo de salida solamente los resultados del análisis de Bonferroni.
- `"/DESTINATION FORMAT = TABTEXT "`, Indica el formato de la salida de los datos, en este caso un archivo txt.
- `"IMAGES = NO OUTFILE = TXTPath."`, Indica la ruta donde ubicaremos el archivo de salida y que no queremos que contenga imágenes.
- `"ONEWAY NotaFinal BY GRUPO_NUM /MISSING ANALYSIS /POSTHOC=BONFERRONI ALPHA(0.05)."`, Con este comando ejecutamos el análisis de Bonferroni.
- `"OMSEND."`, indica el final de la serie de comandos.

3.4.- JDBC: Acceso a Bases de Datos

JDBC es el acrónimo de *Java Database Connectivity*, un API incluida en Java, que describe o define una librería estándar para acceso a fuentes de datos, principalmente orientado a Bases de Datos relacionales que usan SQL.

Consiste en un conjunto de clases e interfaces, escritas en Java, que ofrecen un completo API para la programación de bases de datos, por tanto, se trata de una solución bien por cien Java para el acceso a bases de datos.

JDBC no sólo provee un interfaz para acceso a motores de bases de datos, sino que también define una arquitectura estándar, para que los fabricantes puedan crear los drivers que permitan a las aplicaciones Java el acceso a los datos.

Básicamente el API JDBC hace posible realizar las siguientes tareas:

- Establecer una conexión con una base de datos.
- Enviar sentencias SQL.
- Manipular los datos.
- Procesar los resultados de la ejecución de sentencias.

Debido a que JDBC está escrito completamente en Java también posee la ventaja de ser independiente de la plataforma. No será necesario escribir un programa para cada tipo de base de datos, una misma aplicación escrita usando JDBC puede manejar bases de datos Oracle, Sybase, o SQL Server.

Además podrá ejecutarse en cualquier sistema que posea una Máquina Virtual de Java, es decir, serán aplicaciones completamente independientes de la plataforma.



Packages del API JDBC

El API JDBC proporciona una interfaz de programación para acceso a datos de Base de Datos Relacionales desde el lenguaje de programación Java a través de los paquetes (packages):

- El paquete `java.sql` es el corazón de la API JDBC
- El paquete `javax.sql` es la API de extensión estándar JDBC; y proporciona la funcionalidad de fuente de datos (objeto `DataSource`), y agrupación de conexiones (connection pooling)

Package JAVA.SQL

A continuación podemos ver el índice del paquete `java.sql` (núcleo central de JDBC) en el que se muestra todas las clases, interfaces, y excepciones que se encuentran dentro de este paquete:

Interfaces: <ul style="list-style-type: none">• <code>CallableStatement</code>• <code>Connection</code>• <code>DatabaseMetaData</code>• <code>Driver</code>• <code>PreparedStatement</code>• <code>ResultSet</code>• <code>ResultSetMetaData</code>• <code>Statement</code>• <code>SQLData</code>• <code>SQLInput</code>	<ul style="list-style-type: none">• <code>SQLOutput</code>• <code>Array</code>• <code>Blob</code>• <code>Clob</code>• <code>Ref</code>• <code>Struct</code> Clases: <ul style="list-style-type: none">• <code>Date</code>• <code>DriverManager</code>• <code>DriverPropertyInfo</code>	<ul style="list-style-type: none">• <code>Time</code>• <code>Timestamp</code>• <code>Types</code>• <code>SQLPermission</code> Excepciones: <ul style="list-style-type: none">• <code>DataTruncation</code>• <code>SQLException</code>• <code>SQLWarning</code>• <code>BatchUpdateException</code>
--	--	--

De esta enumeración de interfaces, clases y excepciones los más importantes son:

- **`java.sql.DriverManager`:** es la clase gestora de los drivers, se encarga de cargar y seleccionar el driver adecuado para realizar la conexión con una base de datos.
- **`java.sql.Connection`:** representa la conexión con una base de datos.
- **`java.sql.Statement`:** actúa como un contenedor para ejecutar sentencias SQL, sobre una base de datos, contiene dos subtipos:
 - **`java.sql.PreparedStatement`** para la ejecución de sentencias SQL precompiladas a las que se le pueden pasar parámetros de entrada.
 - **`java.sql.CallableStatement`** que permite ejecutar procedimientos almacenados de una base de datos.
- **`java.sql.ResultSet`:** controla el acceso a los resultados de la ejecución de una consulta, es decir, de un objeto `Statement`, permite también la modificación de estos resultados.

- **java.sql.SQLException:** para tratar excepciones que se produzcan al manipular la base de datos, ya sea durante la conexión, desconexión u obtención y modificación de los datos.
- **java.sql.ResultMetaData:** este interfaz obtiene información detallada relativa a un objeto ResultSet determinado.
- **java.sql.DatabaseMetaData:** ofrece información detallada sobre la base de datos a la que nos encontramos conectados.

Package JAVAX.SQL

Respecto al paquete **javax.sql** nos permite:

- Usar el interface **DataSource** como alternativa a **java.sql.DriverManager** para establecer la conexión con la base de datos.
- Uso de pools de conexiones con la base de datos.
- Uso de transacciones distribuidas entre varias bases de datos.
- Uso de Rowsets para trabajar por eventos contra la base de datos, o trabajar en modo desconectado.

3.5.- JFreeChart

JFreeChart es una biblioteca gráfica de código abierto para la plataforma Java, que permite crear gráficos complejos de forma simple. Está diseñado para el uso en aplicaciones, applets, servlets y JSP. JFreeChart se distribuye con código fuente completo sujeto a los términos de GNU Lesser General Public Licence

JFreeChart también trabaja con GNU Classpath, una implementación en software libre de la norma estándar de biblioteca de clases para el lenguaje de programación Java.2.

JFreeChart puede generar gráficos XY (línea, spline y dispersión), gráficos circulares, Diagramas de Gantt, gráficos de barras (horizontales y verticales, apiladas e independientes). También tiene incorporado un dibujador de histogramas, gráficos metro (línea, brújula y termómetro) y Varias gráficas específicas (tabla de viento, gráfica polar, burbujas de diferentes tamaños, etc.)

JFreeChart exporta a formatos de archivo de imagen JPEG (o puede utilizar la biblioteca ImageIO de Java para exportar PNG y a cualquier formato compatible con ImageIO). También puede exportar a cualquier formato con una aplicación Graphics2D. Los datos son accesibles desde cualquier aplicación de las interfaces definidas y puede generar mapas de imagen HTML.

3.6.- HTML



HTML(*HyperText Markup Language*) hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que, en sus diferentes versiones, define una estructura básica y un código para la definición de contenido de una página web, como texto, imágenes, etc. Es un estándar a cargo de la W3C, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación.

Los documentos HTML son ficheros de texto plano (también conocidos como ASCII) que pueden ser editados con cualquier editor de texto o cualquier programa procesador de textos, guardando el documento con la extensión .html o .htm

El lenguaje HTML se basa en la sintaxis SGML (*Standard Generalized Markup Language*), los diferentes elementos (párrafos, encabezamientos, tablas, listas...) del documento se señalan intercalando etiquetas que indican al navegador cómo debe mostrarlo.

El lenguaje HTML basa su filosofía de desarrollo en la referenciación. Para añadir un elemento externo a la página (imagen, vídeo, script, etc.), este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. De este modo, es el navegador web quien se encarga de interpretar el código uniendo todos los elementos para visualizar la página final.

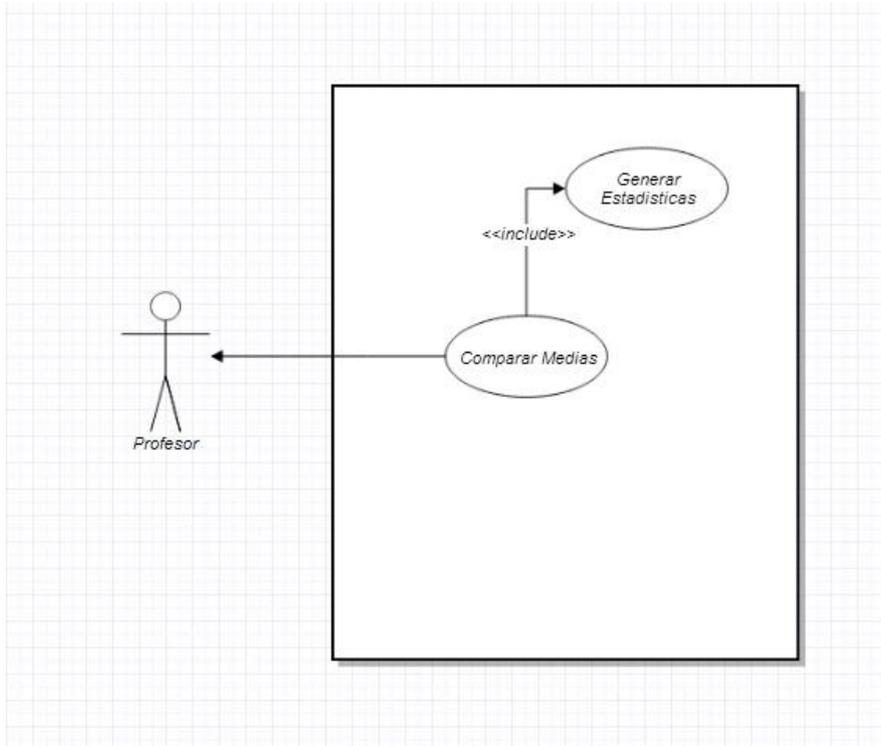
3.7.-PHP

PHP es un lenguaje de scripting, el cual puede ser embebido dentro de páginas HTML. Gran parte de su sintaxis fue tomada de C, Java y Perl con un par de características específicas propias de PHP. Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. PHP puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

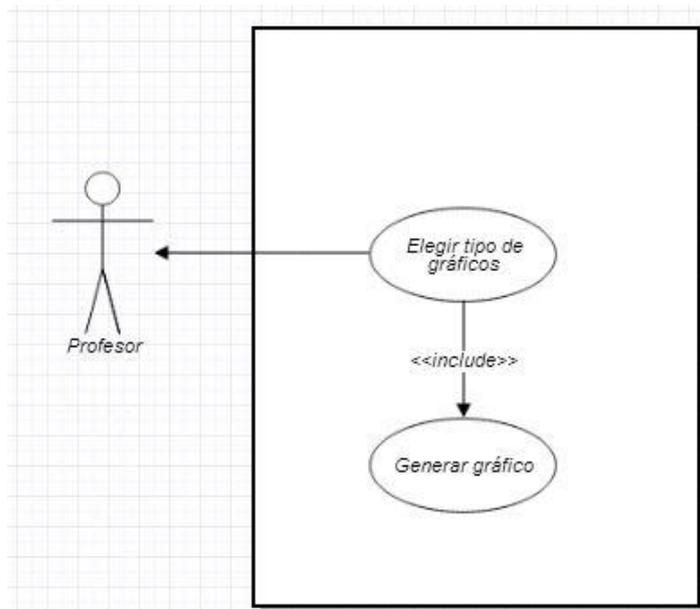
Este lenguaje forma parte del software libre publicado bajo la licencia PHP que es incompatible con la Licencia Pública General de GNU debido a las restricciones del uso del término *PHP*.

3. Casos de uso

4.1.-Comparar medias.



4.2.-Elegir tipo de gráfico.



4. Estructura de la aplicación

A continuación detallaremos la estructura de la aplicación.

El usuario verá una página web en lenguaje html que le permite elegir el tipo de gráfico en que quiere que se muestren los datos. Generamos un botón en código php para poder guardarnos en la variable `tipo_grafico` el tipo de gráfico que ha elegido el usuario y procedemos a hacer una llamada a la aplicación SPSSCall.

```
new SPSSCall(String UrlBD,String login,String password,String anyo,String TXTPath);
```

Una vez que hemos llamado a la clase SPSSCall esta procede a crear la conexión con la base de datos que le hemos pasado por parámetros, una vez establece la conexión recupera los datos de la tabla correspondiente y los guarda en dos ArrayList, uno con las notas y otro con los grupos, una vez hemos guardado todos los datos, procedemos a liberar los recursos, cerrando así la conexión con la base de datos.

Ahora inicializamos el procesador del SPSS con el plug-in de integración que nos proporciona el propio programa y mediante el uso de un dataset creamos una estructura que representará en memoria los datos extraídos de la base de datos, de este modo, nos evitamos generar archivos innecesarios.

Una vez tenemos el dataset activo con los datos en él procedemos a ejecutar el test de Bonferroni mediante el procesador del SPSS, esto nos devuelve un archivo de texto con los datos estadísticos en la ruta que antes hemos especificado.

Acabaremos la ejecución de la clase, parando el procesador del SPSS, ahora con los datos ubicados en el archivo de texto procedemos a generar los gráficos.

Para generar los gráficos utilizaremos las librerías `jcommon-1.0.8.jar` y `jfreechart-1.0.4.jar` del paquete JfreeChart. En el código añadiremos, dependiendo de la gráfica elegida, anteriormente, por el usuario, la siguiente línea de código para crear la gráfica:

```
JFreeChart chart = createChart(dataset);
```

Dependiendo del tipo de gráfico variaran los parámetros que se le pasan la clase. En este caso hemos ejemplificado una llamada generalizada, que crea un gráfico a partir de una base de datos. En este proyecto sólo hemos utilizado dos tipos de gráficos, de barras y de dispersión; la llamada que crea el gráfico de barras es:

```
JFreeChart chart = ChartFactory.createBarChart(titulo, leyenda_eje_x, leyenda_eje_y, datos, orientacion, incluir_leyenda, tooltips, URLs);
```

Y la que crea el gráfico de dispersión es:

```
JFreeChart chart = ChartFactory.createXYLineChart(titulo, leyenda_eje_x,  
leyenda_eje_y, datos, orientacion, incluir_leyenda, tooltips, URLs);
```

Donde incluir_leyenda, tooltips y URLs son de tipo boolean.

Cuando tengamos el gráfico lo guardaremos en disco en formato png con la clase ChartUtilities, también en Java, utilizaremos el formato png ya que es un formato más eficiente para guardar gráficos.

Para guardar los gráficos hemos añadido la siguiente línea de código:

```
ChartUtilities.saveChartAsPNG(archivo, chart, ancho, altura);
```

Donde archivo es la ruta donde se guardará el gráfico, chart es el gráfico que hemos creado anteriormente y ancho y altura corresponden al formato del gráfico

Dependiendo del valor de la variable tipo_grafico se mostrara por pantalla el archivo grafica_barras.png o grafico_dispersion.png, que hemos creado con el código anterior.



5. Pruebas

Para comprobar el correcto funcionamiento de nuestra aplicación vamos a proceder a comparar los resultados extraídos del programa SPSS Statistics para la obtención del test de Bonferroni y los resultados extraídos a través de la ejecución de nuestra aplicación.

Los datos que utilizaremos para probar el correcto funcionamiento de nuestra aplicación son los siguientes, donde cada grupo representa una clase y las notas son la media de las notas de cada alumno.

1																										
2,4	2,7	3	3	3	3	3	3	3	3	3	3	3	5,3	5,5	5,5	5,6	5,9	6,2	6,3	6,6	6,7	6,7	6,9	7,2	7,3	7,5

2																								
9,6	1,3	3	3	3	5	5	5	5,4	5,8	5,9	5,9	6,1	6,2	6,3	6,3	6,4	6,4	6,4	6,7	6,9	6,9	7	7,2	7,4

3																								
2,8	3	3	4	4	4,4	4,4	5	5	5	5,3	5,6	5,6	5,9	5,9	6,1	6,3	6,3	6,3	6,4	6,5	6,6	7,1	7,2	7,2

4																								
2	2,8	3	3	3	3	4,3	4,6	4,7	5	5	5,2	5,7	5,8	5,9	6	6,2	6,7	6,8	6,8	7,1	7,2	7,4	7,5	7,5

5																								
7,2	7,2	7,6	7,6	7,7	7,7	7,9	8	8,1	8,3	8,4	8,5	8,5	8,5	8,6	8,7	8,7	9	9	9,3	9,5	9,5	9,6	10	10

Una vez procesados los datos a través del programa IBM SPSS Statistics con el test de Bonferroni obtenemos los siguientes resultados:

Comparaciones múltiples

Variable dependiente: NotaFinal
Bonferroni

(I) GRUPO	(J) GRUPO	Diferencia de medias (I-J)	Error típico	Sig.	Intervalo de confianza al 95%	
					Límite inferior	Límite superior
1	2	-,91200	,43132	,365	-2,1455	,3215
	3	-,54400	,43132	1,000	-1,7775	,6895
	4	-,43600	,43132	1,000	-1,6695	,7975
	5	-3,67200*	,43132	,000	-4,9055	-2,4385
2	1	,91200	,43132	,365	-,3215	2,1455
	3	,36800	,43132	1,000	-,8655	1,6015
	4	,47600	,43132	1,000	-,7575	1,7095
	5	-2,76000*	,43132	,000	-3,9935	-1,5265
3	1	,54400	,43132	1,000	-,6895	1,7775
	2	-,36800	,43132	1,000	-1,6015	,8655
	4	,10800	,43132	1,000	-1,1255	1,3415
	5	-3,12800*	,43132	,000	-4,3615	-1,8945
4	1	,43600	,43132	1,000	-,7975	1,6695
	2	-,47600	,43132	1,000	-1,7095	,7575
	3	-,10800	,43132	1,000	-1,3415	1,1255
	5	-3,23600*	,43132	,000	-4,4695	-2,0025
5	1	3,67200*	,43132	,000	2,4385	4,9055
	2	2,76000*	,43132	,000	1,5265	3,9935
	3	3,12800*	,43132	,000	1,8945	4,3615
	4	3,23600*	,43132	,000	2,0025	4,4695

*. La diferencia de medias es significativa al nivel 0.05.

Ahora procedemos a analizar los mismos datos con nuestra aplicación para comprobar que ejecuta correctamente el análisis mediante el test de Bonferroni, con lo que obtenemos los siguientes resultados:

Diseño e implementación de una aplicación para realizar análisis estadístico de evaluaciones en las asignaturas de Grado de ETSINF

Multiple Comparisons
Dependent Variable
NotaFinal
Bonferroni

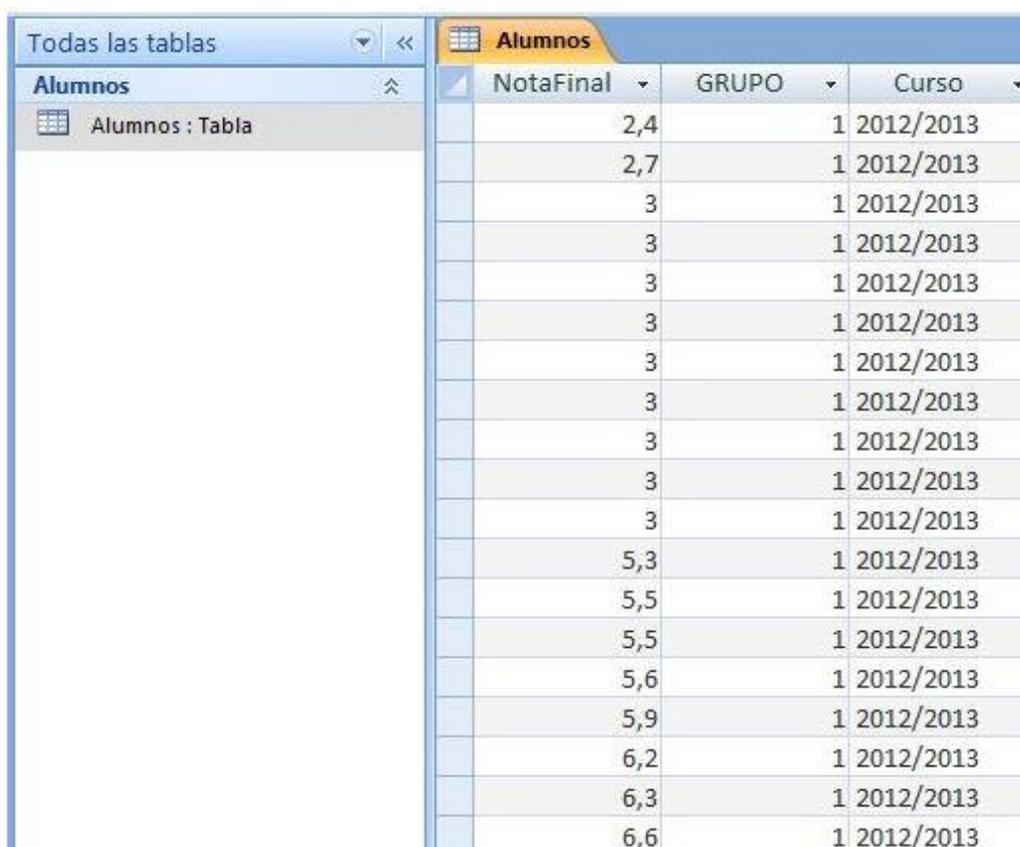
(I) GRUPO	(J) GRUPO	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
1	2	-,91200	,43132	,365	-2,1455	,3215
	3	-,54400	,43132	1,000	-1,7775	,6895
	4	-,43600	,43132	1,000	-1,6695	,7975
	5	-3,67200(*)	,43132	,000	-4,9055	-2,4385
2	1	,91200	,43132	,365	-,3215	2,1455
	3	,36800	,43132	1,000	-,8655	1,6015
	4	,47600	,43132	1,000	-,7575	1,7095
	5	-2,76000(*)	,43132	,000	-3,9935	-1,5265
3	1	,54400	,43132	1,000	-,6895	1,7775
	2	-,36800	,43132	1,000	-1,6015	,8655
	4	,10800	,43132	1,000	-1,1255	1,3415
	5	-3,12800(*)	,43132	,000	-4,3615	-1,8945
4	1	,43600	,43132	1,000	-,7975	1,6695
	2	-,47600	,43132	1,000	-1,7095	,7575
	3	-,10800	,43132	1,000	-1,3415	1,1255
	5	-3,23600(*)	,43132	,000	-4,4695	-2,0025
5	1	3,67200(*)	,43132	,000	2,4385	4,9055
	2	2,76000(*)	,43132	,000	1,5265	3,9935
	3	3,12800(*)	,43132	,000	1,8945	4,3615
	4	3,23600(*)	,43132	,000	2,0025	4,4695

*. The mean difference is significant at the 0.05 level.

Tras observar ambos resultados llegamos a la conclusión de que la aplicación funciona correctamente.

6. Manual de usuario

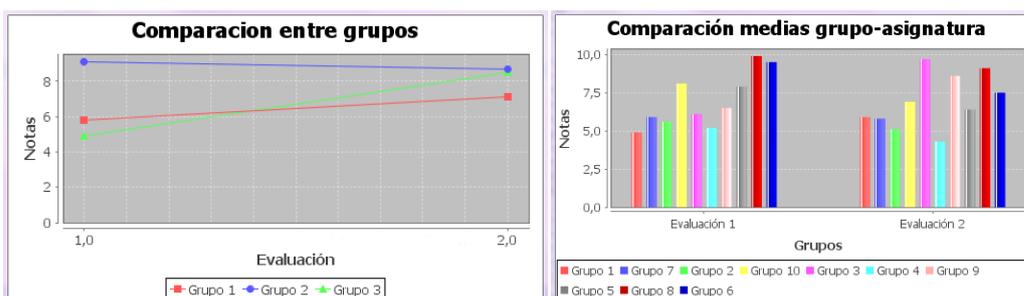
Para poder usar correctamente esta aplicación necesitamos una base de datos con al menos tres variables que deberán estar siempre, que son la nota final, el grupo y el curso, de esta forma cada tupla de la base de datos nos informa de la nota media final de un alumno que pertenece a un grupo en un curso determinado.



NotaFinal	GRUPO	Curso
2,4	1	2012/2013
2,7	1	2012/2013
3	1	2012/2013
3	1	2012/2013
3	1	2012/2013
3	1	2012/2013
3	1	2012/2013
3	1	2012/2013
3	1	2012/2013
3	1	2012/2013
3	1	2012/2013
3	1	2012/2013
5,3	1	2012/2013
5,5	1	2012/2013
5,5	1	2012/2013
5,6	1	2012/2013
5,9	1	2012/2013
6,2	1	2012/2013
6,3	1	2012/2013
6,6	1	2012/2013

Una vez tenemos la Base de Datos definida y con los correspondientes datos en la misma, procederemos a abrir desde un navegador web la herramienta EvalTICS y accedemos al apartado que generará los gráficos y le indicaremos el curso y el tipo de gráfico que deseamos para comprobar visualmente el nivel de los alumnos de los grupos.

Se podrá elegir entre estos dos tipos de gráficos:



Diseño e implementación de una aplicación para realizar análisis estadístico de evaluaciones en las asignaturas de Grado de ETSINF

Una vez hemos elegido el tipo de gráfico la aplicación accederá a la base de datos antes descrita, realizará los cálculos estadísticos correspondientes y los transformará en el tipo de gráfico elegido anteriormente, de esta forma la visualización de los resultados procesados por la aplicación será mucho más cómoda y fácil para el usuario que la esté utilizando.

7. Conclusiones

En este proyecto se han añadido las funcionalidades de generación de estadísticas y generación de gráficos estadísticos. Las nuevas funcionalidades de EvalTICS tras este proyecto son:

1. Generación de estadísticas basadas en el contraste de medias, que nos servirán para comparar grupos.

La aplicación es un componente desarrollado en Java con librerías propias, lo cual facilita su instalación en cualquier entorno.

Durante la realización de este proyecto he aprendido a utilizar entornos y lenguajes de programación como Eclipse y Java y librerías de gestión y de acceso a bases de datos como JDBC.

2. Generación de gráficos estadísticos para una visualización más cómoda y fácil de los resultados.

La aplicación es un componente desarrollado en HTML, PHP y librerías Java de código abierto, por lo que puede ser adaptado a cualquier necesidad.

Durante la realización de este proyecto he aprendido a utilizar lenguajes de programación como Java y herramientas como PHP, HTML y diferentes librerías de gestión de gráficos estadísticos como JFreeChart.



8. Bibliografía

Definición de media:

[http://es.wikipedia.org/wiki/Media_\(matem%C3%A1ticas\)](http://es.wikipedia.org/wiki/Media_(matem%C3%A1ticas))

Definición de desviación típica:

http://es.wikipedia.org/wiki/Desviaci%C3%B3n_est%C3%A1ndar

Test F Fisher:

http://es.wikipedia.org/wiki/Prueba_F_de_Fisher

Test de Bonferroni:

<http://estadisticaorquestainstrumento.wordpress.com/2013/01/28/test-de-bonferroni/>

Portal oficial de Eclipse:

<http://www.eclipse.org/>

Portal oficial de IBM SPSS Statistics:

<http://www-01.ibm.com/software/es/analytics/spss/products/statistics/>

Guía de Usuario del Plug-in de integración con Java:

ftp://public.dhe.ibm.com/software/analytics/spss/documentation/statistics/21.0/en/server/Manuals/Java_Plug-in_User_Guide_for_IBM_SPSS_Statistics.pdf

Portal oficial de Oracle sobre JDBC (Acceso a bases de datos):

<http://www.oracle.com/technetwork/java/overview-141217.html>

Portal oficial de JFreeChart

<http://www.jfree.org/jfreechart/>

Portal oficial de PHP

<http://php.net/>

Descripción de HTML

<http://es.wikipedia.org/wiki/HTML>