

Universidad Autónoma Metropolitana
Unidad Azcapotzalco
División de Ciencias Básicas e Ingeniería
Licenciatura en Ingeniería en Computación

Sistema para la visualización de estadísticas relacionadas con la seriación de
Unidades de Enseñanza y aprendizaje

Modalidad: Proyecto Tecnológico

Trimestre 2018 Primavera

Esteban Araujo de la Rosa
Matrícula: 206300230
Correo electrónico: srearndc@hotmail.com

Josué Figueroa González
Profesor Asociado
Departamento de Sistemas
jfgo@correo.azc.uam.mx

26 de Julio de 2018

Declaratoria

Yo, Josué Figueroa González, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de la Universidad Autónoma Metropolitana Unidad Azcapotzalco.



Josué Figueroa González

Yo, Esteban Araujo de la Rosa, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco.



Esteban Araujo de la Rosa

Resumen

La visualización consiste en transformar información en imágenes que facilitan la extracción de significado. A veces, esa información es cuantitativa, y la visualización creada puede ser llamada “de datos”, en ese caso, el objetivo del gráfico es permitir identificar patrones y tendencias que serían invisibles si esos mismos datos se presentasen en una tabla numérica [1].

Este proyecto utiliza la idea anterior al crear una aplicación Java que, partiendo de una base de datos, facilita el observar la relación entre el desempeño escolar de una UEA y los resultados académicos de la o las UEA seriadas anteriores a través de la visualización de información, en este caso haciendo uso de gráficas de barras.

Tabla de contenido

Declaratoria.....	2
Resumen.....	3
Índice de Figuras.....	6
Introducción.....	7
Antecedentes.....	8
Proyectos Terminales.....	8
Proyectos externos.....	8
Justificación.....	9
Objetivos.....	10
Objetivo General.....	10
Objetivos Específicos.....	10
Marco Teórico.....	11
Visualización.....	11
Java Swing.....	11
JFreeChart.....	11
Hibernate.....	12
Desarrollo del proyecto.....	13
Diseño de base de datos.....	13
Diseño del proyecto.....	14
Interfaz gráfica.....	15
Resultados.....	20
Análisis y discusión de resultados.....	23
Conclusiones.....	24
Referencias.....	25
Anexo A: Código Fuente documentado.....	26
Documentación de las clases.....	26
Referencia de la Clase hibernate.beans.ComparativoAlumnos.....	26
Referencia de la Clase hibernate.beans.Computacion.....	31
Referencia de la Clase hibernate.beans.InfoUEA.....	33
Referencia de la Clase hibernate.beans.InfoUEAAdicional.....	36
Referencia de la Clase hibernate.beans.Kardex.....	39
Referencia de la Clase hibernate.beans.ListasDeIntentos.....	41
Referencia de la Clase hibernate.beans.RequisitosUea.....	43
Referencia de la Clase hibernate.conexion.CrearConexion.....	44
Referencia de la Clase pt.sw.seleccion.formas.FrmBienvenida.....	45
Referencia de la Clase pt.sw.seleccion.formas.FrmVentana1.....	50
Referencia de la Clase pt.sw.seleccion.formas.FrmVentana2.....	62
Referencia de la Clase pt.sw.seleccion.formas.FrmVentana4.....	68
Referencia de la Clase hibernate.beans.OpcionesOportunidadComboBox.....	74
Referencia de la Clase hibernate.operaciones.OperacionesComparativo.....	76
Referencia de la Clase hibernate.operaciones.OperacionesComputacion.....	84
Referencia de la Clase hibernate.operaciones.OperacionesGraficas.....	86
Referencia de la Clase hibernate.operaciones.OperacionesInfoUEA.....	89

Referencia de la Clase hibernate.operaciones.OperacionesKardex.....	93
Referencia de la Clase hibernate.operaciones.OperacionesListasDeIntentos.....	95
Referencia de la Clase hibernate.operaciones.OperacionesRequisitosUea.....	99
Referencia de los mapeos hibernate.mapeos.....	101
Referencia del archivo de configuración hibernate.....	102

Índice de Figuras

Figura 1. Modelo de base de datos.....	13
Figura 2. Funcionalidades del proyecto.....	14
Figura 3. Ventana de Bienvenida y menú principal.....	15
Figura 4. Ventana de bienvenida al ejecutarse la aplicación por primera vez.....	16
Figura 5. Ventana de selección de UEA.....	16
Figura 6. Ventana de selección con UEA seriadas.....	17
Figura 7. Ventana de selección de UEA con lista de seriación.....	17
Figura 8. Gráfico de desempeño.....	18
Figura 9. Gráfico de desempeño con lista de selección.....	18
Figura 10. Gráfica de primera oportunidad.....	19
Figura 11. Gráfica de primera oportunidad vs tercera oportunidad.....	19
Figura 12. Gráfica de primera oportunidad vs primera oportunidad.....	20
Figura 13. Gráfica de primera oportunidad vs segunda oportunidad.....	21
Figura 14. Gráfica de primera oportunidad vs tercera oportunidad.....	21
Figura 15. Gráfica de primera oportunidad vs cuarta oportunidad.....	22
Figura 16. Gráfica de primera oportunidad vs quinta oportunidad.....	22

Introducción

La visualización consiste en transformar información en imágenes que facilitan la extracción de significado. A veces, esa información es cuantitativa, y la visualización creada puede ser llamada “de datos”, en ese caso, el objetivo del gráfico es permitir identificar patrones y tendencias que serían invisibles si esos mismos datos se presentasen en una tabla numérica [1].

Existen numerosas situaciones en las cuales usamos representaciones visuales para entender una variedad de información. Esto puede involucrar cualquier cosa, desde la tendencia del mercado de valores de la semana pasada hasta un itinerario de viaje o incluso la predicción del clima para distintas áreas geográficas. Gracias a nuestra habilidad de percepción visual, una representación visual es a menudo más efectiva que un texto escrito.

Dado que los seres humanos perciben muy bien los atributos visuales, podemos representar un gran número de datos diferentes al “mapearlos” a distintos atributos visuales, por ejemplo, al representar valores numéricos a barras de longitud proporcional al número, con un vistazo rápido (sin leer los números) podemos identificar fácilmente el valor máximo y el mínimo. En este caso, las representaciones visuales, si están bien construidas, pueden ser útiles no solamente para percibir información rápidamente si no también para procesar varias piezas de información al mismo tiempo. No hay que olvidar que el cerebro humano es una “máquina” que procesa constantemente grandes volúmenes de información y datos de manera simultánea [2].

Bajo la premisa de que la visualización de información permite obtener información a partir de datos que, vistos de manera no gráfica, resultarían complejos y que en la Universidad Autónoma Metropolitana (UAM) se tienen datos sobre las calificaciones de los alumnos pero no se ha analizado la relación entre la calificación de un alumno y su desempeño en materias anteriores (seriaciones), se propone un proyecto que busca utilizar la visualización de información para ayudar en la identificación de patrones entre diferentes materias que pudieran servir para la toma de decisiones respecto a los planes de estudio.

Antecedentes

Proyectos Terminales

Visualización de algoritmos de ordenamiento y búsqueda interna con TikZ [3]

Este proyecto se enfoca al desarrollo de una herramienta de software que permite la visualización de algoritmos de ordenamiento y búsqueda interna para facilitar la comprensión y aprendizaje de los mismos.

Visualización de la información generada por la aplicación UAMmero [4]

El propósito del proyecto es mostrar de una manera gráfica, clara y dinámica los resultados de la herramienta computacional UAMmero, que sirve para evaluar el consumo de combustible y de emisiones de vehículos pesados típicos en México, y gracias a esto facilitar la toma de decisiones.

Aplicación Web para la visualización y análisis de datos extraídos del sistema ODBII para la gestión de una flota de camiones [5]

En este proyecto se crea una aplicación web que utiliza los datos obtenidos por un dispositivo ODBII de un vehículo, entre los cuales está el consumo de combustible, ruta recorrida, fallas, etc. para hacer un análisis y mostrar al usuario la información necesaria sobre el manejo de sus recursos.

Proyectos externos

Tableau [6]

Es un software de Inteligencia de Negocio usado para visualización de datos. Es una herramienta interactiva que permite crear visualizaciones en forma de tablas, hojas de trabajo para obtener información del negocio y mejorar el desarrollo de la compañía.

Datawrapper [7]

Es un software amigable, de código abierto para creación de gráficas básicas interactivas, utilizando un conjunto de datos, introducidos ya sea por medio de un archivo CSV o de forma manual, y a partir de los datos, el usuario puede elegir el tipo de gráfica que necesite, finalmente se obtiene un código embebido para publicar el gráfico en el sitio del usuario.

Infogram [8]

Es una plataforma web de visualización de datos y también una plataforma de infográficas, provee un software para creación de infográficas, mapas y reportes de manera intuitiva de suscripción tanto libre como de pago.

Justificación

El desempeño de un alumno en una Unidad de Enseñanza y Aprendizaje (UEA) puede verse afectado por muchas razones, una de ellas podría estar relacionada con su desempeño en UEA anteriores del plan de estudios, por ejemplo, sería interesante saber si aquellos alumnos que tuvieron un buen desempeño en la UEA Taller de Matemáticas tienen un buen desempeño en la siguiente, Introducción al Cálculo. Sin embargo dada la gran cantidad de datos que se tienen, obtener este tipo de información puede resultar complicado. El proyecto propuesto tiene como objetivo hacer uso de visualización de la información para identificar patrones entre los datos de desempeño de los alumnos en ramas de seriación de planes de estudio.

Objetivos

Objetivo General

Diseñar e implementar un sistema para la visualización de estadísticas relacionadas con la seriación de Unidades de Enseñanza y Aprendizaje.

Objetivos Específicos

1. Diseñar e implementar una base de datos con la información académica, plan de estudios y seriación de UEA.
2. Diseñar e implementar un módulo que permita elegir una ruta de seriación y que cargue la información académica relacionada a esa ruta.
3. Diseñar e implementar un módulo que se encargue de obtener y presentar las estadísticas de los alumnos que ya hayan cursado las UEA que formen parte de la seriación.

Marco Teórico

Visualización

No hace mucho tiempo el término visualización significaba construir una imagen visual en la mente (Shorter Oxford English Dictionary, 1972). Ahora, el término ha cambiado a algo más parecido a: la representación gráfica de datos o conceptos. En consecuencia, de ser una construcción interna de la mente, la visualización se ha convertido en un artefacto externo que apoya la toma de decisiones.

Las visualizaciones tienen un rol pequeño pero crucial y creciente en los sistemas cognitivos. Las representaciones visuales proveen el ancho de banda más amplio entre el canal formado por la computadora y los humanos. Adquirimos más información a través de la visión que utilizando los otros sentidos combinados. Los 20 billones o más de neuronas del cerebro dedicadas a analizar la información visual proveen un mecanismo de búsqueda-de-patrones que es un componente fundamental en la mayoría de nuestra actividad cognitiva. Mejorar los sistemas cognitivos a menudo significa amarrar el lazo entre una persona, las herramientas basadas-en-computadora, y otras personas.

Uno de los grandes beneficios de la visualización de datos es la enorme cantidad de información que puede ser interpretada si es presentada correctamente. [10]

Java Swing

Es un paquete parte de la Java Foundation Classes (JFC), la cual provee herramientas o facilidades para la construcción de GUI's o interfaces Gráficas de Usuario (graphical user interface).

Swing es la evolución del AWT (Abstract Window Toolkit), la cual al igual que Swing es un conjunto de bibliotecas enfocadas a la construcción de interfaces, que mejoraron no solo en aspectos visuales sino también en portabilidad y comportamiento.

JFreeChart

JFreeChart es una biblioteca Java para la creación de gráficos. Es de código abierto y una herramienta que facilita implementar gráficos de calidad en aplicaciones Java, está bajo licencia GNU LGPL.

El proyecto JFreeChart inició en Febrero de 2000, soporta varios tipos de gráficas, como son de barras, de pastel, de dispersión entre otros y es compatible con Java Swing y JavaFX.

Hibernate

Hibernate es una herramienta de mapeo objeto-relacional (ORM) para la plataforma Java principalmente que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones.

Hibernate es software libre, distribuido bajo los términos de la licencia GNU LGPL.

Desarrollo del proyecto

Diseño de base de datos

El proyecto se centra en el uso y análisis de los datos académicos de los alumnos de la UAM Azcapotzalco, es importante tener en cuenta que en orden de proteger la privacidad de los estudiantes las matriculas originales se cambiaron y no se usó ningún dato personal, el asesor proporcionó los mencionados datos ya modificados. Los datos utilizados en el desarrollo del proyecto son:

Matrícula del alumno.

Clave de la UEA cursada.

Trimestre en el que cursó la UEA.

Calificación obtenida (MB, B, S, NA, EQ).

Tipo de calificación (global, recuperación).

Seriación de UEA (Sólo se consideró la carrera de Ingeniería en Computación y se ignoraron las UEA en co-registro).

La base de datos a usar se muestra en la Figura 1:

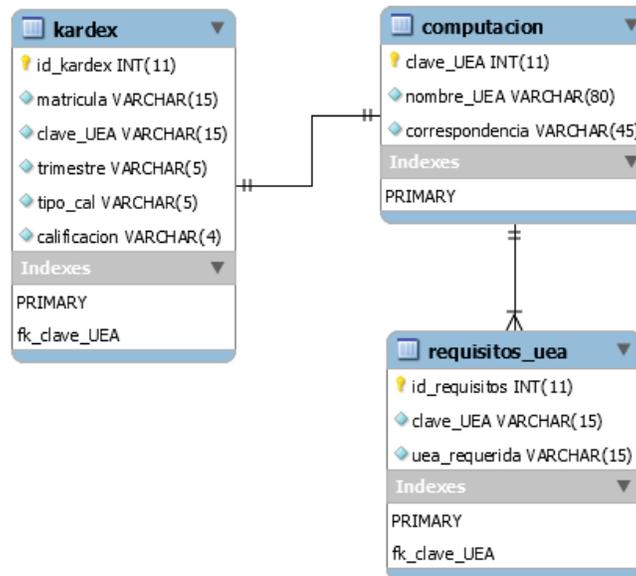


Figura 1. Modelo de base de datos.

Diseño del proyecto

La funcionalidad del proyecto se representa en la Figura 2.



Figura 2. Funcionalidades del proyecto

Módulo de Carga:

Este módulo se encarga de leer(cargar) desde una base de datos tanto la información académica de los alumnos así como la información relacionada con los planes de estudio, en particular el de ingeniería en Computación vigente desde el trimestre 2016 Invierno.

Módulo de Selección:

El objetivo de este módulo es permitir la selección de una UEA almacenada en la base de datos, una vez seleccionada, el sistema carga los datos académicos asociados a la UEA elegida y los mantiene disponibles para el siguiente módulo.

Módulo de Obtención de estadísticas y Visualización:

Este módulo es el encargado de obtener y crear las estadísticas para generar las gráficas correspondientes. Las gráficas estarán basadas en los datos estadísticos de las UEA que forman parte de la seriación de la UEA seleccionada en el módulo anterior. Se presentará información relacionada con el desempeño de los alumnos en la UEA seleccionada y su relación con las UEA que forman parte de la seriación, por ejemplo, si se seleccionó la UEA de “Cálculo Integral”, se mostrará información de la UEA “Cálculo Diferencial” mostrando, de los alumnos que obtuvieron MB en “Cálculo Integral”, que calificación obtuvieron en la inmediata anterior (“Cálculo Diferencial”) y cuantos alumnos la aprobaron y de ellos cuantos obtuvieron cual calificación (MB, B, S). Así para todas las UEA que forman parte de la cadena de seriación.

Especificación Técnica

El proyecto se desarrolló en lenguaje Java, para la parte gráfica se usaron las bibliotecas de Java Swing y para el manejo de datos el framework Hibernate versión 5.2.13 y el servidor MySQL versión 5.7.21, todo integrado con el IDE Netbeans 8.2.

El proyecto se consideró finalizado cuando se cumplieron los siguientes puntos:

- El módulo de Carga almacene los datos correspondientes al historial académico y al plan de estudios y que estén disponibles para el uso de los módulos posteriores.
- El módulo de Selección muestre en pantalla las UEA del plan de estudio y permita seleccionar las UEA que estarán disponibles para trabajar con ellas, como ya se mencionó, son aquellas que solo tienen una seriación inmediata.
- El módulo de Obtención y Visualización procese la información y muestre las gráficas en pantalla. Se consideran sólo gráficas de barra.

Interfaz gráfica

El proyecto se compone de cuatro interfaces gráficas o ventanas, cada una con un propósito específico.

La ventana de bienvenida y menú principal, la cual permite acceder a las diferentes funciones de la aplicación.

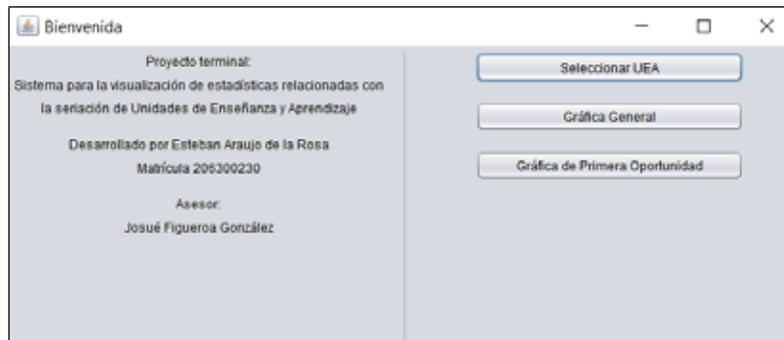


Figura 3. Ventana de Bienvenida y menú principal.

La Figura 3 es la ventana de bienvenida y menú principal, aquí el usuario va a elegir la UEA con la cual quiere trabajar, hay que notar que los botones que nos mandan a las interfaces de graficación no se mostraran a menos que se seleccione una UEA que tenga seriaciones y que sea válida como se muestra en la Figura 4.



Figura 4. Ventana de bienvenida al ejecutarse la aplicación por primera vez.

La siguiente interfaz es la ventana de selección de UEA mostrada en la Figura 5, en esta ventana podemos elegir cualquier UEA que pertenezca al plan de estudios utilizado (En el proyecto solo se utilizó Ingeniería en computación).



Figura 5. Ventana de selección de UEA.

Al ingresar por primera vez a esta ventana, se selecciona la primera UEA dentro de las listas de UEA almacenadas en la base de datos. Además, al seleccionar una UEA se muestra información general de la o las UEA seleccionadas tal y como se observa en la Figura 6.

Si la UEA elegida tiene UEA seriadas anteriores estas se muestran en una lista de selección y además habilitan los botones que nos permiten ir a las interfaces de graficación como se muestra en la Figura 7.

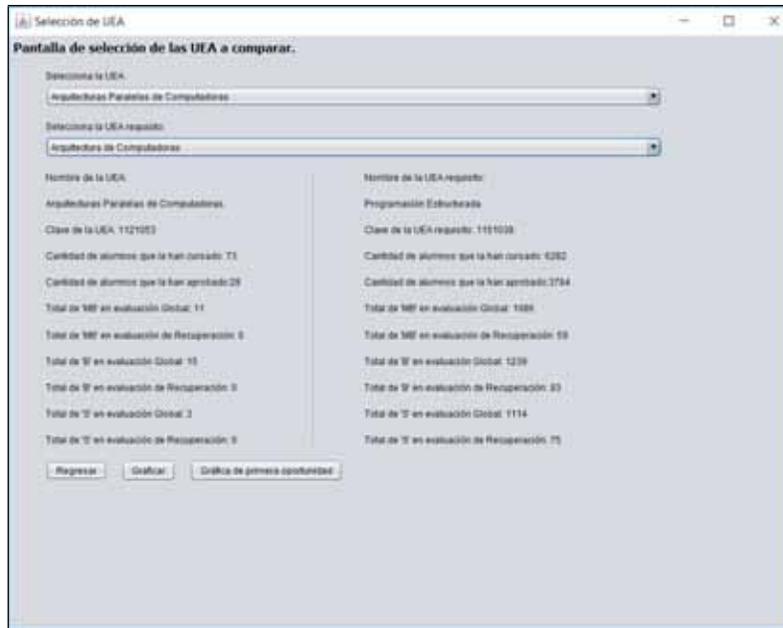


Figura 6. Ventana de selección con UEA seriadas.

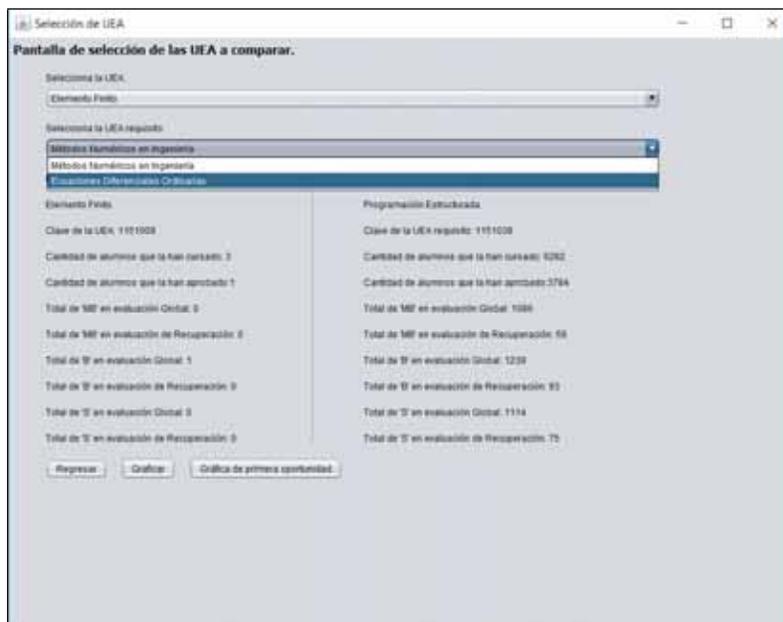


Figura 7. Ventana de selección de UEA con lista de seriación.

La interfaz de graficación o ventana de gráfico de desempeño muestra la comparación entre el desempeño de los alumnos que han aprobado las UEA elegidas, en esta interfaz se grafica el número de alumnos que aprobaron con cierta calificación (MB, B o S) en la UEA elegida vs el número de alumnos que aprobaron con cierta calificación (MB, B o S) en la UEA seriada anterior elegida.

Aquí se puede seleccionar en qué oportunidad se quiere ver la comparación, por ejemplo, si se desea saber la cantidad de alumnos que aprobaron Cinemática y dinámica de partículas la primera vez que la cursaron y que también hayan pasado a la primera la UEA anterior: Introducción a la física, véase la Figura 8.

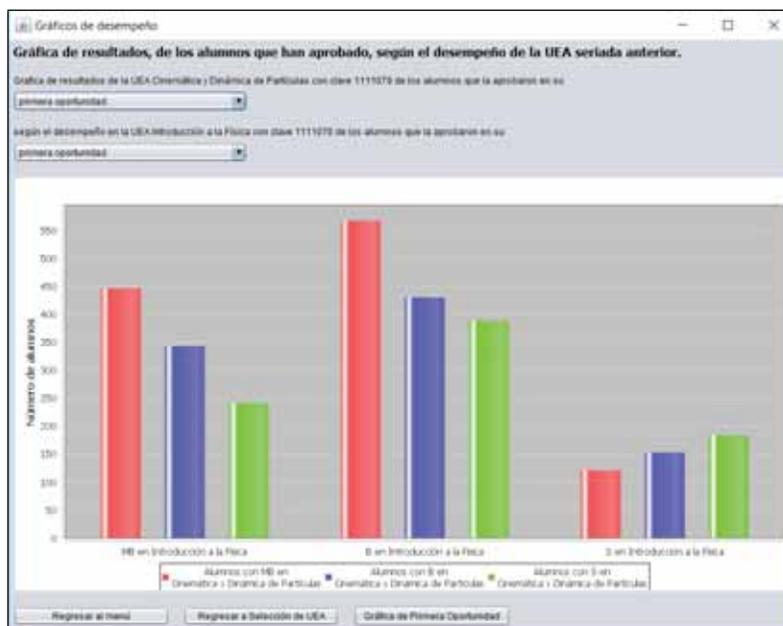


Figura 8. Gráfico de desempeño.

Así pues, se puede seleccionar la comparación de los alumnos que aprobaron una UEA en su tercera oportunidad y que aprobaron la UEA seriada anterior en su segunda oportunidad, o en la primera o en la quinta (recordando que un alumno puede aprobar una UEA a lo más hasta la 5ª vez, haciendo uso de exámenes de recuperación), como ejemplo la Figura 9.

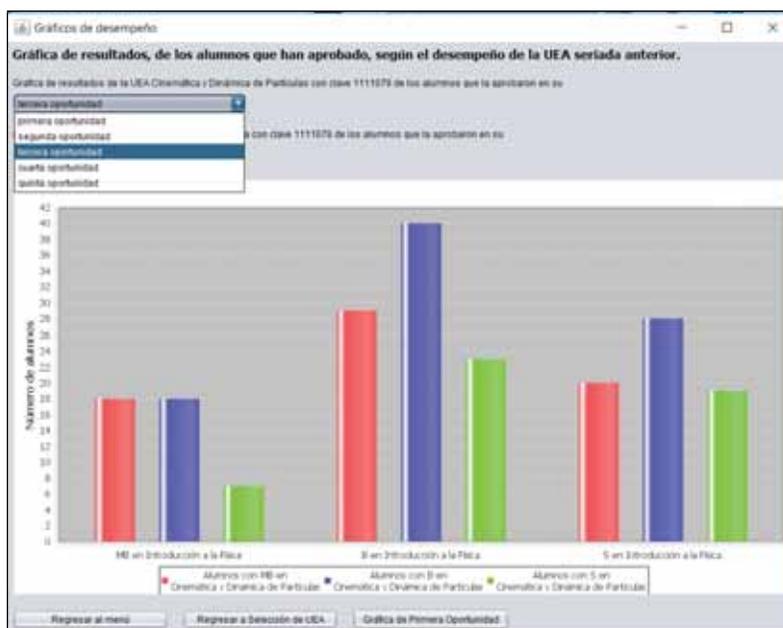


Figura 9. Gráfico de desempeño con lista de selección.

Por último se muestra la interfaz de gráfico de primera oportunidad, mostrada en las Figuras 10 y 11, donde se presenta sólo la comparación entre alumnos que cursaron una

cierta UEA y obtuvieron una cierta calificación (MB, B, S y NA) contra los alumnos que cursaron la UEA seriada anterior en su X oportunidad.

Esta gráfica nos muestra de manera clara como se afectan las calificaciones de los alumnos según el desempeño de la UEA seriada anterior elegida.

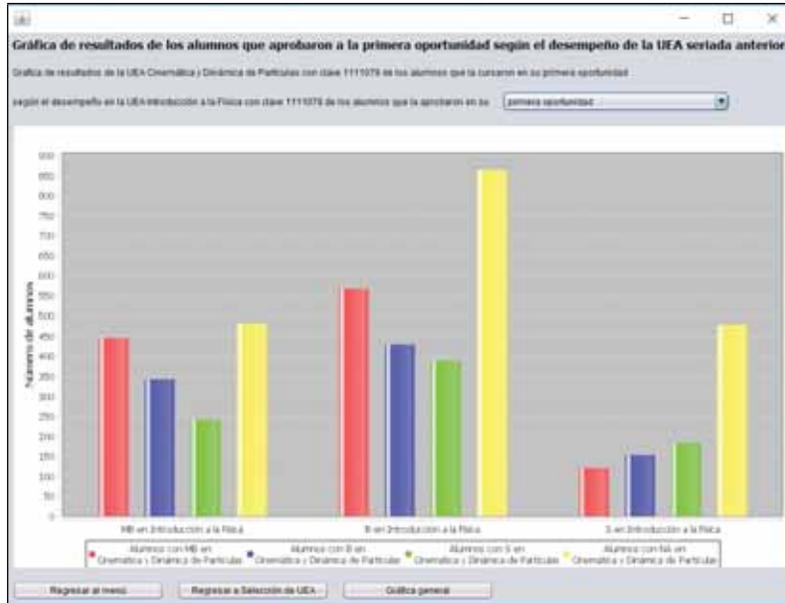


Figura 10. Gráfica de primera oportunidad.

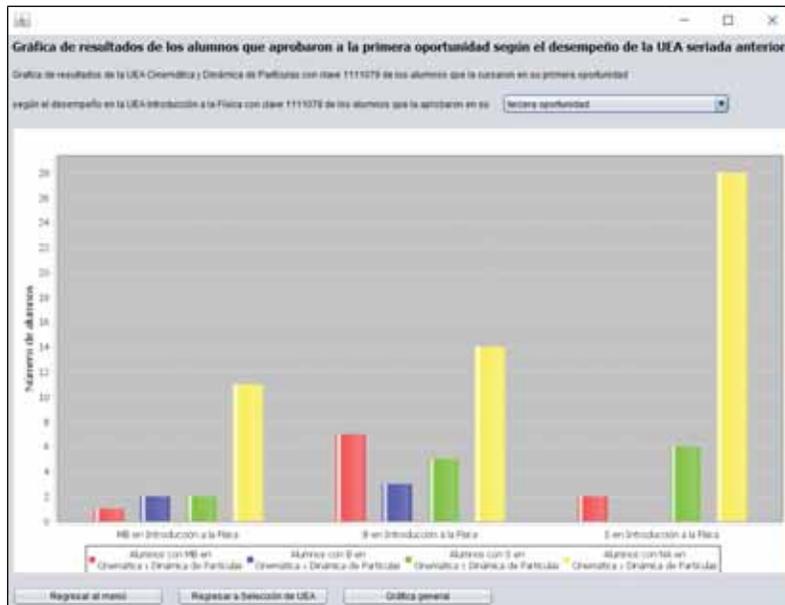


Figura 11. Gráfica de primera oportunidad vs tercera oportunidad.

Resultados

Se logró cumplir con los objetivos planteados: se creó exitosamente una base de datos con la información necesaria, se diseñaron e implementaron los módulos de selección, generación de estadísticas y presentación de gráficas.

Más aún, se puede ver ciertas tendencias en el comportamiento académico de los alumnos, tómesese como ejemplo la UEA Introducción al cálculo, que está seriada con Taller de matemáticas y utilizando las gráficas de primera oportunidad se muestran los resultados.

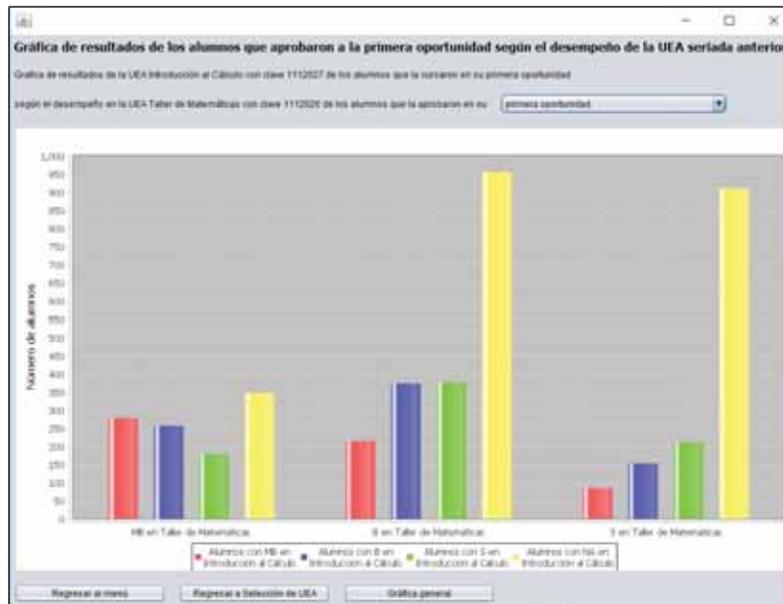


Figura 12. Gráfica de primera oportunidad vs primera oportunidad.

Véase la Figura 12, aquí se ve rápidamente como aumenta el número de reprobados entre los alumnos que obtuvieron una calificación más baja en la materia anterior, es decir, la cantidad de alumnos que reprobaron introducción al cálculo es de menos de 350 para aquellos que obtuvieron MB en taller de matemáticas, pero más de 950 para aquellos que sacaron B, y la proporción de alumnos que reprobaron introducción al cálculo y que obtuvieron S en Taller es más del 50% de la suma total de alumnos que aprobaron.



Figura 13. Gráfica de primera oportunidad vs segunda oportunidad.

Ahora observe la Figura 13, aquí aumenta la proporción de alumnos que han reprobado en Introducción al cálculo, tome en cuenta que esta gráfica muestra aquellos alumnos que cursaron por primera vez introducción al cálculo pero que pasaron Taller de matemáticas (la UEA seriada anterior) a la segunda oportunidad (es decir, no pasaron taller la primera vez que la cursaron).



Figura 14. Gráfica de primera oportunidad vs tercera oportunidad.



Figura 15. Gráfica de primera oportunidad vs cuarta oportunidad.



Figura 16. Gráfica de primera oportunidad vs quinta oportunidad.

Observando las Figuras 14, 15 y 16 es fácil ver que la cantidad de estudiantes que obtuvieron buena calificación en Introducción al cálculo se reduce considerablemente conforme al desempeño en taller de matemáticas pues haber pasado a la quinta oportunidad significa que reprobaron 4 veces la UEA de taller.

Cabe mencionar que la cantidad de entradas en la base de datos de la UEA introducción al cálculo es de 9833 (y la de Taller de matemáticas es de 9487), aunque se filtran los resultados con calificación de EQ (equivalente, ocasionados por cambios en los programas de estudios), la cantidad de datos que se manejan es significativa.

Análisis y discusión de resultados

Como se pudo apreciar en la sección anterior, el objetivo del proyecto ha sido alcanzado y los resultados presentados son una muestra pequeña de lo útil que puede resultar visualizar la enorme cantidad de datos haciendo uso de gráficas, sin embargo, no se puede concluir nada de manera rotunda, es decir, no se puede decir que un alumno con mal desempeño en UEA seriadas anteriores necesariamente vaya a tener un mal desempeño en las siguientes UEA, lo contrario también es cierto, ya que el desempeño no sólo se ve afectado por motivos académicos, existen una multitud de variables que pueden afectar el mismo que van desde el estado emocional, aspectos económicos, sociales, el tener buenos profesores, etc.

Conclusiones

La utilidad de la visualización de datos se puede apreciar de buena manera en este proyecto, al permitir observar relaciones de manera inmediata a través del uso de gráficas y partiendo de un conjunto numeroso de información, en particular de los datos académicos de los estudiantes de la UAM.

El proyecto desarrollado constituye una excelente herramienta si se desean estudiar el desempeño académico de los estudiantes de las materias seriadas, en conjunto con otras herramientas podría dar resultados excepcionales para la planeación y mejoramiento del plan de estudios. Aunque el proyecto está probado con un solo plan de estudios, el de Ingeniería en computación, éste se puede extender a otros planes, siempre que se modifique y/o agregue la información correspondiente en la base de datos.

Referencias

[1] Ignasi Alcalde, “Visualización de la información de los datos al conocimiento,” primera edición. Editorial UOC, 2015.

[2] Riccardo Mazza, “Introduction to Information Visualization,” primera edición. Editorial Springer, Suiza, 2009.

[3] Y. Tovar González y J. A. Bocanegra Rodríguez, “Visualización de algoritmos de ordenamiento y búsqueda interna con TikZ,” Proyecto terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2011.

[4] F. Guadarrama Salto, “Visualización de la información generada por la aplicación UAMmero,” Proyecto terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2016.

[5] A. Luviano Zaldivar, “Aplicación Web para la visualización y análisis de datos extraídos del sistema ODBII para la gestión de una flota de camiones,” Proyecto tecnológico, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2014.

[6] Tableau. [En línea]. Disponible: <https://www.tableau.com/>

[7] Datawrapper (version 4). [En línea]. Disponible: <https://www.datawrapper.de/>

[8] Infogram. [En línea]. Disponible: <https://infogram.com/>

[9] Everac99, “Sueldos IT en México 2015: de startups y desarrollo móvil,” [En línea]. Disponible: <https://everac99.wordpress.com/2015/03/20/sueldos-it-en-mexico-2015-de-startups-y-desarrollo-movil/>

[10] Colin Ware, “Information Visualization, Perception for design,” segunda edición. Editorial Elsevier, 2004.

Anexo A: Código Fuente documentado

Documentación de las clases

Referencia de la Clase `hibernate.beans.ComparativoAlumnos`

Métodos públicos

- `int getAlumnosMBaMBb ()`
 - `void setAlumnosMBaMBb (int alumnosMBaMBb)`
 - `int getAlumnosMBaBb ()`
 - `void setAlumnosMBaBb (int alumnosMBaBb)`
 - `int getAlumnosMBaSb ()`
 - `void setAlumnosMBaSb (int alumnosMBaSb)`
 - `int getAlumnosMBaNAb ()`
 - `void setAlumnosMBaNAb (int alumnosMBaNAb)`
 - `int getAlumnosBaMBb ()`
 - `void setAlumnosBaMBb (int alumnosBaMBb)`
 - `int getAlumnosBaBb ()`
 - `void setAlumnosBaBb (int alumnosBaBb)`
 - `int getAlumnosBaSb ()`
 - `void setAlumnosBaSb (int alumnosBaSb)`
 - `int getAlumnosBaNAb ()`
 - `void setAlumnosBaNAb (int alumnosBaNAb)`
 - `int getAlumnosSaMBb ()`
 - `void setAlumnosSaMBb (int alumnosSaMBb)`
 - `int getAlumnosSaBb ()`
 - `void setAlumnosSaBb (int alumnosSaBb)`
 - `int getAlumnosSaSb ()`
 - `void setAlumnosSaSb (int alumnosSaSb)`
 - `int getAlumnosSaNAb ()`
 - `void setAlumnosSaNAb (int alumnosSaNAb)`
 - `int getAlumnosNAaMBb ()`
 - `void setAlumnosNAaMBb (int alumnosNAaMBb)`
 - `int getAlumnosNAaBb ()`
 - `void setAlumnosNAaBb (int alumnosNAaBb)`
 - `int getAlumnosNAaSb ()`
 - `void setAlumnosNAaSb (int alumnosNAaSb)`
 - `int getAlumnosNAaNAb ()`
 - `void setAlumnosNAaNAb (int alumnosNAaNAb)`
 - **ComparativoAlumnos** (int alumnosMBaMBb, int alumnosMBaBb, int alumnosMBaSb, int alumnosMBaNAb, int alumnosBaMBb, int alumnosBaBb, int alumnosBaSb, int alumnosBaNAb, int alumnosSaMBb, int alumnosSaBb, int alumnosSaSb, int alumnosSaNAb, int alumnosNAaMBb, int alumnosNAaBb, int alumnosNAaSb, int alumnosNAaNAb)
-

Descripción

Clase para almacenar la cantidad de alumnos que obtuvieron calificación X en la UEA y Y en la UEA seriada anterior.

Código

```
package hibernate.beans;

/**
 *
 * @author Esteban
 */
public class ComparativoAlumnos {
    int alumnosMBaMBb=0;
    int alumnosMBaBb=0;
    int alumnosMBaSb=0;
    int alumnosMBaNAb=0;
    int alumnosBaMBb=0;
    int alumnosBaBb=0;
    int alumnosBaSb=0;
    int alumnosBaNAb=0;
    int alumnosSaMBb=0;
    int alumnosSaBb=0;
    int alumnosSaSb=0;
    int alumnosSaNAb=0;
    int alumnosNAaMBb=0;
    int alumnosNAaBb=0;
    int alumnosNAaSb=0;
    int alumnosNAaNAb=0;

    public int getAlumnosMBaMBb() {
        return alumnosMBaMBb;
    }

    public void setAlumnosMBaMBb(int alumnosMBaMBb) {
        this.alumnosMBaMBb = alumnosMBaMBb;
    }

    public int getAlumnosMBaBb() {
        return alumnosMBaBb;
    }

    public void setAlumnosMBaBb(int alumnosMBaBb) {
        this.alumnosMBaBb = alumnosMBaBb;
    }

    public int getAlumnosMBaSb() {
        return alumnosMBaSb;
    }

    public void setAlumnosMBaSb(int alumnosMBaSb) {
        this.alumnosMBaSb = alumnosMBaSb;
    }
}
```

```
public int getAlumnosMBaNAb() {
    return alumnosMBaNAb;
}

public void setAlumnosMBaNAb(int alumnosMBaNAb) {
    this.alumnosMBaNAb = alumnosMBaNAb;
}

public int getAlumnosBaMBb() {
    return alumnosBaMBb;
}

public void setAlumnosBaMBb(int alumnosBaMBb) {
    this.alumnosBaMBb = alumnosBaMBb;
}

public int getAlumnosBaBb() {
    return alumnosBaBb;
}

public void setAlumnosBaBb(int alumnosBaBb) {
    this.alumnosBaBb = alumnosBaBb;
}

public int getAlumnosBaSb() {
    return alumnosBaSb;
}

public void setAlumnosBaSb(int alumnosBaSb) {
    this.alumnosBaSb = alumnosBaSb;
}

public int getAlumnosBaNAb() {
    return alumnosBaNAb;
}

public void setAlumnosBaNAb(int alumnosBaNAb) {
    this.alumnosBaNAb = alumnosBaNAb;
}

public int getAlumnosSaMBb() {
    return alumnosSaMBb;
}

public void setAlumnosSaMBb(int alumnosSaMBb) {
    this.alumnosSaMBb = alumnosSaMBb;
}

public int getAlumnosSaBb() {
    return alumnosSaBb;
}

public void setAlumnosSaBb(int alumnosSaBb) {
    this.alumnosSaBb = alumnosSaBb;
}
```

```

public int getAlumnosSaSb() {
    return alumnosSaSb;
}

public void setAlumnosSaSb(int alumnosSaSb) {
    this.alumnosSaSb = alumnosSaSb;
}

public int getAlumnosSaNAb() {
    return alumnosSaNAb;
}

public void setAlumnosSaNAb(int alumnosSaNAb) {
    this.alumnosSaNAb = alumnosSaNAb;
}

public int getAlumnosNAaMBb() {
    return alumnosNAaMBb;
}

public void setAlumnosNAaMBb(int alumnosNAaMBb) {
    this.alumnosNAaMBb = alumnosNAaMBb;
}

public int getAlumnosNAaBb() {
    return alumnosNAaBb;
}

public void setAlumnosNAaBb(int alumnosNAaBb) {
    this.alumnosNAaBb = alumnosNAaBb;
}

public int getAlumnosNAaSb() {
    return alumnosNAaSb;
}

public void setAlumnosNAaSb(int alumnosNAaSb) {
    this.alumnosNAaSb = alumnosNAaSb;
}

public int getAlumnosNAaNAb() {
    return alumnosNAaNAb;
}

public void setAlumnosNAaNAb(int alumnosNAaNAb) {
    this.alumnosNAaNAb = alumnosNAaNAb;
}

public ComparativoAlumnos(
    int alumnosMBaMBb,
    int alumnosMBaBb,
    int alumnosMBaSb,
    int alumnosMBaNAb,
    int alumnosBaMBb,
    int alumnosBaBb,
    int alumnosBaSb,

```

```

    int alumnosBaNAb,
    int alumnosSaMBb,
    int alumnosSaBb,
    int alumnosSaSb,
    int alumnosSaNAb,
    int alumnosNAaMBb,
    int alumnosNAaBb,
    int alumnosNAaSb,
    int alumnosNAaNAb) {
this.alumnosMBaMBb=alumnosMBaMBb;
this.alumnosMBaBb=alumnosMBaBb;
this.alumnosMBaSb=alumnosMBaSb;
this.alumnosMBaNAb=alumnosMBaNAb;
this.alumnosBaMBb=alumnosBaMBb;
this.alumnosBaBb=alumnosBaBb;
this.alumnosBaSb=alumnosBaSb;
this.alumnosBaNAb=alumnosBaNAb;
this.alumnosSaMBb=alumnosSaMBb;
this.alumnosSaBb=alumnosSaBb;
this.alumnosSaSb=alumnosSaSb;
this.alumnosSaNAb=alumnosSaNAb;
this.alumnosNAaMBb=alumnosNAaMBb;
this.alumnosNAaBb=alumnosNAaBb;
this.alumnosNAaSb=alumnosNAaSb;
this.alumnosNAaNAb=alumnosNAaNAb;
}

public ComparativoAlumnos() {
}
}

```

La documentación para esta clase fue generada a partir del siguiente fichero:

- ComparativoAlumnos.java

Referencia de la Clase hibernate.beans.Computacion

Métodos públicos

- **Computacion** (int claveUEA, String nombreUEA, String correspondencia)
- int **getClaveUEA** ()
- void **setClaveUEA** (int claveUEA)
- String **getNombreUEA** ()
- void **setNombreUEA** (String nombreUEA)
- String **getCorrespondencia** ()
- void **setCorrespondencia** (String correspondencia)
- String **toString** ()
- boolean **CompareTo** (**Computacion** computacion)

Descripción

Clase básica para almacenar información de una UEA: Nombre, clave y correspondencia.

Código

```
package hibernate.beans;

/**
 *
 * @author Esteban
 */
public class Computacion {

    private int claveUEA=0;
    private String nombreUEA="",correspondencia="";

    public Computacion() {
    }

    public Computacion(int claveUEA, String nombreUEA, String correspondencia) {
        this.claveUEA = claveUEA;
        this.nombreUEA = nombreUEA;
        this.correspondencia = correspondencia;
    }

    public int getClaveUEA() {
        return claveUEA;
    }

    public void setClaveUEA(int claveUEA) {
        this.claveUEA = claveUEA;
    }

    public String getNombreUEA() {
        return nombreUEA;
    }

    public void setNombreUEA(String nombreUEA) {
        this.nombreUEA = nombreUEA;
    }

    public String getCorrespondencia() {
```

```
    return correspondencia;
}

public void setCorrespondencia(String correspondencia) {
    this.correspondencia = correspondencia;
}

@Override
public String toString() {
    return this.nombreUEA; //To change body of generated methods, choose Tools | Templates.
}

public boolean CompareTo(Computacion computacion) {
    boolean iguales = false;
    if (this.claveUEA==computacion.claveUEA) {
        if (this.nombreUEA.compareToIgnoreCase(computacion.nombreUEA)==0) {
            iguales = true;
        }
    }
    return iguales;
}
}
```

La documentación para esta clase fue generada a partir del siguiente fichero:

- Computacion.java

Referencia de la Clase hibernate.beans.InfoUEA

Métodos públicos

- **InfoUEA** (int cursado, int aprobado, int mbG, int mbR, int bG, int bR, int sG, int sR)
 - int **getAlumnosQueHanCursado** ()
 - void **setAlumnosQueHanCursado** (int alumnosQueHanCursado)
 - int **getAlumnosQueHanAprobado** ()
 - void **setAlumnosQueHanAprobado** (int alumnosQueHanAprobado)
 - int **getTotalMBGlobal** ()
 - void **setTotalMBGlobal** (int totalMBGlobal)
 - int **getTotalMBRecuperacion** ()
 - void **setTotalMBRecuperacion** (int totalMBRecuperacion)
 - int **getTotalBGlobal** ()
 - void **setTotalBGlobal** (int totalBGlobal)
 - int **getTotalBRecuperacion** ()
 - void **setTotalBRecuperacion** (int totalBRecuperacion)
 - int **getTotalSGlobal** ()
 - void **setTotalSGlobal** (int totalSGlobal)
 - int **getTotalSRecuperacion** ()
 - void **setTotalSRecuperacion** (int totalSRecuperacion)
-

Descripción

Clase para almacenar información concerniente a una UEA.

Código

```
package hibernate.beans;

/**
 *
 * @author Esteban
 */
public class InfoUEA {
    int alumnosQueHanCursado=0;
    int alumnosQueHanAprobado=0;
    int totalMBGlobal=0;
    int totalMBRecuperacion=0;
    int totalBGlobal=0;
    int totalBRecuperacion=0;
    int totalSGlobal=0;
    int totalSRecuperacion=0;

    public InfoUEA() {
    }

    public InfoUEA(int cursado,int aprobado, int mbG, int mbR, int bG, int bR,int sG, int sR) {
```

```
this.alumnosQueHanCursado=cursado;
this.alumnosQueHanAprobado=aprobado;
this.totalMBGlobal=mbG;
this.totalMBRecuperacion=mbR;
this.totalBGlobal=bG;
this.totalBRecuperacion=bR;
this.totalSGlobal=sG;
this.totalSRecuperacion=sR;
}

public int getAlumnosQueHanCursado() {
    return alumnosQueHanCursado;
}

public void setAlumnosQueHanCursado(int alumnosQueHanCursado) {
    this.alumnosQueHanCursado = alumnosQueHanCursado;
}

public int getAlumnosQueHanAprobado() {
    return alumnosQueHanAprobado;
}

public void setAlumnosQueHanAprobado(int alumnosQueHanAprobado) {
    this.alumnosQueHanAprobado = alumnosQueHanAprobado;
}

public int getTotalMBGlobal() {
    return totalMBGlobal;
}

public void setTotalMBGlobal(int totalMBGlobal) {
    this.totalMBGlobal = totalMBGlobal;
}

public int getTotalMBRecuperacion() {
    return totalMBRecuperacion;
}

public void setTotalMBRecuperacion(int totalMBRecuperacion) {
    this.totalMBRecuperacion = totalMBRecuperacion;
}
```

```
public int getTotalBGlobal() {
    return totalBGlobal;
}

public void setTotalBGlobal(int totalBGlobal) {
    this.totalBGlobal = totalBGlobal;
}

public int getTotalBRecuperacion() {
    return totalBRecuperacion;
}

public void setTotalBRecuperacion(int totalBRecuperacion) {
    this.totalBRecuperacion = totalBRecuperacion;
}

public int getTotalSGlobal() {
    return totalSGlobal;
}

public void setTotalSGlobal(int totalSGlobal) {
    this.totalSGlobal = totalSGlobal;
}

public int getTotalSRecuperacion() {
    return totalSRecuperacion;
}

public void setTotalSRecuperacion(int totalSRecuperacion) {
    this.totalSRecuperacion = totalSRecuperacion;
}
}
```

La documentación para esta clase fue generada a partir del siguiente fichero:

- InfoUEA.java

Referencia de la Clase hibernate.beans.InfoUEAAadicional

Métodos públicos

- **InfoUEAAadicional** (int aprobado, int reprobado, int mbG, int mbR, int bG, int bR, int sG, int sR, int i)
- int **getAlumnosQueHanAprobado** ()
- void **setAlumnosQueHanAprobado** (int alumnosQueHanAprobado)
- int **getAlumnosQueHanReprobado** ()
- void **setAlumnosQueHanReprobado** (int alumnosQueHanReprobado)
- int **getTotalMBGlobal** ()
- void **setTotalMBGlobal** (int totalMBGlobal)
- int **getTotalMBRecuperacion** ()
- void **setTotalMBRecuperacion** (int totalMBRecuperacion)
- int **getTotalBGlobal** ()
- void **setTotalBGlobal** (int totalBGlobal)
- int **getTotalBRecuperacion** ()
- void **setTotalBRecuperacion** (int totalBRecuperacion)
- int **getTotalSGlobal** ()
- void **setTotalSGlobal** (int totalSGlobal)
- int **getTotalSRecuperacion** ()
- void **setTotalSRecuperacion** (int totalSRecuperacion)
- int **getIntento** ()
- void **setIntento** (int intento)

Descripción

Clase para el almacenamiento relacionada a la información de la UEA requisito.

Código

```
package hibernate.beans;
/**
 *
 * @author Esteban
 */
public class InfoUEAAadicional {

    int alumnosQueHanAprobado=0;
    int alumnosQueHanReprobado=0;
    int totalMBGlobal=0;
    int totalMBRecuperacion=0;
    int totalBGlobal=0;
    int totalBRecuperacion=0;
    int totalSGlobal=0;
    int totalSRecuperacion=0;
    int intento=0;

    public InfoUEAAadicional() {
    }
}
```

```

public InfoUEAAAdicional(int aprobado, int reprobado, int mbG, int mbR, int bG, int bR,int sG, int sR, int i) {
    this.alumnosQueHanAprobado=aprobado;
    this.alumnosQueHanReprobado=reprobado;
    this.totalMBGlobal=mbG;
    this.totalMBRecuperacion=mbR;
    this.totalBGlobal=bG;
    this.totalBRecuperacion=bR;
    this.totalSGlobal=sG;
    this.totalSRecuperacion=sR;
    this.intento=i;
}

public int getAlumnosQueHanAprobado() {
    return alumnosQueHanAprobado;
}

public void setAlumnosQueHanAprobado(int alumnosQueHanAprobado) {
    this.alumnosQueHanAprobado = alumnosQueHanAprobado;
}

public int getAlumnosQueHanReprobado() {
    return alumnosQueHanReprobado;
}

public void setAlumnosQueHanReprobado(int alumnosQueHanReprobado) {
    this.alumnosQueHanReprobado = alumnosQueHanReprobado;
}

public int getTotalMBGlobal() {
    return totalMBGlobal;
}

public void setTotalMBGlobal(int totalMBGlobal) {
    this.totalMBGlobal = totalMBGlobal;
}

public int getTotalMBRecuperacion() {
    return totalMBRecuperacion;
}

public void setTotalMBRecuperacion(int totalMBRecuperacion) {
    this.totalMBRecuperacion = totalMBRecuperacion;
}

public int getTotalBGlobal() {
    return totalBGlobal;
}

```

```
}

public void setTotalBGlobal(int totalBGlobal) {
    this.totalBGlobal = totalBGlobal;
}

public int getTotalBRecuperacion() {
    return totalBRecuperacion;
}

public void setTotalBRecuperacion(int totalBRecuperacion) {
    this.totalBRecuperacion = totalBRecuperacion;
}

public int getTotalSGlobal() {
    return totalSGlobal;
}

public void setTotalSGlobal(int totalSGlobal) {
    this.totalSGlobal = totalSGlobal;
}

public int getTotalSRecuperacion() {
    return totalSRecuperacion;
}

public void setTotalSRecuperacion(int totalSRecuperacion) {
    this.totalSRecuperacion = totalSRecuperacion;
}

public int getIntento() {
    return intento;
}

public void setIntento(int intento) {
    this.intento = intento;
}
}
```

La documentación para esta clase fue generada a partir del siguiente fichero:

- InfoUEAAadicional.java

Referencia de la Clase hibernate.beans.Kardex

Métodos públicos

- int **getIdKardex** ()
 - void **setIdKardex** (int idKardex)
 - String **getMatricula** ()
 - void **setMatricula** (String matricula)
 - String **getUea** ()
 - void **setUea** (String uea)
 - String **getTrimestre** ()
 - void **setTrimestre** (String trimestre)
 - String **getTipoCal** ()
 - void **setTipoCal** (String tipoCal)
 - String **getCalificacion** ()
 - void **setCalificacion** (String calificacion)
 - String **getTipoAlumno** ()
 - void **setTipoAlumno** (String tipoAlumno)
-

Descripción

Clase para almacenar la información de los alumnos

Código

```
package hibernate.beans;

/**
 *
 * @author Esteban
 */
public class Kardex {

    private int idKardex;
    private String matricula,uea,trimestre,tipoCal,calificacion,tipoAlumno;

    public int getIdKardex() {
        return idKardex;
    }

    public void setIdKardex(int idKardex) {
        this.idKardex = idKardex;
    }

    public String getMatricula() {
        return matricula;
    }

    public void setMatricula(String matricula) {
        this.matricula = matricula;
    }

    public String getUea() {
        return uea;
    }
}
```

```
public void setUea(String uea) {
    this.uea = uea;
}

public String getTrimestre() {
    return trimestre;
}

public void setTrimestre(String trimestre) {
    this.trimestre = trimestre;
}

public String getTipoCal() {
    return tipoCal;
}

public void setTipoCal(String tipoCal) {
    this.tipoCal = tipoCal;
}

public String getCalificacion() {
    return calificacion;
}

public void setCalificacion(String calificacion) {
    this.calificacion = calificacion;
}

public String getTipoAlumno() {
    return tipoAlumno;
}

public void setTipoAlumno(String tipoAlumno) {
    this.tipoAlumno = tipoAlumno;
}
}
```

La documentación para esta clase fue generada a partir del siguiente fichero:

- Kardex.java

Referencia de la Clase hibernate.beans.ListasDeIntentos

Métodos públicos

- List< Kardex > **getLista1** ()
 - void **setLista1** (List< Kardex > lista1)
 - List< Kardex > **getLista2** ()
 - void **setLista2** (List< Kardex > lista2)
 - List< Kardex > **getLista3** ()
 - void **setLista3** (List< Kardex > lista3)
 - List< Kardex > **getLista4** ()
 - void **setLista4** (List< Kardex > lista4)
 - List< Kardex > **getLista5** ()
 - void **setLista5** (List< Kardex > lista5)
-

Descripción

Clase para almacenar y separar la información de los alumnos por intento

Código

```
package hibernate.beans;

import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author Esteban
 */
public class ListasDeIntentos {

    List<Kardex> lista1 = new ArrayList<Kardex>();
    List<Kardex> lista2 = new ArrayList<Kardex>();
    List<Kardex> lista3 = new ArrayList<Kardex>();
    List<Kardex> lista4 = new ArrayList<Kardex>();
    List<Kardex> lista5 = new ArrayList<Kardex>();

    public List<Kardex> getLista1() {
        return lista1;
    }

    public void setLista1(List<Kardex> lista1) {
        this.lista1 = lista1;
    }

    public List<Kardex> getLista2() {
```

```
    return lista2;
}

public void setLista2(List<Kardex> lista2) {
    this.lista2 = lista2;
}

public List<Kardex> getLista3() {
    return lista3;
}

public void setLista3(List<Kardex> lista3) {
    this.lista3 = lista3;
}

public List<Kardex> getLista4() {
    return lista4;
}

public void setLista4(List<Kardex> lista4) {
    this.lista4 = lista4;
}

public List<Kardex> getLista5() {
    return lista5;
}

public void setLista5(List<Kardex> lista5) {
    this.lista5 = lista5;
}
}
```

La documentación para esta clase fue generada a partir del siguiente fichero:

- ListasDeIntentos.java

Referencia de la Clase hibernate.beans.RequisitosUea

Métodos públicos

- int **getIdRequisitos** ()
 - void **setIdRequisitos** (int idRequisitos)
 - String **getUea** ()
 - void **setUea** (String uea)
 - String **getUeaRequerida** ()
 - void **setUeaRequerida** (String ueaRequerida)
-

Descripción

Clase para almacenar las UEA que tienen UEA seriadas requeridas.

Código

```
package hibernate.beans;

/**
 *
 * @author Esteban
 */
public class RequisitosUea {

    private int idRequisitos;
    private String uea,ueaRequerida;

    public int getIdRequisitos() {
        return idRequisitos;
    }

    public void setIdRequisitos(int idRequisitos) {
        this.idRequisitos = idRequisitos;
    }

    public String getUea() {
        return uea;
    }

    public void setUea(String uea) {
        this.uea = uea;
    }

    public String getUeaRequerida() {
        return ueaRequerida;
    }

    public void setUeaRequerida(String ueaRequerida) {
        this.ueaRequerida = ueaRequerida;
    }
}
```

La documentación para esta clase fue generada a partir del siguiente fichero:

- RequisitosUea.java

Referencia de la Clase `hibernate.conexion.CrearConexion`

Métodos públicos estáticos

- static `SessionFactory` `getSessionFactory ()`
 - static void `closeSessionFactory ()`
-

Descripción

Clase de utilidad Hibernate para manejar la conexión con la base de datos.

Código

```
package hibernate.conexion;

import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

/**
 * Hibernate Utility class with a convenient method to get Session Factory
 * object.
 *
 * @author Esteban
 */
public class CrearConexion {

    private static final SessionFactory sessionFactory = buildSessionFactory();

    private static SessionFactory buildSessionFactory() {
        try {
            return new Configuration().configure().buildSessionFactory();
        }
        catch (Throwable ex) {
            System.err.println("Error al crear sessionFactory "+ex);
            throw new ExceptionInInitializerError(ex);
        }
    }

    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }

    public static void closeSessionFactory() {
        sessionFactory.close();
    }
}
```

La documentación para esta clase fue generada a partir del siguiente fichero:

- `CrearConexion.java`

Referencia de la Clase pt.sw.seleccion.formas.FrmBienvenida

Herencias JFrame.

Métodos públicos

- FrmBienvenida (Computacion c1, Computacion c2)

Métodos públicos estáticos

- static void main (String args[])

Descripción

Interfaz de bienvenida y menú principal

Documentación de las funciones miembro

static void pt.sw.seleccion.formas.FrmBienvenida.main (String args[]) [static]

Código

```
package pt.sw.seleccion.formas;

import hibernate.beans.Computacion;
import java.awt.Frame;
import java.awt.Toolkit;
import javax.swing.JFrame;

/**
 *
 * @author Esteban
 */
public class FrmBienvenida extends javax.swing.JFrame {

    /**
     * Creates new form FrmBienvenida
     */
    Computacion cUEA1 = new Computacion();
    Computacion cUEA2 = new Computacion();

    public FrmBienvenida() {
        initComponents();
        this.setExtendedState(Frame.MAXIMIZED_BOTH);
        btnGraficar.setVisible(false);
        btnGraficaPO.setVisible(false);
    }

    public FrmBienvenida(Computacion c1,Computacion c2) {
        initComponents();
        this.setExtendedState(Frame.MAXIMIZED_BOTH);
        cUEA1 = c1;
        cUEA2 = c2;
        btnGraficar.setVisible(true);
    }
}
```

```

    btnGraficaPO.setVisible(true);
}
/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN: initComponents
private void initComponents() {

    lblProyectoTerminal = new javax.swing.JLabel();
    btnSeleccionarUEA = new javax.swing.JButton();
    lblTitulo1 = new javax.swing.JLabel();
    lblTitulo2 = new javax.swing.JLabel();
    lblEAR = new javax.swing.JLabel();
    lblMatrícula = new javax.swing.JLabel();
    lblAsesor = new javax.swing.JLabel();
    lblJFG = new javax.swing.JLabel();
    jSeparator1 = new javax.swing.JSeparator();
    btnGraficar = new javax.swing.JButton();
    btnGraficaPO = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setTitle("Bienvenida");
    setMaximumSize(new java.awt.Dimension(647, 300));
    setMinimumSize(new java.awt.Dimension(647, 300));

    lblProyectoTerminal.setText("Proyecto terminal:");

    btnSeleccionarUEA.setText("Seleccionar UEA");
    btnSeleccionarUEA.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnSeleccionarUEAActionPerformed(evt);
        }
    });

    lblTitulo1.setText("Sistema para la visualización de estadísticas relacionadas con");

    lblTitulo2.setText("la seriación de Unidades de Enseñanza y Aprendizaje");

    lblEAR.setText("Desarrollado por Esteban Araujo de la Rosa");

    lblMatrícula.setText("Matrícula 206300230");

    lblAsesor.setText("Asesor:");

    lblJFG.setText("Josué Figueroa González");

    jSeparator1.setBackground(new java.awt.Color(240, 240, 240));
    jSeparator1.setForeground(new java.awt.Color(240, 240, 240));
    jSeparator1.setOrientation(javax.swing.SwingConstants.VERTICAL);

    btnGraficar.setText("Gráfica General");
    btnGraficar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnGraficarActionPerformed(evt);
        }
    });
});

```

```

btnGraficaPO.setText("Gráfica de Primera Oportunidad");
btnGraficaPO.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnGraficaPO.actionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 50,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                    .addComponent(btnSeleccionarUEA, javax.swing.GroupLayout.DEFAULT_SIZE,
                        javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(btnGraficar, javax.swing.GroupLayout.DEFAULT_SIZE,
                        javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(btnGraficaPO, javax.swing.GroupLayout.DEFAULT_SIZE, 250,
                        Short.MAX_VALUE))
                .addGap(52, Short.MAX_VALUE))
        );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jSeparator1)
                .addGroup(layout.createSequentialGroup()
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(btnSeleccionarUEA)
                            .addGap(18, 18, 18)
                            .addComponent(btnGraficar)
                            .addGap(18, 18, 18)
                            .addComponent(btnGraficaPO))
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(lblProyectoTerminal)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                            .addComponent(lblTitulo1)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                            .addComponent(lblTitulo2)
                            .addGap(18, 18, 18)
                            .addComponent(lblEAR)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                            .addComponent(lblMatrícula)
                            .addGap(18, 18, 18)
                            .addComponent(lblAsesor))
                    )
                )
            )
        );

```

```

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(lblJFG)))
        .addGap(0, 98, Short.MAX_VALUE)))
        .addContainerGap()
    );

    pack();
} // </editor-fold> // GEN-END: initComponents

    private void btnSeleccionarUEAActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_btnSeleccionarUEAActionPerformed
        this.dispose();
        FrmVentana1 ventana1;
        if (cUEA1.getClaveUEA() != 0) ventana1 = new FrmVentana1(cUEA1, cUEA2);
        else ventana1 = new FrmVentana1();
        ventana1.setVisible(true);
    } // GEN-LAST:event_btnSeleccionarUEAActionPerformed

    private void btnGraficarActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_btnGraficarActionPerformed
        if ((cUEA1.getClaveUEA() != 0) && (cUEA2.getClaveUEA() != 0)) {
            this.dispose();
            FrmVentana2 ventana2 = new FrmVentana2(cUEA1, cUEA2);
            ventana2.setVisible(true);
        }
    } // GEN-LAST:event_btnGraficarActionPerformed

    private void btnGraficaPOActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_btnGraficaPOActionPerformed
        if ((cUEA1.getClaveUEA() != 0) && (cUEA2.getClaveUEA() != 0)) {
            this.dispose();
            FrmVentana4 ventana4 = new FrmVentana4(cUEA1, cUEA2);
            ventana4.setVisible(true);
        }
    } // GEN-LAST:event_btnGraficaPOActionPerformed

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    // <editor-fold defaultstate="collapsed" desc="Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(FrmBienvenida.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(FrmBienvenida.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
    } catch (IllegalAccessException ex) {

```

```

        java.util.logging.Logger.getLogger(FrmBienvenida.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(FrmBienvenida.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new FrmBienvenida().setVisible(true);
        }
    });
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton btnGraficaPO;
private javax.swing.JButton btnGraficar;
private javax.swing.JButton btnSeleccionarUEA;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JLabel lblAsesor;
private javax.swing.JLabel lblEAR;
private javax.swing.JLabel lblJFG;
private javax.swing.JLabel lblMatrícula;
private javax.swing.JLabel lblProyectoTerminal;
private javax.swing.JLabel lblTitulo1;
private javax.swing.JLabel lblTitulo2;
// End of variables declaration//GEN-END:variables
}

```

La documentación para esta clase fue generada a partir del siguiente fichero:
FrmBienvenida.java

Referencia de la Clase pt.sw.seleccion.formas.FrmVentana1

Herencias JFrame.

Métodos públicos

- **FrmVentana1** (Computacion c1, Computacion c2)

Métodos públicos estáticos

- static void **main** (String args[])

Descripción

Interfaz de selección.

Documentación de las funciones miembro

static void pt.sw.seleccion.formas.FrmVentana1.main (String *args*[]) [*static*]

Código

```
package pt.sw.seleccion.formas;

import hibernate.beans.Computacion;
import hibernate.beans.InfoUEA;
import hibernate.beans.RequisitosUea;
import hibernate.operaciones.OperacionesComputacion;
import hibernate.operaciones.OperacionesInfoUEA;
import hibernate.operaciones.OperacionesRequisitosUea;
import java.awt.Component;
import java.awt.Frame;
import java.util.Collections;
import java.util.List;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JLabel;
import javax.swing.JList;
import javax.swing.JOptionPane;
import javax.swing.ListCellRenderer;

/**
 *
 * @author Esteban
 */
public class FrmVentana1 extends javax.swing.JFrame {

    /**
     * Creates new form FrmVentana1
     */
    Computacion computacion1 = new Computacion();
    Computacion computacion2 = new Computacion();

    private String claveUEA1,ClaveUEA1Re;
    public FrmVentana1() {
        initComponents();
        this.setExtendedState(Frame.MAXIMIZED_BOTH);
        cmbListaUea.removeAllItems();
        cmbListaUeaReq.removeAllItems();
        ocultarComponentes();
    }
}
```

```

    ocultarInfoUEA();
    llenarListaUeas(computacion1);
}

public FrmVentana1(Computacion c1, Computacion c2) {
    computacion1=c1;
    computacion2=c2;
    initComponents();
    this.setExtendedState(Frame.MAXIMIZED_BOTH);
    cmbListaUea.removeAllItems();
    cmbListaUeaReq.removeAllItems();
    ocultarComponentes();
    ocultarInfoUEA();
    llenarListaUeas(c1);
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-BEGIN: initComponents
private void initComponents() {

    lblListaUea = new javax.swing.JLabel();
    lblListaUeaReq = new javax.swing.JLabel();
    cmbListaUeaReq = new javax.swing.JComboBox<>();
    cmbListaUea = new javax.swing.JComboBox<>();
    lblClaveUEA = new javax.swing.JLabel();
    lblNumCursado = new javax.swing.JLabel();
    lblNumAprobado = new javax.swing.JLabel();
    lblNumMBGlobal = new javax.swing.JLabel();
    lblNumMBRec = new javax.swing.JLabel();
    lblNumBGlobal = new javax.swing.JLabel();
    lblNumBRec = new javax.swing.JLabel();
    lblNumSGlob = new javax.swing.JLabel();
    lblNumSRec = new javax.swing.JLabel();
    btnComparar = new javax.swing.JButton();
    sepBarraDeSeparacion = new javax.swing.JSeparator();
    lblClaveUEAre = new javax.swing.JLabel();
    lblNumCursadoRe = new javax.swing.JLabel();
    lblNumAprobadoRe = new javax.swing.JLabel();
    lblNumMBGlobalRe = new javax.swing.JLabel();
    lblNumMBRecRe = new javax.swing.JLabel();
    lblNumBGlobalRe = new javax.swing.JLabel();
    lblNumBRecRe = new javax.swing.JLabel();
    lblNumSGlobRe = new javax.swing.JLabel();
    lblNumSRecRe = new javax.swing.JLabel();
    lblTitulo = new javax.swing.JLabel();
    lblNombreUEA = new javax.swing.JLabel();
    lblNombreUEAre = new javax.swing.JLabel();
    lblNombreUEA1 = new javax.swing.JLabel();
    lblNombreUEAre1 = new javax.swing.JLabel();
    btnRegresar = new javax.swing.JButton();
    btnGraficaPO = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setTitle("Selección de UEA");
    setMaximumSize(new java.awt.Dimension(1024, 768));
    setMinimumSize(new java.awt.Dimension(1024, 768));

    lblListaUea.setText("Selecciona la UEA:");

    lblListaUeaReq.setText("Selecciona la UEA requisito:");

    cmbListaUeaReq.addItemListener(new java.awt.event.ItemListener() {

```

```

public void itemStateChanged(java.awt.event.ItemEvent evt) {
    cmbListaUeaReqItemStateChanged(evt);
}
});
cmbListaUeaReq.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cmbListaUeaReqActionPerformed(evt);
    }
});

cmbListaUea.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cmbListaUeaActionPerformed(evt);
    }
});
cmbListaUea.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        cmbListaUeaKeyTyped(evt);
    }
});

lblClaveUEA.setText("Información de la UEA:");

lblNumCursado.setText("Cantidad de alumnos que la han cursado: ");
lblNumAprobado.setText("Cantidad de alumnos que la han aprobado:");
lblNumMBGlobal.setText("Total de MB en evaluación Global: ");
lblNumMBRec.setText("Total de MB en evaluación de Recuperación: ");
lblNumBGlobal.setText("Total de B en evaluación Global: ");
lblNumBRec.setText("Total de B en evaluación de Recuperación: ");
lblNumSGlob.setText("Total de S en evaluación Global: ");
lblNumSRec.setText("Total de S en evaluación de Recuperación: ");

btnComparar.setText("Graficar");
btnComparar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCompararActionPerformed(evt);
    }
});

sepBarraDeSeparacion.setOrientation(javax.swing.SwingConstants.VERTICAL);
sepBarraDeSeparacion.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));

lblClaveUEARe.setText("Información de la UEA requisito:");
lblNumCursadoRe.setText("Cantidad de alumnos que la han cursado: ");
lblNumAprobadoRe.setText("Cantidad de alumnos que la han aprobado:");
lblNumMBGlobalRe.setText("Total de MB en evaluación Global: ");
lblNumMBRecRe.setText("Total de MB en evaluación de Recuperación: ");
lblNumBGlobalRe.setText("Total de B en evaluación Global: ");
lblNumBRecRe.setText("Total de B en evaluación de Recuperación: ");
lblNumSGlobRe.setText("Total de S en evaluación Global: ");
lblNumSRecRe.setText("Total de S en evaluación de Recuperación: ");

```

```

lblTitulo.setFont(new java.awt.Font("Tahoma", 1, 16)); // NOI18N
lblTitulo.setText("Pantalla de selección de las UEA a comparar.");

lblNombreUEA.setText("Nombre de la UEA:");

lblNombreUEARe.setText("Nombre de la UEA requisito:");

lblNombreUEA1.setText("Nombre de UEA");

lblNombreUEARe1.setText("Nombre UEA Requisito");

btnRegresar.setText("Regresar");
btnRegresar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnRegresarActionPerformed(evt);
    }
});

btnGraficaPO.setText("Gráfica de primera oportunidad");
btnGraficaPO.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnGraficaPOActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .add(layout.createSequentialGroup()
                        .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .add(layout.createSequentialGroup()
                                .addContainerGap()
                                .addComponent(lblTitulo))
                            .add(layout.createSequentialGroup()
                                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                    .add(layout.createSequentialGroup()
                                        .addGap(48, 48, 48)
                                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                            .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                                                .addComponent(lblListaUeaReq, javax.swing.GroupLayout.Alignment.LEADING)
                                                .addComponent(lblListaUea, javax.swing.GroupLayout.Alignment.LEADING)
                                                .addComponent(cmbListaUea, javax.swing.GroupLayout.Alignment.LEADING, 0,
                                                    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                                .addComponent(cmbListaUeaReq, javax.swing.GroupLayout.Alignment.LEADING,
                                                    javax.swing.GroupLayout.PREFERRED_SIZE, 800, javax.swing.GroupLayout.PREFERRED_SIZE))
                                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
                                                        false)
                                                        .addComponent(lblClaveUEA, javax.swing.GroupLayout.DEFAULT_SIZE,
                                                            javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                                        .addComponent(lblNumCursado, javax.swing.GroupLayout.DEFAULT_SIZE,
                                                            javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                                        .addComponent(lblNumAprobado, javax.swing.GroupLayout.DEFAULT_SIZE,
                                                            javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                                        .addComponent(lblNumMBGlobal, javax.swing.GroupLayout.DEFAULT_SIZE,
                                                            javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                                        .addComponent(lblNumMBRec, javax.swing.GroupLayout.DEFAULT_SIZE,
                                                            javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                                        .addComponent(lblNumBGlobal, javax.swing.GroupLayout.DEFAULT_SIZE,
                                                            javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                                        .addComponent(lblNumBRec, javax.swing.GroupLayout.DEFAULT_SIZE,
                                                            javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                                        .addComponent(lblNumSGlob, javax.swing.GroupLayout.DEFAULT_SIZE,
                                                            javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                                        .addComponent(lblNumSRRec, javax.swing.GroupLayout.PREFERRED_SIZE, 328,
                                                            javax.swing.GroupLayout.PREFERRED_SIZE))
                                                    .addComponent(lblNombreUEA)
                                                )
                                            )
                                        .addComponent(lblNombreUEA)
                                    )
                                .addComponent(lblNombreUEA)
                            )
                        .addComponent(lblNombreUEA)
                    )
                )
            )
        )
    )
    .addComponent(lblNombreUEA)
);

```

```

        .addComponent(lblNombreUEA1))
        .addGap(18, 18, 18)
        .addComponent(sepBarraDeSeparacion, javax.swing.GroupLayout.PREFERRED_SIZE, 50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                .addComponent(lblClaveUEARe, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(lblNumCursadoRe, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(lblNumAprobadoRe, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(lblNumMBGglobalRe, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(lblNumMBRecRe, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(lblNumBGglobalRe, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(lblNumBRecRe, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(lblNumSGlobRe, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(lblNumSRecRe, javax.swing.GroupLayout.PREFERRED_SIZE, 328,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addComponent(lblNombreUEARe1)
                .addComponent(lblNombreUEARe)))
        .addGroup(layout.createSequentialGroup()
            .addComponent(btnRegresar)
            .addGap(18, 18, 18)
            .addComponent(btnComparar)
            .addGap(18, 18, 18)
            .addComponent(btnGraficaPO))))
        .addContainerGap(176, Short.MAX_VALUE))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(lblTitulo)
            .addGap(18, 18, 18)
            .addComponent(lblListaUea)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(cmbListaUea, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(lblListaUeaReq)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(cmbListaUeaReq, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
                    .addComponent(lblNombreUEA)
                    .addGap(18, 18, 18)
                    .addComponent(lblNombreUEA1)
                    .addGap(18, 18, 18)
                    .addComponent(lblClaveUEA)
                    .addGap(18, 18, 18)
                    .addComponent(lblNumCursado)
                    .addGap(18, 18, 18)
                    .addComponent(lblNumAprobado)
                    .addGap(18, 18, 18)
                    .addComponent(lblNumMBGglobal)
                    .addGap(18, 18, 18)
                    .addComponent(lblNumMBRec)

```

```

        .addGap(18, 18, 18)
        .addComponent(lblNumBGlobal)
        .addGap(18, 18, 18)
        .addComponent(lblNumBRec)
        .addGap(18, 18, 18)
        .addComponent(lblNumSGlob)
        .addGap(18, 18, 18)
        .addComponent(lblNumSRec))
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
        .addComponent(sepBarraDeSeparacion, javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(lblNombreUEAre)
            .addGap(18, 18, 18)
            .addComponent(lblNombreUEAre1)
            .addGap(18, 18, 18)
            .addComponent(lblClaveUEAre)
            .addGap(18, 18, 18)
            .addComponent(lblNumCursadoRe)
            .addGap(18, 18, 18)
            .addComponent(lblNumAprobadoRe)
            .addGap(18, 18, 18)
            .addComponent(lblNumMBGlobalRe)
            .addGap(18, 18, 18)
            .addComponent(lblNumMBRecRe)
            .addGap(18, 18, 18)
            .addComponent(lblNumBGlobalRe)
            .addGap(18, 18, 18)
            .addComponent(lblNumBRecRe)
            .addGap(18, 18, 18)
            .addComponent(lblNumSGlobRe)
            .addGap(18, 18, 18)
            .addComponent(lblNumSRecRe))))
    .addGap(18, 18, 18)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(btnRegresar, javax.swing.GroupLayout.PREFERRED_SIZE, 25,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(btnComparar)
        .addComponent(btnGraficaPO))
    .addContainerGap(194, Short.MAX_VALUE)
);

    pack();
} // </editor-fold> // GEN-END: initComponents

        private void cmbListaUeaKeyTyped(java.awt.event.KeyEvent evt) { // GEN-
FIRST:event_cmbListaUeaKeyTyped
        // TODO add your handling code here:
        } // GEN-LAST:event_cmbListaUeaKeyTyped

        private void cmbListaUeaActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_cmbListaUeaActionPerformed
        cmbListaUeaReq.removeAllItems();
        Computacion compu = (Computacion)cmbListaUea.getSelectedItem();
        if (compu.getClaveUEA()!=0) {
            //System.out.println(compu.getClaveUEA()+" y "+compu.getNombreUEA());
            llenarListaUeasReq(compu.getClaveUEA(),compu);
            cmbListaUeaReq.setEnabled(true);
            llenarInfoUEA(compu);
            mostrarInfoUEA();
        }
        else {
            cmbListaUeaReq.setEnabled(false);
            ocultarInfoUEA();
            ocultarComponentes();
        }
        } // GEN-LAST:event_cmbListaUeaActionPerformed

```

```

        private void cmbListaUeaReqActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_cmbListaUeaReqActionPerformed
            if (cmbListaUeaReq.isValid()) {
                Computacion compu = (Computacion) cmbListaUeaReq.getSelectedItem();
                llenarInfoUEARe(compu);
                //System.out.println("hola");
            }
        } //GEN-LAST:event_cmbListaUeaReqActionPerformed

        private void cmbListaUeaReqItemStateChanged(java.awt.event.ItemEvent evt) { //GEN-FIRST:event_cmbListaUeaReqItemStateChanged

        } //GEN-LAST:event_cmbListaUeaReqItemStateChanged

        private void btnCompararActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_btnCompararActionPerformed
            if ((cmbListaUea.isValid()) && (cmbListaUeaReq.isValid())) {
                if (cmbListaUeaReq.isVisible()) {
                    this.dispose();
                    Computacion c1, c2;
                    c1 = (Computacion) cmbListaUea.getSelectedItem();
                    c2 = (Computacion) cmbListaUeaReq.getSelectedItem();
                    FrmVentana2 ventana2 = new FrmVentana2(c1, c2);
                    ventana2.setVisible(true);
                    /*FrmVentana3 ventana3 = new FrmVentana3(c1, c2);
                    ventana3.setVisible(true);
                    FrmVentana4 ventana4 = new FrmVentana4(c1, c2);
                    ventana4.setVisible(true);*/
                }
                else {
                    JOptionPane.showMessageDialog(this, "La UEA elegida no tiene UEA anteriores seriadas", "Error",
                    JOptionPane.ERROR_MESSAGE);
                }
            }
            else {
                JOptionPane.showMessageDialog(this, "Por favor elija una UEA", "Error",
                JOptionPane.ERROR_MESSAGE);
            }
        } //GEN-LAST:event_btnCompararActionPerformed

        private void btnRegresarActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_btnRegresarActionPerformed
            if ((cmbListaUea.isValid()) && (cmbListaUeaReq.isValid())) {
                if (cmbListaUeaReq.isVisible()) {
                    this.dispose();
                    Computacion c1, c2;
                    c1 = (Computacion) cmbListaUea.getSelectedItem();
                    c2 = (Computacion) cmbListaUeaReq.getSelectedItem();
                    FrmBienvenida bienvenida = new FrmBienvenida(c1, c2);
                    bienvenida.setVisible(true);
                    /*FrmVentana3 ventana3 = new FrmVentana3(c1, c2);
                    ventana3.setVisible(true);
                    FrmVentana4 ventana4 = new FrmVentana4(c1, c2);
                    ventana4.setVisible(true);*/
                }
                else {
                    this.dispose();
                    FrmBienvenida bienvenida = new FrmBienvenida();
                    bienvenida.setVisible(true);
                }
            }
            else {
                this.dispose();
                FrmBienvenida bienvenida = new FrmBienvenida();
                bienvenida.setVisible(true);
            }
        } //GEN-LAST:event_btnRegresarActionPerformed

        private void btnGraficaPOActionPerformed(java.awt.event.ActionEvent evt) { //GEN-

```

```

FIRST:event_btnGraficaPOActionPerformed
    if ((cmbListaUea.isValid())&&(cmbListaUeaReq.isValid())) {
        if (cmbListaUeaReq.isVisible()) {
            this.dispose();
            Computacion c1,c2;
            c1 = (Computacion)cmbListaUea.getSelectedItem();
            c2 = (Computacion)cmbListaUeaReq.getSelectedItem();
            FrmVentana4 ventana4 = new FrmVentana4(c1,c2);
            ventana4.setVisible(true);
            /*FrmVentana3 ventana3 = new FrmVentana3(c1,c2);
            ventana3.setVisible(true);
            FrmVentana4 ventana4 = new FrmVentana4(c1,c2);
            ventana4.setVisible(true);*/
        }
        else {
            JOptionPane.showMessageDialog(this, "La UEA elegida no tiene UEA anteriores seriadas", "Error",
JOptionPane.ERROR_MESSAGE);
        }
    } else {
        JOptionPane.showMessageDialog(this, "Por favor elija una UEA", "Error",
JOptionPane.ERROR_MESSAGE);
    }
}
//GEN-LAST:event_btnGraficaPOActionPerformed

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    <!--editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) -->
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
    * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(FrmVentana1.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(FrmVentana1.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(FrmVentana1.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(FrmVentana1.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    }
    <!--/editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new FrmVentana1().setVisible(true);
        }
    });
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton btnComparar;
private javax.swing.JButton btnGraficaPO;

```

```

private javax.swing.JButton btnRegresar;
private javax.swing.JComboBox<Computacion> cmbListaUea;
private javax.swing.JComboBox<Computacion> cmbListaUeaReq;
private javax.swing.JLabel lblClaveUEA;
private javax.swing.JLabel lblClaveUEAre;
private javax.swing.JLabel lblListaUea;
private javax.swing.JLabel lblListaUeaReq;
private javax.swing.JLabel lblNombreUEA;
private javax.swing.JLabel lblNombreUEA1;
private javax.swing.JLabel lblNombreUEAre;
private javax.swing.JLabel lblNombreUEAre1;
private javax.swing.JLabel lblNumAprobado;
private javax.swing.JLabel lblNumAprobadoRe;
private javax.swing.JLabel lblNumBGlobal;
private javax.swing.JLabel lblNumBGlobalRe;
private javax.swing.JLabel lblNumBRec;
private javax.swing.JLabel lblNumBRecRe;
private javax.swing.JLabel lblNumCursado;
private javax.swing.JLabel lblNumCursadoRe;
private javax.swing.JLabel lblNumMBGlobal;
private javax.swing.JLabel lblNumMBGlobalRe;
private javax.swing.JLabel lblNumMBRec;
private javax.swing.JLabel lblNumMBRecRe;
private javax.swing.JLabel lblNumSGlob;
private javax.swing.JLabel lblNumSGlobRe;
private javax.swing.JLabel lblNumSRec;
private javax.swing.JLabel lblNumSRecRe;
private javax.swing.JLabel lblTitulo;
private javax.swing.JSeparator sepBarraDeSeparacion;
// End of variables declaration//GEN-END:variables

private void llenarListaUeas(Computacion compu1) {

    OperacionesComputacion operaciones = new OperacionesComputacion();
    List <Computacion> lista = operaciones.listComputacion();
    System.out.println("Número de UEAS: "+lista.size());
    if (lista.size()>0) {
        Collections.sort(lista, (final Computacion c1, final Computacion c2) ->
c1.getNombreUEA().compareTo(c2.getNombreUEA()));
        for (int i=0;i<lista.size();i++) {
            Computacion compu = lista.get(i);
            cmbListaUea.addItem(compu);
            boolean igual = compu.CompareTo(compu1);
            if (igual) {
                cmbListaUea.setSelectedIndex(i);
            }
        }
        mostrarInfoUEA();
    } else {
        Computacion computacion = new Computacion();
        computacion.setClaveUEA(0);
        computacion.setNombreUEA("La base de datos no contine ningún registro");
        computacion.setCorrespondencia(" ");
        cmbListaUea.addItem(computacion);
        ocultarInfoUEA();
    }
}

private void llenarListaUeasReq(int claveUEA,Computacion compu2) {
    String clave = Integer.toString(claveUEA);
    OperacionesRequisitosUea op = new OperacionesRequisitosUea();
    List <RequisitosUea> listaTmp = op.ueasRequeridas(clave);
    if (listaTmp.size()>0) {
        for (int i=0;i<listaTmp.size();i++) {
            RequisitosUea req = listaTmp.get(i);
            OperacionesComputacion opC = new OperacionesComputacion();
            Computacion compu;

```

```

        compu = opC.datosComputacion(req.getUeaRequerida());
        cmbListaUeaReq.addItem(compu);
        if (compu.CompareTo(compu2)) {
            //System.out.println("iguales");
            cmbListaUeaReq.setSelectedIndex(i);
            llenarInfoUEARE(compu);
        }
    }
    if (compu2.getClaveUEA()==0) {
        cmbListaUeaReq.setSelectedIndex(0);
        llenarInfoUEARE(cmbListaUeaReq.getItemAt(0));
    }
    mostrarComponentes();
} else {
    OperacionesComputacion opC = new OperacionesComputacion();
    Computacion computacion = opC.noUEA();
    cmbListaUeaReq.addItem(computacion);
    ocultarComponentes();
}
}

```

```

private void ocultarInfoUEA() {
    lblNumAprobado.setVisible(false);
    lblNumBGlobal.setVisible(false);
    lblNumBRec.setVisible(false);
    lblClaveUEA.setVisible(false);
    lblNumCursado.setVisible(false);
    lblNumMBGlobal.setVisible(false);
    lblNumMBRec.setVisible(false);
    lblNumSGlob.setVisible(false);
    lblNumSRec.setVisible(false);
    btnComparar.setVisible(false);
    lblNombreUEA.setVisible(false);
    lblNombreUEA1.setVisible(false);
}

```

```

private void mostrarInfoUEA() {
    lblNumAprobado.setVisible(true);
    lblNumBGlobal.setVisible(true);
    lblNumBRec.setVisible(true);
    lblClaveUEA.setVisible(true);
    lblNumCursado.setVisible(true);
    lblNumMBGlobal.setVisible(true);
    lblNumMBRec.setVisible(true);
    lblNumSGlob.setVisible(true);
    lblNumSRec.setVisible(true);
    lblNombreUEA.setVisible(true);
    lblNombreUEA1.setVisible(true);
}

```

```

private void ocultarComponentes() {
    lblListaUeaReq.setVisible(false);
    cmbListaUeaReq.setVisible(false);
    sepBarraDeSeparacion.setVisible(false);
    lblNumAprobadoRe.setVisible(false);
    lblNumBGlobalRe.setVisible(false);
    lblNumBRecRe.setVisible(false);
    lblClaveUEARE.setVisible(false);
    lblNumCursadoRe.setVisible(false);
    lblNumMBGlobalRe.setVisible(false);
    lblNumMBRecRe.setVisible(false);
    lblNumSGlobRe.setVisible(false);
    lblNumSRecRe.setVisible(false);
    btnComparar.setVisible(false);
    lblNombreUEARE.setVisible(false);
    lblNombreUEARE1.setVisible(false);
    btnGraficaPO.setVisible(false);
}

```

```

}

private void mostrarComponentes() {
    lblListaUeaReq.setVisible(true);
    cmbListaUeaReq.setVisible(true);
    sepBarraDeSeparacion.setVisible(true);
    lblNumAprobadoRe.setVisible(true);
    lblNumBGlobalRe.setVisible(true);
    lblNumBRecRe.setVisible(true);
    lblClaveUEARe.setVisible(true);
    lblNumCursadoRe.setVisible(true);
    lblNumMBGlobalRe.setVisible(true);
    lblNumMBRecRe.setVisible(true);
    lblNumSGlobRe.setVisible(true);
    lblNumSRecRe.setVisible(true);
    btnComparar.setVisible(true);
    lblNombreUEARe.setVisible(true);
    lblNombreUEARe1.setVisible(true);
    btnGraficaPO.setVisible(true);
}

private void llenarInfoUEA(Computacion computacion) {
    OperacionesInfoUEA operacionesInfoUEA = new OperacionesInfoUEA();
    InfoUEA infoUEA =
operacionesInfoUEA.adquirirInfoUEA(Integer.toString(computacion.getClaveUEA()));
    lblNombreUEA1.setText(computacion.getNombreUEA());
    lblClaveUEA.setText("Clave de la UEA: "+computacion.getClaveUEA());
    lblNumCursado.setText("Cantidad de alumnos que la han cursado:
"+infoUEA.getAlumnosQueHanCursado());
    lblNumAprobado.setText("Cantidad de alumnos que la han
aprobado:"+infoUEA.getAlumnosQueHanAprobado());
    lblNumBGlobal.setText("Total de 'B' en evaluación Global: "+infoUEA.getTotalBGlobal());
    lblNumBRec.setText("Total de 'B' en evaluación de Recuperación: "+infoUEA.getTotalBRecuperacion());
    lblNumMBGlobal.setText("Total de 'MB' en evaluación Global: "+infoUEA.getTotalMBGlobal());
    lblNumMBRec.setText("Total de 'MB' en evaluación de Recuperación:
"+infoUEA.getTotalMBRecuperacion());
    lblNumSGlob.setText("Total de 'S' en evaluación Global: "+infoUEA.getTotalSGlobal());
    lblNumSRec.setText("Total de 'S' en evaluación de Recuperación: "+infoUEA.getTotalSRecuperacion());
}

private void llenarInfoUEARe(Computacion computacion) {
    String clave = Integer.toString(computacion.getClaveUEA());
    OperacionesInfoUEA operacionesInfoUEA = new OperacionesInfoUEA();
    InfoUEA infoUEA = operacionesInfoUEA.adquirirInfoUEA(clave);
    lblNombreUEARe1.setText(computacion.getNombreUEA());
    lblClaveUEARe.setText("Clave de la UEA requisito: "+clave);
    lblNumCursadoRe.setText("Cantidad de alumnos que la han cursado:
"+infoUEA.getAlumnosQueHanCursado());
    lblNumAprobadoRe.setText("Cantidad de alumnos que la han
aprobado:"+infoUEA.getAlumnosQueHanAprobado());
    lblNumBGlobalRe.setText("Total de 'B' en evaluación Global: "+infoUEA.getTotalBGlobal());
    lblNumBRecRe.setText("Total de 'B' en evaluación de Recuperación: "+infoUEA.getTotalBRecuperacion());
    lblNumMBGlobalRe.setText("Total de 'MB' en evaluación Global: "+infoUEA.getTotalMBGlobal());
    lblNumMBRecRe.setText("Total de 'MB' en evaluación de Recuperación:
"+infoUEA.getTotalMBRecuperacion());
    lblNumSGlobRe.setText("Total de 'S' en evaluación Global: "+infoUEA.getTotalSGlobal());
    lblNumSRecRe.setText("Total de 'S' en evaluación de Recuperación: "+infoUEA.getTotalSRecuperacion());
}
}

class CmbListaUeaRenderer extends JLabel implements ListCellRenderer {
    @Override
    public Component getListCellRendererComponent(JList list, Object value, int index, boolean isSelected,
boolean cellHasFocus) {

        if (value instanceof Computacion) {
            Computacion compu = (Computacion)value;

```

```
        setText(compu.getNombreUEA());
    }
    return this;
}
}
```

La documentación para esta clase fue generada a partir del siguiente fichero:
FrmVentana1.java

Referencia de la Clase pt.sw.seleccion.formas.FrmVentana2

Herencias JFrame.

Métodos públicos

- FrmVentana2 (Computacion c1, Computacion c2)
- String seleccionCalificacion ()

Métodos públicos estáticos

- static void main (String args[])

Descripción

Interfaz de graficación.

Documentación del constructor y destructor

pt.sw.seleccion.formas.FrmVentana2.FrmVentana2 (Computacion c1, Computacion c2)

Parámetros:

<i>c1</i>	UEA
<i>c2</i>	UEA requisito

Documentación de las funciones miembro

static void pt.sw.seleccion.formas.FrmVentana2.main (String args[]) [static]

Código

```
package pt.sw.seleccion.formas;

import hibernate.beans.Computacion;
import hibernate.beans.OpcionesOportunidadComboBox;
import hibernate.operaciones.OperacionesGraficas;
import java.awt.Dimension;
import java.awt.Frame;
import java.util.ArrayList;
import java.util.Enumeration;
import java.util.List;
import javax.swing.AbstractButton;
import javax.swing.JRadioButton;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.data.category.DefaultCategoryDataset;

/**
 *
 */
```

```

* @author Esteban
*/
public class FrmVentana2 extends javax.swing.JFrame {

    /**
     * Creates new form FrmVentana2
     */
    Computacion computacion1 = new Computacion();
    Computacion computacion2 = new Computacion();

    private javax.swing.ButtonGroup bgroupCalificacion;
    public FrmVentana2() {
        initComponents();
    }

    /**
     *
     * @param c1
     * @param c2
     */
    public FrmVentana2(Computacion c1, Computacion c2) {
        initComponents();
        this.setExtendedState(Frame.MAXIMIZED_BOTH);
        lblTitulo1.setText("Grafica de resultados de la UEA "+c1.getNombreUEA()+" con clave "+c1.getClaveUEA()+" de los alumnos que la aprobaron en su");
        lblTitulo2.setText("según el desempeño en la UEA "+c2.getNombreUEA()+" con clave "+c2.getClaveUEA()+" de los alumnos que la aprobaron en su");
        cmbOportunidadUEA.removeAllItems();
        cmbOportunidadUEARE.removeAllItems();
        llenarComboBox(c1,c2);
        crearGraficaDeBarra(c1,1,c2,1);
        computacion1=c1;
        computacion2=c2;
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN: initComponents
    private void initComponents() {

        lblTitulo1 = new javax.swing.JLabel();
        lblTitulo2 = new javax.swing.JLabel();
        pnlContenedor = new javax.swing.JPanel();
        btnRegresarASeleccion = new javax.swing.JButton();
        btnGraficaPO = new javax.swing.JButton();
        cmbOportunidadUEA = new javax.swing.JComboBox<>();
        cmbOportunidadUEARE = new javax.swing.JComboBox<>();
        btnRegresarAMenu = new javax.swing.JButton();
        lblTitulo = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Gráficos de desempeño");
        setMinimumSize(new java.awt.Dimension(1024, 768));

        lblTitulo1.setText("Grafica de resultados de la UEA X de los alumnos que la aprobaron en su.");
        lblTitulo1.setVerticalAlignment(javax.swing.SwingConstants.TOP);

        lblTitulo2.setText("según el desempeño en la UEA Y cuyos alumnos la aprobaron en su.");

        javax.swing.GroupLayout pnlContenedorLayout = new javax.swing.GroupLayout(pnlContenedor);
        pnlContenedor.setLayout(pnlContenedorLayout);
        pnlContenedorLayout.setHorizontalGroup(
            pnlContenedorLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

.addGap(0, 0, Short.MAX_VALUE)
);
pnlContenedorLayout.setVerticalGroup(
    pnlContenedorLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGap(0, 535, Short.MAX_VALUE)
);

btnRegresarASeleccion.setText("Regresar a Selección de UEA");
btnRegresarASeleccion.setMaximumSize(new java.awt.Dimension(200, 25));
btnRegresarASeleccion.setMinimumSize(new java.awt.Dimension(200, 25));
btnRegresarASeleccion.setPreferredSize(new java.awt.Dimension(200, 25));
btnRegresarASeleccion.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnRegresarASeleccionActionPerformed(evt);
    }
});

btnGraficaPO.setText("Gráfica de Primera Oportunidad");
btnGraficaPO.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnGraficaPOActionPerformed(evt);
    }
});

cmbOportunidadUEA.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cmbOportunidadUEAActionPerformed(evt);
    }
});

cmbOportunidadUEARe.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cmbOportunidadUEAReActionPerformed(evt);
    }
});

btnRegresarAMenu.setText("Regresar al menú");
btnRegresarAMenu.setMaximumSize(new java.awt.Dimension(200, 25));
btnRegresarAMenu.setMinimumSize(new java.awt.Dimension(200, 25));
btnRegresarAMenu.setPreferredSize(new java.awt.Dimension(200, 25));
btnRegresarAMenu.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnRegresarAMenuActionPerformed(evt);
    }
});

lblTitulo.setFont(new java.awt.Font("Tahoma", 1, 16)); // NOI18N
lblTitulo.setText("Gráfica de resultados, de los alumnos que han aprobado, según el desempeño de la UEA
seriada anterior.");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(pnlContenedor, javax.swing.GroupLayout.DEFAULT_SIZE,
                Short.MAX_VALUE)
            .addComponent(lblTitulo1, javax.swing.GroupLayout.DEFAULT_SIZE, 1000, Short.MAX_VALUE)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(btnRegresarAMenu, javax.swing.GroupLayout.PREFERRED_SIZE,
                        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(18, 18, 18)
                    .addComponent(btnRegresarASeleccion, javax.swing.GroupLayout.PREFERRED_SIZE,

```

```

javafx.swing.GroupLayout.DEFAULT_SIZE, javafx.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(18, 18, 18)
    .addComponent(btnGraficaPO))
    .addComponent(lblTitulo2)
    .addComponent(cmbOportunidadUEA, javafx.swing.GroupLayout.PREFERRED_SIZE, 300,
javafx.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(cmbOportunidadUEARE, javafx.swing.GroupLayout.PREFERRED_SIZE, 300,
javafx.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(lblTitulo))
    .addGap(0, 0, Short.MAX_VALUE)))
    .addContainerGap())
);
layout.setVerticalGroup(
    layout.createParallelGroup(javafx.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(lblTitulo)
        .addGap(18, 18, 18)
        .addComponent(lblTitulo1)
        .addPreferredGap(javafx.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(cmbOportunidadUEA, javafx.swing.GroupLayout.PREFERRED_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE, javafx.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(lblTitulo2)
        .addPreferredGap(javafx.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(cmbOportunidadUEARE, javafx.swing.GroupLayout.PREFERRED_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE, javafx.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
            .addComponent(pnlContenedor, javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(javafx.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(btnGraficaPO)
            .addComponent(btnRegresarASeleccion, javafx.swing.GroupLayout.PREFERRED_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE, javafx.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(btnRegresarAMenu, javafx.swing.GroupLayout.PREFERRED_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE, javafx.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap())
    );

pack();
} // </editor-fold> // GEN-END: initComponents

private void btnRegresarASeleccionActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_btnRegresarASeleccionActionPerformed
    this.dispose();
    FrmVentana1 ventana1 = new FrmVentana1(computacion1, computacion2);
    ventana1.setVisible(true);
} // GEN-LAST:event_btnRegresarASeleccionActionPerformed

private void cmbOportunidadUEAActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_cmbOportunidadUEAActionPerformed
    if (cmbOportunidadUEA.isValid() && cmbOportunidadUEARE.isValid()) {
        OpcionesOportunidadComboBox opUEA = (OpcionesOportunidadComboBox)
cmbOportunidadUEA.getSelectedItem();
        OpcionesOportunidadComboBox opUEARE = (OpcionesOportunidadComboBox)
cmbOportunidadUEARE.getSelectedItem();
        crearGraficaDeBarra(opUEA.getComputacion(), opUEA.getNumero(), opUEARE.getComputacion(),
opUEARE.getNumero());
    }
} // GEN-LAST:event_cmbOportunidadUEAActionPerformed

private void cmbOportunidadUEAREActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_cmbOportunidadUEAREActionPerformed
    if (cmbOportunidadUEA.isValid() && cmbOportunidadUEARE.isValid()) {
        OpcionesOportunidadComboBox opUEA = (OpcionesOportunidadComboBox)
cmbOportunidadUEA.getSelectedItem();

```

```

        OpcionesOportunidadComboBox opUEARE = (OpcionesOportunidadComboBox)
cmbOportunidadUEARE.getSelectedItem();
        crearGraficaDeBarra(opUEA.getComputacion(), opUEA.getNumero(), opUEARE.getComputacion(),
opUEARE.getNumero());
    }
} //GEN-LAST:event_cmbOportunidadUEAREActionPerformed

    private void btnGraficaPOActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_btnGraficaPOActionPerformed
        this.dispose();
        FrmVentana4 ventana4 = new FrmVentana4(computacion1,computacion2);
        ventana4.setVisible(true);
    } //GEN-LAST:event_btnGraficaPOActionPerformed

    private void btnRegresarAMenuActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_btnRegresarAMenuActionPerformed
        this.dispose();
        FrmBienvenida bienvenida = new FrmBienvenida(computacion1, computacion2);
        bienvenida.setVisible(true);
    } //GEN-LAST:event_btnRegresarAMenuActionPerformed

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    <!--editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) -->
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
    * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(FrmVentana2.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(FrmVentana2.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(FrmVentana2.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(FrmVentana2.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    }
}
</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new FrmVentana2().setVisible(true);
        }
    });
}

// Variables declaration - do not modify //GEN-BEGIN:variables
private javax.swing.JButton btnGraficaPO;
private javax.swing.JButton btnRegresarAMenu;
private javax.swing.JButton btnRegresarASeleccion;
private javax.swing.JComboBox<OpcionesOportunidadComboBox> cmbOportunidadUEA;
private javax.swing.JComboBox<OpcionesOportunidadComboBox> cmbOportunidadUEARE;

```

```

private javax.swing.JLabel lblTitulo;
private javax.swing.JLabel lblTitulo1;
private javax.swing.JLabel lblTitulo2;
private javax.swing.JPanel pnlContenedor;
// End of variables declaration//GEN-END:variables

private void crearGraficaDeBarra(Computacion c1,Integer inta, Computacion c2, Integer intb) {
    pnlContenedor.removeAll();
    OperacionesGraficas op = new OperacionesGraficas();
    //DefaultCategoryDataset dataset = op.datosBarra(c1, c2);
    DefaultCategoryDataset dataset = op.datosBarra(c1,inta,c2,intb);
    String tituloGrafica = " ";
    String tituloEjeX = "";//Calificaciones en "+c2.getNombreUEA());
        JFreeChart barChart = ChartFactory.createBarChart(tituloGrafica, tituloEjeX, "Número de
alumnos",dataset);
    ChartPanel cp = new ChartPanel(barChart);
    Dimension dim = pnlContenedor.getSize();
    cp.setBounds(2, 2, dim.width, dim.height);
    pnlContenedor.add(cp);
    cp.setVisible(true);
    repaint();
}

private String textoG(Integer numero) {
    String conversion="";
    if (null!=numero) switch (numero) {
        case 1:
            conversion="primera oportunidad";
            break;
        case 2:
            conversion="segunda oportunidad";
            break;
        case 3:
            conversion="tercera oportunidad";
            break;
        case 4:
            conversion="cuarta oportunidad";
            break;
        case 5:
            conversion="quinta oportunidad";
            break;
        default:
            break;
    }
    return conversion;
}

private void llenarComboBox(Computacion c1, Computacion c2) {
    for (int i=1;i<6;i++) {
        OpcionesOportunidadComboBox tmp1 = new OpcionesOportunidadComboBox(i,textoG(i),c1);
        OpcionesOportunidadComboBox tmp2 = new OpcionesOportunidadComboBox(i,textoG(i),c2);
        cmbOportunidadUEA.addItem(tmp1);
        cmbOportunidadUEARE.addItem(tmp2);
    }
    cmbOportunidadUEA.setSelectedIndex(0);
    cmbOportunidadUEARE.setSelectedIndex(0);
}
}

```

La documentación para esta clase fue generada a partir del siguiente fichero:
FrmVentana2.java

Referencia de la Clase pt.sw.seleccion.formas.FrmVentana4

Herencias JFrame.

Métodos públicos

- **FrmVentana4** (Computacion c1, Computacion c2)

Métodos públicos estáticos

- static void **main** (String args[])

Descripción

Interfaz de gráfica de primera oportunidad.

Documentación de las funciones miembro

static void pt.sw.seleccion.formas.FrmVentana4.main (String *args*[]) [*static*]

Código

```
package pt.sw.seleccion.formas;

import hibernate.beans.Computacion;
import hibernate.beans.OpcionesOportunidadComboBox;
import hibernate.operaciones.OperacionesGraficas;
import java.awt.Dimension;
import java.awt.Frame;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.data.category.DefaultCategoryDataset;

/**
 *
 * @author Esteban
 */
public class FrmVentana4 extends javax.swing.JFrame {

    /**
     * Creates new form FrmVentana4
     */
    Computacion computacion1 = new Computacion();
    Computacion computacion2 = new Computacion();

    public FrmVentana4() {
        initComponents();
    }

    public FrmVentana4(Computacion c1, Computacion c2) {
        initComponents();
        this.setExtendedState(Frame.MAXIMIZED_BOTH);
        lblTitulo1.setText("Grafica de resultados de la UEA "+c1.getNombreUEA()+" con clave "+c1.getClaveUEA()+" de los alumnos que la cursaron en su primera oportunidad");
        lblTitulo2.setText("según el desempeño en la UEA "+c2.getNombreUEA()+" con clave "+c2.getClaveUEA()+" de los alumnos que la aprobaron en su");
        cmbOportunidadUEARe.removeAllItems();
        llenarComboBox(c2);
    }
}
```

```

    crearGraficaDeBarra(c1,1,c2,1);
    computacion1=c1;
    computacion2=c2;
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents
private void initComponents() {

    lblTitulo1 = new javax.swing.JLabel();
    lblTitulo2 = new javax.swing.JLabel();
    pnlContenedor = new javax.swing.JPanel();
    btnRegresarASeleccion = new javax.swing.JButton();
    cmbOportunidadUEARe = new javax.swing.JComboBox<>();
    btnRegresarAMenu = new javax.swing.JButton();
    lblTitulo = new javax.swing.JLabel();
    btnGraficaGeneral = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setMaximumSize(new java.awt.Dimension(1024, 768));
    setMinimumSize(new java.awt.Dimension(1024, 768));

    lblTitulo1.setText("Grafica de resultados de la UEA X de los alumnos que la aprobaron en su primera
oportunidad");
    lblTitulo1.setVerticalAlignment(javax.swing.SwingConstants.TOP);

    lblTitulo2.setText("según el desempeño en la UEA Y cuyos alumnos la aprobaron en su:");

    javax.swing.GroupLayout pnlContenedorLayout = new javax.swing.GroupLayout(pnlContenedor);
    pnlContenedor.setLayout(pnlContenedorLayout);
    pnlContenedorLayout.setHorizontalGroup(
        pnlContenedorLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 0, Short.MAX_VALUE)
    );
    pnlContenedorLayout.setVerticalGroup(
        pnlContenedorLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 587, Short.MAX_VALUE)
    );

    btnRegresarASeleccion.setText("Regresar a Selección de UEA");
    btnRegresarASeleccion.setMaximumSize(new java.awt.Dimension(200, 25));
    btnRegresarASeleccion.setMinimumSize(new java.awt.Dimension(200, 25));
    btnRegresarASeleccion.setPreferredSize(new java.awt.Dimension(200, 25));
    btnRegresarASeleccion.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnRegresarASeleccionActionPerformed(evt);
        }
    });

    cmbOportunidadUEARe.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            cmbOportunidadUEAReActionPerformed(evt);
        }
    });

    btnRegresarAMenu.setText("Regresar al menú");
    btnRegresarAMenu.setMaximumSize(new java.awt.Dimension(200, 25));
    btnRegresarAMenu.setMinimumSize(new java.awt.Dimension(200, 25));
    btnRegresarAMenu.setPreferredSize(new java.awt.Dimension(200, 25));
    btnRegresarAMenu.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnRegresarAMenuActionPerformed(evt);
        }
    });

```

```

    }
    });

    lblTitulo.setFont(new java.awt.Font("Tahoma", 1, 16)); // NOI18N
    lblTitulo.setText("Gráfica de resultados de los alumnos que aprobaron a la primera oportunidad según el
    desempeño de la UEA seriada anterior.");

    btnGraficaGeneral.setText("Gráfica general");
    btnGraficaGeneral.setMaximumSize(new java.awt.Dimension(200, 25));
    btnGraficaGeneral.setMinimumSize(new java.awt.Dimension(200, 25));
    btnGraficaGeneral.setPreferredSize(new java.awt.Dimension(200, 25));
    btnGraficaGeneral.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnGraficaGeneralActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(pnlContenedor, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(lblTitulo1, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addGroup(layout.createSequentialGroup()
                        .addContainerGap()
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(btnRegresarAMenu, javax.swing.GroupLayout.PREFERRED_SIZE,
                                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGap(18, 18, 18)
                            .addComponent(btnRegresarASeleccion, javax.swing.GroupLayout.PREFERRED_SIZE,
                                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGap(18, 18, 18)
                            .addComponent(btnGraficaGeneral, javax.swing.GroupLayout.PREFERRED_SIZE,
                                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addGroup(layout.createSequentialGroup()
                            .addContainerGap()
                            .addComponent(lblTitulo2)
                            .addGap(18, 18, 18)
                            .addComponent(cmbOportunidadUEARe, javax.swing.GroupLayout.PREFERRED_SIZE,
                                300, javax.swing.GroupLayout.PREFERRED_SIZE))
                            .addComponent(lblTitulo))
                        .addGap(0, 2, Short.MAX_VALUE)))
                    .addContainerGap())
                .addContainerGap()
            );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(lblTitulo)
                    .addGap(18, 18, 18)
                    .addComponent(lblTitulo1)
                    .addGap(18, 18, 18)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(lblTitulo2)
                        .addComponent(cmbOportunidadUEARe, javax.swing.GroupLayout.PREFERRED_SIZE,
                            javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGap(18, 18, 18)
                    .addComponent(pnlContenedor, javax.swing.GroupLayout.DEFAULT_SIZE,
                        javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addGap(18, 18, 18)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(btnRegresarASeleccion, javax.swing.GroupLayout.PREFERRED_SIZE,
                            javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(btnRegresarAMenu, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(btnGraficaGeneral, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap()
    );

    pack();
} // </editor-fold> // GEN-END: initComponents

    private void btnRegresarASeleccionActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_btnRegresarASeleccionActionPerformed
        this.dispose();
        FrmVentana1 ventana1 = new FrmVentana1(computacion1, computacion2);
        ventana1.setVisible(true);
    } // GEN-LAST:event_btnRegresarASeleccionActionPerformed

    private void cmbOportunidadUEAreActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_cmbOportunidadUEAreActionPerformed

        OpcionesOportunidadComboBox opUEAre = (OpcionesOportunidadComboBox)
cmbOportunidadUEAre.getSelectedItem();
        crearGraficaDeBarra(computacion1, 1, opUEAre.getComputacion(), opUEAre.getNumero());

    } // GEN-LAST:event_cmbOportunidadUEAreActionPerformed

    private void btnRegresarAMenuActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_btnRegresarAMenuActionPerformed
        this.dispose();
        FrmBienvenida bienvenida = new FrmBienvenida(computacion1, computacion2);
        bienvenida.setVisible(true);
    } // GEN-LAST:event_btnRegresarAMenuActionPerformed

    private void btnGraficaGeneralActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_btnGraficaGeneralActionPerformed
        this.dispose();
        FrmVentana2 ventana2 = new FrmVentana2(computacion1, computacion2);
        ventana2.setVisible(true);
    } // GEN-LAST:event_btnGraficaGeneralActionPerformed

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(FrmVentana4.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(FrmVentana4.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(FrmVentana4.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

```

```

        java.util.logging.Logger.getLogger(FrmVentana4.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    }
//</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new FrmVentana4().setVisible(true);
        }
    });
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton btnGraficaGeneral;
private javax.swing.JButton btnRegresarAMenu;
private javax.swing.JButton btnRegresarASeleccion;
private javax.swing.JComboBox<OpcionesOportunidadComboBox> cmbOportunidadUEARE;
private javax.swing.JLabel lblTitulo;
private javax.swing.JLabel lblTitulo1;
private javax.swing.JLabel lblTitulo2;
private javax.swing.JPanel pnlContenedor;
// End of variables declaration//GEN-END:variables

private void crearGraficaDeBarra(Computacion c1, int in1, Computacion c2, int in2) {

    pnlContenedor.removeAll();
    OperacionesGraficas op = new OperacionesGraficas();
    DefaultCategoryDataset dataset = op.datosBarraConNA(c1,in1,c2,in2);
    String tituloGrafica = " ";
    JFreeChart barChart = ChartFactory.createBarChart(tituloGrafica,"", "Número de alumnos",dataset);
    ChartPanel cp = new ChartPanel(barChart);
    cp.setVisible(true);
    Dimension dim = pnlContenedor.getSize();
    cp.setBounds(2, 2, dim.width, dim.height);
    pnlContenedor.add(cp);
    repaint();
}

private void llenarComboBox(Computacion c2) {
    for (int i=1;i<6;i++) {
        OpcionesOportunidadComboBox tmp = new OpcionesOportunidadComboBox(i,textoG(i),c2);
        cmbOportunidadUEARE.addItem(tmp);
    }
    cmbOportunidadUEARE.setSelectedIndex(0);
}

private String textoG(Integer numero) {
    String conversion="";
    if (null!=numero) switch (numero) {
        case 1:
            conversion="primera oportunidad";
            break;
        case 2:
            conversion="segunda oportunidad";
            break;
        case 3:
            conversion="tercera oportunidad";
            break;
        case 4:
            conversion="cuarta oportunidad";
            break;
        case 5:
            conversion="quinta oportunidad";
            break;
        default:
            break;
    }
}

```

```
}  
    return conversion;  
}  
}
```

La documentación para esta clase fue generada a partir del siguiente fichero:
FrmVentana4.java

Referencia de la Clase

hibernate.beans.OpcionesOportunidadComboBox

Métodos públicos

- **OpcionesOportunidadComboBox** (Integer n, String t)
 - **OpcionesOportunidadComboBox** (Integer n, String t, **Computacion** c)
 - int **getNumero** ()
 - void **setNumero** (int numero)
 - String **getTexto** ()
 - void **setTexto** (String texto)
 - **Computacion** **getComputacion** ()
 - void **setComputacion** (**Computacion** computacion)
 - String **toString** ()
-

Descripción

Clase para facilitar el manejo de la lista de UEA con un JComboBox de java swing.

Código

```
package hibernate.beans;

/**
 *
 * @author Esteban
 */
public class OpcionesOportunidadComboBox {
    int numero=0;
    String texto="";
    Computacion computacion = new Computacion();

    public OpcionesOportunidadComboBox() {
    }

    public OpcionesOportunidadComboBox(Integer n, String t) {
        this.numero=n;
        this.texto=t;
    }

    public OpcionesOportunidadComboBox(Integer n, String t, Computacion c) {
        this.numero=n;
        this.texto=t;
        this.computacion=c;
    }
}
```

```
public int getNumero() {
    return numero;
}

public void setNumero(int numero) {
    this.numero = numero;
}

public String getTexto() {
    return texto;
}

public void setTexto(String texto) {
    this.texto = texto;
}

public Computacion getComputacion() {
    return computacion;
}

public void setComputacion(Computacion computacion) {
    this.computacion = computacion;
}

@Override
public String toString() {
    return this.texto;
}
}
```

La documentación para esta clase fue generada a partir del siguiente fichero:

- OpcionesOportunidadComboBox.java

Referencia de la Clase

hibernate.operaciones.OperacionesComparativo

Métodos públicos

- **ComparativoAlumnos obtenerDatos** (Integer a, Integer b)
- **ComparativoAlumnos obtenerDatos** (Integer a, Integer inta, Integer b, Integer intb)
- **ComparativoAlumnos obtenerDatosConNA** (Integer a, Integer inta, Integer b, Integer intb)
- **List elegirLista** (ListasDeIntentos lista, Integer intento)

Descripción

Clase que contiene los métodos para obtener las estadísticas entre dos UEAs.

Código

```
package hibernate.operaciones;

import hibernate.beans.ComparativoAlumnos;
import hibernate.beans.Kardex;
import hibernate.beans.ListasDeIntentos;
import hibernate.conexion.CrearConexion;
import java.util.Iterator;
import java.util.List;
import org.apache.log4j.Logger;
import org.apache.log4j.PropertyConfigurator;
import org.hibernate.Session;

/**
 *
 * @author Esteban
 */
public class OperacionesComparativo {

    private final static Logger log = Logger.getLogger(OperacionesComparativo.class);

    public ComparativoAlumnos obtenerDatos (Integer a, Integer b) {
        ComparativoAlumnos comparativoAlumnostmp;

        mbmb=0,mbb=0,mbs=0,mbna=0,bmb=0,bb=0,bs=0,bna=0,smb=0,sb=0,ss=0,sna=0,namb=0,nab=0,nas=0,nana=0;
        String claveUea = Integer.toString(a);
        String claveUeaRe = Integer.toString(b);

        PropertyConfigurator.configure("log4j.properties");
        Session session = null;
    }
}
```

```

try {
    //Session session = CrearConexion.getSessionFactory().openSession();
    session = CrearConexion.getSessionFactory().openSession();
    session.beginTransaction();
    log.info("Leyendo Kardex");
    String instruccionSql1 = "FROM Kardex K WHERE K.uea = "+claveUea;
    List kardexList1 = session.createQuery(instruccionSql1).list();
    String instruccionSql2 = "FROM Kardex K WHERE K.uea = "+claveUeaRe;
    List kardexList2 = session.createQuery(instruccionSql2).list();
    log.info("Se leyó Kardex");
    int aprobados=0,mbG=0,mbR=0,bG=0,bR=0,sG=0,sR=0;
    for (Iterator iterator1 = kardexList1.iterator(); iterator1.hasNext();){
        Kardex kardex = (Kardex) iterator1.next();
        if (kardex.getCalificacion().compareToIgnoreCase("MB")==0) {
            for (Iterator iterator2 = kardexList2.iterator(); iterator2.hasNext();) {
                Kardex k2 = (Kardex) iterator2.next();
                if(k2.getMatricula().compareToIgnoreCase(kardex.getMatricula())==0) {
                    if (k2.getCalificacion().compareToIgnoreCase("MB")==0) {
                        mbmb++;
                    } else if (k2.getCalificacion().compareToIgnoreCase("B")==0) {
                        mbb++;
                    } else if (k2.getCalificacion().compareToIgnoreCase("S")==0) {
                        mbs++;
                    } else if (k2.getCalificacion().compareToIgnoreCase("NA")==0) {
                        mbna++;
                    }
                }
            }
        }
        } else if (kardex.getCalificacion().compareToIgnoreCase("B")==0) {
            for (Iterator iterator2 = kardexList2.iterator(); iterator2.hasNext();) {
                Kardex k2 = (Kardex) iterator2.next();
                if(k2.getMatricula().compareToIgnoreCase(kardex.getMatricula())==0) {
                    if (k2.getCalificacion().compareToIgnoreCase("MB")==0) {
                        bmb++;
                    } else if (k2.getCalificacion().compareToIgnoreCase("B")==0) {
                        bb++;
                    } else if (k2.getCalificacion().compareToIgnoreCase("S")==0) {
                        bs++;
                    } else if (k2.getCalificacion().compareToIgnoreCase("NA")==0) {
                        bna++;
                    }
                }
            }
        }
    }
}

```

```

    }
} else if (kardex.getCalificacion().compareToIgnoreCase("S")==0) {
    for (Iterator iterator2 = kardexList2.iterator(); iterator2.hasNext();) {
        Kardex k2 = (Kardex) iterator2.next();
        if(k2.getMatricula().compareToIgnoreCase(kardex.getMatricula())==0) {
            if (k2.getCalificacion().compareToIgnoreCase("MB")==0) {
                smb++;
            } else if (k2.getCalificacion().compareToIgnoreCase("B")==0) {
                sb++;
            } else if (k2.getCalificacion().compareToIgnoreCase("S")==0) {
                ss++;
            } else if (k2.getCalificacion().compareToIgnoreCase("NA")==0) {
                sna++;
            }
        }
    }
}
} else if (kardex.getCalificacion().compareToIgnoreCase("NA")==0) {
    for (Iterator iterator2 = kardexList2.iterator(); iterator2.hasNext();) {
        Kardex k2 = (Kardex) iterator2.next();
        if(k2.getMatricula().compareToIgnoreCase(kardex.getMatricula())==0) {
            if (k2.getCalificacion().compareToIgnoreCase("MB")==0) {
                namb++;
            } else if (k2.getCalificacion().compareToIgnoreCase("B")==0) {
                nab++;
            } else if (k2.getCalificacion().compareToIgnoreCase("S")==0) {
                nas++;
            } else if (k2.getCalificacion().compareToIgnoreCase("NA")==0) {
                nana++;
            }
        }
    }
}
}

if ((kardexList1.size(>0)&&(kardexList2.size(>0))) {
    comparativoAlumnostmp=new ComparativoAlumnos(mbmb, mbb, mbs, mbna, bmb, bb, bs, bna,smb,
sb, ss, sna,namb, nab, nas, nana);
} else comparativoAlumnostmp = new ComparativoAlumnos(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);

session.getTransaction().commit();
session.close();
return comparativoAlumnostmp;
} catch (Throwable ex) {

```

```

        log.error("No se pudo crear la sesión ");
        System.err.println("No se pudo crear la sesión "+ex);
        throw new ExceptionInInitializerError(ex);
    }
}

public ComparativoAlumnos obtenerDatos (Integer a, Integer inta, Integer b, Integer intb) {
    ComparativoAlumnos comparativoAlumnostmp;

                                                                                               int
    mbmb=0,mbb=0,mbs=0,mbna=0,bmb=0,bb=0,bs=0,bna=0,smb=0,sb=0,ss=0,sna=0,namb=0,nab=0,nas=0,nana=0;
    String claveUea = Integer.toString(a);
    String claveUeaRe = Integer.toString(b);
    OperacionesListasDeIntentos opUEA = new OperacionesListasDeIntentos();
    OperacionesListasDeIntentos opUEARE = new OperacionesListasDeIntentos();
    ListasDeIntentos listaUEA = opUEA.crearListas(a);
    ListasDeIntentos listaUEARE = opUEARE.crearListas(b);

    List kardexList1 = elegirLista(listaUEA, inta);
    List kardexList2 = elegirLista(listaUEARE, intb);

    int aprobados=0,mbG=0,mbR=0,bG=0,bR=0,sG=0,sR=0,encontrado=0;
    for (Iterator iterator1 = kardexList1.iterator(); iterator1.hasNext();){
        Kardex kardex = (Kardex) iterator1.next();
        if (kardex.getCalificacion().compareToIgnoreCase("MB")==0) {
            for (Iterator iterator2 = kardexList2.iterator(); iterator2.hasNext();) {
                Kardex k2 = (Kardex) iterator2.next();
                if(k2.getMatricula().compareToIgnoreCase(kardex.getMatricula())==0) {
                    encontrado=1;
                    if (k2.getCalificacion().compareToIgnoreCase("MB")==0) {
                        mbmb++;
                    } else if (k2.getCalificacion().compareToIgnoreCase("B")==0) {
                        mbb++;
                    } else if (k2.getCalificacion().compareToIgnoreCase("S")==0) {
                        mbs++;
                    } else if (k2.getCalificacion().compareToIgnoreCase("NA")==0) {
                        mbna++;
                    }
                }
            }
        }
    }
    } else if (kardex.getCalificacion().compareToIgnoreCase("B")==0) {
        for (Iterator iterator2 = kardexList2.iterator(); iterator2.hasNext();) {
            Kardex k2 = (Kardex) iterator2.next();
            if(k2.getMatricula().compareToIgnoreCase(kardex.getMatricula())==0) {

```

```

    encontrado=1;
    if (k2.getCalificacion().compareToIgnoreCase("MB")==0) {
        bmb++;
    } else if (k2.getCalificacion().compareToIgnoreCase("B")==0) {
        bb++;
    } else if (k2.getCalificacion().compareToIgnoreCase("S")==0) {
        bs++;
    } else if (k2.getCalificacion().compareToIgnoreCase("NA")==0) {
        bna++;
    }
}
}
} else if (kardex.getCalificacion().compareToIgnoreCase("S")==0) {
    for (Iterator iterator2 = kardexList2.iterator(); iterator2.hasNext();) {
        Kardex k2 = (Kardex) iterator2.next();
        if(k2.getMatricula().compareToIgnoreCase(kardex.getMatricula())==0) {
            encontrado=1;
            if (k2.getCalificacion().compareToIgnoreCase("MB")==0) {
                smb++;
            } else if (k2.getCalificacion().compareToIgnoreCase("B")==0) {
                sb++;
            } else if (k2.getCalificacion().compareToIgnoreCase("S")==0) {
                ss++;
            } else if (k2.getCalificacion().compareToIgnoreCase("NA")==0) {
                sna++;
            }
        }
    }
}
} else if (kardex.getCalificacion().compareToIgnoreCase("NA")==0) {
    for (Iterator iterator2 = kardexList2.iterator(); iterator2.hasNext();) {
        Kardex k2 = (Kardex) iterator2.next();
        if(k2.getMatricula().compareToIgnoreCase(kardex.getMatricula())==0) {
            encontrado = 1;
            if (k2.getCalificacion().compareToIgnoreCase("MB")==0) {
                namb++;
            } else if (k2.getCalificacion().compareToIgnoreCase("B")==0) {
                nab++;
            } else if (k2.getCalificacion().compareToIgnoreCase("S")==0) {
                nas++;
            } else if (k2.getCalificacion().compareToIgnoreCase("NA")==0) {
                nana++;
            }
        }
    }
}
}
}

```

```

    }
    }
    }
    if ((kardexList1.size(>0)&&(kardexList2.size(>0)) {
        comparativoAlumnostmp=new ComparativoAlumnos(mbmb, mbb, mbs, mbna, bmb, bb, bs, bna,smb, sb,
        ss, sna,namb, nab, nas, nana);
        } else comparativoAlumnostmp = new ComparativoAlumnos(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
        return comparativoAlumnostmp;
    }

    public ComparativoAlumnos obtenerDatosConNA (Integer a, Integer inta, Integer b, Integer intb) {
        ComparativoAlumnos comparativoAlumnostmp;

        int
        mbmb=0,mbb=0,mbs=0,mbna=0,bmb=0,bb=0,bs=0,bna=0,smb=0,sb=0,ss=0,sna=0,namb=0,nab=0,nas=0,nana=0;
        String claveUea = Integer.toString(a);
        String claveUeaRe = Integer.toString(b);
        OperacionesListasDeIntentos opUEA = new OperacionesListasDeIntentos();
        OperacionesListasDeIntentos opUEARE = new OperacionesListasDeIntentos();
        ListasDeIntentos listaUEA = opUEA.crearListasConNA(a);
        ListasDeIntentos listaUEARE = opUEARE.crearListas(b);

        List kardexList1 = elegirLista(listaUEA, inta);
        List kardexList2 = elegirLista(listaUEARE, intb);

        int aprobados=0,mbG=0,mbR=0,bG=0,bR=0,sG=0,sR=0,encontrado=0;
        for (Iterator iterator1 = kardexList1.iterator(); iterator1.hasNext();){
            Kardex kardex = (Kardex) iterator1.next();
            if (kardex.getCalificacion().compareToIgnoreCase("MB")==0) {
                for (Iterator iterator2 = kardexList2.iterator(); iterator2.hasNext();) {
                    Kardex k2 = (Kardex) iterator2.next();
                    if(k2.getMatricula().compareToIgnoreCase(kardex.getMatricula())==0) {
                        encontrado=1;
                        if (k2.getCalificacion().compareToIgnoreCase("MB")==0) {
                            mbmb++;
                        } else if (k2.getCalificacion().compareToIgnoreCase("B")==0) {
                            mbb++;
                        } else if (k2.getCalificacion().compareToIgnoreCase("S")==0) {
                            mbs++;
                        } else if (k2.getCalificacion().compareToIgnoreCase("NA")==0) {
                            mbna++;
                        }
                    }
                }
            }
        }
    }
}

```

```

} else if (kardex.getCalificacion().compareToIgnoreCase("B")==0) {
    for (Iterator iterator2 = kardexList2.iterator(); iterator2.hasNext();) {
        Kardex k2 = (Kardex) iterator2.next();
        if(k2.getMatricula().compareToIgnoreCase(kardex.getMatricula())==0) {
            encontrado=1;
            if (k2.getCalificacion().compareToIgnoreCase("MB")==0) {
                bmb++;
            } else if (k2.getCalificacion().compareToIgnoreCase("B")==0) {
                bb++;
            } else if (k2.getCalificacion().compareToIgnoreCase("S")==0) {
                bs++;
            } else if (k2.getCalificacion().compareToIgnoreCase("NA")==0) {
                bna++;
            }
        }
    }
}
} else if (kardex.getCalificacion().compareToIgnoreCase("S")==0) {
    for (Iterator iterator2 = kardexList2.iterator(); iterator2.hasNext();) {
        Kardex k2 = (Kardex) iterator2.next();
        if(k2.getMatricula().compareToIgnoreCase(kardex.getMatricula())==0) {
            encontrado=1;
            if (k2.getCalificacion().compareToIgnoreCase("MB")==0) {
                smb++;
            } else if (k2.getCalificacion().compareToIgnoreCase("B")==0) {
                sb++;
            } else if (k2.getCalificacion().compareToIgnoreCase("S")==0) {
                ss++;
            } else if (k2.getCalificacion().compareToIgnoreCase("NA")==0) {
                sna++;
            }
        }
    }
}
} else if (kardex.getCalificacion().compareToIgnoreCase("NA")==0) {
    for (Iterator iterator2 = kardexList2.iterator(); iterator2.hasNext();) {
        Kardex k2 = (Kardex) iterator2.next();
        if(k2.getMatricula().compareToIgnoreCase(kardex.getMatricula())==0) {
            encontrado = 1;
            if (k2.getCalificacion().compareToIgnoreCase("MB")==0) {
                namb++;
            } else if (k2.getCalificacion().compareToIgnoreCase("B")==0) {
                nab++;
            } else if (k2.getCalificacion().compareToIgnoreCase("S")==0) {
                nas++;
            }
        }
    }
}

```


Referencia de la Clase

hibernate.operaciones.OperacionesComputacion

Métodos públicos

- List< Computacion > listComputacion ()
- Computacion datosComputacion (String clave)
- Computacion noUEA ()

Descripción detallada

Clase que contiene los métodos para obtener la lista de UEA de la carrera de Ingeniería en computación a partir de la base de datos.

Código

```
package hibernate.operaciones;

import hibernate.beans.Computacion;
import hibernate.conexion.CrearConexion;
import java.util.Iterator;
import java.util.List;
import org.apache.log4j.Logger;
import org.apache.log4j.PropertyConfigurator;
import org.hibernate.Session;

/**
 *
 * @author Esteban
 */
public class OperacionesComputacion {

    private final static Logger log = Logger.getLogger(OperacionesComputacion.class);

    public List<Computacion> listComputacion(){
        //String hql="SELECT clave_UEA,nombre_UEA,correspondencia FROM Computacion";

        PropertyConfigurator.configure("log4j.properties");
        Session session = null;

        try {
            session = CrearConexion.getSessionFactory().openSession();
            session.beginTransaction();
            log.info("Leyendo Computacion");
            List compuList = session.createQuery("FROM Computacion").list();
            log.info("Se leyó Computacion");
            session.getTransaction().commit();
            session.close();
            return compuList;
        } catch (Throwable ex) {
            log.error("No se pudo crear la sesión ");
            System.err.println("No se pudo crear la sesión "+ex);
            throw new ExceptionInInitializerError(ex);
        }
    }
}
```

```

public Computacion datosComputacion(String clave) {

    PropertyConfigurator.configure("log4j.properties");
    Session session = null;

    try {
        Computacion computacion;
        session = CrearConexion.getSessionFactory().openSession();
        session.beginTransaction();
        log.info("Leyendo Computacion");
        String instruccionSql = "FROM Computacion C WHERE C.claveUEA = "+clave;
        List compuList = session.createQuery(instruccionSql).list();
        log.info("Se leyó Computacion");
        session.getTransaction().commit();
        session.close();
        if (compuList.size()>0) {
            computacion=(Computacion)compuList.get(0);
            return computacion;
        }
        else return null;
    } catch (Throwable ex) {
        log.error("No se pudo crear la sesión ");
        System.err.println("No se pudo crear la sesión "+ex);
        throw new ExceptionInInitializerError(ex);
    }
}

public Computacion noUEA() {
    Computacion computacion = new Computacion();
    computacion.setClaveUEA(0);
    computacion.setCorrespondencia(" ");
    computacion.setNombreUEA("No hay UEA requerida para la UEA seleccionada.");
    return computacion;
}
}

```

La documentación para esta clase fue generada a partir del siguiente fichero:

- OperacionesComputacion.java

Referencia de la Clase hibernate.operaciones.OperacionesGraficas

Métodos públicos

- DefaultCategoryDataset **datosBarra** (**Computacion** c1, **Computacion** c2)
- DefaultCategoryDataset **datosBarra** (**Computacion** c1, Integer inta, **Computacion** c2, Integer intb)
- DefaultCategoryDataset **datosBarraConNA** (**Computacion** c1, Integer inta, **Computacion** c2, Integer intb)

Descripción

Clase que procesa la información para obtener gráficas de barras.

Código

```
package hibernate.operaciones;

import hibernate.beans.ComparativoAlumnos;
import hibernate.beans.Computacion;
import org.jfree.data.category.DefaultCategoryDataset;

/**
 *
 * @author Esteban
 */
public class OperacionesGraficas {
    public DefaultCategoryDataset datosBarra(Computacion c1,Computacion c2){

        String elementoUno = "MB";
        String elementoDos = "B";
        String elementoTres = "S";
        String elementoCuatro = "NA";
        String categoriaA = "MB en "+c2.getClaveUEA();
        String categoriaB = "B en "+c2.getClaveUEA();
        String categoriaC = "S en "+c2.getClaveUEA();
        //String CategoriaD = "NA en "+c2.getClaveUEA();

        OperacionesComparativo op = new OperacionesComparativo();
        ComparativoAlumnos comparativoAlumnos = op.obtenerDatos(c1.getClaveUEA(),
c2.getClaveUEA());

        DefaultCategoryDataset ds = new DefaultCategoryDataset( );

        ds.addValue( comparativoAlumnos.getAlumnosMBaMBb() , elementoUno , categoriaA );
        ds.addValue( comparativoAlumnos.getAlumnosMBaBb() , elementoUno , categoriaB );
        ds.addValue( comparativoAlumnos.getAlumnosMBaSb() , elementoUno , categoriaC );
        //ds.addValue( comparativoAlumnos.getAlumnosMBaNAb() , elementoUno , CategoriaD );

        ds.addValue( comparativoAlumnos.getAlumnosBaMBb() , elementoDos , categoriaA );
        ds.addValue( comparativoAlumnos.getAlumnosBaBb() , elementoDos , categoriaB );
        ds.addValue( comparativoAlumnos.getAlumnosBaSb() , elementoDos , categoriaC );
        //ds.addValue( comparativoAlumnos.getAlumnosBaNAb() , elementoDos , CategoriaD );

        ds.addValue( comparativoAlumnos.getAlumnosSaMBb() , elementoTres , categoriaA );
```

```

ds.addValue( comparativoAlumnos.getAlumnosSaBb() , elementoTres , categoriaB );
ds.addValue( comparativoAlumnos.getAlumnosSaSb() , elementoTres , categoriaC );
//ds.addValue( comparativoAlumnos.getAlumnosSaNAb() , elementoTres , CategoriaD );

ds.addValue( comparativoAlumnos.getAlumnosNAaMBb() , elementoCuatro , categoriaA );
ds.addValue( comparativoAlumnos.getAlumnosNAaBb() , elementoCuatro , categoriaB );
ds.addValue( comparativoAlumnos.getAlumnosNAaSb() , elementoCuatro , categoriaC );
//ds.addValue( comparativoAlumnos.getAlumnosNAaNAb() , elementoCuatro , CategoriaD );

return ds;
}

public DefaultCategoryDataset datosBarra(Computacion c1,Integer inta,Computacion c2, Integer
intb){

    String elementoUno = "Alumnos con MB en\n"+c1.getNombreUEA();
    String elementoDos = "Alumnos con B en\n"+c1.getNombreUEA();
    String elementoTres = "Alumnos con S en\n"+c1.getNombreUEA();
    //String elementoCuatro = "Alumnos con NA \nen la materia \n"+c1.getClaveUEA();
    String categoriaA = "MB en "+c2.getNombreUEA();
    String categoriaB = "B en "+c2.getNombreUEA();
    String categoriaC = "S en "+c2.getNombreUEA();
    //String CategoriaD = "NA en "+c2.getClaveUEA();

    OperacionesComparativo op = new OperacionesComparativo();
    ComparativoAlumnos comparativoAlumnos = op.obtenerDatos(c1.getClaveUEA(), inta,
c2.getClaveUEA(),intb);

    DefaultCategoryDataset ds = new DefaultCategoryDataset( );

    ds.addValue( comparativoAlumnos.getAlumnosMBaMBb() , elementoUno , categoriaA );
    ds.addValue( comparativoAlumnos.getAlumnosMBaBb() , elementoUno , categoriaB );
    ds.addValue( comparativoAlumnos.getAlumnosMBaSb() , elementoUno , categoriaC );
    //ds.addValue( comparativoAlumnos.getAlumnosMBaNAb() , elementoUno , CategoriaD );

    ds.addValue( comparativoAlumnos.getAlumnosBaMBb() , elementoDos , categoriaA );
    ds.addValue( comparativoAlumnos.getAlumnosBaBb() , elementoDos , categoriaB );
    ds.addValue( comparativoAlumnos.getAlumnosBaSb() , elementoDos , categoriaC );
    //ds.addValue( comparativoAlumnos.getAlumnosBaNAb() , elementoDos , CategoriaD );

    ds.addValue( comparativoAlumnos.getAlumnosSaMBb() , elementoTres , categoriaA );
    ds.addValue( comparativoAlumnos.getAlumnosSaBb() , elementoTres , categoriaB );
    ds.addValue( comparativoAlumnos.getAlumnosSaSb() , elementoTres , categoriaC );
    //ds.addValue( comparativoAlumnos.getAlumnosSaNAb() , elementoTres , CategoriaD );

    /*ds.addValue( comparativoAlumnos.getAlumnosNAaMBb() , elementoCuatro , categoriaA );
    ds.addValue( comparativoAlumnos.getAlumnosNAaBb() , elementoCuatro , categoriaB );
    ds.addValue( comparativoAlumnos.getAlumnosNAaSb() , elementoCuatro , categoriaC );*/
    //ds.addValue( comparativoAlumnos.getAlumnosNAaNAb() , elementoCuatro , CategoriaD );

    return ds;
}

public DefaultCategoryDataset datosBarraConNA(Computacion c1,Integer inta,Computacion c2,
Integer intb){

    String elementoUno = "Alumnos con MB en\n"+c1.getNombreUEA();
    String elementoDos = "Alumnos con B en\n"+c1.getNombreUEA();

```

```

String elementoTres = "Alumnos con S en\n"+c1.getNombreUEA();
String elementoCuatro = "Alumnos con NA en\n"+c1.getNombreUEA();
String categoriaA = "MB en "+c2.getNombreUEA();
String categoriaB = "B en "+c2.getNombreUEA();
String categoriaC = "S en "+c2.getNombreUEA();
//String CategoriaD = "NA en "+c2.getClaveUEA();

OperacionesComparativo op = new OperacionesComparativo();
ComparativoAlumnos comparativoAlumnos = op.obtenerDatosConNA(c1.getClaveUEA(),
inta, c2.getClaveUEA(),intb);

DefaultCategoryDataset ds = new DefaultCategoryDataset( );

ds.addValue( comparativoAlumnos.getAlumnosMBaMBb() , elementoUno , categoriaA );
ds.addValue( comparativoAlumnos.getAlumnosMBaBb() , elementoUno , categoriaB );
ds.addValue( comparativoAlumnos.getAlumnosMBaSb() , elementoUno , categoriaC );
//ds.addValue( comparativoAlumnos.getAlumnosMBaNAb() , elementoUno , CategoriaD );

ds.addValue( comparativoAlumnos.getAlumnosBaMBb() , elementoDos , categoriaA );
ds.addValue( comparativoAlumnos.getAlumnosBaBb() , elementoDos , categoriaB );
ds.addValue( comparativoAlumnos.getAlumnosBaSb() , elementoDos , categoriaC );
//ds.addValue( comparativoAlumnos.getAlumnosBaNAb() , elementoDos , CategoriaD );

ds.addValue( comparativoAlumnos.getAlumnosSaMBb() , elementoTres , categoriaA );
ds.addValue( comparativoAlumnos.getAlumnosSaBb() , elementoTres , categoriaB );
ds.addValue( comparativoAlumnos.getAlumnosSaSb() , elementoTres , categoriaC );
//ds.addValue( comparativoAlumnos.getAlumnosSaNAb() , elementoTres , CategoriaD );

ds.addValue( comparativoAlumnos.getAlumnosNAaMBb() , elementoCuatro , categoriaA );
ds.addValue( comparativoAlumnos.getAlumnosNAaBb() , elementoCuatro , categoriaB );
ds.addValue( comparativoAlumnos.getAlumnosNAaSb() , elementoCuatro , categoriaC );
//ds.addValue( comparativoAlumnos.getAlumnosNAaNAb() , elementoCuatro , CategoriaD );

return ds;
}
}

```

La documentación para esta clase fue generada a partir del siguiente fichero:

- OperacionesGraficas.java

Referencia de la Clase hibernate.operaciones.OperacionesInfoUEA

Métodos públicos

- **InfoUEA** `adquirirInfoUEA` (String claveUea)
 - `List< InfoUEAAdicional >` **adquirirInfoUEAAdicional** (String claveUea)
 - **InfoUEAAdicional** `obtenerestadísticas` (List list, int intento)
-

Descripción

Clase para obtener las estadísticas relacionadas con la UEA

Código

```
package hibernate.operaciones;

import hibernate.beans.InfoUEA;
import hibernate.beans.InfoUEAAdicional;
import hibernate.beans.Kardex;
import hibernate.beans.ListasDeIntentos;
import hibernate.conexion.CrearConexion;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import org.apache.log4j.Logger;
import org.apache.log4j.PropertyConfigurator;
import org.hibernate.Session;

/**
 *
 * @author Esteban
 */
public class OperacionesInfoUEA {

    private final static Logger log = Logger.getLogger(OperacionesInfoUEA.class);

    public InfoUEA adquirirInfoUEA(String claveUea){

        PropertyConfigurator.configure("log4j.properties");
        Session session = null;
        InfoUEA infoUEA;

        try {
            session = CrearConexion.getSessionFactory().openSession();
            session.beginTransaction();
```

```

String instruccionSql = "FROM Kardex K WHERE K.uea = "+claveUea;
log.info("Leyendo Kardex");
List kardexList = session.createQuery(instruccionSql).list();
log.info("Se leyó Kardex");
int aprobados=0,mbG=0,mbR=0,bG=0,bR=0,sG=0,sR=0;
for (Iterator iterator1 = kardexList.iterator(); iterator1.hasNext();){
    Kardex kardex = (Kardex) iterator1.next();
    if ((kardex.getCalificacion().compareToIgnoreCase("NA")!=0)&&(kardex.getCalificacion().compareToIgnore-
Case("EQ")!=0)) aprobados++;
    if (kardex.getTipoCal().compareToIgnoreCase("GLO.")==0) {
        if (kardex.getCalificacion().compareToIgnoreCase("MB")==0) {
            mbG++;
        } else if(kardex.getCalificacion().compareToIgnoreCase("B")==0) {
            bG++;
        } else if (kardex.getCalificacion().compareToIgnoreCase("s")==0) {
            sG++;
        }
    } else if(kardex.getTipoCal().compareToIgnoreCase("REC.")==0) {
        if (kardex.getCalificacion().compareToIgnoreCase("MB")==0) {
            mbR++;
        } else if(kardex.getCalificacion().compareToIgnoreCase("B")==0) {
            bR++;
        } else if (kardex.getCalificacion().compareToIgnoreCase("s")==0) {
            sR++;
        }
    }
}
if (kardexList.size(>0) {
    infoUEA = new InfoUEA(kardexList.size(), aprobados, mbG, mbR, bG, bR, sG, sR);
} else infoUEA = new InfoUEA();

session.getTransaction().commit();
session.close();
return infoUEA;
} catch (Throwable ex) {
    log.error("No se pudo crear la sesión ");
    System.err.println("No se pudo crear la sesión "+ex);
    throw new ExceptionInInitializerError(ex);
}
}

public List<InfoUEAA adicional> adquirirInfoUEAA adicional(String claveUea){

```

```

int clave = Integer.parseInt(claveUea);
List <InfoUEAAadicional> lista = new ArrayList<InfoUEAAadicional>();
OperacionesListasDeIntentos op = new OperacionesListasDeIntentos();
ListasDeIntentos listas = op.crearListas(clave);
InfoUEAAadicional tmp1,tmp2,tmp3,tmp4,tmp5;
tmp1 = obtenerestadísticas(listas.getLista1(), 1);
tmp2 = obtenerestadísticas(listas.getLista2(), 2);
tmp3 = obtenerestadísticas(listas.getLista3(), 3);
tmp4 = obtenerestadísticas(listas.getLista4(), 4);
tmp5 = obtenerestadísticas(listas.getLista5(), 5);
tmp4.setAlumnosQueHanReprobado(listas.getLista5().size());
tmp3.setAlumnosQueHanReprobado(listas.getLista4().size());
tmp2.setAlumnosQueHanReprobado(listas.getLista3().size());
tmp1.setAlumnosQueHanReprobado(listas.getLista2().size());
lista.add(tmp1);
lista.add(tmp2);
lista.add(tmp3);
lista.add(tmp4);
lista.add(tmp5);
return lista;
}

```

```

public InfoUEAAadicional obtenerestadísticas(List list,int intento) {

InfoUEAAadicional infoUEAAadicional;
int aprobados=0,reprobados=0,mbG=0,mbR=0,bG=0,bR=0,sG=0,sR=0,i=0;

for (Iterator iterator1 = list.iterator(); iterator1.hasNext();){
Kardex kardex = (Kardex) iterator1.next();
if (kardex.getCalificacion().compareToIgnoreCase("NA")!=0) {
aprobados++;
}
if (kardex.getTipoCal().compareToIgnoreCase("GLO.")==0) {
if (kardex.getCalificacion().compareToIgnoreCase("MB")==0) {
mbG++;
} else if(kardex.getCalificacion().compareToIgnoreCase("B")==0) {
bG++;
} else if (kardex.getCalificacion().compareToIgnoreCase("S")==0) {
sG++;
} else if (kardex.getCalificacion().compareToIgnoreCase("NA")==0) {
reprobados++;
}
} else if(kardex.getTipoCal().compareToIgnoreCase("REC.")==0) {

```

```
if (kardex.getCalificacion().compareToIgnoreCase("MB")==0) {
    mbR++;
} else if (kardex.getCalificacion().compareToIgnoreCase("B")==0) {
    bR++;
} else if (kardex.getCalificacion().compareToIgnoreCase("S")==0) {
    sR++;
} else if (kardex.getCalificacion().compareToIgnoreCase("NA")==0) {
    reprobados++;
}
}
}

if (list.size()>0) {
    infoUEAAadicional = new InfoUEAAadicional(aprobados, reprobados, mbG, mbR, bG, bR, sG, sR,intento);
} else infoUEAAadicional = new InfoUEAAadicional();
return infoUEAAadicional;
}
}
```

La documentación para esta clase fue generada a partir del siguiente fichero:

- OperacionesInfoUEA.java

Referencia de la Clase hibernate.operaciones.OperacionesKardex

Métodos públicos

- List< Kardex > listKardex ()
- List< Kardex > infoUEAKardex (String claveUea)

Descripción

Clase que contiene los métodos para la creación de las listas que contenga los datos académicos relacionados a una UEA específica.

Código

```
package hibernate.operaciones;

import hibernate.beans.Kardex;
import hibernate.conexion.CrearConexion;
import java.util.Iterator;
import java.util.List;
import org.apache.log4j.Logger;
import org.apache.log4j.PropertyConfigurator;
import org.hibernate.Session;

/**
 *
 * @author Esteban
 */
public class OperacionesKardex {

    private final static Logger log = Logger.getLogger(OperacionesKardex.class);

    public List<Kardex> listKardex( ){

        PropertyConfigurator.configure("log4j.properties");
        Session session = null;

        try {
            session = CrearConexion.getSessionFactory().openSession();
            session.beginTransaction();
            log.info("Leyendo Kardex");
            List kardexList = session.createQuery("FROM Kardex").list();
            log.info("Se leyó Kardex");
            session.getTransaction().commit();
            session.close();
            return kardexList;
        } catch (Throwable ex) {
            log.error("No se pudo crear la sesión ");
            System.err.println("No se pudo crear la sesión "+ex);
            throw new ExceptionInInitializerError(ex);
        }
    }

    public List<Kardex> infoUEAKardex(String claveUea ){

        PropertyConfigurator.configure("log4j.properties");
        Session session = null;
```

```
try {
    session = CrearConexion.getSessionFactory().openSession();
    session.beginTransaction();
    log.info("Leyendo Kardex");
    String instruccionSql = "FROM Kardex K WHERE K.uea = "+claveUea;
    List kardexList = session.createQuery(instruccionSql).list();
    log.info("Se leyó Kardex");
    session.getTransaction().commit();
    session.close();
    return kardexList;
} catch (Throwable ex) {
    log.error("No se pudo crear la sesión ");
    System.err.println("No se pudo crear la sesión "+ex);
    throw new ExceptionInInitializerError(ex);
}
}
```

La documentación para esta clase fue generada a partir del siguiente fichero:

- OperacionesKardex.java

Referencia de la Clase

hibernate.operaciones.OperacionesListasDeIntentos

Métodos públicos

- **ListasDeIntentos crearListas** (Integer clave)
- **ListasDeIntentos crearListasConNA** (Integer clave)
- **int checarAlumno** (List list, String matricula)

Descripción

Clase con los métodos para filtrar de entre los datos de una UEA el número de intento y la calificación obtenida.

Código

```
package hibernate.operaciones;

import hibernate.beans.Kardex;
import hibernate.beans.ListasDeIntentos;
import hibernate.conexion.CrearConexion;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import org.apache.log4j.Logger;
import org.apache.log4j.PropertyConfigurator;
import org.hibernate.Session;

/**
 *
 * @author Esteban
 */
public class OperacionesListasDeIntentos {

    private final static Logger log = Logger.getLogger(OperacionesListasDeIntentos.class);

    public ListasDeIntentos crearListas(Integer clave) {

        String claveUEA = clave.toString();
        ListasDeIntentos listasDeIntentos = new ListasDeIntentos();
        List<Kardex> lista1 = new ArrayList<Kardex>();
        List<Kardex> lista2 = new ArrayList<Kardex>();
        List<Kardex> lista3 = new ArrayList<Kardex>();
        List<Kardex> lista4 = new ArrayList<Kardex>();
        List<Kardex> lista5 = new ArrayList<Kardex>();
        int contador=0,ini=0;
        String matriculaActual="";
        String matriculaAnterior="";

        PropertyConfigurator.configure("log4j.properties");
        Session session = null;

        try {
            session = CrearConexion.getSessionFactory().openSession();
            session.beginTransaction();
            String instruccionSql1 = "FROM Kardex K WHERE K.uea = "+claveUEA;
```

```

log.info("Leyendo Kardex");
List kardexList1 = session.createQuery(instruccionSql1).list();
log.info("Se leyó Kardex");
for (Iterator iterator1 = kardexList1.iterator(); iterator1.hasNext());{
    Kardex kardex = (Kardex) iterator1.next();
    if (ini==0) {
        ini++;
        matriculaAnterior=kardex.getMatricula();
    }
    matriculaActual=kardex.getMatricula();
    if (matriculaActual.compareToIgnoreCase(matriculaAnterior)==0){
        contador++;
    } else {
        matriculaAnterior = matriculaActual;
        contador = 1;
    }
    String cal = kardex.getCalificacion();
    if ((cal.compareToIgnoreCase("MB")==0)||((cal.compareToIgnoreCase("B")==0)||
(cal.compareToIgnoreCase("S")==0)) {
        if (contador==1) {
            lista1.add(kardex);
        } else if (contador==2) {
            lista2.add(kardex);
        } else if (contador==3) {
            lista3.add(kardex);
        } else if (contador==4) {
            lista4.add(kardex);
        } else if (contador==5) {
            lista5.add(kardex);
        }
    } else if ((cal.compareToIgnoreCase("NA")==0)&&(contador==5)) {
        lista5.add(kardex);
    }
}
listasDeIntentos.setLista1(lista1);
listasDeIntentos.setLista2(lista2);
listasDeIntentos.setLista3(lista3);
listasDeIntentos.setLista4(lista4);
listasDeIntentos.setLista5(lista5);
session.getTransaction().commit();
session.close();
} catch (Throwable ex) {
    log.error("No se pudo crear la sesión ");
    System.err.println("No se pudo crear la sesión "+ex);
    throw new ExceptionInInitializerError(ex);
}
return listasDeIntentos;
}

public ListasDeIntentos crearListasConNA(Integer clave) {

String claveUEA = clave.toString();
ListasDeIntentos listasDeIntentos = new ListasDeIntentos();
List<Kardex> lista1 = new ArrayList<Kardex>();
List<Kardex> lista2 = new ArrayList<Kardex>();
List<Kardex> lista3 = new ArrayList<Kardex>();
List<Kardex> lista4 = new ArrayList<Kardex>();
List<Kardex> lista5 = new ArrayList<Kardex>();
int contador=0,ini=0;
String matriculaActual="";

```

```

String matriculaAnterior="";

PropertyConfigurator.configure("log4j.properties");
Session session = null;

try {
    session = CrearConexion.getSessionFactory().openSession();
    session.beginTransaction();
    String instruccionSql1 = "FROM Kardex K WHERE K.uea = "+claveUEA;
    log.info("Leyendo Kardex");
    List kardexList1 = session.createQuery(instruccionSql1).list();
    log.info("Se leyó Kardex");
    for (Iterator iterator1 = kardexList1.iterator(); iterator1.hasNext();){
        Kardex kardex = (Kardex) iterator1.next();
        if (ini==0) {
            ini++;
            matriculaAnterior=kardex.getMatricula();
        }
        matriculaActual=kardex.getMatricula();
        if (matriculaActual.compareToIgnoreCase(matriculaAnterior)==0){
            contador++;
        } else {
            matriculaAnterior = matriculaActual;
            contador = 1;
        }
        String cal = kardex.getCalificacion();
        if ((cal.compareToIgnoreCase("MB")==0)||((cal.compareToIgnoreCase("B")==0)||
(cal.compareToIgnoreCase("S")==0)) {
            if (contador==1) {
                lista1.add(kardex);
            } else if (contador==2) {
                lista2.add(kardex);
            } else if (contador==3) {
                lista3.add(kardex);
            } else if (contador==4) {
                lista4.add(kardex);
            } else if (contador==5) {
                lista5.add(kardex);
            }
        } else if (cal.compareToIgnoreCase("NA")==0) {
            if (contador==1) lista1.add(kardex);
            else if (contador==5) lista5.add(kardex);
        }
    }
    listasDeIntentos.setLista1(lista1);
    listasDeIntentos.setLista2(lista2);
    listasDeIntentos.setLista3(lista3);
    listasDeIntentos.setLista4(lista4);
    listasDeIntentos.setLista5(lista5);
    session.getTransaction().commit();
    session.close();
} catch (Throwable ex) {
    log.error("No se pudo crear la sesión ");
    System.err.println("No se pudo crear la sesión "+ex);
    throw new ExceptionInInitializerError(ex);
}
return listasDeIntentos;
}

public int checarAlumno(List list, String matricula) {

```

```
List<Kardex> tempi = new ArrayList<Kardex>();
int intentos=0;
for (Iterator iterator1 = list.iterator(); iterator1.hasNext());{
    Kardex kardex = (Kardex) iterator1.next();
    if (kardex.getMatricula().compareToIgnoreCase(matricula)==0) {
        tempi.add(kardex);
    }
}
if (tempi.size(>0) intentos=tempi.size();
return intentos;
}
```

La documentación para esta clase fue generada a partir del siguiente fichero:

- OperacionesListasDeIntentos.java

Referencia de la Clase hibernate.operaciones.OperacionesRequisitosUea

Métodos públicos

- List< RequisitosUea > listaRequisitosUeas ()
- List< RequisitosUea > ueasRequeridas (String claveUea)

Descripción

Clase que contiene los métodos para saber si una UEA tiene UEA seriadas y si las tiene obtener la lista de esas UEA.

Código

```
package hibernate.operaciones;

import hibernate.beans.RequisitosUea;
import hibernate.conexion.CrearConexion;
import java.util.Iterator;
import java.util.List;
import org.apache.log4j.Logger;
import org.apache.log4j.PropertyConfigurator;
import org.hibernate.Session;

/**
 *
 * @author Esteban
 */
public class OperacionesRequisitosUea {

    private final static Logger log = Logger.getLogger(OperacionesRequisitosUea.class);

    public List<RequisitosUea> listaRequisitosUeas( ){

        PropertyConfigurator.configure("log4j.properties");
        Session session = null;

        try {
            session = CrearConexion.getSessionFactory().openSession();
            session.beginTransaction();
            log.info("Leyendo RequisitosUea");
            List requisitosUeaList = session.createQuery("FROM RequisitosUea").list();
            log.info("Se leyó RequisitosUea");
            session.getTransaction().commit();
            session.close();
            return requisitosUeaList;
        } catch (Throwable ex) {
            log.error("No se pudo crear la sesión ");
            System.err.println("No se pudo crear la sesión "+ex);
            throw new ExceptionInInitializerError(ex);
        }
    }

    public List<RequisitosUea> ueasRequeridas(String claveUea){

        PropertyConfigurator.configure("log4j.properties");
```

```
Session session = null;

try {
    session = CrearConexion.getSessionFactory().openSession();
    session.beginTransaction();
    log.info("Leyendo RequisitosUea");
    String instruccionSql = "FROM RequisitosUea R WHERE R.uea = "+claveUea;
    List requisitosUeaList = session.createQuery(instruccionSql).list();
    log.info("Se leyó RequisitosUea");
    session.getTransaction().commit();
    session.close();
    return requisitosUeaList;
} catch (Throwable ex) {
    log.error("No se pudo crear la sesión ");
    System.err.println("No se pudo crear la sesión "+ex);
    throw new ExceptionInInitializerError(ex);
}
}
```

La documentación para esta clase fue generada a partir del siguiente fichero:

- OperacionesRequisitosUea.java

Referencia de los mapeos hibernate.mapeos

Descripción

Mapeo para relacionar las referencias Hibernate con las tablas de la base de datos.

Mapeo Computacion.hbm.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<hibernate-mapping>
  <class name="hibernate.beans.Computacion" table="computacion">
    <id name="claveUEA" column="clave_UEA" type="int">
      </id>
    <property name="nombreUEA" column="nombre_UEA" type="string"/>
    <property name="correspondencia" column="correspondencia" type="string"/>
  </class>
</hibernate-mapping>
```

Mapeo Kardex.hbm.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<hibernate-mapping>
  <class name="hibernate.beans.Kardex" table="kardex">
    <id name="idKardex" column="id_kardex" type="int">
      <generator class="native"/>
    </id>
    <property name="matricula" column="matricula" type="string"/>
    <property name="uea" column="uea" type="string"/>
    <property name="trimestre" column="trimestre" type="string"/>
    <property name="tipoCal" column="tipo_cal" type="string"/>
    <property name="calificacion" column="calificacion" type="string"/>
    <property name="tipoAlumno" column="tipo_alumno" type="string"/>
  </class>
</hibernate-mapping>
```

Mapeo RequisitosUea.hbm.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<hibernate-mapping>
  <class name="hibernate.beans.RequisitosUea" table="requisitos_uea">
    <id name="idRequisitos" column="id_requisitos" type="int">
      <generator class="native"/>
    </id>
    <property name="uea" column="uea" type="string"/>
    <property name="ueaRequerida" column="uea_requerida" type="string"/>
  </class>
</hibernate-mapping>
```

Referencia del archivo de configuración hibernate

Descripción

Ajustes para poder establecer la conexión a la base de datos.

hibernate.cfg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<hibernate-configuration>
  <session-factory>
    <property name="hibernate.connection.driver_class">
      com.mysql.jdbc.Driver
    </property>
    <property name="hibernate.connection.url">
      jdbc:mysql://localhost:3306/datos_uam?
verifyServerCertificate=false&useSSL=true
    </property>
    <property name="hibernate.connection.username">
      root
    </property>
    <property name="hibernate.connection.password">
      root
    </property>
    <property name="hibernate.dialect">
      org.hibernate.dialect.MySQLDialect
    </property>
    <mapping resource="hibernate/mapeos/Kardex.hbm.xml"/>
    <mapping resource="hibernate/mapeos/Computacion.hbm.xml"/>
    <mapping resource="hibernate/mapeos/RequisitosUea.hbm.xml"/>
  </session-factory>
</hibernate-configuration>
```