



UNIVERSIDAD DE GUAYAQUIL

FACULTAD DE CIENCIAS MATEMATICAS Y FISICAS

CARRERA DE INGENIERIA EN SISTEMAS

COMPUTACIONALES

“DISEÑO Y DESARROLLO DE LA BANDEJA DE TRABAJO QUE PROCESARA
SOLICITUDES Y CERTIFICADOS PARA EL PROTOTIPO DE SISTEMA
ACADEMICO DE LA UNIVERSIDAD DE GUAYAQUIL.”.

TESIS DE GRADO

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS COMPUTACIONALES

AUTOR: ALVARO LUIS VIZUETA LOJAN

TUTOR: ING. MARY CABANILLA

GUAYAQUIL – ECUADOR

2015



Presidencia
de la República
del Ecuador



Plan Nacional
de Ciencia, Tecnología,
Innovación y Saberes



SENESCYT
Secretaría Nacional de Educación Superior,
Ciencia, Tecnología e Innovación

REPOSITORIO NACIONAL EN CIENCIAS Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS

TÍTULO "DISEÑO Y DESARROLLO DE LA BANDEJA DE TRABAJO QUE PROCESARA SOLICITUDES Y CERTIFICADOS PARA EL PROTOTIPO DE SISTEMA ACADEMICO DE LA UNIVERSIDAD DE GUAYAQUIL."

		REVISORES:	
INSTITUCIÓN: Universidad de Guayaquil		FACULTAD: Ciencias Matemáticas y Físicas	
CARRERA: INGENIERÍA EN SISTEMAS COMPUTACIONALES			
FECHA DE PUBLICACIÓN:		Nº DE PÁGS.: 126	
ÁREA TEMÁTICA: BASE DE DATOS			
PALABRAS CLAVES: TRAMITES, PROCESAMIENTO, REGISTRO			
<p>RESUMEN: Este documento contiene el desarrollo de tema de tesis propuesto y aceptado por la universidad de guayaquil para su sustentacion. Este desarrollo esta basado en analisi de factores que afectan directamente a los estudiantes de las facultades de administracion y matematicas, exponiendo el probelma actual asi como tambien el desarrollo de una solucion vaible y factible que vaya en por de mejorar la situacion actual. Todo esto en base a levantamiento previo de informacion que se ha venido realizando en el campo donde se desarrolla el problema. Este contexto pretende exponer y explicar la mejor manera de solucionar el inconveninte que generar no contar con un sistema de manejo centralizado de las solicitudes y certificados que emiten las carreras, razon principal del estudio a exponer.</p>			
Nº DE REGISTRO(en base de datos):		Nº DE CLASIFICACIÓN:	
DIRECCIÓN URL (tesis en la web):			
ADJUNTO PDF		<input type="checkbox"/> SI	<input type="checkbox"/> NO
CONTACTO CON AUTOR: ALVARO LUIS VIZUETA LOJAN		Teléfono: 0998086607	E-mail: alvizueta@gmail.com
CONTACTO DE LA INSTITUCIÓN		Nombre:	
		Teléfono:	

APROBACION DEL TUTOR

En mi calidad de Tutor del trabajo de investigación, “Desarrollo del prototipo del sistema académico para la universidad de Guayaquil; Diseño de bandeja de trabajo”. elaborado por el Sr. ALVARO LUIS VIZUETA LOJAN, egresado de la Carrera de Ingeniería en Sistemas Computacionales, Facultad de Ciencias Matemáticas y Físicas de la Universidad de Guayaquil, previo a la obtención del Título de Ingeniero en Sistemas, me permito declarar que luego de haber orientado, estudiado y revisado, la Apruebo en todas sus partes.

Atentamente

Ing. Mary Cabanilla

TUTOR

DEDICATORIA

DEDICO ESTE TRABAJO A MI ADORADA HIJA IVANNA, PORQUE HA SIDO EL EJE DE MIS FUERZAS, A MI MADRE POR SER ESA PALABRA DE ALIENTO, A MI ESPOSA QUIEN ME HA COMPRENDIO Y APOYADO EN TODO ESTE PROCESO, A MI PADRE, QUIEN SIN SERLO BIOLOGICAMENTE, SUPO CUMPLIR TAN DELICADA LABOR, A MIS HERMANOS QUE DE UNA U OTRA MANERA ESTUVIERON AHÍ CONMIGO DANDOME FUERZAS, A MIS COMPAÑEROS DE LABORES, QUE SIEMPRE ME APOYARON Y CREYERON EN MI, Y EN ESPECIAL A MI QUERIDA ABUELA, MI ANGEL QUIEN ME CUIDA Y PROTEGE, Y AUNQUE YA NO ESTE CON NOSOTROS, NUNCA SE A ALEJADO DE MI.

AGRADECIMIENTO

PRIMERAMENTE A DIOS, A MI
FAMILIA Y A LA UNIVERSIDAD
DE GUAYAQUIL, CARRERA
DE INGENIERIA EN
SISTEMAS
COMPUTACIONALES, Y A
LOS PROFESORES QUE NOS
SUPIERON GUIAR EN ESTE
DURO ANDAR.

TRIBUNAL DE GRADO

Ing. Eduardo Santos Baquerizo. Msc.
DECANO DE LA FACULTAD
CIENCIAS MATEMATICAS Y FISICAS

Ing. Inelda Martillo Alcívar Mgs
DIRECTORA
CISC, CIN

Ing. Bernardo Iñiguez Mgs.
DIRECTOR DE TESIS

Nombre y Apellidos
PROFESOR DEL ÁREA -TRIBUNAL

Ab. Juan Chávez A.
SECRETARIO

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Tesis de Grado, me corresponden exclusivamente; y el patrimonio intelectual de la misma a la UNIVERSIDAD DE GUAYAQUIL”

ALVARO LUIS VIZUETA LOJAN



UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS

**CARRERA DE INGENIERIA EN SISTEMAS
COMPUTACIONALES**

DISEÑO Y DESARROLLO DE LA BANDEJA DE TRABAJO QUE PROCESARA
SOLICITUDES Y CERTIFICADOS PARA EL PROTOTIPO DE SISTEMA
ACADEMICO DE LA UNIVERSIDAD DE GUAYAQUIL.

Tesis de Grado que se presenta como requisito para optar por el título de
INGENIERO en SISTEMAS COMPUTACIONALES

Auto/a: ALVARO LUIS VIZUETA LOJAN

C.I. 0927197491

Tutor: ING. MARY CABANILLA

Guayaquil, Julio de 2015

CERTIFICADO DE ACEPTACIÓN DEL TUTOR

En mi calidad de Tutor de Tesis de Grado, nombrado por el Consejo Directivo de la Facultad de Ciencias Matemáticas y Físicas de la Universidad de Guayaquil.

CERTIFICO:

Que he analizado el Proyecto de Grado presentado por el/la estudiante ALVARO LUIS VIZUETA LOJAN, como requisito previo para optar por el título de Ingeniero en Sistemas Computacionales cuyo problema es:

Considero aprobado el trabajo en su totalidad.

Presentado por:

Álvaro Luis Vizueta Loján

Cedula de ciudadanía N° 0927197491

Tutor: Ing. Mary Cabanilla.

Guayaquil, Julio de 2015



UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS
CARRERA DE INGENIERIA EN SISTEMAS COMPUTACIONALES

Autorización para Publicación de Tesis en Formato Digital

1. Identificación de la Tesis

Nombre Alumno: Álvaro Luis Vizueta Loján	
Dirección: Ciudadela Martha de Roldos, Mz. 109 Villa. 9	
Teléfono: 0998086607	E-mail: alvizueta@gmail.com
Facultad: Ciencias Matemáticas y Físicas	
Carrera: Ingeniería en Sistemas Computacionales	
Título al que opta: Ingeniería en Sistemas Computacionales	
Profesor guía: Ing. Mary Cabanilla	
Título de la Tesis: Diseño y desarrollo de la bandeja de trabajo que procesara solicitudes y certificados para el prototipo de sistema académico de la universidad de Guayaquil.	
Temas Tesis: Bandeja de Trabajo.	

2. Autorización de Publicación de Versión Electrónica de la Tesis

A través de este medio autorizo a la Biblioteca de la Universidad de Guayaquil y a la Facultad de Ciencias Matemáticas y Físicas a publicar la versión electrónica de esta tesis.

Publicación electrónica:

Inmediata	<input checked="" type="checkbox"/>	Después de 1 año	<input type="checkbox"/>
-----------	-------------------------------------	------------------	--------------------------

Firma Alumno:

3. Forma de envío:

El texto de la Tesis debe ser enviado en formato Word, como archivo .Doc. O .RTF y .Puf para PC. Las imágenes que la acompañen pueden ser: .gif, .jpg o .TIFF.

DVDROM

CDROM

ÍNDICE GENERAL

APROBACION DEL TUTOR.....	I
DEDICATORIA	II
AGRADECIMIENTO	III
TRIBUNAL DE GRADO	IV
DECLARACIÓN EXPRESA	V
CERTIFICADO DE ACEPTACIÓN DEL TUTOR.....	VII
ÍNDICE DE TABLAS	XV
ÍNDICE DE GRÁFICOS	XVI
ABSTRACT	XVIII
INTRODUCCIÓN.....	1
CAPÍTULO I.....	2
EL PROBLEMA	2
PLANTEAMIENTO DEL PROBLEMA	2
SITUACIÓN CONFLICTO NUDOS CRÍTICOS	2
CAUSAS Y CONSECUENCIAS DEL PROBLEMA	3
DELIMITACIÓN DEL PROBLEMA.....	3
FORMULACIÓN DEL PROBLEMA.....	3
EVALUACIÓN DEL PROBLEMA	4
OBJETIVOS.....	6
OBJETIVO GENERAL	6
OBJETIVOS ESPECÍFICOS	6
ALCANCES DEL PROBLEMA.....	7
JUSTIFICACION E IMPORTANCIA.....	8
CAPÍTULO II.....	10

MARCO TEÓRICO	10
ANTECEDENTES.....	10
FUNDAMENTACIÓN TEÓRICA	11
CSS3	13
BOOTSTRAP.....	15
JAVA.....	16
ZK 7.0	19
HTML5.....	20
HIBERNATE	22
SQL SEVER 2012.....	23
APACHE TOMCAT 7	25
MVC	27
FLUJOS DE TRABAJO – WORKFLOW.....	30
HERRAMIENTA DE DESARROLLO.....	31
ECLIPSE LUNA	31
BASE DE DATOS.....	34
FUNDAMENTACIÓN LEGAL.....	35
HIPÓTESIS PREGUNTA A CONTESTARSE	38
VARIABLES DE LA INVESTIGACIÓN	39
CAPÍTULO III.....	40
METODOLOGÍA.....	40
DISEÑO DE LA INVESTIGACIÓN	40
MODALIDAD DE LA INVESTIGACIÓN.....	40
PROYECTO FACTIBLE.....	40
INSTRUMENTOS DE RECOLECCIÓN DE DATOS	42

CONTENIDO	42
IDENTIFICACIÓN DE LA INSTITUCIÓN:	42
MODO DE RESPUESTA:	42
INSTRUMENTO DE INVESTIGACIÓN	42
LA ENTREVISTA.....	43
ENTREVISTA A PERSONAL ADMINISTRATIVO.....	43
ANÁLISIS FODA.....	50
CRITERIO PARA LA ELABORACION DE LA PROPUESTA	54
CRITERIOS DE VALIDEZ DE LA PROPUESTA.....	54
OPERACIONALIZACIÓN DE VARIABLES	55
MAPA DE PROCESO DE LA BANDEJA DE TRABAJO	56
MAPA DE PROCESO DEL WORKFLOW.....	57
CAPÍTULO IV	58
MARCO ADMINISTRATIVO	58
DIAGRAMA DE GANTT.....	58
PRESUPUESTO.....	59
DETALLE DE EGRESOS DEL PROYECTO.....	59
CAPÍTULO V	60
CONCLUSIONES Y RECOMENDACIONES	60
CONCLUSIONES	60
RECOMENDACIONES.....	61
REFERENCIAS BIBLIOGRÁFICAS	61
BandejaTrabajo.java	67
BandejaTrabajoBO.java.....	84
OpcionesBO.java.....	91

EnviarCorreo.java	98
ProcesoPersonaEAO.java	99
ProcesoEAO.java	99
ProcesosEAO.java.....	100
DetalleRegistroEAO.java	101
ProcesoEAO.java	103
ProcesoSiguienteEAO.java	103
RegistroTramitesEAO.java.....	104
TipoTramiteEAO.java.....	106

ABREVIATURAS

ABP	Aprendizaje Basado en Problemas
UG	Universidad de Guayaquil
CISC	Carrera de Ingeniería en Sistemas Computacionales.
CINT	Carrera de Ingeniería en Networking y Telecomunicaciones.
FTP	Archivos de Transferencia
HTML	Lenguaje de Marca de salida de Hyper Texto
HTTP	Protocolo de transferencia de Hyper Texto
IDE	Integrated development environment.
MVC	Modelo Vista Controlador.
T-SQL	Transact-SQL
SQL	Structured Query Language(Lenguaje de Acceso a Base de Datos).
DDL	Lenguaje de Definición de Datos
CC.MM.FF	Facultad de Ciencias Matemáticas y Físicas
ISP	Proveedor de Servicio de Internet
POO	Programación Orientada a Objetos
Mtra.	Maestra
Msc.	Master
URL	Localizador de Fuente Uniforme
WWW	world wide web (red mundial)

SIMBOLOGÍA

S	Desviación estándar
E	Error
E	Espacio muestral
$E(Y)$	Esperanza matemática de la v.a. Y
S	Estimador de la desviación estándar
E	Exponencial

ÍNDICE DE TABLAS

Tabla 1. CUADRO DE ANALISIS FODA.....	51
Tabla 2.- Análisis de matriz FODA.....	53
Tabla 3. Operacionalización de las Variables	55
Tabla 4. Ingresos	59
Tabla 5. Detalle Egresos.....	59

ÍNDICE DE GRÁFICOS

Gráfico 1.- Logo JSON.....	12
Gráfico 2.-Logo CSS3.....	13
Gráfico 3.-Logo Bootstrap.....	15
Gráfico 4. Java	16
Gráfico 5.- Zk Framework	19
Gráfico 6. Logo HTML5.....	20
Gráfico 7. Secciones de una página web en HTML 5	21
Gráfico 8. Logo Hibernate.....	22
Gráfico 9. Logo SQL Server 2012.....	23
Gráfico 10. Logo Apache Tomcat	25
Gráfico 11. Modelo Vista Controlador	27
Gráfico 12.- Proceso MVC	29
Gráfico 13. Logo Eclipse Luna.....	31
Gráfico 14.- Pregunta 1.....	43
Gráfico 15.- Pregunta 2.....	44
Gráfico 16.- Pregunta 3.....	45
Gráfico 17.- Pregunta 4.....	45
Gráfico 18.- Pregunta 5.....	46
Gráfico 19.- Pregunta 6.....	47
Gráfico 20.- Pregunta 7.....	47
Gráfico 21.- Pregunta 8.....	48
Gráfico 22.- Pregunta 9.....	49
Gráfico 23.- Pregunta 10.....	49
Gráfico 24.- Mapa de proceso Bandeja de Trabajo.....	56
Gráfico 25.- Mapa de proceso WorkFlow	57
Gráfico 26.- CRONOGRAMA DE ACTIVIDADES	58



UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMATICAS Y FISICAS
CARRERA DE INGENIERIA EN SISTEMAS COMPUTACIONALES

**“DESARROLLO DEL PROTOTIPO DEL SISTEMA ACADÉMICO
PARA LA UNIVERSIDAD DE GUAYAQUIL;
DISEÑO DE BANDEJA
DE TRABAJO”**

Autor: ALVARO LUIS VIZUETA LOJAN

Tutor: ING. MARY CABANILLA

RESUMEN

Este documento contiene el desarrollo de tema de tesis propuesto y aceptado por la universidad de guayaquil para su sustentacion.

Este desarrollo esta basado en analisis de factores que afectan directamente a los estudiantes de las facultades de administracion y matematicas, exponiendo el probelma actual asi como tambien el desarrollo de una solucion vaible y factible que vaya en por de mejorar la situacion actual.

Todo esto en base a levantamiento previo de informacion que se ha venido realizando en el campo donde se desarrolla el problema.

Este contexto pretende exponer y explicar la mejor manera de solucionar el inconveninte que generar no contar con un sistema de manejo centralizado de las solicitudes y certificados que emiten las carreras, razon principal del estudio a exponer.



UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMATICAS Y FISICAS
CARRERA DE INGENIERIA EN SISTEMAS COMPUTACIONALES

“DEVELOPMENT ACADEMIC PROTOTYPE SYSTEM

UNIVERSITY OF GUAYAQUIL;

DESIGN TRAY WORK”

ABSTRACT

This document contains the development of proposed thesis topic and accepted by the University of Guayaquil for their support.

This development is based on factors affecting analisis directly to students of the powers of directors and mathematics, exposing the makeshift current as well as the development of a solution viable and feasible go in for improving the current situation.

All this on the basis of information prior to lift has been carried out in the field where the problem develops.

Expose this context and aims to explain the best way to solve the inconveniente not have to create a system of central management applications and certificates issuing the races, main reason of study to expose.

INTRODUCCIÓN

Este documento contiene información explícita del desarrollo y fundamentación en la que está basado este estudio de tesis, todo esto parte de una necesidad evidente como lo es la falta de un sistema que ayude a llevar un control más óptimo de las peticiones de certificados y de la revisión de solicitudes, dos conceptos que los englobaremos en un solo termino que es **“Tramites”**.

Todo el desarrollo está enfocado en plantear una solución a la problemática presente, problemática a la que se desea dar solución mediante el diseño y desarrollo de un sistema bajo ambiente Web.

Ambiente que permitirá y brindara un acceso remoto y seguro a los registros de las bases de datos de la Universidad de Guayaquil, todo desarrollado de manera centralizada.

El desarrollo de esta solución es parte de un prototipo integral que pretende automatizar los procesos que sean posibles dentro de la universidad para así brindar una mejor atención al estudiante, servir como herramienta fiable y practica que permita agilizar de las actividades dentro de lo administrativo.

CAPÍTULO I
EL PROBLEMA
PLANTEAMIENTO DEL PROBLEMA
UBICACIÓN DEL PROBLEMA EN UN CONTEXTO

En la actualidad la Universidad de Guayaquil no cuenta con un sistema centralizado, actualizado y unificado para la revisión y procesamiento de certificados y de solicitudes que generan los estudiantes de las facultades de Administración y Ciencias Matemáticas y Físicas de la Universidad de Guayaquil.

Estos procesos aún se los sigue revisando y procesando manualmente, revisando las actas de cada alumno, o por medio de pequeños sistemas no integrados, lo que hace que se corra riesgo de pérdida de información importante, además de generar pérdida de tiempo al revisar cada uno de los “trámites generados”.

SITUACIÓN CONFLICTO NUDOS CRÍTICOS

Al momento de requerir de un documento como lo es un certificado, y de obtener una respuesta a una solicitud de anulación de materia, anulación de inscripción, etc. Estos deben ser realizados manualmente, y entregado en las diferentes ventanillas de cada facultad o carrera para su revisión y posterior aprobación o negación de las mismas. Lo que para muchos de los estudiantes genera pérdidas de tiempo importantes ya que muchos de los que en la Universidad de Guayaquil se instruyen, son personas laboralmente activas.

Y qué decir del proceso posterior que es la revisión del mismo que toma otra cierta cantidad de tiempo para que sea tomada en cuenta y revisada por la persona que ha sido designada para este trabajo.

CAUSAS Y CONSECUENCIAS DEL PROBLEMA

La falta de un sistema centralizado integral y funcional que permita la revisión fácil y rápida de todos los “tramites” que son generados por los estudiantes de las facultades, crean inconvenientes de tiempo tanto para los encargados de este trabajo y para las personas que se verán beneficiadas por el resultado del mismo, genera malestar e inconveniente en muchos de los casos.

Otro tipo de consecuencia aún más grave, se podría decir es la probable perdida de información. Perdida de información ocasionada por errores humanos involuntarios o por negligencia del personal que labora dentro de la institución. Que puede ser evitada en gran parte al utilizar un sistema informático que permita el control y almacenamiento adecuado de toda la información de los tipos “tramites” contemplados.

DELIMITACIÓN DEL PROBLEMA

Este problema actualmente va enfocado específicamente al área encargada del proceso de trámites de solicitud y de certificados y a los estudiantes que generan estos trámites para sus actividades estudiantiles dentro de la carrera de Ingeniería en Sistemas de la Universidad de Guayaquil.

FORMULACIÓN DEL PROBLEMA

¿Qué inconvenientes causa no tener sistemas informáticos centralizados para el control de Trámites generados?

A más de crear un gran inconveniente que es la pérdida de tiempo incensario al momento de revisar todo lo que se ha generado dentro del periodo laboral, es la perdida de información histórica de todas las revisiones que se han llevado acabo, perdida de información que no es respaldada adecuadamente, tener que seguir obligatoriamente llevando manualmente el control de todo lo que se realiza con los tramites, tener poco control sobre el trabajo que una persona realiza, desencadenando en confusiones y malestar hacia la población estudiantil.

EVALUACIÓN DEL PROBLEMA

Estos son los principales aspectos tomados en cuenta para la evaluación del problema en cuestión.

Delimitado: Este problema no solo abarca o compromete a las personas que están detrás de los procesos de revisión de los tramites que se generan, también involucra a los estudiantes como afectados o beneficiados directos según el caso, ya que a falta de sistemas encargados de facilitar las tareas de revisión, estos generan tiempos de espera innecesarios, o verificación poco certera de los procesos que se realizan para la revisión de cada uno de los tramites.

Claro: el problema de no contar con sistemas informáticos de control en la actualidad, resulta difícil de creer, y más aún cuando una institución de tercer nivel, en alguna de sus carreras, enseña a utilizar estas herramientas en beneficio y mejoramiento de procesos que controlados manualmente o que están es sistemas aislados, y este es uno de los ejemplos claros de la poca puesta en práctica de lo que se enseña en las aulas.

Evidente: Es difícil no percibir o darse cuenta del malestar que se genera al momento de estar al pendiente de los tramites que cada uno de los usuarios (estudiantes) genera a lo largo del ciclo de estudio, esto debido a la falta un ágil e integral sistema de revisión y despacho de trámites.

Factible: Este problema es completamente factible de investigar y solucionar con las herramientas que se cuenta en la actualidad, herramientas de desarrollo en ambientes web, fáciles y seguros de acceder y que hoy en día son una gran solución para los problemas de trámites en todo ámbito de trabajo. Todo esto realizado y dirigido con recursos propios tales como los estudiantes que se instruyen diariamente en la Universidad de Guayaquil y que ven el problema desde un punto de vista más directo.

La solución de este problema no llevara de mucho tiempo de desarrollo ya que se cuenta con el recurso humano suficiente, y con recursos de hardware que no representaran valor alguno para la Universidad de Guayaquil.

Identifica los productos esperados: Este problema a más de claro indirectamente determina lo que se debe desarrollar o crear a fin de obtener un mejor desempeño en las funciones que cada una de las personas tiene dentro de una institución. Dejando también la posibilidad de poder integrar nuevos requerimientos de una manera ágil y rápida sin que esto afecte el normal desempeño de las funciones ya establecidas.

Variables: Las variables que se pueden destacar dentro de este contexto son las siguientes:

Tiempo: Que se pierde al esperar que un trámite sea revisado para obtener una posterior respuesta, ya sea este a favor o nula.

Satisfacción: Sabiendo que estos procesos general inconvenientes no solo de tiempo, generan un ambiente de insatisfacción, que desencadena en malas referencias del personal administrativo.

Seguridad: Como es de conocimiento general, los procesos manuales o separados, contienen bajo grado de seguridad, ya que son propensos a pérdidas de datos, tergiversación de información, ya sea mal intencionada o involuntaria.

Los autores Correa, Saavedra y Arévalo (2009), “muestran que gracias a la utilización de los sistemas de información se obtienen los siguientes beneficios”

- a. Acceso inmediato a la información ya sea de personas, datos, software o hardware.
- b. Mayor motivación para anticipar las solicitudes de las directivas.
- c. Evitar pérdida de tiempo en la recopilación de información.
- d. Impulsos para crear grupos de investigación.

e. Se generan más dinámicas, gracias a los medios informáticos; como el correo electrónico.

Según López (2006), "las organizaciones necesitan de la información para tener vida y prosperar, creciendo de esta manera hacia lugares lejanos, modificando así la forma de manejar los negocios".

Senn (2005, p.23), considera que : "las finalidades de los sistemas de información, como la de cualquier sistema dentro de una organización son ,procesar entradas, mantener archivos de datos relacionados con la organización y producir información, reportes y otras salidas"

OBJETIVOS

OBJETIVO GENERAL

Analizar y desarrollar el prototipo de la bandeja de entrada de solicitudes y certificados que generan los estudiantes, de acuerdo a los procesos levantados en las facultades de Administración y Matemáticas de la Universidad de Guayaquil.

OBJETIVOS ESPECÍFICOS

Diseñar la base de datos que contemplara y contendrá los datos de esta bandeja de entrada tomando como referencia toda la documentación obtenida previamente del levantamiento de información, para un correcto almacenamiento de la información.

Diseño de la interfaz gráfica y de los componentes de negocio que procesaran y direccionara los certificados y solicitudes de acuerdo a los flujos de trabajo que se definan, automatizando el flujo de revisión de cada tramite.

Desarrollar la bandeja de entrada de solicitudes y certificados de acuerdo a la información levantada, que ayude a centralizar estos dos tipos de peticiones.

Integrar la revisión de solicitudes y certificados en una sola bandeja de trabajo, las cuales serán revisadas por las personas designadas en el flujo de trabajo, en pro de mejorar de los tiempos de respuesta.

Desarrollar y documentar las funciones y procesos en el manual de usuarios de este módulo, de forma clara, objetiva y precisa que ayude a un correcto uso de la “Bandeja de Trabajo”.

ALCANCES DEL PROBLEMA

Se realizara el análisis y diseño de la interfaz gráfica y los componentes de desarrollo para la administración de la “Bandeja De Trabajo”.

De acuerdo con el resultado que muestra el análisis de los procesos y de la información levantada para el desarrollo de la bandeja de trabajo se destacan los siguientes puntos:

Análisis y desarrollo de una “Bandeja De Trabajo” estandarizada para la revisión y administración de solicitudes y certificados que se genere.

Presentación de esta Bandeja De Trabajo bajo perfiles debidamente registrados en la base de datos de seguridad.

Visión general y detallada en la Bandeja de Trabajo de todos los tramites generados en el Sistema Académico, con la capacidad de procesarlos en base a los flujos de trabajo determinados para cada tramite, flujos de trabajo que ya han sido determinados en del levantamiento de procesos.

Todo esto se desarrollara bajo los diseños de interfaces ya establecidos por el personal encargado de la arquitectura principal del Sistema Académico.

JUSTIFICACION E IMPORTANCIA

Es de suma importancia que se pueda proponer y desarrollar un sistema que centralice los tramites de certificados y solicitudes, pero que también a su vez los procese correctamente y dependiendo de la forma de trabajo de cada tramite, agilizando tiempos de respuesta, así como la seguridad y la veracidad de la información que sea emitida hacia el estudiante y sin que esto atente o sea propenso a perdida involuntaria de información. De manera colateral al mejorar los tiempos de respuesta y las facilidades de consulta que brinde el sistema se mejorara la visión general que tiene el estudiante acerca del servicio que se brinda para el bien de la población estudiantil.

Este estudio y desarrollo de este sistema beneficiara directamente, al personal administrativo que realiza la labor de revisión dentro de la Universidad y de manera colateral a los estudiantes que requieren estos trámites.

Para explicarlo concretamente expreso los siguientes puntos:

1.- Este desarrollo ayudara en gran medida al personal que realiza las funciones de control y verificación de los trámites que se generen, incluyendo en el desarrollo todos los conceptos y puntos de vista necesarios para llevar a cabo de manera más ágil y controlada.

2.- Esto ayudara directamente a controlar y centralizar procesos que se ha venido efectuando durante mucho tiempo manualmente, y que por obvias razones es obsoleto y propenso a fallos.

3.- Dejar presente que los profesionales que se forman dentro de esta institución son capaces de aportar con valores técnicos y científicos que ayudaran al mejoramiento de los sistemas y de las estructuras y que de alguna manera retribuyen lo que han recibido a lo largo de su desarrollo profesional, el conocimiento.

4.- La utilidad de los sistemas informáticos cada vez toma mayor fuerza, debido al gran aporte que brindan al desarrollo normal de los procesos, y este será uno de aquellos casos en que un proceso se verá potencializado por esta herramienta tan potente, y que cada vez más son parte fundamental del desarrollo de nuestras actividades diarias

CAPÍTULO II

MARCO TEÓRICO

ANTECEDENTES

Actualmente la revisión y emisión de certificados y la revisión de las solicitudes que ofrecen las diferentes facultades de la Universidad de Guayaquil, desde sus inicios se los ha manejado como procesos separados, y teniendo como problemática principal el tiempo de esperar hasta obtener un resultado en el caso de los certificados, y en el caso de las solicitudes, tener que esperar a ser atendido personalmente y que se pueda procesar la solicitud.

Todo esto especificado bajo un servicio que ofrece la universidad de Guayaquil a sus alumnos, servicio que como el de emisión de certificados ayuda a los alumnos a realizar diversos trámites fuera de la universidad que requieran de ello.

Las solicitudes con un relativo menor valor que el de los certificados, pero de igual importancia, utilizado como elemento interno para realizar peticiones de cambios de curso, materias etc. Es un servicio de gran demanda, especialmente al inicio de los semestres, ciclos o periodos de cada entidad dentro de la Universidad de Guayaquil.

Estos dos elementos, de gran demanda e importancia dentro de las actividades diarias de la Universidad, se siguen sometiendo a procesos manuales, aun en estos días donde la tecnología nos brinda la posibilidad de eliminar procesos manuales y sistematizar las actividades y mejorar de gran manera el tiempo de atención.

Este antecedente es lo que actualmente vive la mayoría de las entidades que pertenecen a la institución en mención, y a la actualidad no hay proyecto que plantee mejorar esta situación.

FUNDAMENTACIÓN TEÓRICA

Los tramites de solicitudes y certificados están definidos en base a las necesidades de los estudiantes para realizar las diversas actividades o peticiones que requieren siguiendo las normas y leyes actuales.

El proceso de revisión de los tramites están estipulados para cada persona que encargada dentro del proceso que conlleva cada tipo de trámites establecidos, de manera segura y procurando seguir la estructura central que se maneja, todo mediante la asignación de perfiles.

Esto dará la confianza y seguridad de que los tramite que se generen serán revisados correcta y específicamente por las personas encargadas, ayudando así a mejorar la agilidad de estos procesos, sin que ello afecte a la veracidad del mismo.

El desarrollo de la bandeja de trabajo está unido con los principio de revisión que conlleva cada una de las solicitudes y certificados establecidos y vigentes actualmente, asegurando así un correcto flujo de trabajo.

Dentro del manejo de este desarrollo tenemos algunos conceptos como:

- **JSON**
- **HIBERNATE**
- ECLIPSE LUNA**
- MVC**
- CSS3**
- APACHE TOMCAT 7**
- SQL SEVER 2012**
- ZK 7.0**
- JAVA 7**
- BOOTSTRAP**
- HTML5**
- BASE DE DATOS**
- METODO DE DESARROLLO DE PROTOTIPO**

Gráfico 1.- Logo JSON



Elaboración: Álvaro Vizueta

Fuente: ccia.ei.uvigo.es

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, Standard ECMA-262 3rd Edition - Diciembre 1999. (www.json.org, 2010)

JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos. (www.json.org, 2010)

JSON está constituido por dos estructuras: Una colección de pares de nombre/valor. En varios lenguajes esto es conocidos como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo. (www.json.org, 2010)

Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias. Estas son estructuras universales; virtualmente todos los lenguajes de programación las soportan de una forma u otra. (www.json.org, 2010)

Es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras. (www.json.org, 2010)

CSS3

Gráfico 2.-Logo CSS3



Elaboración: Álvaro Vizqueta
Fuente: pedroventura.com

"CSS" es un acrónimo de Cascading Style Sheets, se podría decir que CSS es un tipo de lenguaje que nos ayuda a establecer o diseñar el estilo de nuestras páginas web, que son hechas en HTML o en base a archivos XML.

CSS está creado para separar el contenido principal del estilo, pero permitiendo mantener a los diseñadores de manera simultáneamente un control más preciso sobre el diseño de las páginas.

CSS en sus inicios tuvo como objetivo principal el de separar controladamente el contenido de la página con la forma de la misma, esto desde sus inicios ha sido realizado satisfactoriamente, pero ahora se plantea un objetivo más complejo que es de permitir al diseñador obtener el control total sobre todos los elementos de la página, que ha sido difícil satisfacer

Especificaciones anteriores de CSS permitían utilizar diversas utilidades para los estilos de las páginas web pero aun con todo eso los encargados de desarrollo

siguen aplicando trucos para obtener efectos tan comunes como efectos más complejos.

La primera versión de CSS ya contribuyó considerablemente con avances en los diseños de páginas web, brindando gran control sobre los elementos de la página. Pero aun con todos los avances disponibles quedaban ciertas cosas que los encargados del diseño deseaban realizar pero que CSS aún no tenía especificado, y todo lo que no estaba contemplado debía ser realizado a base de trucos en el diseño.

La desventaja de realizar al tipo de truco era que se tenía que modificar el contenido de la página para tratar de incorporar nuevas etiquetas HTML que logre los efectos y los estilos deseados por los diseñadores. Con todos estos inconvenientes no se estaba logrando lo que en sí es la utilidad de CSS.

A raíz y necesidad nació CSS 2 que incorporaba nuevas utilidades que aún siguen siendo utilizadas, aun con estos avances se dio a conocer CSS3 que avanza aun un poco más en la meta de aportar mucho más control sobre los elementos de la página.

Dentro de lo que brinda CSS3, la novedad más importante que brinda a los diseñadores web es la inclusión de nuevos sistemas de control sobre el estilo, que ayudara de gran manera a mantener el control sobre el diseño de los elementos que se muestran en las páginas.

Todas las bondades que brinda CSS3 para el diseño se los pueden conseguir sin la necesidad de recurrir a hacks o trucos de programación que aparte de dar el resultado esperado complican el código de la web.

BOOTSTRAP

Gráfico 3. Logo Bootstrap



Elaboración: Álvaro Vizueta
Fuente: www.albertchurch.com

Framework originario de los diseñadores de Twitter que sirve de intermediario de conexión web con plantillas CSS y JavaScript. Su principal uso u bondad es la de poder adecuar las interfaces del web site a la del tamaño del Pc, o de darse el caso de que esta interface sea accedida desde un dispositivo móvil, poder adaptarse al tamaño de pantalla de la misma.

Para realizar o cumplir con esta propiedad Bootstrap utiliza una técnica de desarrollo que se conoce como “responsive desing” o llamado de otra manera “diseño adaptativo”, como se lo menciono anteriormente el trabajo principal que realiza esta técnica es la de adaptar automáticamente el diseño web al dispositivo desde donde se acceda.

Media Queries, módulo de CSS3, es el que permite que la interfaz gráfica y todo su contenido se adapte a la resolución del dispositivo, lo que permite obtener un diseño web fluido, acorde a casi cualquier tamaño de pantalla, todo automáticamente.

Aunque si se desea se puede trabajar con framework Bootstrap para el desarrollo de interfaces web de tipo adaptativos. Y después de todo lo que brinda el desarrollo en el framework en Bootstrap para diseño de sitios, estos son poco complejos e intuitivos, lo que brinda la agilidad necesaria para poder cargar estos elementos en casi cualquier dispositivo.

Bootstrap incluye una serie de componentes con estilos predefinidos y fáciles de configurar como lo son: Botones, menús, desplegables, formularios que incluyen

todos los elementos de integración de JQuery que ofrecen ventanas y tooltips dinámicos.

El soporte para HTML5 y CSS3 es relativamente incompleto, pero en contra de esa desventaja es que Bootstrap es compatible con casi todos los navegadores web, ya que toda la información principal para que sea compatible con los sitios o aplicaciones esta libre para todos los dispositivos y navegadores.

Se aplica un concepto de compatibilidad parcial que pone a disposición la información principal de un web site y que sea compatible con todos los dispositivos y navegadores.

Propiedades tales como las que permiten esquinas redondeadas, gradientes y sombras son utilizadas por Bootstrap aun cuando los navegadores antiguos no los soporten.

Desde la segunda versión de Bootstrap se incluye el soporte para diseños sensibles, lo que nos explica que el diseño gráfico se ajustara automática y dinámicamente, como siempre teniendo en cuenta las características del dispositivo usado.

JAVA

Gráfico 4. Java



Elaboración: Álvaro Vizuela
Fuente: www.oracle.com

Java es un lenguaje de **Programación Orientado A Objetos (POO)**, que desde sus inicios (la primera versión de JDK 1.0 fue lanzado en 1996), hasta ahora en la

actualidad se ha ido popularizando en el mundo de los programadores, hasta llegar a ser uno de los lenguajes más usados.

La base principal, o como también se lo podría denominar como antecedente los podremos ubicar en los lenguajes de programación C y C++. Destacando que lenguaje C fue desarrollado en los años 70, lenguaje robusto pero que no admite la orientación a objetos, también considerado un lenguaje poco amigable si tomamos en cuenta la sintaxis que utiliza y los elementos de programación que emplea como lo es el manejo directo de memoria, lo que lo hace considerarse un lenguaje difícil de aprender.

C++, considerado otro de los pilares de JAVA, es una evolución de C en los años 80. C++ comienza a introducir el diseño orientado a objetos, pero que sigue manteniendo compatibilidad con C.

El cambio suponía que C++ soportara la estructura tradicional de desarrollo y a su vez la de desarrollo orientado a objetos.

La evolución de C++ fue parcial ya que siguió manteniendo características de su lenguaje original tales como el manejo directo de la memoria, sentencias goto, y el uso de variables globales.

Un paso final si se podría decir de esta "evolución" es JAVA, lenguaje predecesor de C y C++, que quita por completo varios aspectos de cada uno de los lenguajes citados, constituyéndose en un lenguaje orientado a objetos.

El factor importante que coloca a Java como un lenguaje popular y de amplia gama de usos es el de romper aspectos de C++, que utilizado incorrectamente por parte de los desarrolladores generaba problemas en las aplicaciones que se desarrollen bajo este lenguaje

Punto principal o vista práctica de este contexto es determinar si es necesario saber C o C++ para poder programar en Java, pues, aunque C y C++ son los pilares o parte de la evolución hasta llegar al producto final que es JAVA, pues la respuesta es no, no es necesario saber de estos lenguajes antecesores.

No obstante se pudiere considerar que el conocimiento previo de estos dos lenguajes supone una ventaja sobre otras personas que no la conozcan, no es necesario el estudio en profundidad como paso previo al aprendizaje de Java.

Java es considerado un lenguaje muy útil para casi todo ámbito de problemas que se puedan resolver mediante programación:

1. Aplicaciones “cliente”: o aplicaciones locales, programas que se ejecutan en el ordenador del usuario, sin que sea necesario conectividad con otros terminales.

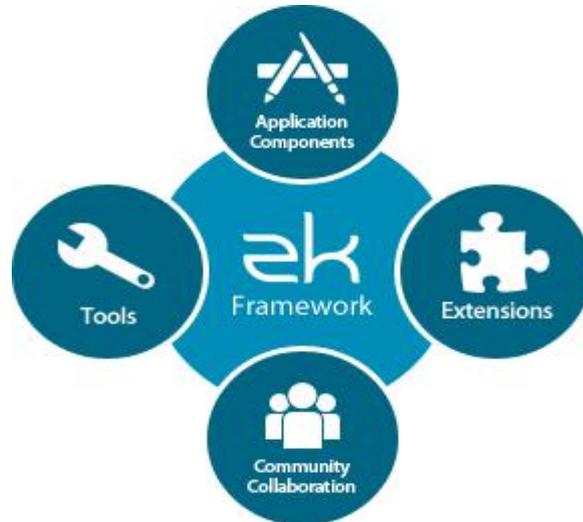
2. Aplicaciones “cliente/servidor”: como su nombre nos da a entender, son aplicaciones que requieren que un cliente se pueda conectar a un servidor y que este a su vez le dé una respuesta en base a las peticiones creadas desde la aplicación cliente.

3. Aplicaciones web: programas desarrollados en java, alojados en un servidor web y ejecutados desde una página web, estas aplicaciones reciben las solicitudes del cliente las procesan y devuelven una respuesta, todo mediante un explorador de internet o navegador (Internet Explorer, Firefox, Safari, etc.)

Estos son solo tres ejemplos comunes y cotidianos del uso del lenguaje de programación Java. Lenguaje de gran potencial y versatilidad que brinda grandes beneficios si se lo desarrolla con todo su potencial.

ZK 7.0

Gráfico 5.- Zk Framework



Elaboración: Álvaro Vizueta
Fuente: www.zkoss.org

ZK framework, es un framework de desarrollo de código abierto, ZK fue desarrollado gracias al aporte de desarrolladores propuestos a que la utilización de interfaces de usuario desarrollados bajo Ajax sea más cómoda y fácil de implementar y a su vez de desarrollar, ya que para muchos resulta algo complejo el desarrollo en este ambiente.

Uno de los aspectos más importantes del desarrollo en Zk es que el programador se “olvide” literalmente de JavaScript en la capa de presentación. Con esta ventaja se da a entender que el desarrollador solo se preocupe por desarrollar. También planteando la idea de desarrollar de manera tradicional sin tener que crear o manipular archivos varios de configuración y las capas de la aplicación.

Esto le permite a Zk desarrollar aplicaciones web Ajax con contenido similar a las comúnmente conocidas aplicaciones de escritorio tales como desarrollar con Swing en Java. Esto hace que Zk se torne un framework interesante para aquellos programadores interesados en seguir las tendencias actuales de desarrollo.

HTML5

Gráfico 6. Logo HTML5



Elaboración: Álvaro Vizueta
Fuente: www.trazos-web.com

Hyper Text Markup Language o **HTML**, determinado así por sus siglas en inglés que traducido al español es Lenguaje de Marcas de Hipertexto, es un lenguaje referenciado para el desarrollo de páginas web. Este estándar sirve de referencia en el diseño y desarrollo de páginas web.

El principio básico de desarrollo de HTML es la de poder referenciar un elemento externo para añadirlo, esto quiere decir que el elemento no se incrusta directamente en el código fuente, sino que para utilizarlo hace una referencia a la ubicación de cualquiera elemento mediante texto.

De esta manera el contenido de la página web es solo texto, que es interpretado por el navegador web y que también es el encargado de reunir todo el contenido y mostrarlo al usuario.

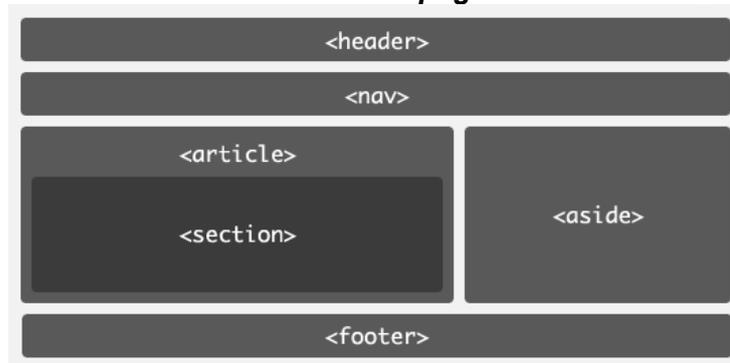
El estándar HTML lo que busca es que, cualquier página escrita en este estándar pueda ser interpretado por cualquier navegador actualizado.

La versión de **HTML5** nos brinda la posibilidad de una interacción mucho más amplia entre páginas web y el contenido multimedia que se quiera presentar, con mayor facilidad al momento de codificarlo.

Esta versión está basado en la recopilación de diseños más comunes de páginas web alrededor del mundo, lo que le permita llegar a un estándar de etiquetas que realicen las mismas tareas rápidas y eficientes:

- Para esta versión se han agregado las etiquetas: <header>, <footer>, <nav>, <section>, <article> la cuales pueden ser utilizadas para reemplazar los <div> en cada parte del contexto, y en cambio, tener etiquetas específicas para ello.

Gráfico 7. Secciones de una página web en HTML 5



*Elaboración: **Álvaro Vizueta**
Fuente: www.trazos-web.com*

- La nueva etiqueta <video> utilizado para insertar un reproductor de video, con esto se utiliza un reproductor más actualizado que el de la etiqueta <embed> evitando así la utilización de <object>, eliminado la necesidad de tener Flash Player
- La utilización de la nueva etiqueta <audio> para insertar audio en nuestro sitio web, reemplazando <embed> con igual características.
- Etiqueta <canvas> Utilizada para el manejo de gráficos, dibujos de vectores o realizar animaciones.

HIBERNATE

Gráfico 8. Logo Hibernate



Elaboración: Álvaro Vizuela

Fuente: www.json.org

¿Qué es un ORM?

Un ORM o Mapeador Objeto-Relacional es un framework que propone una nueva forma de modelar los datos, y que permite solucionar la diferencia que existe entre los paradigmas de la POO (Programación Orientada a Objetos) y el modelo Relacional. (www.es.slideshare.net, 2015)

El modelo Relacional trata con relaciones y conjuntos por lo cual tiende a ser de carácter matemático, mientras que el modelo de la POO trata con objetos y las asociaciones entre ellos; el problema entre estos dos modelos surge en el momento de querer persistir los objetos de negocio. (www.es.slideshare.net, 2015)

Los ORMs además de mapear, permite reducir susceptiblemente el código necesario para llevar a cabo las operaciones de persistencia y recuperación de objetos. Proporcionan interfaces más simples para el manejo de objetos a través de su propio lenguaje de consulta. (www.es.slideshare.net, 2015)

Proveen al programador de configuraciones que le permiten optimizar los tiempos de respuesta en sus correspondientes aplicaciones. (www.es.slideshare.net, 2015)

Hibernate herramienta de **JAVA** que facilita el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer las relaciones. (www.es.slideshare.net, 2015)

Parte de una filosofía de mapear objetos Java “normales”, también conocidos en la comunidad como “POJOs” (Plain Old Java Objects). (www.es.slideshare.net, 2015)

Proporciona un potente lenguaje de consulta (HQL): subquery, outer joins, ordering, projection (report query), paginación, etc. (www.es.slideshare.net, 2015)

Lenguaje intermedio que según la base de datos usaremos y el dialecto que especifiquemos será traducido al SQL dependiente de cada base de datos de forma automática y transparente. (www.es.slideshare.net, 2015)

SQL SEVER 2012

Gráfico 9. Logo SQL Server 2012



Elaboración: Álvaro Vizueta
Fuente: reckspace.com

Microsoft SQL Server es un sistema para la gestión de base de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consulta son T-SQL y ANSI SQL. (www.es.slideshare.net, 2015)

T-SQL, lenguaje de programación del SQL Server, a través de él podemos realizar muchas operaciones relacionadas con el SQL sin tener que volver a pasar por código ASP o VB, esto simplifica el código y ganara en rapidez dado que el T-SQL se ejecuta dentro del SQL Server y es código compilado, se compila la primera vez que se ejecuta el Store Procedure. (www.es.slideshare.net, 2015)

El T-SQL se puede utilizar desde multitud de aplicaciones y desde diferentes lenguajes de programación.

- Desde Visual Basic.
- Desde Visual C++.
- Desde Active Server Pages (ASP).

Características de SQL Server

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red solo acceden a la información.
- Además permite administrar información de otros servidores de datos.

Entorno Grafico SQL Server

En el entorno gráfico de SQL server, podemos administrar las bases de datos que tenemos, así como ver Log de errores, exportar listas, o realizar respaldos de las bases de datos, entre otras muchas funciones. (www.es.slideshare.net, 2015)

DDL. Un lenguaje de definición de datos es un lenguaje que permite a los usuarios llevar a cabo las tareas de definición de los procedimientos o funciones que permitan consultarlos. (www.es.slideshare.net, 2015)

El lenguaje de programación SQL, admite las siguientes sentencias de definición:

- **CREATE**
- **DROP**
- **ALTER**

Cada una de las cuales se puede aplicar a las tablas, vistas, procedimientos almacenados y triggers de la base de datos.

DCL (Lenguaje de Manipulación de Datos) o Data Manipulation Language, DML) es un lenguaje que permite a los usuarios llevar a cabo las tareas de consulta o manipulación de los datos, organizados por el modelo de datos

adecuado. Es usado para recuperar y manipular datos en una base de datos relacional. (www.es.slideshare.net, 2015)

Desventajas

- Usa Address Windowing extension (AWE), para hacer el direccionamiento de 64Bit esto le impide usar la administración dinámica de la memoria y solo le permite alojar lo más 64Gb de memoria compartida.
- Solo permite 16 instancias distintas concurrentes aproximadamente en una máquina.
- No maneja compresión de datos por tanto ocupa mucho espacio en disco.
- Está atado a la plataforma de sistema operativo sobre la cual puede instalarse.

APACHE TOMCAT 7

Gráfico 10. Logo Apache Tomcat



Elaboración: Álvaro Vizueta
Fuente: ribnix.com

Apache Tomcat tiene un contenedor de servlets (catalina) que fue desarrollado bajo el proyecto Jakarta en Apache Software Foundation, creado directamente por Duncan Davidson. (www.ajpdssoft.com, 2015)

Este servidor puede implementar las especificaciones de los servlets y de JavaServer Pages (JSP) que son nativos de Sun Microsystems. Como muchos pueden llegar a pensar o creer Apache Tomcat no es un servidor de aplicaciones, pero que puede ser utilizado en varias plataformas. (www.ajpdssoft.com, 2015)

De manera simplificada, diremos que Apache Tomcat (o Jakarta Tomcat) es un software utilizado como servidor de aplicaciones Web, desarrolladas en Java que puede ser ejecutado en cualquier sistema operativo gracias a la Máquina Virtual de java, lo que le da una gran versatilidad de operación.

Tomcat como tal es administrado y mantenido por desarrolladores de la Apache Software Foundation y por desarrolladores voluntarios en todo el mundo que también aportan en su desarrollo.

Apache Tomcat es de libre uso, todos los usuarios que deseen tiene a su disposición el código fuente y a su vez también la distribución binaria en los términos establecidos en la Apache Software License.

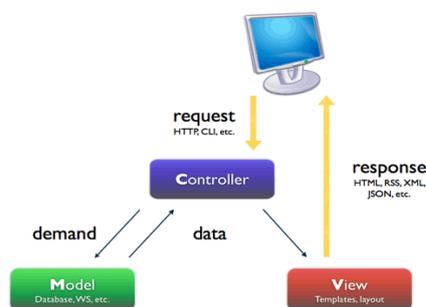
La versión con la que cuenta actualmente Apache Tomcat es la versión 7.x. en las cuales usamos las especificaciones de Servlet 3.0 y Jsp 2.2. el contenedor de servlets que actualmente se lo conoce como Catalina fue puesto en la versión 4.0

Tomcat como servidor web es capaz de soportar servlets y JSP, pero esto no quiere decir que Tomcat sea un servidor de aplicaciones como JBoos o JOnAS. También contiene el compilador Jasper, que se encarga de compilar los JSPs para a su vez convertirlos en servlets.

En la actualidad Tomcat es utilizado como servidor web independiente en entornos complejos, de alto índice de tráfico y que requieren una alta disponibilidad. Esto vale recalcar ya que en sus inicios Tomcat fue considerado como solo un servidor de desarrollo o de entornos pequeños con requerimientos bajos.

MVC

Gráfico 11. Modelo Vista Controlador



Elaboración: Álvaro Vizuela
Fuente: librosweb.es

MVC, modelo denominado así por la abreviación de las palabras **Modelo Vista Controlador**, es un diseño de desarrollo de software, que propone que los datos que vaya a utilizar la aplicación se manejen separado de la parte que se muestra al usuario o también denominado interfaz gráfica y a su vez de la lógica de control, todo esto mediante la utilización de tres componentes distintos.

Este modelo de desarrollo ha mostrado su eficacia dentro del periodo que ha sido utilizado, y esto lo has demostrado ya que muchos desarrolladores de software han comprobado su funcionalidad en diversas plataformas de desarrollo y lenguajes de programación.

- Se denomina componente **Modelo** al encargado de dar una representación de todo lo que utiliza de parte del sistema (sistemas o partes de acceso y la forma de comportamiento de los datos o lógica de negocios).
- Se denomina **Vista**, parte grafica o interfaz de usuario, al que se encarga de interactuar con el cliente que utilice la información del sistema, y de cómo este es capaz de ser entendido por el mismo.
- Se denomina **Controlador**, al que trabaja o actúa como un intermediario de comunicación entre el **Modelo** y la **Vista**, ya que interactúa entre ellos realizando transformaciones que permitan la adaptabilidad de la información que se maneje entre uno y otro ente.

Modelo, está encargado de:

- Mediante una conexión a una base de datos específica, ingresa donde está alojada la información. Un punto importante del modelo es que este independiente del medio de almacenamiento.
- Establece las reglas a seguir para la lógica de negocios.
- Un modelo activo, es capaz de avisar cualquier cambio que se pueda producir desde un medio externo.

Controlador, está encargado de:

- Receptar todos los eventos de entrada, ya sea este un simple cambio de palabras en una caja de texto, clic en algún botón, carga de datos en un listbox, etc.)
- En este componente se ubicaran las reglas que se seguirán para la administración de eventos.

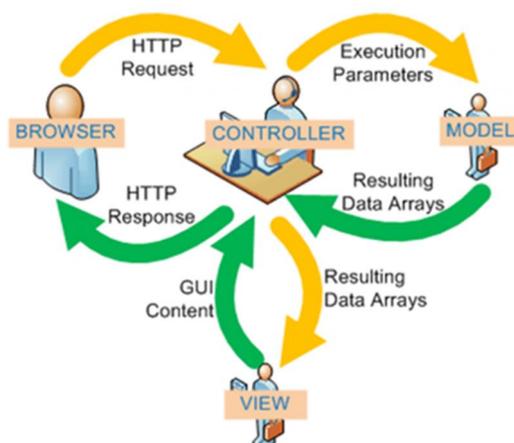
Las vistas son responsables de:

- Mostrar al cliente los datos que pudiese recibir del modelo, dependiendo de la acción que se realice.
- Son capaces de brindar un servicio de tipo actualización, para que estos puedan ser llamados desde el modelo o el controlador.

Patrón característico de funcionamiento de Control:

1. Un usuario trabaja con la interfaz de la forma posible establecida (pulsando un botón, link, etc.).
2. La interfaz provee al controlador la notificación solicitada por el cliente. Este evento es administrado por el controlador que de manera común se lo realiza a través de un gestor de eventos llamado handler o callback.
3. El siguiente paso es que el controlador acceda al modelo, modificando correctamente en base a la acción solicitada por el usuario.

Gráfico 12.- Flujo MVC



Elaboración: Álvaro Vizuela

Fuente: www.si.ua.es

- Este a su vez se encarga de administrar los objetos de la vista para que se pueda crear la vista de usuario. En esta parte la vista recibe los datos del modelo para visualizar la interfaz correcta para el cliente en la cual se reflejan los cambios que se han realizado en el modelo.

La vista no tiene que ser del conocimiento del modelo, pero se podría usar un patrón de Observador para suministrar cierto tipo de dirección entre la vista y el modelo, dejando que se le pueda notificar cualquier cambio de interés al modelo.

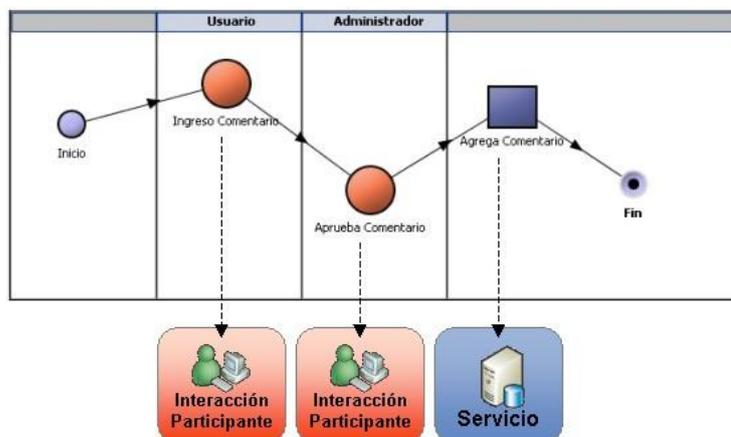
El controlador como tal no es capaz de suministrar algún objeto que se encuentre en el dominio a la vista, pero si es capaz de dar algún tipo de orden para refrescar la vista.

Muchas implementaciones la vista no tiene acceso al modelo de forma directa, todo esto se lo deja al controlador para que realice esta labor.

- Este ciclo se repite nuevamente cuando el cliente realiza alguna interacción con la interfaz del sistema.

FLUJOS DE TRABAJO – WORKFLOW

Gráfico 13.- Ejemplo de Flujo de Trabajo



Elaboración: Álvaro Vizuela

Fuente: <http://72.55.164.244:8090/web/guest/bpm>

El flujo de trabajo (workflow en inglés) es el estudio de los aspectos operacionales de una actividad de trabajo: cómo se estructuran las tareas, cómo se realizan, cuál es su orden correlativo, cómo se sincronizan, cómo fluye la información que soporta las tareas y cómo se le hace seguimiento al cumplimiento de las tareas. Generalmente los problemas de flujo de trabajo se modelan con redes de Petri.

Si bien el concepto de flujo de trabajo no es específico a la tecnología de la información, una parte esencial del software para trabajo colaborativo(groupware) es justamente el flujo de trabajo.

Una aplicación de flujos de trabajo automatiza la secuencia de acciones, actividades o tareas utilizadas para la ejecución del proceso, incluyendo el seguimiento del estado de cada una de sus etapas y la aportación de las herramientas necesarias para gestionarlo

Se pueden distinguir tres tipos de actividad:

- **Actividades colaborativas:** Un conjunto de usuarios trabajan sobre un mismo repositorio de datos para obtener un resultado común. Tiene entidad el trabajo de cada uno de ellos en sí mismo.
- **Actividades cooperativas:** Un conjunto de usuarios trabajan sobre su propio conjunto particular, estableciendo los mecanismos de cooperación entre

ellos. No tiene entidad el trabajo de ninguno de ellos si no es visto desde el punto de vista global del resultado final.

- Actividades de coordinación.

OBJETIVOS

- Reflejar, mecanizar y automatizar los métodos y organización en el sistema de información.
- Establecer los mecanismos de control y seguimiento de los procedimientos organizativos.
- Independizar el método y flujo de trabajo de las personas que lo ejecutan.
- Facilitar la movilidad del personal.
- Soportar procesos de reingeniería de negocio.
- Agilizar el proceso de intercambio de información y agilizar la toma de decisiones de una organización, empresa o institución.
- Adicionalmente optimizar el servicio

HERRAMIENTA DE DESARROLLO

La herramienta de desarrollo seleccionada ha sido ECLIPSE LUNA, la cual la describimos a continuación.

ECLIPSE LUNA

Gráfico 14. Logo Eclipse Luna



Elaboración: Álvaro Vizuela
Fuente: www.bsiag.com

Eclipse de Luna incluye soporte oficial para Java TM 8 en las herramientas de desarrollo Java, plug-in de herramientas de desarrollo, objeto Equipos, Eclipse

Framework Comunicación, integración Maven, xtext, Xtend, Web Tools Platform, y Memory Analyzer. El compilador de Eclipse incluye mejoras en el idioma, búsqueda y refactorización, Quick Assist y limpiar migrar clases anónimas a las expresiones lambda y la espalda, y nuevas opciones formateador para lambdas. (www.bsiag.com, 2013)

Destacados Luna incluyen:

Soporte completo para Java [™] 8 el workbench de trabajo ofrece un nuevo tema oscuro, editores divididas, números de línea activada de forma predeterminada, la reducción de los espacios en blanco en la presentación por defecto, y la capacidad para ocultar la barra de "acceso rápido". (www.bsiag.com, 2013)

Sirius permite a los arquitectos para crear fácilmente sus propios bancos de trabajo de modelado gráfico mediante el aprovechamiento de las tecnologías de modelado de Eclipse, incluyendo Eclipse Modeling Framework (EMF) y gráfica Modeling Framework (GMF)

Equinox incluye una implementación completa del Marco de base de R6, así como varias implementaciones de servicios compendio Aplicación de ECF de servicio remoto OSGi / estándar Servicio de administración remota se ha mejorado para utilizar CompleteableFuture de Java 8 para servicios remotos asíncronos. (www.bsiag.com, 2013)

Código recomendado integra el motor de búsqueda de fragmento de código Snipmatch y añade la posibilidad de contribuir fácilmente nuevos fragmentos de un repositorio compartido. (www.bsiag.com, 2013)

El nuevo depurador QVTi extiende el nuevo depurador OCL. La OPS proporciona implementaciones de código abierto de los protocolos de mensajería abiertas y estándar que apoyan las necesidades actuales y emergentes de la integración M2M con Web y middleware y aplicaciones empresariales. (www.bsiag.com, 2013).

Formas EMF como parte de la plataforma de cliente EMF ofrece una forma completamente nueva para el desarrollo eficiente de las interfaces de usuario basadas en formularios. (www.bsiag.com, 2013)

La especificación UML 2.5 es ahora compatible y diagramas UML se puede validar contra el OCL incrustado en perfiles UML.

EcoreTools, el modelador gráfico para Ecore, ha sido completamente re-implementado en base a Sirius y ofrece un diseño gráfico más agradable, el apoyo para la documentación de su modelo, especificando las limitaciones de dominio y el modelado de tipos genéricos. (www.bsiag.com, 2013)

El workbench de Eclipse ofrece un nuevo tema oscuro que incluye un ajuste resaltador de sintaxis apropiado para el editor de varios lenguajes de programación.

TCF Terminal soporta una vista de línea de comandos integrada en Windows, Linux y Mac. Las herramientas de desarrollo de PHP vienen con soporte para PHP 5.5 y un mejor desempeño en el editor de PHP. (www.bsiag.com, 2013)

El paquete de "Eclipse para desarrolladores de PHP" en el sitio de descarga de Eclipse proporciona una manera fácil de empezar a desarrollar aplicaciones PHP.

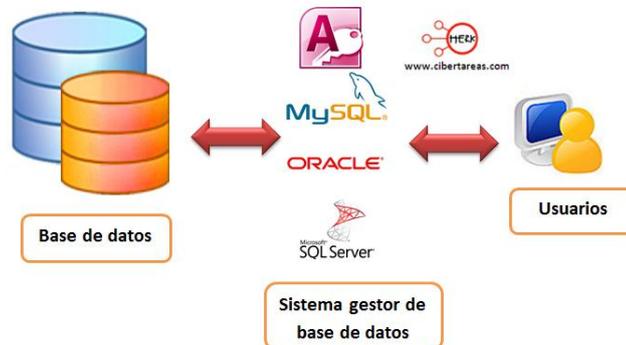
La liberación Gyrex 1.3 ahora incluye un bus de eventos distribuida que aprovecha el nuevo soporte WebSocket completa para una comunicación más fácil en los nodos de un clúster. (www.bsiag.com, 2013)

Memory Analyzer ahora analiza vuelcos de almacenamiento dinámico de Java 8, incluyendo las clases de colecciones Java 8. Ahora hay una aplicación independiente C / C ++ depurador añade a la CDT.

Ahora es posible para depurar una aplicación a través de la línea de comandos y tener una interfaz de usuario de Eclipse depurador emergente. (www.bsiag.com, 2013)

BASE DE DATOS

Gráfico 15.- Base de Datos



Elaboración: Álvaro Vizuela
Fuente: www.azucenamarez.blogspot.com

Una base de datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Una base de datos es un sistema de archivos electrónico.

Las bases de datos tradicionales se organizan por campos, registros y archivos. Un campo es una pieza única de información; un registro es un sistema completo de campos; y un archivo es una colección de registros. Por ejemplo, una guía de teléfono es análoga a un archivo. Contiene una lista de registros, cada uno de los cuales consiste en tres campos: nombre, dirección, y número de teléfono.

Componentes básicos de una base de datos

Documentos, constituyen la entidad físico/cognitiva compleja que alberga la estructura formal, basada en los datos físicos necesarios para su identificación (título, autor, lugar de publicación, fecha, edición,...) y la estructura lógico-cognitiva, centrada en el contenido y en las propiedades semánticas.

Representación de documentos, tanto de sus propiedades físicas como semánticas se hace mediante palabras clave, frases, etc. que servirán de puntos de acceso cuando interroguemos al sistema.

Necesidades de información de los usuarios, manifestadas en la solicitud de información.

Representación de las necesidades de información, expresadas también con palabras clave o frases.

Comparación de la representación de información con la representación de los documentos. Las bases de datos, basadas en la función semejanza comparan, a través de un índice, ambas representaciones para seleccionar los documentos relevantes.

FUNDAMENTACIÓN LEGAL

El tema propuesto está en base a leyes que contemplan la viabilidad del tema propuesto:

LEY DE EDUCACIÓN SUPERIOR

PRINCIPIOS DEL SISTEMA DE EDUCACIÓN SUPERIOR

Art. 13.- Funciones del Sistema de Educación Superior.- Son funciones del Sistema de Educación Superior:

- a)** Garantizar el derecho a la educación superior mediante la docencia, la investigación y su vinculación con la sociedad, y asegurar crecientes niveles de calidad, excelencia académica y pertinencia.
- b)** Promover la creación, desarrollo, transmisión y difusión de la ciencia, la técnica, la tecnología y la cultura.
- c)** Formar académicos, científicos y profesionales responsables, éticos y solidarios, comprometidos con la sociedad, debidamente preparados para que sean capaces de generar y aplicar sus conocimientos y métodos científicos, así como la creación y promoción cultural y artística.

LEY DE PROPIEDAD INTELECTUAL

SECCION V

DISPOSICIONES ESPECIALES SOBRE CIERTAS OBRAS

PARAGRAFO PRIMERO

DE LOS PROGRAMAS DE ORDENADOR

Art. 28.- Los programas de ordenador se consideran obras literarias y se protegen como tales. Dicha protección se otorga independientemente de que hayan sido incorporados en un ordenador y cualquiera sea la forma en que estén expresados, ya sea en forma legible por el hombre (código fuente) o en forma legible por máquina (código objeto), ya sean programas operativos y programas aplicativos, incluyendo diagramas de flujo, planos, manuales de uso, y en general, aquellos elementos que conformen la estructura, secuencia y organización del programa.

Art. 29.- Es titular de un programa de ordenador, el productor, esto es la persona natural o jurídica que toma la iniciativa y responsabilidad de la realización de la obra. Se considerará titular, salvo prueba en contrario, a la persona cuyo nombre conste en la obra o sus copias de la forma usual. 58 Dicho titular está además legitimado para ejercer en nombre propio los derechos morales sobre la obra, incluyendo la facultad para decidir sobre su divulgación. El productor tendrá el derecho exclusivo de realizar, autorizar o prohibir la realización de modificaciones o versiones sucesivas del programa, y de programas derivados del mismo. Las disposiciones del presente artículo podrán ser modificadas mediante acuerdo entre los autores y el productor.

Art. 30.- La adquisición de un ejemplar de un programa de ordenador que haya circulado lícitamente, autoriza a su propietario a realizar exclusivamente:

a) Una copia de la versión del programa legible por máquina (código objeto) con fines de seguridad o resguardo;

b) Fijar el programa en la memoria interna del aparato, ya sea que dicha fijación desaparezca o no al apagarlo, con el único fin y en la medida necesaria para utilizar el programa; y,

c) Salvo prohibición expresa, adaptar el programa para su exclusivo uso personal, siempre que se limite al uso normal previsto en la licencia. El 59 adquirente no podrá transferir a ningún título el soporte que contenga el programa así adaptado, ni podrá utilizarlo de ninguna otra forma sin autorización expresa, según las reglas generales. Se requerirá de autorización del titular de los derechos para cualquier otra utilización, inclusive la reproducción para fines de uso personal o el aprovechamiento del programa por varias personas, a través de redes u otros sistemas análogos, conocidos o por conocerse.

Art. 31. -No se considerará que exista arrendamiento de un programa de ordenador cuando éste no sea el objeto esencial de dicho contrato. Se considerará que el programa es el objeto esencial cuando la funcionalidad del objeto materia del contrato, dependa directamente del programa de ordenador suministrado con dicho objeto; como cuando se arrienda un ordenador con programas de ordenador instalados previamente.

Art. 32. -Las excepciones al derecho de autor establecidas en los artículos 30 y 31 son las únicas aplicables respecto a los programas de ordenador. Las normas contenidas en el presente Parágrafo se interpretarán de manera que su aplicación no perjudique la normal explotación de la obra o los intereses legítimos del titular de los derechos.

DECRETO 1014

SOBRE EL USO DEL SOFTWARE LIBRE

Art. 1.- Establecer como política pública para las entidades de administración Pública central la utilización del Software Libre en sus sistemas y equipamientos informáticos.

Art. 2.- Se entiende por software libre, a los programas de computación que se pueden utilizar y distribuir sin restricción alguna, que permitan el acceso a los códigos fuentes y que sus aplicaciones puedan ser mejoradas. Estos programas de computación tienen las siguientes libertades:

- Utilización de programa con cualquier propósito de uso común.

- Distribución de copias sin restricción alguna.
- Estudio y modificación de programa (Requisito: código fuente disponible).
- Publicación del programa mejorado (Requisito: código fuente disponible)

Art. 3.- Las entidades de la administración pública central previa a la instalación del software libre en sus equipos, deberán verificar la existencia de capacidad técnica que brinde el soporte necesario para este tipo de software.

Art. 4.- Se faculta la utilización de software propietario (no libre) únicamente cuando no exista una solución de software libre que supla las necesidades requeridas, o cuando esté en riesgo de seguridad nacional, o cuando el proyecto informático se encuentre en un punto de no retorno.

Art. 5.- Tanto para software libre como software propietario, siempre y cuando se satisfagan los requerimientos.

Art. 6.- La subsecretaría de Informática como órgano regulador y ejecutor de las políticas y proyectos informáticos en las entidades de Gobierno Central deberá realizar el control y seguimiento de este Decreto.

Art. 7.- Encargue de la ejecución de este decreto los señores Ministros Coordinadores y el señor Secretario General de la Administración Pública y Comunicación.

LEY ORGANICA DE EDUCACIÓN SUPERIOR

Art. 32.- Programas informáticos.- Las empresas que distribuyan programas informáticos tienen la obligación de conceder tarifas preferenciales para el uso de las licencias obligatorias de los respectivos programas, a favor de las instituciones de educación superior, para fines académicos. Las instituciones de educación superior obligatoriamente incorporarán el uso de programas informáticos con software libre.

HIPÓTESIS PREGUNTA A CONTESTARSE

- ¿El prototipo de la Bandeja de Trabajo ayudara a revisar y gestionar correcta y ordenadamente las solicitudes y certificados que se generen por los estudiantes de la de las facultades de Administracion y Ciencias Matematicas y Fisicas de la Universidad de Guayaquil, agilitando el tiempo de respuesta que se tiene actualmente?

VARIABLES DE LA INVESTIGACIÓN

VARIABLES INDEPENDIENTES

Diseño de la base de datos para el prototipo de la Bandeja de Trabajo para el procesamiento de Certificados y Solicitudes.

VARIABLES DEPENDIENTES

Diseño y Desarrollo del Prototipo de la Bandeja de Trabajo para el procesamiento de Certificado y Solicitudes del Sistema Académico.

CAPÍTULO III

METODOLOGÍA

DISEÑO DE LA INVESTIGACIÓN

MODALIDAD DE LA INVESTIGACIÓN

Este trabajo de investigación está fundamentado en plantear el prototipo de Bandeja de Trabajo para el procesamiento de solicitudes y certificados, como una herramienta funcional de trabajo para las personas encargadas de revisar estos procesos, aprovechando las bondades y facilidades que nos brinda las tecnologías de desarrollo web, determinando que lo que se está planteando es una investigación factible, ya que de manera porcentual el proyecto se desglosa en : 20% investigativo, 20% bibliográfico, y un 60% de la propuesta.

El objetivo principal de este o de cualquier otro sistema que se plantee es la de poder eliminar en lo posible la mayor cantidad de procesos manuales, y hacerlos sistematizados.

Todo esto se desarrolla en busca del objetivo principal que es cumplir todo lo planteado, y dentro de los límites propuesto

PROYECTO FACTIBLE

Esta propuesta para el desarrollo del prototipo de la bandeja de trabajo que servirá de integrador de las solicitudes y certificados generados por los estudiantes que aquí se instruyen como propuesta de tesis del área educativa de la Universidad de Guayaquil, se rige a pasos que ayudaran a seguir la estructura principal de desarrollo. Y que nos dará las bases para establecer y satisfacer las necesidades de la muestra que parte de este estudio.

Todo esto para plantear un cambio que signifique la mejora de los procesos actuales, proyecto de tipo educativo, enfocado bajo una investigación de tipos de

Proyectos Factibles, todo esto determinado en base al planteamiento y detalle de problemas, para generar así una propuesta de posibles cambios.

PROCEDIMIENTO DE LA INVESTIGACION

Explicado de manera secuencial el orden del desarrollo de la investigación, desde sus orígenes hasta los criterios para la validez de la propuesta.

EL PROBLEMA

Planteamiento del problema

Situación conflicto nudos críticos

Causas y consecuencias del problema

Delimitación del problema

Formulación del problema

Evaluación del problema

Objetivos

Objetivo general

Objetivos específicos

Alcances del problema

Justificación e importancia

MARCO TEORICO

Antecedentes

Fundamentación teórica

Fundamentación legal

Hipótesis preguntas a contestarse

Variables de la investigación

METODOLOGIA

Diseño de la investigación

Modalidad de la investigación

Proyecto factible

Operacionalización de variables

Instrumentos de recolección de datos

Criterio para la elaboracion de la propuesta

Criterios de validez de la propuesta

INSTRUMENTOS DE RECOLECCIÓN DE DATOS

Este trabajo investigativo está realizada y sustentada en herramientas que permiten la recopilación de información que proviene de las diversas áreas parte del estudio que están involucradas en el problema planteado.

CONTENIDO

IDENTIFICACIÓN DE LA INSTITUCIÓN:

Universidad de Guayaquil – Facultades de Ciencias Físicas y Matemáticas.

OBJETIVO:

Conocer el concepto común del personal administrativo de cara a lo que actualmente presenta la institución y de una futura proyección de parte de los mismos. Recopilando información necesaria sobre si las facultades se encuentran preparadas para captar las peticiones que se receptan y dirigirlas al sistema académico existente.

MODO DE RESPUESTA:

Preguntas específicas y de respuesta abierta, que permite saber con exactitud las opiniones expresadas en base a la entrevista planteada, y así definir también si la solución planteada tiene el grado de aceptación suficiente por parte del personal entrevistado.

INSTRUMENTO DE INVESTIGACIÓN

Entrevista: Dirigida al personal administrativo encargado de la recepción y procesamiento de tramites (solicitudes y certificados) de la Universidad de Guayaquil, específicamente a la facultad de ciencias Matemáticas y Físicas, campo de desarrollo de la problemática.

LA ENTREVISTA

Entrevista realizada al personal encargado de revisión y posterior procesamiento de tramites (solicitudes y certificados) de la carrera de Ingeniería en sistemas y Networking de la facultad de Ciencias Matemáticas y Físicas de la Universidad de Guayaquil, que busca como objetivo principal conocer las opiniones sobre el prototipo del sistema que se desea implementar, y a su vez también saber que concepto se tiene de los métodos actuales.

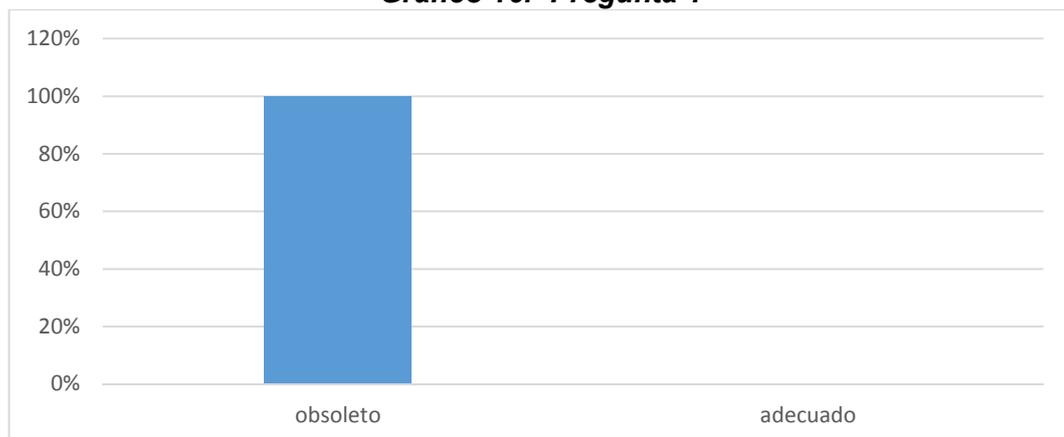
Tema: Prototipo de Bandeja de Trabajo para el proceso de certificados y solicitudes.

ENTREVISTA A PERSONAL ADMINISTRATIVO

1.- ¿Cómo ve actualmente la manera de revisar y procesar los tramites?

Del total de entrevistados, el **100%** concuerdan que la manera de llevar el proceso de revisión de los tramites es obsoleto, y que por tal motivo hay mucho retraso al procesar dichos tramites, esto debido a la gran demanda de la población estudiantil, que de manera directa se ve afectada por los tiempos el tiempo que deben esperar para obtener una respuesta, y a su vez afecta al personal administrativo ya revisar esto conlleva en muchos de los casos bastante tiempo de revisión en su jornada diaria.

Gráfico 16.- Pregunta 1

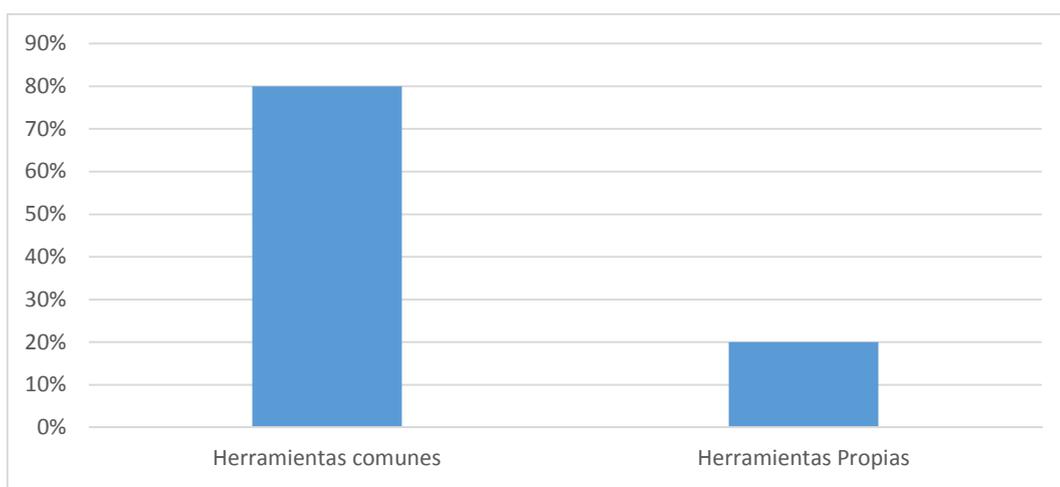


Elaboración: Álvaro Vizueta
Fuente: Entrevista

2.- ¿Tiene alguna herramienta de apoyo que le ayude a realizar las labores de revisión de los tramites generados?

Del total de entrevistados la mayoría apunta a que utiliza como herramienta de apoyo programas de escritorio comunes, ninguna que sea específica y diseñada para su labor diaria, esto complementado con procesos manuales que se consideran obsoletos.

Gráfico 17.- Pregunta 2

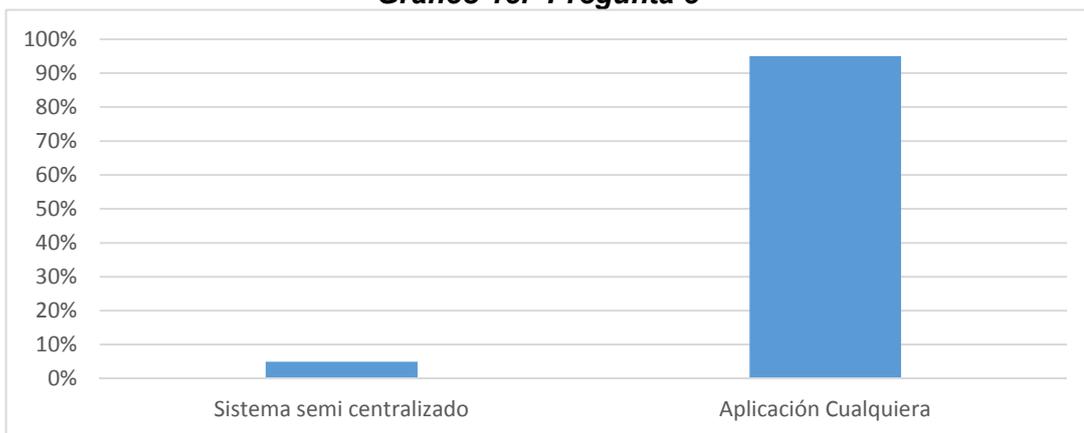


Elaboración: Álvaro Vizueta
Fuente: Entrevista

3.- ¿En la actualidad, cuentan con un sistema integral, y centralizado que le permita llevar un control de los trámites que se generan?

Todos respondieron que un sistema integral que trabaje específicamente para los trámites no poseen, en su defecto se ayudan de cierta manera con aplicaciones sencillas para llevar de alguna manera cierto control, pero todo esto no es centralizado y visible para todos los que trabajan en estos trámites.

Gráfico 18.- Pregunta 3



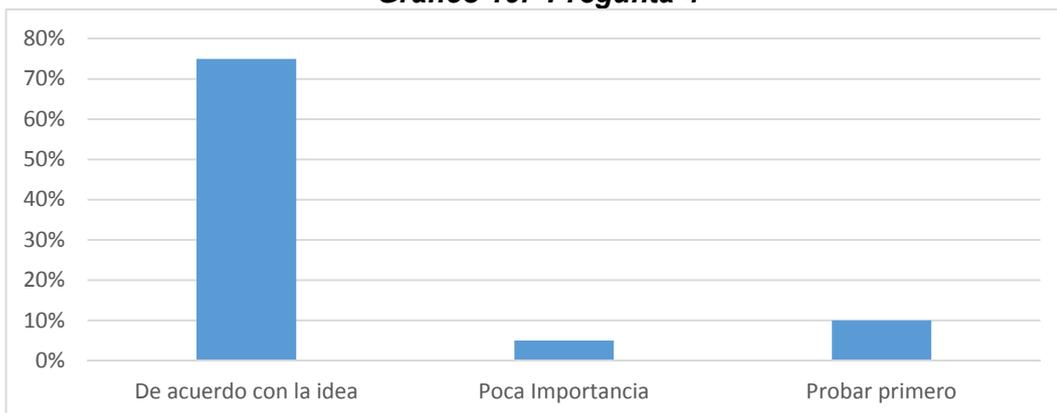
Elaboración: Álvaro Vizueta

Fuente: Entrevista

4.- ¿Qué tan importante consideraría la posibilidad de poseer un sistema que le permita registrar y almacenar sus actividades y que le permita llevar un control del trabajo realizado?

Los entrevistados piensan que el planteamiento de un sistema centralizado se considera muy importante, ya que mediante este sistema, de cumplir lo que su labor necesita, ayudaría a llevar un registro ordenado de los tramites que se realizan, y agilizaría el flujo de trabajo que cada uno tiene, y colateralmente también beneficiaría a los alumnos.

Gráfico 19.- Pregunta 4



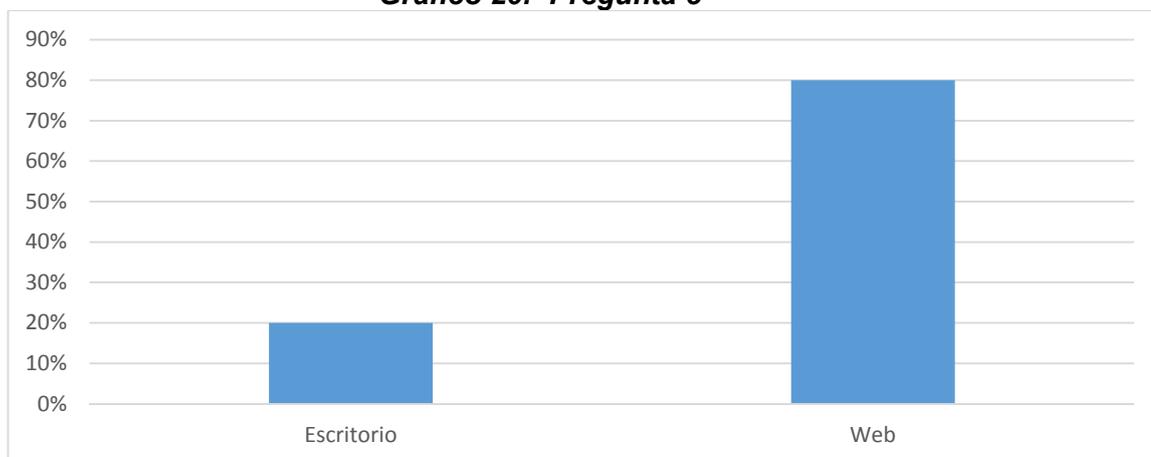
Elaboración: Álvaro Vizueta

Fuente: Entrevista

5.- ¿De ser el caso de poseer un sistema centralizado, de que tipo preferiría que sea este: desde una aplicación de escritorio o en un ambiente web, brevemente indique el porqué de su respuesta?

El 80% de los entrevistados, preferiría que de darse el caso este sistema nuevo sea desarrollado en ambiente web, ya que a su criterio esto les ayudaría a acceder desde fuera de la Universidad de darse el caso para poder culminar, avanzar o ayudar con su gestión a un alumno que así lo requiera de manera urgente.

Gráfico 20.- Pregunta 5

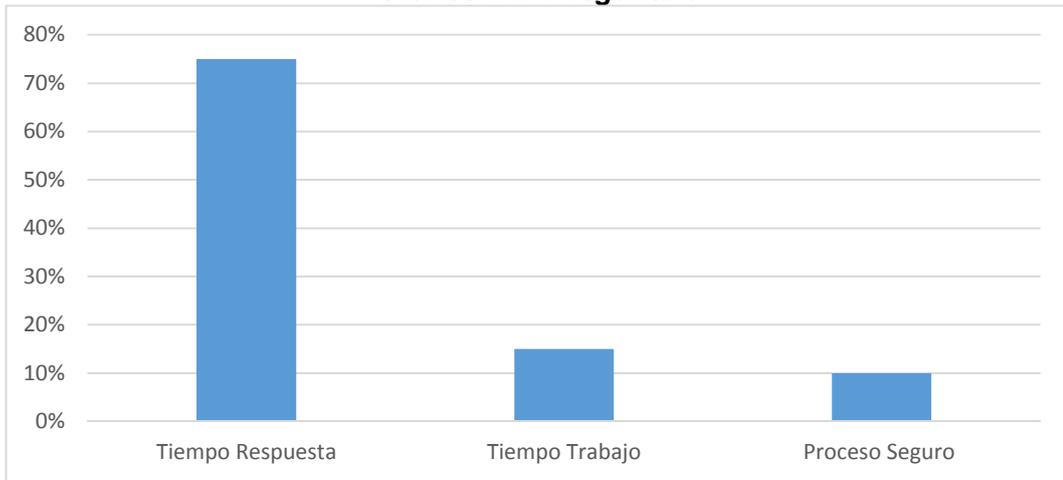


*Elaboración: Álvaro Vizueta
Fuente: Entrevista*

6.- ¿Qué beneficios cree obtendría Ud. si se contara con un sistema centralizado para la revisión de los trámites?

La mayor respuesta que se tuvo en base a esta respuesta fue que el beneficio principal que se obtendría es el de mejorar los tiempos de respuesta, ya que con este sistema se propone eliminar procesos manuales por procesos automáticos.

Gráfico 21.- Pregunta 6

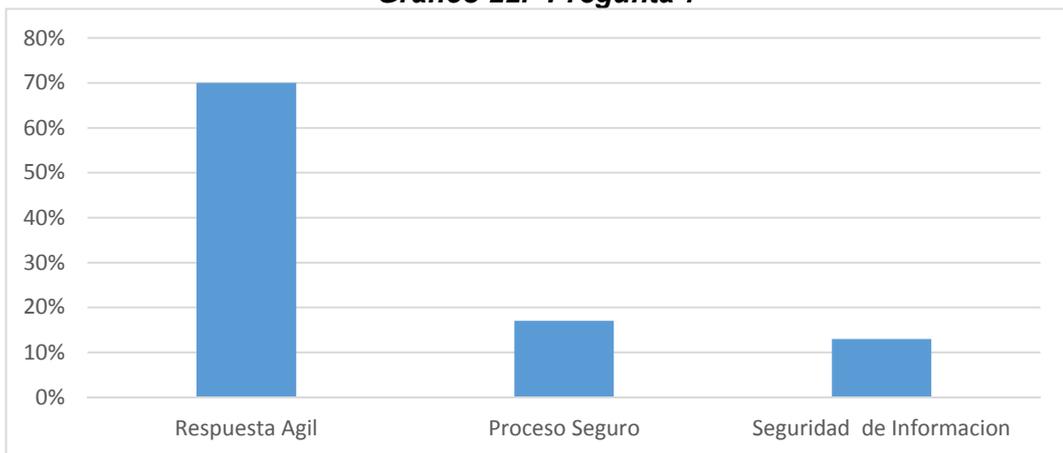


Elaboración: Álvaro Vizueta
Fuente: Entrevista

7.- ¿Qué beneficios cree que tendrían los alumnos al contar con un sistema diseñado para la revisión de los trámites?

De la entrevista se describe que lo más beneficioso sería el que el estudiante podría realizar de manera más rápida sus trámites, de forma segura y con la fiabilidad de que no habrá pérdida de información en el proceso, y que todo lo que se realice quede registrado en el sistema.

Gráfico 22.- Pregunta 7

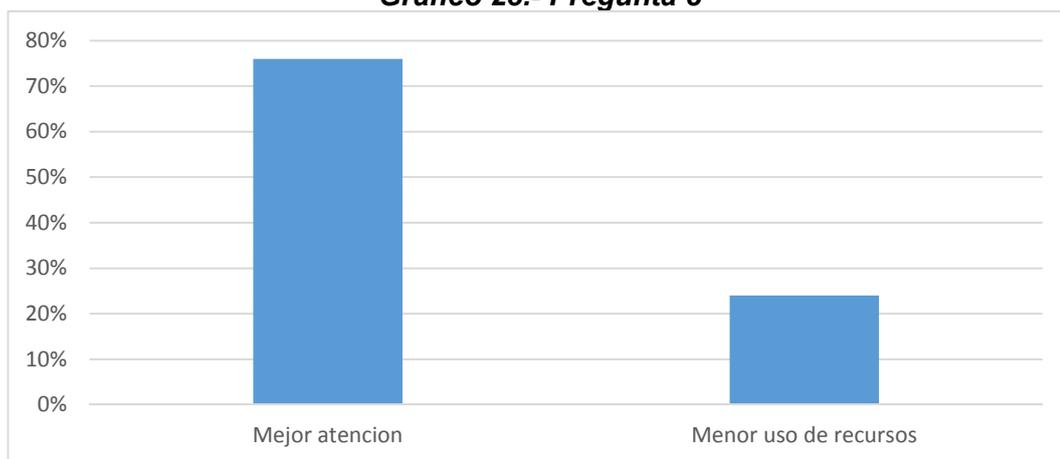


Elaboración: Álvaro Vizueta
Fuente: Entrevista

8.- ¿Qué beneficios obtendría la Universidad en base a la idea planteada de un sistema centralizado para la revisión de trámites?

Dentro del contenido de la entrevista tenemos que según el criterio de los entrevistados el beneficio para la Universidad está en que sus empleados al atender las peticiones de manera más rápida, esto se reflejarían en una mejor atención hacia los alumnos.

Gráfico 23.- Pregunta 8

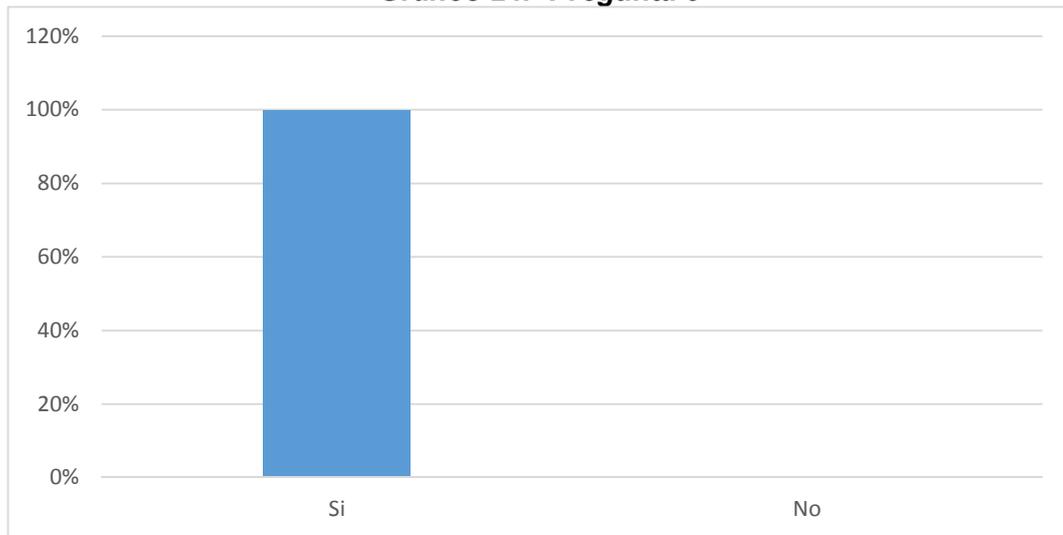


Elaboración: Álvaro Vizueta
Fuente: Entrevista

9.- ¿Cree Ud Que actualmente se necesita de un sistema nuevo que ayude a realizar la revisión de trámites?

Según esta pregunta la un porcentaje alto indica que si es necesario contar con un sistema dedicado a la revisión de los tramites generados ya que actualmente no hay sistema de este tipo, algo que vaya acorde con las nuevas tendencias informáticas.

Gráfico 24.- Pregunta 9

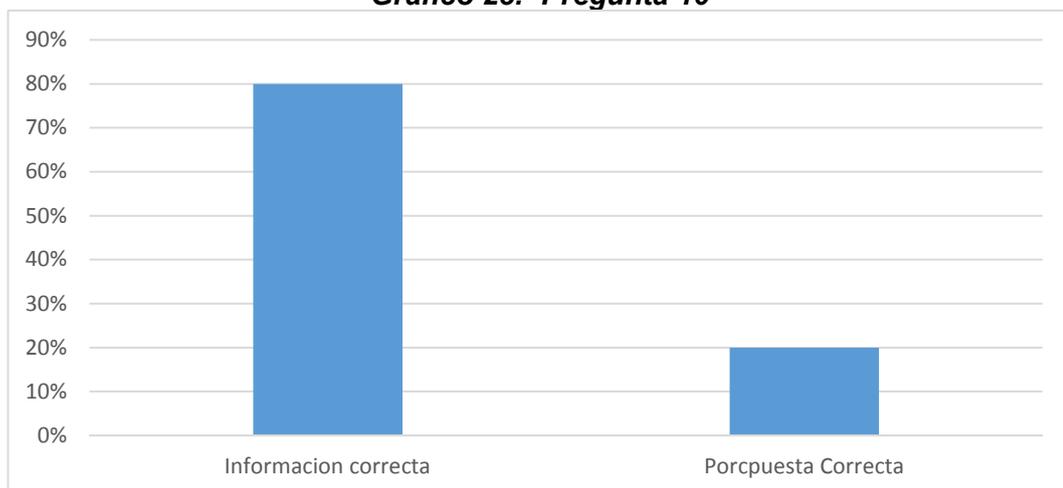


Elaboración: Álvaro Vizuela
Fuente: Entrevista

10.- ¿Qué recomendaciones o sugerencia daría para el desarrollo de un prototipo que cumpla los requerimientos para la revisión y posterior notificación de los trámites?

Según los resultados las recomendaciones que se dan son en base al levantamiento correcto de la información, ya que de ello depende que lo que se vaya a realizar sea correcto y orientado a lo que la Universidad requiere.

Gráfico 25.- Pregunta 10



Elaboración: Álvaro Vizuela
Fuente: Entrevista

ANÁLISIS FODA

En esta parte estableceremos las fortalezas, oportunidades, debilidades y amenazas, de tal forma que podamos determinar las estrategias que nos ayuden a aprovechar las oportunidades a partir de las fortalezas, minimizar debilidades y evitar las amenazas.

Fortalezas

1. Propuesta de desarrollo hecha en ambiente web.
2. La idea de un cambio tecnológico en pro de mejorar el servicio al estudiante.
3. Visión a futuro por parte de los Administradores de la Universidad.
4. La necesidad de un cambio que permita agilizar los procesos.
5. Propuesta que no tiene costo para la Universidad.
6. Funcionalidad y requerimientos establecidos para la Universidad

Oportunidades

1. No existe proyecto o propuesta igual actualmente.
2. Poder adaptar el mismo sistema a otras Facultades.
3. Otras entidades podrían interesarse en el funcionamiento de este sistema.
4. Mejorar la perspectiva de la Universidad hacia entes externos
5. Poder promocionar con este desarrollo lo que se imparte dentro de la institución.

Debilidades

1. Cambios inesperados en los procesos.
2. Cambio de requerimientos fuera de tiempo.
3. Falta de comunicación con las personas encargadas de los procesos.
4. Falta de tiempo para desarrollo por cuestiones laborales.
5. Reuniones escasas por falta de tiempo del docente.
6. Desarrollo no pudiera estar a tiempo para presentación.
7. Resistencia de alguna persona al cambio.

Amenazas

1. Que algún director externo a la Facultad exprese que no es necesario un planteamiento de este tipo.
2. Alguien externo se interese en desarrollar algo similar y crear un clima de competencia.
3. Jefes de lugares de trabajo no estén de acuerdo en dar tiempo para que se desarrolle este proyecto.
4. Mayor carga laboral en el lugar de trabajo.
5. Personas consultadas externamente opinen que el proyecto no es lo suficientemente bueno para ser evaluado.

Tabla 1. CUADRO DE ANALISIS FODA

FACTORES INTERNOS	
FORTALEZAS (F)	DEBILIDADES(D)
<ol style="list-style-type: none"> 1. Propuesta de desarrollo hecha en ambiente web. 2. La idea de un cambio tecnológico en pro de mejorar el servicio al estudiante. 3. Visión a futuro por parte de los Administradores de la Universidad. 4. La necesidad de un cambio que permita agilizar los procesos. 5. Propuesta que no tiene costo para la Universidad. 6. Funcionalidad y requerimientos establecidos para la Universidad 	<ol style="list-style-type: none"> 1. Cambios inesperados en los procesos. 2. Cambio de requerimientos fuera de tiempo. 3. Falta de comunicación con las personas encargadas de los procesos. 4. Falta de tiempo para desarrollo por cuestiones laborales. 5. Reuniones escasas por falta de tiempo del docente. 6. Desarrollo no pudiera estar a tiempo para presentación. 7. Resistencia de alguna persona al cambio.
FACTORES EXTERNOS	
OPORTUNIDADES (O)	AMENAZAS (A)
<ol style="list-style-type: none"> 1. No existe proyecto o propuesta igual actualmente. 2. Poder adaptar el mismo sistema a otras Facultades. 3. Otras entidades podrían interesarse en el funcionamiento de este sistema. 4. Mejorar la perspectiva de la Universidad hacia entes externos 5. Poder promocionar con este desarrollo lo que se imparte dentro de la institución. 	<ol style="list-style-type: none"> 1. Que algún director externo a la Facultad exprese que no es necesario un planteamiento de este tipo. 2. Alguien externo se interese en desarrollar algo similar y crear un clima de competencia. 3. Jefes de lugares de trabajo no estén de acuerdo en dar tiempo para que se desarrolle este proyecto. 4. Mayor carga laboral en el lugar de trabajo. 5. Personas consultadas externamente opinen que el proyecto no es lo suficientemente bueno para ser evaluado.

Elaboración: Álvaro Vizuela

Fuente: Álvaro Vizuela

Después de haber identificado las fortalezas, debilidades, oportunidades y amenazas resultantes del análisis efectuado al proyecto planteado, se sugiere se realicen acciones que permitan minimizar las debilidades y amenazas que presenta el proyecto.

Acciones recomendadas para el desarrollo del proyecto.

E1: Elaboración de cronogramas y planes de reuniones con firmas de compromiso.

E2: Documentos de apoyo para los estudiantes que puedan presentar en sus respectivos lugares de trabajo, para así comprometer al empleador a brindar el apoyo necesario.

E3: Comprometer y concientizar al personal administrativo que el proyecto que se plantea también va en pro de mejorar su desarrollo de labores diarias

E4: Establecer requerimientos claros y definitivos que no requieran cambio en un momento poco propicio.

E5: Concientizar que se debe valorar más el trabajo del estudiante que el de una persona externa a la Universidad.

E6: Tener un grupo de Docentes que apoye y despeje dudas a lo largo del desarrollo de las actividades que engloban este proyecto.

E7: Evaluación periódica del producto que se está desarrollando.

Después de enlistar las diferentes acciones posibles a tomar y que nos ayudarían a minimizar las debilidades y amenazas de esta propuesta, se hará un análisis sistemático mediante matriz FODA que nos ayude a detectar las relaciones entre las variables más importantes.

Tabla 2.- Análisis de matriz FODA

ANÁLISIS DE MATRIZ FODA		
FACTORES INTERNOS	FORTALEZAS (F)	DEBILIDADES (D)
FACTORES EXTERNOS	ESTRATEGIAS FO: MAXI-MAXI	ESTRATEGIAS DO: MINI-MAXI
OPORTUNIDADES	F1 aprovechar O1 y O2 F2 aprovechar O3 y O4 F3 aprovechar O5 y O3 F4 aprovechar O1 y O3 F5 aprovechar O1 F6 aprovechar O1 y O2	E1 para superar D1,D2,D6 y aprovechar O2 E2 para superar D4 y aprovechar O4 E3 para superar D5,D7 y aprovechar O1 E4 para superar D1,D2,D6 y aprovechar O5 y O2 E5 para superar D6,D7 y aprovechar O4 y O5 E6 para superar D6 y aprovechar O2, O1 E7 para superar D3,D6 y aprovechar O4,O1
	ESTRATEGIAS FA: MAXI-MINI	ESTRATEGIAS DA: MINI-MINI
AMENAZAS	Optimizar F1 para reducir A1 Optimizar F2 para reducir A1,A5 Optimizar F3 para reducir A2,A3 Optimizar F4 para reducir A1, A5 Optimizar F5 para reducir A5,A1,A2 Optimizar F6 para reducir A1,A5,A2	E1 reducir D1 como A1,A5 E2 reducir D4,D6 como A3,A4 E3 reducir D1,D2 como A1,A5 E4 reducir D1,D2,D6 como A1 E5 reducir D2,D6 como A1,A5 E6 reducir D3,D5 como A2 E7 reducir D1,D2,D6 como A1,A5

Elaboración: Álvaro Vizuela

Fuente: Álvaro Vizuela

De acuerdo a cada cuadrante se establece la interpretación de una estrategia alternativa de cada tipo, de acuerdo con el cuadrante.

Estrategia FO: El hecho de ser una propuesta nueva que anteriormente no ha sido planteado de una gran expectativa de aceptación por parte de los que están directamente involucrados con su manejo.

Estrategia FA: Optimizar el tiempo y las reuniones que se tiene con el personal al cual va dirigido esta investigación y aumentar el interés y las expectativas de este desarrollo

Estrategia DO: Plantear planes y soportes que permitan tener el tiempo suficiente para desarrollar la propuesta y que esta esté apegada a las exigencias de la Universidad

Estrategia DA: Elaboración de planes estratégicos que permitan reducir las amenazas tanto dentro de la Universidad como en los lugares de Trabajo, para así lograr el objetivo principal de este desarrollo.

CRITERIO PARA LA ELABORACION DE LA PROPUESTA

La propuesta como tal nos ha brindado la posibilidad de generar varios criterios que nos dan la oportunidad de conseguir el objetivo principal que es el desarrollo del trabajo investigativo expuesto. Que aparte de brindar datos importantes nos dan las bases para respaldar un trabajo viable y fiable, todo por medio de pasos específicos y determinados para la culminación de este contexto.

Como cualquier proyecto informático, se debe tener en cuenta aspectos muy importantes como son los requerimientos necesarios para el desarrollo de la solución en la plataforma apropiada, seleccionando correctamente el hardware y software necesario, así como los diseños de las interfaces que van a ser utilizados por los usuarios así como los diseños de las bases de datos para cada uno de los módulos.

El desarrollo de esta solución contempla la necesidad de contar con una herramienta de ambiente web, ya determinada por los que en un inicio diseñaron la arquitectura que actualmente se utiliza, que permita obtener beneficios tanto principalmente a docentes y de manera simultánea a estudiantes que forman parte de los procesos que aun en estos días siguen siendo manuales. Resaltando el objetivo principal de este proyecto que es la revisión unificada de Certificados y Solicitudes.

CRITERIOS DE VALIDEZ DE LA PROPUESTA

Este proyecto una vez culminado en su totalidad debe ser verificado y validado en su correcto funcionamiento. Todo esto mediante los distintos escenarios de pruebas que se puedan definir para cada uno de los objetivos planteados. Y dependiendo de los resultados de las pruebas se determinara la validez de la propuesta planteada.

OPERACIONALIZACIÓN DE VARIABLES

Tabla 3. Operacionalización de las Variables

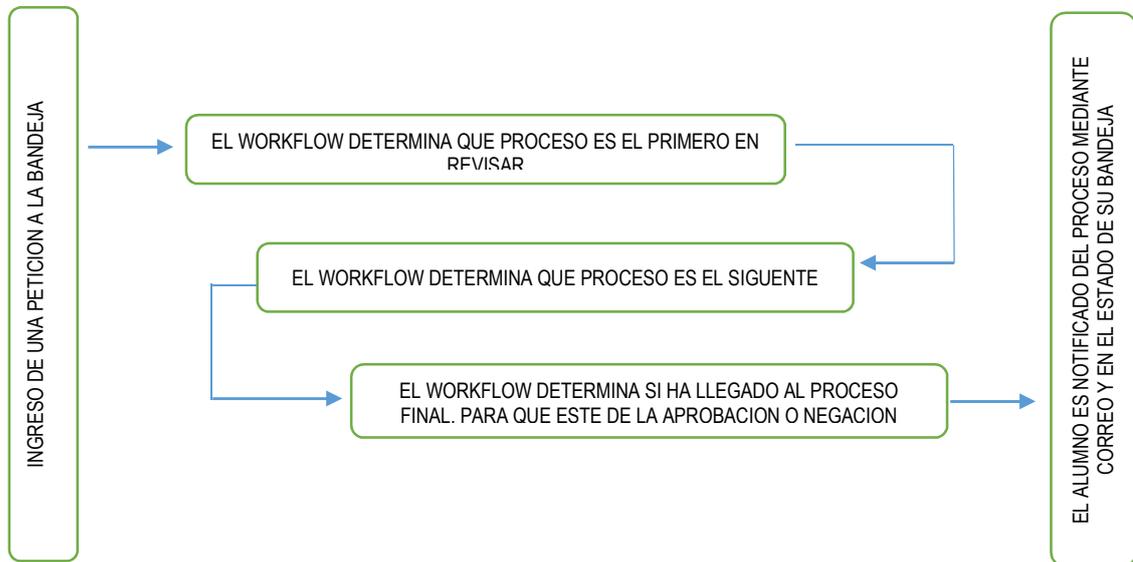
Variables	Dimensiones	Indicadores	Técnicas y/o Instrumentos
V.I. Proceso de desarrollo de la base de datos para el prototipo de la Bandeja de Trabajo	Ambiente de instalación	100%.Implantación del ambiente para la instalación	Información sobre la estructura y estándar de la base de datos.
	Desarrollo de la base de datos que funcionara con el resto del sistema	100%. Estructura de base de datos	Estándares del resto del sistema de base de datos
V.D. Proceso de Desarrollo del Prototipo	Ambiente de desarrollo	100%.Implantación del ambiente a ser utilizado para el desarrollo.	Información sobre la estructura de desarrollo del sistema
	Proceso de desarrollo de software prototipo	100%Desarrollo de software prototipo	Estándares de desarrollo ya estipulados y encuesta

Elaboración: Álvaro Vizuela

Fuente: Álvaro Vizuela

MAPA DE PROCESO DE LA BANDEJA DE TRABAJO

Gráfico 26.- Mapa de proceso Bandeja de Trabajo



Elaboración: Álvaro Vizueta

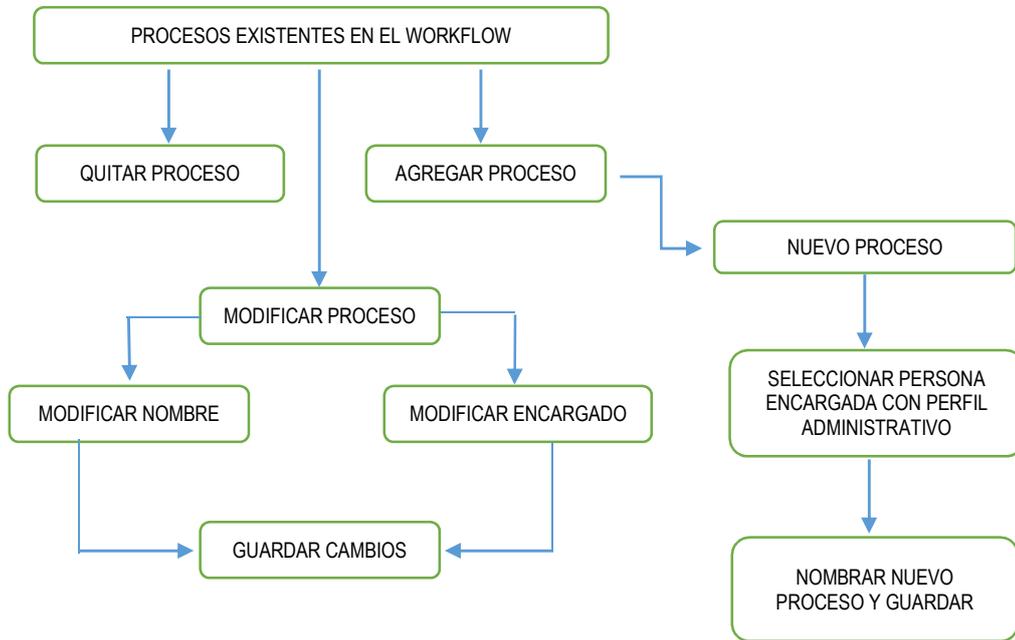
Fuente: Álvaro Vizueta

Proceso que explica cómo se lleva a cabo la revisión en la bandeja de trabajo, que según el usuario que se conecte al sistema se le mostrara el proceso indicado a ser tomado en cuenta por el para su revisión.

La bandeja de trabajo en conjunto con el Workflow, determinan el flujo del trámite según lo que este especificado en la base de datos, para así determinar el proceso siguiente y de esta manera automatizar el proceso de cada uno de los tramites.

MAPA DE PROCESO DEL WORKFLOW

Gráfico 27.- Mapa de proceso WorkFlow



Elaboración: **Álvaro Vizuela**
Fuente: **Álvaro Vizuela**

Este mapa de proceso explica cómo se lleva a cabo la revisión de los trámites mediante el WorkFlow, desde la parte de ingreso del mismo desde el ambiente web, pasando por el proceso de revisión hasta la parte final de la notificación en donde se indica al estudiante la respuesta a su solicitud.

Todo este proceso es registrado en la base de datos, lo que permite pueda ser notificado según su conveniencia.

CAPÍTULO IV

MARCO ADMINISTRATIVO

DIAGRAMA DE GANTT

Gráfico 28.- CRONOGRAMA DE ACTIVIDADES

Id		Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1		Duracion del Proyecto	110 días?	lun 15/12/14	vie 15/05/15	
2		Reuniones de Trabajo para	1 día?	lun 15/12/14	lun 15/12/14	
3		Asignacion de Alumnos pa	1 día?	mar 16/12/14	mar 16/12/14	2
4		Explicación de Todos los n	1 día?	lun 22/12/14	lun 22/12/14	3
5		Distribucion de temas para	1 día?	lun 05/01/15	lun 05/01/15	4
6		Explicación de Todos los n	1 día?	vie 16/01/15	vie 16/01/15	5
7		Levantamiento de informac	1 día?	vie 30/01/15	vie 30/01/15	6
8		Levantamiento de informac	1 día?	lun 02/02/15	lun 02/02/15	7
9		Entrega de Información de	1 día?	mar 03/02/15	mar 03/02/15	8
10		Capacitación sobre el man	1 día?	mié 04/02/15	mié 04/02/15	9
11		Entrega de Bases de Datos	1 día?	jue 05/02/15	jue 05/02/15	10
12		Entrega de Bases de Datos	1 día?	vie 06/02/15	vie 06/02/15	11
13		Asignación de Docentes pa	1 día?	jue 12/02/15	jue 12/02/15	12
14		Primera Versió	1 día?	vie 20/02/15	vie 20/02/15	13
15		Revision 1	1 día?	vie 27/02/15	vie 27/02/15	14
16		Segunda Versio	1 día?	vie 06/03/15	vie 06/03/15	15
17		Revision 2	1 día?	vie 13/03/15	vie 13/03/15	16
18		Unificación de Solicitudes y	1 día?	lun 16/03/15	lun 16/03/15	17
19		Unificacion para la revisión	1 día?	lun 13/04/15	lun 13/04/15	18
20		Designación de Tareas al r	1 día?	mar 14/04/15	mar 14/04/15	19
21		Tercera Versió	1 día?	mié 15/04/15	mié 15/04/15	20
22		Revisión 3	1 día?	jue 16/04/15	jue 16/04/15	21
23		Analisis de observaciones	1 día?	vie 17/04/15	vie 17/04/15	22
24		Integración Primera Prueba	1 día?	lun 20/04/15	lun 20/04/15	23
25		Integración Segunda Prueba	1 día?	mar 21/04/15	mar 21/04/15	24
26		Asignación de Tutores y Re	1 día?	mié 22/04/15	mié 22/04/15	25
27		Version Final, Revision Fin	1 día?	jue 23/04/15	jue 23/04/15	26
28		Revisiones y entrega de do	1 día?	jue 30/04/15	jue 30/04/15	27
29		Revisiones de documento	1 día?	vie 08/05/15	vie 08/05/15	28
30		Versión para Preparar am	1 día?	vie 15/05/15	vie 15/05/15	29

Elaboración: Álvaro Vizueta
Fuente: ING. BERNARDO IÑIGUEZ

PRESUPUESTO

Esta propuesta esta daba bajo financiamiento propio.

Tabla 4. Ingresos

INGRESOS	
Financiamiento propio	\$1035,00
TOTAL	

Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

DETALLE DE EGRESOS DEL PROYECTO

Todos los ingresos son propios, no hay fuentes externas de financiamiento para lo que se está desarrollando.

Tabla 5. Detalle Egresos

EGRESOS	
Suministros de oficina	\$25,00
Fotocopias	\$5,00
Servicios de Internet	\$30,00
Transporte	\$25,00
Refrigerio	\$65,00
Computador Dell	\$800.00
Empastado, anillado de tesis de grado	\$80.00
TOTAL	\$1035,00

Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

Realizado con éxito el desarrollo del prototipo propuesto ya que:

- Se plantea un prototipo para el manejo de las solicitudes y certificados dentro de una misma plataforma, ayudando a mejorar la calidad y tiempo de servicio.
- Se deja planteada la idea de una herramienta para eliminar en su gran mayoría los procesos manuales.
- Todo esto está respaldado bajo el diseño de una base de datos que ayudara a evitar la duplicidad de datos.
- Se desarrollo un medio de notificaciones via mail que servirá para indicar las novedades de la solicitud o certificado generado.
- Se facilita la incorporación de nuevo certificados en la herramienta que se ha desarrollado.

Este prototipo está desarrollado bajo plataforma Java en conjunto con la herramienta HIBERNATE para la persistencia de Datos. Se cuenta con Apache TOMCAT como servidor WEB y servidor WS. El servidor de Base de Datos central es SQL SERVER 2012. Framework Zk, potente herramienta para el desarrollo de la parte visual que incluye CSS3, JSON utilizado para encriptar los datos de forma segura, todo esto bajo la estructura de desarrollo Modelo Vista Controlador o MVC.

El desarrollo de este prototipo está basado en el análisis que se obtuvo en el levantamiento de procesos que rigen actualmente a los certificados y solicitudes generados con la iniciativa de mejorar los tiempos de respuesta, y que se pueda visualizar vía web el estado de los certificados y solicitudes. Generando un documento digital o una respuesta por correo.

RECOMENDACIONES

- Todo aquel desarrollo que implique levantamiento de información en pro de mejorar un servicio, deber ser realizado a fondo y acorde con lo que se maneja en la actualidad.
- El desarrollo de este y de otros prototipos debe ser acorde a las especificaciones básicas de manejo, así como la estructura de la base de dato que se ha diseñado para este fin
- Toda la información obtenida de cualquier medio, deber ser considerada importante, ya que de todo lo que se genere dentro del normal desenvolvimiento es una pauta de desarrollo y de futuras mejoras.

REFERENCIAS BIBLIOGRÁFICAS

ehu.eus. (2015). ehu.eus Obtenido de ehu.eus:

<http://www.ehu.eus/mrodriguez/archivos/csharp/pdf/ServiciosWeb/WebServices.pdf>

ug.edu.ec (2015). ug.edu.ec Obtenido de ug.edu.ec:

<http://www.ug.edu.ec/nosotros/>

json.org (2014). json.org Obtenido de json.org:

<http://www.json.org>

es.slideshare.net (2014) es.slideshare.net Obtenido de es.slideshare.net:

<http://es.slideshare.net/EdurekaIN/webinar-hibernate>

trazos-web.com (2014) trazos-web.com Obtenido de trazos-web.com:

<http://www.trazos-web.com/2010/02/01/html5-que-es-y-como-usarlo/>

albertchurch.com (2014) albertchurch.com Obtenido de albertchurch.com:

<http://www.albertchurch.com/blog/haz-tu-diseno-responsive-con-bootstrap/>

projects.eclipse.org (2014) projects.eclipse.org Obtenido de projects.eclipse.org:
<https://projects.eclipse.org/releases/luna>

tiposdeinvestigacion.com (2014): tiposdeinvestigacion.com Obtenido de
tiposdeinvestigacion.com: <http://www.tiposdeinvestigacion.com>

ANEXOS

Universidad de Guayaquil
Facultada de Ciencias Matemáticas y Físicas
Carrera de Ingeniería en Sistemas Computacionales

ENTREVISTA

“Diseño y desarrollo de la bandeja de trabajo que procesara solicitudes y certificados para el prototipo de Sistema Académico de la Universidad de Guayaquil.”.

Tramites: concepto que engloba a solicitudes y certificados.

1.- ¿Cómo ve actualmente la manera de revisar los tramites?

2.- ¿Tiene alguna herramienta de apoyo que le ayude a realizar las labores de revisión de los tramites generados?

3.- ¿En la actualidad, cuentan con un sistema integral, y centralizado que le permita llevar un control de los trámites que se generan?

4.- ¿Qué tan importante consideraría la posibilidad de poseer un sistema que le permita registrar y almacenar sus actividades y que le permita llevar un control del trabajo realizado y por qué?

5.- ¿De ser el caso de poseer un sistema centralizado, de que tipo preferiría que sea este: desde una aplicación de escritorio o en un ambiente web, brevemente indique el porqué de su respuesta?

6.- ¿Qué beneficios cree obtendría Ud. si se contara con un sistema centralizado para la revisión de los trámites?

7.- ¿Qué beneficios cree que tendrían los alumnos al contar con un sistema diseñado para la revisión de los trámites?

8.- ¿Qué beneficios obtendría la universidad en base a la idea planteada de un sistema centralizado para la revisión de trámites?

9.- ¿Cree Ud Que actualmente se necesita de un sistema nuevo que ayude a realizar la revisión de trámites, explique el porqué de su respuesta?

10.- ¿Qué recomendaciones o sugerencia daría para el desarrollo de un prototipo que cumpla los requerimientos para la revisión y posterior notificación de los trámites?

Clases Java de “Bandeja de Trabajo”

BandejaTrabajo.java

```
package ug.sistema.academico.controladores.tramites;

import java.io.File;
import java.io.FileInputStream;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
import java.util.Random;

import net.sf.jasperreports.engine.JRException;
import net.sf.jasperreports.engine.JasperCompileManager;
import net.sf.jasperreports.engine.JasperExportManager;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;

import org.zkoss.zk.ui.Component;
import org.zkoss.zk.ui.Executions;
import org.zkoss.zk.ui.Sessions;
import org.zkoss.zk.ui.select.annotation.Wire;
import org.zkoss.zk.ui.util.GenericForwardComposer;
import org.zkoss.zul.Button;
import org.zkoss.zul.Caption;
import org.zkoss.zul.Combobox;
import org.zkoss.zul.Div;
//import org.zkoss.zul.Filedownload;
import org.zkoss.zul.Label;
import org.zkoss.zul.ListBox;
import org.zkoss.zul.Textbox;
import org.zkoss.zul.Window;

import ug.sistema.academico.service.ruteador.IRuteadorSRV;
import ug.sistema.academico.service.ruteador.RuteadorSOAP;
import ug.sistema.clases.CertificadosHeaderView;
import ug.sistema.clases.ToastMessage;
import ug.sistema.clases.TransportePerfilesAsignados;
import ug.sistema.libs.Utills;

import com.google.gson.Gson;

import ec.edu.ug.academico.type.MensajesGenericos;

public class BandejaTrabajo extends GenericForwardComposer<Component> {

    private static final long serialVersionUID = 1L;

    @Wire
    private Caption objLblTitle;
    @Wire
    Utills objUtills, objUtillsConstants;
    @Wire
    private Combobox cmbPaging;
    @Wire
    private Textbox bndBuscar;
    @Wire
    private ListBox cmbPolíticas, lstRegistros;
    @Wire
    private Window objWinModalAssignment, winAdminTramites;
    @Wire
    private ListBox cmbEstado;
```

```

@Wire
private TextBox txtClave, txtUsuario, txtCedula, txtIdUsuario, txtEstado, txtIdSolicitud,
txtActionOpcion, txtCarrera,
txtEmail, txtTramite, txtFecha, txtObservaciones, txtTipoTramite;
@Wire
private Button btnAprobar, btnCancelar, btnNew, btnOK;
@Wire
private Label lblAlert, lblDetalles, lblCodeCarrera;
@Wire
IRuteadorSRV portTramites, portSeguridad;
ArrayList<Object> objArrayTransporteParametros;
Div divRegistros, divDetalleTramite;
String vStrSeparador;
String vStrTraza;
MensajesGenericos objMensGenericos;
String vStrAccesoDirecto;
String vStrNombreCarrera;
@Wire
Label objLabelAction;
boolean vBoolFlag;
TransportePerfilesAsignados objTransportePerfilAsig;

String nexProceso;

public BandejaTrabajo() {
    boolean valid = Sessions.getCurrent().getAttribute(
        "objGlobalInformation") != null ? true : false;
    if (!valid) {
        Executions.sendRedirect("SegLoginForm.zul");
    }
}

public void doAfterCompose(Component objComp) {
    try {
        objUtils = new Utils();
        objUtilsConstants = new Utils();
        objUtils.setProperties("/resources/mensajes.properties");
        objUtilsConstants.setProperties("/resources/constants.properties");
        vBoolFlag = false;
        portSeguridad = RuteadorSOAP.getIRuteadorSOAP("seguridad");
        portTramites = RuteadorSOAP.getIRuteadorSOAP("tramites");
        objTransportePerfilAsig = new TransportePerfilesAsignados();

        if ((String) Executions.getCurrent().getAttribute("mainPage") != null)
        {
            super.doAfterCompose(objComp);

            vStrAccesoDirecto = (String) Executions.getCurrent()
                .getAttribute("vStrAccesoDirecto") == null ? ""
                : (String) Executions.getCurrent().getAttribute(
                    "vStrAccesoDirecto");

            objLblTitle.setLabel("Detalle de Tramite");
            vStrSeparador = objUtilsConstants.getObjProperties()
                .getProperty("fStrSeparadorParametros");

            this.lblAlert.setValue(objUtils.getObjProperties().getProperty(
                "fStrMessageClick"));
            this.cargaRegistros(0, "");
        } else {
            Sessions.getCurrent().removeAttribute("objGlobalInformation");
            Executions.sendRedirect("../SegLoginForm.zul");
        }
    } catch (Exception e) {
        System.out.println(e.toString());
        e.printStackTrace();
    }
}

```

```

    }
}

public void buscarRegistros() throws Exception {
    if (!bndBuscar.getValue().trim().equals("")) {
        String vStrBuscarCedula = objUtils.checkNumber(bndBuscar.getValue()
            .trim(), 10);
        String vStrBuscarNombre = objUtils.checkText(bndBuscar.getValue()
            .trim());

        if (vStrBuscarCedula.equals("") || vStrBuscarNombre.equals("")) {
            this.cargaRegistros(0, bndBuscar.getValue().trim());
        } else {
            if (!vStrBuscarCedula.equals("")) {
                ToastMessage.showNotify(
                    vStrBuscarCedula,
                    "alertify."
                );
            } else {
                if (!vStrBuscarNombre.equals("")) {
                    ToastMessage.showNotify(vStrBuscarNombre,
                        "alertify."
                    );
                }
            }
        }
    }
}

objUtils.getObjProperties().getProperty(
    "fStrERROR");
} else {
    if (!vStrBuscarNombre.equals("")) {
        ToastMessage.showNotify(vStrBuscarNombre,
            "alertify."
        );
    }
}
objUtils.getObjProperties()
    .getProperty("fStrERROR");
} else {
    this.cargaRegistros(0, "");
}
}

private void cargaRegistros(int vIntPage, String vStrParametro) throws Exception {
    String vStrUser = objUtilsConstants.getInformacionSession().getvStrGlobalUserID();

    objArrayTransporteParametros = new ArrayList<>();
    objArrayTransporteParametros.add("HQL_TRAMITES_GET_USUARIOS");
    objArrayTransporteParametros.add(vStrUser);
    objArrayTransporteParametros.add(true);
    objArrayTransporteParametros.add("0");

    objArrayTransporteParametros.add(objUtilsConstants.getObjProperties().getProperty("fStrMaxRegist
ers"));
    objArrayTransporteParametros.add(String.valueOf(vIntPage));

    objArrayTransporteParametros.add(objUtilsConstants.getInformacionSession().getvStrGlobalUserID()
);

    MensajesGenericos objMensGenericos = new
Gson().fromJson(portTramites.getGsonFromRoute("tramites", "ug.sistema.academico.bo.tramites.BandejaTraba
joBO",
        "getListRegistros", new
Gson().toJson(objArrayTransporteParametros),
objUtilsConstants.getInformacionSession().getvStrGlobalUserID(), MensajesGenericos.class);

    cmbPaging.getChildren().clear();
    if (objMensGenericos.getvStrCode().equals(
        objUtils.getObjProperties().getProperty("fStrSUCCESS"))) {
        lstRegistros.getChildren().clear();
        lstRegistros.appendChild(CertificadosHeaderView.getHeaderCertificados());
    }
}

```

```

        if (objMensGenericos.getObjContenido().size() >= 1) {
            int vIntCabecera = 0;
            for (ArrayList<String> objRegister :
objMensGenericos.getObjContenido()) {
                if (vIntCabecera == 0) {
                    for (int vIntIndice = 0; vIntIndice < Integer
.parseInt(objRegister.get(3));
vIntIndice++) {
                        cmbPaging.appendItem(String.valueOf(vIntIndice));
                    } else {
                        lstRegistros.appendChild(objUtilsConstants
.getListItem(objRegister));
                    }
                    vIntCabecera++;
                }
                cmbPaging.setSelectedIndex(vIntPage);
            }
        } else {
            ToastMessage.showNotify(objMensGenericos.getvStrMessage(),
"alertify." + objMensGenericos.getvStrCode());
        }
    }

    public void presentarDatosPaging() throws Exception {
        this.cargaRegistros(Integer.parseInt(cmbPaging.getValue()), "");
    }

    public void cargaDetalles() throws Exception{
        System.out.println(String.valueOf(lstRegistros.getSelectedItem().getValue()));
        divRegistros.setVisible(false);
        divDetalleTramite.setVisible(true);
        boolean proceso =
verificarProceso(String.valueOf(lstRegistros.getSelectedItem().getValue()));
        if(!proceso)
        {
            btnAprobar.setVisible(true);
            btnOK.setVisible(false);
        }
        objArrayTransporteParametros = new ArrayList<>();

        objArrayTransporteParametros.add((String.valueOf(lstRegistros.getSelectedItem().getValue())));

        objMensGenericos = new
Gson().fromJson(portTramites.getGsonFromRoute("tramites",
"ug.sistema.academico.bo.tramites.BandejaTrabajoBO","getRegistro",
new
Gson().toJson(objArrayTransporteParametros),
objUtilsConstants.getInformacionSession().getvStrGlobalUserID()), MensajesGenericos.class);

        if
(objMensGenericos.getvStrCode().equals(objUtils.getObjProperties().getProperty("fStrSUCCESS")))
        {
            if (objMensGenericos.getMapMensajesGenericos().size() > 0) {

                //txtIdSolicitud.setValue(objMensGenericos.getMapMensajesGenericos().get("idSolicitud"));
                //txtIdSolicitud.setVisible(false);

                txtCedula.setValue(objMensGenericos.getMapMensajesGenericos().get("cedula"));
            }
        }
    }

```

```

txtUsuario.setValue(objMensGenericos.getMapMensajesGenericos().get("usuario"));
txtTramite.setValue(objMensGenericos.getMapMensajesGenericos().get("tramite"));
txtTipoTramite.setValue(objMensGenericos.getMapMensajesGenericos().get("detalleTramite"));
txtCarrera.setValue(objMensGenericos.getMapMensajesGenericos().get("carrera"));
txtObservaciones.setValue(objMensGenericos.getMapMensajesGenericos().get("comentario"));
System.out.println("Fecha:
"+objMensGenericos.getMapMensajesGenericos().get("fecha"));
String fecha =
objMensGenericos.getMapMensajesGenericos().get("fecha");
txtfecha.setValue(fecha);
}
/*if (objMensGenericos.getMapMensajesGenericos()
.get("estado").trim().equals("APROBADO")) {
btnAprobar.setVisible(false);
}*/
}
}

public void actualizarBandeja() throws Exception{
divDetalleTramite.setVisible(false);
divRegistros.setVisible(true);
cargaRegistros(0, "");
}

public void aprobarCertificado() throws Exception{
Window objWinModal = (Window) Executions.createComponents("doModal.zul",
winAdminTramites, null);
objWinModal.doModal();
if (objLabelAction.getValue().trim().equals("modificar"))
{
objArrayTransporteParametros = new ArrayList<>();
objArrayTransporteParametros.add(txtIdSolicitud.getText().toString());
objArrayTransporteParametros.add("A");
objArrayTransporteParametros.add("");

objArrayTransporteParametros.add(txtObservaciones.getText().toString()
+" Aprobado por:
"+objUtils.getInformacionSession().getvStrGlobalUsername()
+"", ID:
"+objUtils.getInformacionSession().getvStrGlobalUserID());

objMensGenericos = new
Gson().fromJson(portTramites.getGsonFromRoute("Certificados",
"ug.sistema.academico.bo.certificados.AdminCertBO", "actualizarRegistro", new
Gson().toJson(objArrayTransporteParametros),
objUtilsConstants.getInformacionSession().getvStrGlobalUserID(), MensajesGenericos.class);

if
(objMensGenericos.getvStrCode().equals(objUtils.getObjProperties().getProperty("fStrSUCCESS"))) {
//preRegistro();
txtEstado.setValue("APROBADO");
btnAprobar.setVisible(false);
btnOK.setVisible(true);
}
}
}

public boolean verificarProceso(String idTramite) throws Exception{
boolean flag = false;
objArrayTransporteParametros = new ArrayList<>();
objArrayTransporteParametros.add(idTramite);

```

```

        System.out.println("Id registro:"+idTramite);
        objMensGenericos = new Gson().fromJson(portTramites.getGsonFromRoute("tramites",
        "ug.sistema.academico.bo.tramites.BandejaTrabajoBO","procesoSiguiente", new
Gson().toJson(objArrayTransporteParametros),
        objUtilsConstants.getInformacionSession().getvStrGlobalUserID()), MensajesGenericos.class);

        if
(objMensGenericos.getvStrCode().equals(objUtils.getObjProperties().getProperty("fStrSUCCESS"))) {
            //preRegistro();

            nexProceso =
objMensGenericos.getMapMensajesGenericos().get("idProcesoSiguiente");
            if(nexProceso != null){
                flag = true;
            }
        }
        return flag;
    }

    public void revisarRegistro()throws Exception{
        Window objWinModal = (Window) Executions.createComponents("doModal.zul",
winAdminTramites, null);
        objWinModal.doModal();
        if (objLabelAction.getValue().trim().equals("modificar"))
        {
            objArrayTransporteParametros = new ArrayList<>();

            objArrayTransporteParametros.add((String.valueOf(lstRegistros.getSelectedItem().getValue())));
            objArrayTransporteParametros.add(nexProceso);

            objArrayTransporteParametros.add(objUtils.getInformacionSession().getvStrGlobalUserID());
            objArrayTransporteParametros.add(txtObservaciones.getText().toString());

            objMensGenericos = new
Gson().fromJson(portTramites.getGsonFromRoute("tramites",
        "ug.sistema.academico.bo.tramites.BandejaTrabajoBO","actualizarRegistro", new
Gson().toJson(objArrayTransporteParametros),
        objUtilsConstants.getInformacionSession().getvStrGlobalUserID()), MensajesGenericos.class);

            if
(objMensGenericos.getvStrCode().equals(objUtils.getObjProperties().getProperty("fStrSUCCESS"))) {
                //preRegistro();
                btnAprobar.setVisible(false);
                btnOK.setVisible(true);
            }
        }

        //creacion reporte jasper

        public void generarReporte() throws Exception {

            try {
                Connection con = this.sqlConexion();
                //String dirReporte = System.getProperty("user.dir") +
                "\\Reportes\\Certificado de Cursos.jrxml";
                String dirReporte =
                Executions.getCurrent().getDesktop().getWebApp().getRealPath("sistemaCompleto/tramites/reportes/Certificad
                o de Cursos.jrxml");
                System.out.println(dirReporte);
                String vStrPathPDF = dirReporte.replace(".jrxml", new
                Random().nextLong() + ".pdf");
            }
        }
    }
}

```

```

HashMap<String, Object> mp= new HashMap<String,
Object>());
mp.put("p_identificacion",new String("77192216"));
JasperReport report =
JasperCompileManager.compileReport(dirReporte);
JasperPrint print = JasperFillManager.fillReport(report, mp, con);
JasperExportManager.exportReportToPdfFile(print,vStrPathPDF);
File objFile = new File(vStrPathPDF);
byte[]objByteArrayInputStream = new byte[(int) objFile.length()];
FileInputStream fileInputStream;
try {
fileInputStream = new FileInputStream(objFile);
fileInputStream.read(objByteArrayInputStream);
fileInputStream.close();
this.mostrarReporte(vStrPathPDF);
} catch (Exception e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
con.close();
} catch (JRException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
} catch (SQLException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
}

public Connection sqlConexion() throws ClassNotFoundException, SQLException
{
Connection conn = null;
Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
String url =
"jdbc:sqlserver://localhost:1433;databaseName=BdSistemaAcademico;user=sai;password=12345";
conn = DriverManager.getConnection(url);
return conn;
}

//envio de notificacion via mail

public void mostrarReporte(String ruta){
Map<String,Object> arg = new HashMap<String,Object>();
String accion;

String path = ruta;
path.replaceAll("\\\\", "/");

System.out.println(path);
arg.put("rutaPdf",path);
arg.put("enunciado","Vizualizacion de Certificado");

Window objWindow = (Window)
Executions.createComponents("tramites/modalMostrarArchivo.zul", winAdminTramites, arg);
objWindow.doModal();

accion = objLabelAction.getValue();
if(accion.trim().equals("cancelar")){
new File(ruta).delete();
}
}

public void enviarMail(String to,String cc, String contenido,String archivo){

objArrayTransporteParametros = new ArrayList<>();
objArrayTransporteParametros.add("alvizueta_89@yahoo.com");

```

```

        objMensGenericos = new Gson().fromJson(portTramites.getGsonFromRoute("tramites",
            "ug.sistema.academico.bo.tramites.EnviarCorreo","enviarMail", new
Gson()).toJson(objArrayTransporteParametros),

        objUtilsConstants.getInformacionSession().getvStrGlobalUserID()), MensajesGenericos.class);

        if
(objMensGenericos.getvStrCode().equals(objUtils.getObjProperties().getProperty("fStrSUCCESS"))) {
            ToastMessage.showNotify("Registro Enviado", "alertify."
                + objUtils.getObjProperties().getProperty("fStrWARNING"));
        }
        else{
            ToastMessage.showNotify("Fallo Envio :)", "alertify."
                + objUtils.getObjProperties().getProperty("fStrWARNING"));
        }
    }
}

```

Opciones.java

```

package ug.sistema.academico.controladores.tramites;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.HashMap;
import java.util.Map;
import java.util.StringTokenizer;

import org.zkoss.zk.ui.Component;
import org.zkoss.zk.ui.Executions;
import org.zkoss.zk.ui.Sessions;
import org.zkoss.zk.ui.select.annotation.Wire;
import org.zkoss.zk.ui.util.GenericForwardComposer;
import org.zkoss.zul.Button;
import org.zkoss.zul.Caption;
import org.zkoss.zul.ComboBox;
import org.zkoss.zul.Div;
import org.zkoss.zul.Label;
import org.zkoss.zul.ListBox;
import org.zkoss.zul.Listitem;
import org.zkoss.zul.Textbox;
import org.zkoss.zul.Window;

import ug.sistema.academico.service.ruteador.IRuteadorSRV;
import ug.sistema.academico.service.ruteador.RuteadorSOAP;
import ug.sistema.clases.CertificadosHeaderView;
import ug.sistema.clases.SeguridadHeadersView;
import ug.sistema.clases.ToastMessage;
import ug.sistema.clases.TransportePerfilesAsignados;
import ug.sistema.libs.Utils;

import com.google.gson.Gson;

import ec.edu.ug.academico.type.MensajesGenericos;

public class Opciones extends GenericForwardComposer<Component> {

    private static final long serialVersionUID = 1L;
    @Wire
    private Caption objLblTitle;

```

```

@Wire
Utils objUtils, objUtilsConstants;
@Wire
private Combobox cmbPaging,cmbEncargado,cmbPerfiles;
@Wire
private Textbox bndBuscar,txtNombreProceso;
@Wire
private Listbox cmbPolíticas, lstUsuarios,lstCertificados,
lstProcesos,lstEncargados,lstSolicitudes,lstProcesoPersona,lstAdmisitradores;
@Wire
private Window objWinOpciones;
@Wire
private Listbox objLstbxUsuariosPerfiles, cmbEstado,cmbEstadoProceso;
@Wire
private Textbox txtClave, txtUsuario, txtApPaterno, txtApMaterno,
txtSegNombre, txtPrNombre, txtCedula, txtIdUsuario,
txtActionOpcion, txtIdPersona, txtEmail;
private Textbox txtIdCertificado,txtNombreCertificado,txtUsuarioCreacion,
txtFechaCreado,txtDescripcionDoc;

@Wire
private Button btnGuardar, btnDeshacer, btnNew,btnNuevoProceso;
@Wire
private Label lblAlert, lblCarrera, lblCodeCarrera,parametro;
@Wire
IRuteadorSRV portTramites, portMalla,portSeguridad;
ArrayList<Object> objArrayTransporteParametros;
Div divListadoUsarios, divOpcionesMain,divdetalleCertificados;
String vStrSeparador;
String vStrTraza;
MensajesGenericos objMensGenericos;
String vStrAccesoDirecto;
String vStrNombreCarrera;
@Wire
Label objLabelAction,nombreCertificado,ruta;
boolean vBoolFlag;
TransportePerfilesAsignados objTransportePerfilAsig;
String txtEmailAnterior;

public Opciones() {
    boolean valid = Sessions.getCurrent().getAttribute(
        "objGlobalInformation") != null ? true : false;
    if (!valid) {
        Executions.sendRedirect("SegLoginForm.zul");
    }
}

public void doAfterCompose(Component objComp) {
    try {
        objUtils = new Utils();
        objUtilsConstants = new Utils();
        objUtils.setProperties("/resources/mensajes.properties");
        objUtilsConstants.setProperties("/resources/constants.properties");
        vBoolFlag = false;
        portTramites =RuteadorSOAP.getIRuteadorSOAP("tramites");
        portSeguridad =RuteadorSOAP.getIRuteadorSOAP("seguridad");
        objTransportePerfilAsig = new TransportePerfilesAsignados();
        if ((String) Executions.getCurrent().getAttribute("mainPage") != null) {
            super.doAfterCompose(objComp);
            vStrAccesoDirecto = (String) Executions.getCurrent()
                .getAttribute("vStrAccesoDirecto") == null
                ? "" : (String)
                Executions.getCurrent().getAttribute(
                    "vStrAccesoDirecto");
            vStrSeparador = objUtilsConstants.getObjProperties()
                .getProperty("fStrSeparadorParametros");

```

```

        this.chargeView(0, "", "");
        this.listadoTipoTramites();
        this.cargarPerfiles();
        this.listProcesoPersona();
        this.listaProcesos();
    } else {
Sessions.getCurrent().removeAttribute("objGlobalInformation");
        Executions.sendRedirect("../SegLoginForm.zul");
    }

    } catch (Exception e) {
        e.printStackTrace();
    }
}

public void buscarUsuarios() throws Exception {
    String vStrBuscarCargo = cmbPerfiles.getValue().toString();
    System.out.println("Parametro cargo "+vStrBuscarCargo);
    if (!bndBuscar.getValue().trim().equals("")) ||
lcmbPerfiles.getValue().toString().equals("")) {
        String vStrBuscarCedula =
objUtils.checkNumber(bndBuscar.getValue().trim(), 10);
        String vStrBuscarNombre =
objUtils.checkText(bndBuscar.getValue().trim());
        vStrBuscarCargo = cmbPerfiles.getValue().trim();

        System.out.println("Parametro Cedula: "+vStrBuscarCedula);
        System.out.println("Parametro nombre "+vStrBuscarNombre);
        System.out.println("Parametro cargo "+vStrBuscarCargo);

        if (vStrBuscarCedula.equals("") ||
vStrBuscarNombre.equals("") || !vStrBuscarCargo.equals("")) {
            this.chargeView(0,
bndBuscar.getValue().trim(), vStrBuscarCargo.trim());
        } else {
            if (!vStrBuscarCedula.equals("")) {
                ToastMessage.showNotify(
                    vStrBuscarCedula,
                    "alertify."
                +
objUtils.getObjProperties().getProperty(
                    "fStrERROR"));
            } else {
                if (!vStrBuscarNombre.equals("")) {
                    ToastMessage.showNotify(vStrBuscarNombre,
                    "alertify."
                +
objUtils.getObjProperties()
                    .getProperty("fStrERROR"));
                }
            }
        }
    } else {
        this.chargeView(0, "", "");
    }
}

public void presentarDatosPaging() throws Exception {
    this.chargeView(Integer.parseInt(cmbPaging.getValue()), "", "");
}

public void cargarPerfiles(){
    StringTokenizer st;
    objArrayTransporteParametros=new ArrayList<>();
}

```

```

        MensajesGenericos objMensGenericos = new
Gson().fromJson(portSeguridad.getGsonFromRoute("tramites",

        "ug.sistema.academico.bo.tramites.OpcionesBO","getListPerfiles", new
Gson().toJson(objArrayTransporteParametros),

        objUtilsConstants.getInformacionSession().getvStrGlobalUserID()),MensajesGenericos.class);

        cmbPerfiles.getChildren().clear();
        st = new StringTokenizer(objMensGenericos.getvStrMessage(",");
        while (st.hasMoreTokens()){
            cmbPerfiles.appendItem(st.nextToken());
            System.out.println ("perfiles"+st.toString());
        }
    }

private void chargeView(int vIntPage, String vStrParametro,String vStrCargo)
throws Exception {
    String vStrSeparador = objUtilsConstants.getObjProperties()
        .getProperty("fStrSeparadorParametros");
    String vStrTraza = "null"
        + vStrSeparador
        + "% "
        + vStrParametro
        + "% "
        + vStrSeparador
        + "% "
        + vStrParametro
        + "% "
        + vStrSeparador
        + "% "
        + vStrParametro
        + "% "
        + vStrSeparador
        + "% "
        + vStrParametro
        + "% "
        + vStrSeparador
        + "% "
        + vStrCargo+ "%";

    objArrayTransporteParametros = new ArrayList<>();
    objArrayTransporteParametros.add("SQL_USUARIOS_ADMINVIEW");
    objArrayTransporteParametros.add(vStrTraza);
    objArrayTransporteParametros.add(true);
    objArrayTransporteParametros.add("0");
    objArrayTransporteParametros.add(objUtilsConstants.getObjProperties()
        .getProperty("fStrMaxRegisters"));
    objArrayTransporteParametros.add(String.valueOf(vIntPage));
    objArrayTransporteParametros.add(objUtilsConstants
        .getInformacionSession().getvStrGlobalUserID());

    MensajesGenericos objMensGenericos = new Gson().fromJson(portSeguridad
        .getGsonFromRoute("Seguridad",

        "ug.sistema.academico.bo.seguridad.SegControlesBOImpl",
        "getListUsuarios", new Gson()

        .toJson(objArrayTransporteParametros),

        objUtilsConstants.getInformacionSession()

        .getvStrGlobalUserID()),

        MensajesGenericos.class);

    cmbPaging.getChildren().clear();
    if (objMensGenericos.getvStrCode().equals(
        objUtils.getObjProperties().getProperty("fStrSUCCESS"))) {

        lstUsuarios.getChildren().clear();
        lstUsuarios.appendChild(SeguridadHeadersView.getHeaderUsuarios());
    }
}

```

```

        if (objMensGenericos.getObjContenido().size() > 1) {
            int vIntCabecera = 0;
            for (ArrayList<String> objRegister : objMensGenericos
                .getObjContenido()) {
                if (vIntCabecera == 0) {
                    for (int vIntIndice = 0; vIntIndice < Integer
                        .parseInt(objRegister.get(3)); vIntIndice++) {
                        cmbPaging.appendItem(String.valueOf(vIntIndice));
                    } else {
                        lstUsuarios.appendChild(objUtilsConstants
                            .getListItem(objRegister));
                    }
                    vIntCabecera++;
                    cmbPaging.setSelectedIndex(vIntPage);
                }
            } else {
                ToastMessage.showNotify(objMensGenericos.getvStrMessage(),
                    "alertify." + objMensGenericos.getvStrCode());
            }
        }

        public void cleanDatos() throws Exception {
            lblCarrera.setVisible(false);
            txtApMaterno.setValue("");
            txtApPaterno.setValue("");
            txtCedula.setValue("");
            txtCedula.setReadOnly(false);
            txtClave.setValue("");
            txtPrNombre.setValue("");
            txtSegNombre.setValue("");
            txtUsuario.setValue("");
            txtEmail.setValue("");
            cmbPolíticas.setSelectedItem(null);
            cmbEstado.setSelectedItem(null);

            objLstbxUsuariosPerfiles.getChildren().clear();

            objLstbxUsuariosPerfiles.appendChild(SeguridadHeadersView.getHeaderUsuarios());

            if (objLstbxUsuariosPerfiles.getChildren().size() > 1) {
                do {
                    objLstbxUsuariosPerfiles.getChildren().remove(1);
                } while (objLstbxUsuariosPerfiles.getChildren().size() > 1);
            }
            btnGuardar.setVisible(false);
            btnDeshacer.setVisible(false);
            btnNew.setVisible(true);
            objTransportePerfilAsig = new TransportePerfilesAsignados();
        }

        public void activarBotones() {
            btnGuardar.setVisible(false);
            btnDeshacer.setVisible(false);
            btnNew.setVisible(true);
        }

        public void descargarGrid(String vStrPage) {

```

```

        try {
            objUtils.exportTreeToPdf(lstUsuarios, "usuariosSistema_pag"
                + vStrPage + ".pdf");
        } catch (Exception ex) {
            ToastMessage.showNotify(ex.getMessage(), "alertify."
                +
objUtils.getObjProperties().getProperty("fStrERROR"));
        }
    }

    public void activarOpciones(String vStrAction, String vStrValueID,
        String vStrIdPersona) {
        txtActionOpcion.setValue(vStrAction);
        txtIdUsuario.setValue(vStrValueID);
        txtIdPersona.setValue(vStrIdPersona);
        btnDeshacer.setVisible(true);
        btnGuardar.setVisible(true);
        btnNew.setVisible(false);
    }

    public void listadoTipoTramites()throws Exception{

        objArrayTransporteParametros=new ArrayList<>();
        objArrayTransporteParametros.add("HQL_TIPO_TRAMITES");
        objArrayTransporteParametros.add("CERTIFICADO");

        MensajesGenericos objMensGenericos = new
Gson().fromJson(portTramites.getGsonFromRoute("tramites",
        "ug.sistema.academico.bo.tramites.OpcionesBO","getTramite", new
Gson().toJson(objArrayTransporteParametros),
        objUtilsConstants.getInformacionSession().getvStrGlobalUserID()),MensajesGenericos.class);

        if
(objMensGenericos.getvStrCode().equals(objUtils.getObjProperties().getProperty("fStrSUCCESS")))
        {

            lstCertificados.appendChild(CertificadosHeaderView.getHeaderListadoCertificados());
            if (objMensGenericos.getObjContenido().size() > 1)
            {
                for (ArrayList<String> objRegister :
objMensGenericos.getObjContenido())
                {

                    lstCertificados.appendChild(objUtilsConstants.getListItem(objRegister));
                }
            }
        } else {
            ToastMessage.showNotify(objMensGenericos.getvStrMessage(),
                "alertify." + objMensGenericos.getvStrCode());
        }

        objArrayTransporteParametros.clear();
        objArrayTransporteParametros.add("HQL_TIPO_TRAMITES");
        objArrayTransporteParametros.add("SOLICITUD");

        MensajesGenericos objMensGenericos2 = new
Gson().fromJson(portTramites.getGsonFromRoute("tramites",
        "ug.sistema.academico.bo.tramites.OpcionesBO","getTramite", new
Gson().toJson(objArrayTransporteParametros),
        objUtilsConstants.getInformacionSession().getvStrGlobalUserID()),MensajesGenericos.class);

        if
(objMensGenericos.getvStrCode().equals(objUtils.getObjProperties().getProperty("fStrSUCCESS")))

```

```

        {
            lstSolicitudes.appendChild(CertificadosHeaderView.getHeaderListadoCertificados());
            if (objMensGenericos2.getObjContenido().size() > 1)
            {
                for (ArrayList<String> objRegister :
objMensGenericos2.getObjContenido())
                {
                    lstSolicitudes.appendChild(objUtilsConstants.getListItem(objRegister));
                }
            }
        } else {
            ToastMessage.showNotify(objMensGenericos.getvStrMessage(),
"alertify." + objMensGenericos.getvStrCode());
        }
    }

    public void listProcesoPersona()throws Exception {
        objArrayTransporteParametros=new ArrayList<>();
        objArrayTransporteParametros.add("HQL_TRAMITES_PROCESOPERSONA");
        objArrayTransporteParametros.add("ACTIVO");
        objArrayTransporteParametros.add(false);
        objArrayTransporteParametros.add("0");
        objArrayTransporteParametros.add("0");
        objArrayTransporteParametros.add("0");

        MensajesGenericos objMensGenericos = new
Gson().fromJson(portTramites.getGsonFromRoute("Tramites",
"ug.sistema.academico.bo.tramites.OpcionesBO","getProcesoPersona", new
Gson().toJson(objArrayTransporteParametros),
objUtilsConstants.getInformacionSession().getvStrGlobalUserID()),MensajesGenericos.class);

        if
(objMensGenericos.getvStrCode().equals(objUtils.getObjProperties().getProperty("fStrSUCCESS")))
        {
            lstProcesoPersona.getChildren().clear();
            lstAdmisitradores.getChildren().clear();

            lstProcesoPersona.appendChild(CertificadosHeaderView.getHeaderListaProcesos());
            lstAdmisitradores.appendChild(CertificadosHeaderView.getHeaderFimaCertificado());
            if (objMensGenericos.getObjContenido().size() > 1)
            {
                int vIntCabecera = 0;
                for (ArrayList<String> objRegister :
objMensGenericos.getObjContenido())
                {
                    if (vIntCabecera == 0)
                    {
                        for (int vIntIndice = 0;
vIntIndice < Integer.parseInt(objRegister.get(3)); vIntIndice++) {
                            }
                        } else {
                            lstProcesoPersona.appendChild(objUtilsConstants.getListItem(objRegister));
                            lstAdmisitradores.appendChild(objUtilsConstants.getListItem(objRegister));
                        }
                    }
                }
                vIntCabecera++;
            }
        }
    }
}

```

```

    }
} else {
    ToastMessage.showNotify(objMensGenericos.getvStrMessage(),
        "alertify." + objMensGenericos.getvStrCode());
}
}

public void listaProcesos()throws Exception {

    objArrayTransporteParametros=new ArrayList<>();
    objArrayTransporteParametros.add("HQL_TRAMITES_PROCESOS");
    objArrayTransporteParametros.add("ACTIVO");
    objArrayTransporteParametros.add(false);
    objArrayTransporteParametros.add("0");
    objArrayTransporteParametros.add("0");
    objArrayTransporteParametros.add("0");

    MensajesGenericos objMensGenericos = new
Gson().fromJson(portTramites.getGsonFromRoute("tramites",

    "ug.sistema.academico.bo.tramites.OpcionesBO","getProcesos", new
Gson().toJson(objArrayTransporteParametros),

    objUtilsConstants.getInformacionSession().getvStrGlobalUserID()),MensajesGenericos.class);

    if
(objMensGenericos.getvStrCode().equals(objUtils.getObjProperties().getProperty("fStrSUCCESS")))
    {
        lstEncargados.getChildren().clear();

        lstEncargados.appendChild(CertificadosHeaderView.getHeaderListaEncargados());
        if (objMensGenericos.getObjContenido().size() > 1)
        {
            int vIntCabecera = 0;
            for (ArrayList<String> objRegister :
objMensGenericos.getObjContenido())
            {
                if (vIntCabecera == 0)
                {
                    for (int vIntIndice = 0;
vIntIndice < Integer.parseInt(objRegister.get(3)); vIntIndice++) {
                        } else {
                            lstEncargados.appendChild(objUtilsConstants.getListItem(objRegister));
                        }
                    vIntCabecera++;
                }
            }
        } else {
            ToastMessage.showNotify(objMensGenericos.getvStrMessage(),
                "alertify." + objMensGenericos.getvStrCode());
        }
    }

    public void cargarDetalles()throws Exception{

        Window objWinModal = (Window) Executions.createComponents("doModal.zul",
objWinOpciones, null);
        objWinModal.doModal();
        if (objLabelAction.getValue().trim().equals("modificar"))
        {

```

```

        divOpcionesMain.setVisible(false);
        divdetalleCertificados.setVisible(true);
        divListadoUsuarios.setVisible(false);
        String idCertificado =
String.valueOf(lstCertificados.getSelectedItem().getValue());

        objArrayTransporteParametros = new ArrayList<>();
        objArrayTransporteParametros.add(idCertificado);

        objMensGenericos = new
Gson().fromJson(portTramites.getGsonFromRoute("tramites",
        "ug.sistema.academico.bo.certificados.CertListadosBO","getObjCertificado", new
Gson().toJson(objArrayTransporteParametros),
        objUtilsConstants.getInformacionSession()
        .getvStrGlobalUserID()),
        MensajesGenericos.class);
        //this.listaProcesoPersonas();

        if
(objMensGenericos.getvStrCode().equals(objUtils.getObjProperties().getProperty("fStrSUCCESS"))) {
        if
(objMensGenericos.getMapMensajesGenericos().size() > 0) {

            txtIdCertificado.setValue(objMensGenericos.getMapMensajesGenericos().get("id"));
            txtNombreCertificado.setValue(objMensGenericos.getMapMensajesGenericos().get("Documento"));
            txtUsuarioCreacion.setValue(objMensGenericos.getMapMensajesGenericos().get("Creador"));
            txtFechaCreado.setValue(objMensGenericos.getMapMensajesGenericos().get("FechaCreacion"));
            txtDescripcionDoc.setValue(objMensGenericos.getMapMensajesGenericos().get("Descripcion"));
            cmbEstado.getChildren().clear();
            Listitem comboitem = new Listitem();
            comboitem.setLabel("Activo");

            comboitem.setValue(objMensGenericos.getMapMensajesGenericos().get("Estado"));
            cmbEstado.appendChild(comboitem);
            cmbEstado.setSelectedIndex(0);
        }
    }
}

public void modallIngreso(){
    Map<String, Object> arg = new HashMap<String, Object>();
    String accion=null;
    StringTokenizer st = null;

    if(parametro.getValue().trim().equals("CERTIFICADO"))
    {
        arg.put("enunciado","Nuevo Certificado");
        arg.put("maxDias","false");
        arg.put("rutaArchivo","true");

        Window objWindow = (Window)
Executions.createComponents("tramites/modalNuevoTramite.zul", objWinOpciones, arg);
        objWindow.doModal();

        accion =objLabelAction.getValue();
        System.out.println(accion);
    }
}

```

```

        if(accion.trim().equals("cancelar")){
            ToastMessage.showNotify("Accion Cancelada",
                "alertify.warning");
        }else{
            st = new StringTokenizer(accion, ",");
            objArrayTransporteParametros=new ArrayList<>();

            while
(st.hasMoreTokens()){objArrayTransporteParametros.add(st.nextToken());}

            objArrayTransporteParametros.add(parametro.getValue().toString());

            objArrayTransporteParametros.add(objUtils.getInformacionSession().getvStrGlobalUserID());

            objMensGenericos = new
Gson().fromJson(portTramites.getGsonFromRoute("tramites",
                "ug.sistema.academico.bo.tramites.OpcionesBO", "creaCertificado", new
Gson().toJson(objArrayTransporteParametros),
                objUtilsConstants.getInformacionSession().getvStrGlobalUserID()), MensajesGenericos.class);

            if
(objMensGenericos.getvStrCode().equals(objUtils.getObjProperties().getProperty("fStrSUCCESS"))){
                ToastMessage.showNotify("Registro
Insertado Correctamente", "alertify.warning");
            }
        }
    }

    if(parametro.getValue().trim().equals("SOLICITUD")){

        arg.put("enunciado", "Nueva Solicitud");
        arg.put("maxDias", "true");
        arg.put("rutaArchivo", "false");

        Window objWindow = (Window)
Executions.createComponents("tramites/modalNuevoTramite.zul", objWinOpciones, arg);
        objWindow.doModal();

        accion =objLabelAction.getValue();

        if(accion.trim().equals("cancelar")){
            ToastMessage.showNotify("Accion Cancelada",
                "alertify.warning");
        }else{
            st = new StringTokenizer(accion, ",");
            objArrayTransporteParametros=new ArrayList<>();

            while
(st.hasMoreTokens()){objArrayTransporteParametros.add(st.nextToken());}

            objArrayTransporteParametros.add(parametro.getValue().toString());

            objArrayTransporteParametros.add(objUtils.getInformacionSession().getvStrGlobalUserID());

            objMensGenericos = new
Gson().fromJson(portTramites.getGsonFromRoute("tramites",
                "ug.sistema.academico.bo.tramites.OpcionesBO", "creaSolicitud", new
Gson().toJson(objArrayTransporteParametros),
                objUtilsConstants.getInformacionSession().getvStrGlobalUserID()), MensajesGenericos.class);

```

```

        if
(objMensGenericos.getvStrCode().equals(objUtils.getObjProperties().getProperty("fStrSUCCESS"))){
    ToastMessage.showNotify("Registro Insertado
Correctamente","alertify.warning");
    }
    }
}

public void modalIngresoProceso(){
    Map<String, Object> arg = new HashMap<String, Object>();
    String accion=null;
    StringTokenizer st = null;
    Date now = new Date(System.currentTimeMillis());
    SimpleDateFormat date = new SimpleDateFormat("yyyy-MM-dd");
    String fecha = date.format(now);

    arg.put("fecha", fecha);
    arg.put("enunciado", "Nuevo Proceso");
    Window objWindow = (Window)
Executions.createComponents("tramites/modalProcesos.zul", objWinOpciones, arg);
    objWindow.doModal();

    accion =objLabelAction.getValue();
    System.out.println("Parametro modalProceso: "+accion);
    if(accion.trim().equals("cancelar")){
        ToastMessage.showNotify("Accion Cancelada",
            "alertify.warning");
    }else{
        st = new StringTokenizer(accion, ",");
        objArrayTransporteParametros=new ArrayList<>();

        while
(st.hasMoreTokens()){objArrayTransporteParametros.add(st.nextToken());}

        objArrayTransporteParametros.add(objUtils.getInformacionSession().getvStrGlobalUserID());

        objMensGenericos = new
Gson().fromJson(portTramites.getGsonFromRoute("tramites",
            "ug.sistema.academico.bo.tramites.OpcionesBO", "creaProceso", new
Gson().toJson(objArrayTransporteParametros),

            objUtilsConstants.getInformacionSession().getvStrGlobalUserID()), MensajesGenericos.class);

        if
(objMensGenericos.getvStrCode().equals(objUtils.getObjProperties().getProperty("fStrSUCCESS"))){
    ToastMessage.showNotify("Registro Insertado
Correctamente","alertify.warning");
    }
    }
}
}

```

BandejaTrabajoBO.java

```

package ug.sistema.academico.bo.tramites;

import java.util.ArrayList;

import org.hibernate.Session;

import ug.sistema.academico.models.malla.Carrera;
import ug.sistema.academico.models.seguridad.Usuarios;
import ug.sistema.academico.models.tramites.DetalleRegistro;

```

```

import ug.sistema.academico.models.tramites.ProcesoSiguiente;
import ug.sistema.academico.models.tramites.RegistroTramites;
import ug.sistema.academico.utils.ListaGenerica;
import ug.sistema.academico.utils.Utilis;

```

```

import ug.sistema.academico.eao.malla.CarreraEAO;
import ug.sistema.academico.eao.seguridad.UsuariosEAO;
import ug.sistema.academico.eao.tramites.DetalleRegistroEAO;
import ug.sistema.academico.eao.tramites.ProcesoSiguienteEAO;
import ug.sistema.academico.eao.tramites.RegistroTramitesEAO;

```

```

import com.google.gson.Gson;

```

```

import ec.edu.ug.academico.type.MensajesGenericos;

```

```

public class BandejaTrabajoBO {

```

```

    /**
     * Método : getListRegistros.
     * Acceso : público.
     * Descripción : Consulta a la base y obtiene los registros de los tramites ingresados, y los
    retorna.
     * Autor : Alvaro Vizqueta
     * Creación : 10/05/2015
     * Modificación : 19/06/2015
     */
    public String getListRegistros(String vStrQuery, String vStrParametros,
        boolean vBoolPaginador, String vIntMinimo, String vIntMaximo,
        String vIntPagina, String vStrUsuario) {

        String vStrResultado = "";
        RegistroTramitesEAO objRtramitesEAO = new RegistroTramitesEAO();
        objRtramitesEAO.setearConexion("sistemaacademico");
        Session objSessionR = objRtramitesEAO.getSession();

        UsuariosEAO objUsuarioEAO = new UsuariosEAO();
        objUsuarioEAO.setearConexion("seguridad");
        @SuppressWarnings("unused")
        Session objSessionUsuario = objUsuarioEAO.getSession();
        Usuarios objUsuarios = null;

        CarreraEAO objCarreraEAO = new CarreraEAO();
        objCarreraEAO.setearConexion("sistemaacademico");
        @SuppressWarnings("unused")
        Session objSessionMalla = objCarreraEAO.getSession();
        Carrera objCarrera = null;

        MensajesGenericos objMsjGenerico = new MensajesGenericos("", "");
        Utilis objUtilis = new Utilis();
        objUtilis.setProperties("/resources/constants.properties");

        ArrayList<ArrayList<String>> objContenido = new ArrayList<ArrayList<String>>();
        try {
            ArrayList<Object> vArrayParameters = objUtilis.getArrayListObject(
                vStrParametros,
                objUtilis.getObjProperties().getProperty(
                    "fStrSeparadorParametros"));
            ListaGenerica objListaGenerica = objRtramitesEAO
                .getResultListCustomAcademico(vStrQuery,
                vArrayParameters, vBoolPaginador, vIntMinimo, vIntMaximo, vIntPagina);

            ArrayList<String> arrayfilas = new ArrayList<String>();
            arrayfilas.add(String.valueOf(objListaGenerica.getvIntMinimo()));
            arrayfilas.add(String.valueOf(objListaGenerica.getvIntPagina()));
            arrayfilas.add(String.valueOf(objListaGenerica.getvIntMaximo()));
            arrayfilas.add(String.valueOf(objListaGenerica.getvNumPaginas()));
            objContenido.add(arrayfilas);
        }
    }
}

```

```

ArrayList<Object> objArrayContenido = objListaGenerica.getArrayContenido();

if (!vBoolPaginador) {
    for (Object object : objArrayContenido) {
        RegistroTramites objRegistros = (RegistroTramites) object;
        arrayfilas = new ArrayList<String>();
        arrayfilas.add(objRegistros.getId());

        arrayfilas.add(objRegistros.getEspecificacionTramite().getTipoTramite().getNombre()
            +
            "+objRegistros.getEspecificacionTramite().getNombre());
            objUsuarios =
            objUsuarioEAO.getUsuarioByIdentificador(objRegistros.getIdusuario(),"id");
            arrayfilas.add(objUsuarios.getPrimerNombre()+
            "+objUsuarios.getSegundoNombre()+ " "+objUsuarios.getApellidoPaterno()
            + " "+objUsuarios.getApellidoMaterno());
            objCarrera =
            objCarreraEAO.getRegistroIdentificador(objRegistros.getIdCarrera(),"id");
            arrayfilas.add(objCarrera.getFacultad().getNombre()+ " /
            "+objCarrera.getNombre());

        arrayfilas.add(objRegistros.getEspecificacionTramite().getTipoTramite().getNombre());
        arrayfilas.add(objRegistros.getFechaCreacion().toString());
        arrayfilas.add(objRegistros.getProcesos().getDescripcion());
        objContenido.add(arrayfilas);
    }
} else {
    /*ArrayList<Object> arrayCertificados = objCertificadoEAO
    .quicksort(objArrayContenido);*/

    for (Object objTemp : objArrayContenido) {
        RegistroTramites objRegistros = (RegistroTramites) objTemp;
        arrayfilas = new ArrayList<String>();
        arrayfilas.add(objRegistros.getId());

        arrayfilas.add(objRegistros.getEspecificacionTramite().getTipoTramite().getNombre()
            +
            "+objRegistros.getEspecificacionTramite().getNombre());
            objUsuarios =
            objUsuarioEAO.getUsuarioByIdentificador(objRegistros.getIdusuario(),"id");
            arrayfilas.add(objUsuarios.getPrimerNombre()+
            "+objUsuarios.getSegundoNombre()+ " "+objUsuarios.getApellidoPaterno()
            + " "+objUsuarios.getApellidoMaterno());
            objCarrera =
            objCarreraEAO.getRegistroIdentificador(objRegistros.getIdCarrera(),"id");
            arrayfilas.add(objCarrera.getFacultad().getNombre()+ " /
            "+objCarrera.getNombre());

        arrayfilas.add(objRegistros.getEspecificacionTramite().getTipoTramite().getNombre());
        arrayfilas.add(objRegistros.getFechaCreacion().toString());
        arrayfilas.add(objRegistros.getProcesos().getDescripcion());
        objContenido.add(arrayfilas);
    }
}
objUtils.setProperties("/resources/mensajes.properties");
objMsjGenerico.setvStrMessage(objUtils.getMessage("",
    "fStrMessageOK"));
objMsjGenerico.setvStrCode(objUtils.getObjProperties().getProperty(
    "fStrSUCCESS"));
} catch (Exception e) {
    System.out.println(e.toString());
    objMsjGenerico.setvStrMessage(objUtils.getMessage("",
        "fStrMessageErrorDatabase")
        + "----"
        + e.getLocalizedMessage());
    objMsjGenerico.setvStrCode(objUtils.getObjProperties().getProperty(
        "fStrERROR"));
    System.out.println(e);
} finally {

```

```

        objSessionR.close();
    }
    objMsjGenerico.setObjContenido(objContenido);
    vStrResultado = new Gson().toJson(objMsjGenerico);
    return vStrResultado;
}

/*****
* Método           : getRegistro.
* Acceso           : público.
* Descripción      : Consulta a la base y obtiene los registros de los tramites ingresados, y los
retorna.
* Autor            : Alvaro Vizueta
* Creación         : 10/05/2015
* Modificación     : 19/06/2015
* *****/
@SuppressWarnings("unused")
public String getRegistro(String vStrIdRegistro){

    String vStrResultado = "";
    UsuariosEAO objUsuarioEAO =new UsuariosEAO();
    objUsuarioEAO.setearConexion("seguridad");
    Session objSessionUsuario = objUsuarioEAO.getSession();

    DetalleRegistroEAO objDetalleEAO = new DetalleRegistroEAO();
    objDetalleEAO.setearConexion("sistemaacademico");
    Session objSession = objDetalleEAO.getSession();

    CarreraEAO objCarreraEAO = new CarreraEAO();
    objCarreraEAO.setearConexion("sistemaacademico");
    Session objSessionMalla = objCarreraEAO.getSession();

    Utils objUtils = new Utils();
    objUtils.setProperties("/resources/mensajes.properties");
    MensajesGenericos objMsjGenerico = new MensajesGenericos("", "");

    try
    {

        DetalleRegistro objDetalle = null;
        objDetalle = objDetalleEAO.getRegistroIdentificador(vStrIdRegistro, "registroTramites.id");
        String idUsuario = objDetalle.getRegistroTramites().getIdusuario();

        Usuarios objUsuarios = null;
        objUsuarios = objUsuarioEAO.getUsuarioByIdentificador(idUsuario,"id");

        Carrera objCarrera = null;
        objCarrera = objCarreraEAO.getRegistroIdentificador(objUsuarios.getCarreraDefault(),"id");

        if (objDetalle == null || objUsuarios == null || objCarrera == null) {

objMsjGenerico.setvStrMessage(objUtils.getMessage("", "fStrMessageNotFound"));

objMsjGenerico.setvStrCode(objUtils.getObjProperties().getProperty("fStrWARNING"));
        } else {

            objMsjGenerico.setvStrMessage(objUtils.getMessage("",
                "fStrMessageOK"));
            objMsjGenerico.setvStrCode(objUtils.getObjProperties()
                .getProperty("fStrSUCCESS"));
            objMsjGenerico.addMensajeGenerico(new
MensajesGenericos("comentario",objDetalle.getComentarioAdministrativo()));
            objMsjGenerico.addMensajeGenerico(new
MensajesGenericos("detalleTramite",objDetalle.getRegistroTramites().getEspecificacionTramite().getNombre()))
;
            objMsjGenerico.addMensajeGenerico(new
MensajesGenericos("tramite",objDetalle.getRegistroTramites().getEspecificacionTramite().getTipoTramite().get
Nombre()));

```

```

        objMsjGenerico.addMensajeGenerico(new
MensajesGenericos("carrera",objCarrera.getFacultad().getNombre()
        + "-" +objCarrera.getNombre());
        objMsjGenerico.addMensajeGenerico(new
MensajesGenericos("fecha",objDetalle.getRegistroTramites().getFechaCreacion().toString());
        objMsjGenerico.addMensajeGenerico(new
MensajesGenericos("usuario",objUsuarios.getPrimerNombre()+" "+
        objUsuarios.getApellidoPaterno()+
"+objUsuarios.getApellidoMaterno());
        objMsjGenerico.addMensajeGenerico(new
MensajesGenericos("cedula",objUsuarios.getIdentificacion()));
    }
}
catch(Exception e)
{
    vStrResultado = objUtils.getMessage(vStrIdRegistro,
        "fStrMessageErrorDatabase");
    e.printStackTrace();
    objMsjGenerico.setvStrMessage(objUtils.getMessage(vStrIdRegistro,
        "fStrMessageErrorDatabase")
        + "___"
        + e.getLocalizedMessage());
    objMsjGenerico.setvStrCode(objUtils.getObjProperties().getProperty(
        "fStrERROR"));
}
finally {
    objSession.close();
}
vStrResultado = new Gson().toJson(objMsjGenerico);
return vStrResultado;
}

/*****
* Método          : actualizarRegistro.
* Acceso          : público.
* Descripción     : Consulta a la base y Actualiza los registros de los tramites ingresados, y los
retorna.
* Autor          : Alvaro Vizueta
* Creación       : 10/05/2015
* Modificacion  : 19/06/2015
* *****/
public String actualizarRegistro(String vStrIdSolicitud,
    String vStrProcesoSiguiente,String vStrIdPersona,String vStrObservaciones){

    String vStrResultado = "";
    RegistroTramitesEAO objRegistroTramiteEAO = new RegistroTramitesEAO();
    objRegistroTramiteEAO.setearConexion("sistemaacademico");
    Session objSession = objRegistroTramiteEAO.getSession();

    Utils objUtils = new Utils();
    objUtils.setProperties("/resources/mensajes.properties");
    MensajesGenericos objMsjGenerico = new MensajesGenericos("", "");
    boolean flag=false;

    try
    {
        /*vStrIdSolicitud,vStrCriteria,vStrProceso,vStrObservaciones*/
        flag =
objRegistroTramiteEAO.actualizarRegistro(vStrIdSolicitud,vStrProcesoSiguiente,vStrIdPersona,vStrObservacio
nes);

        if (!flag) {
            objMsjGenerico.setvStrMessage(objUtils.getMessage("",
                "fStrMessageNotfound"));
            objMsjGenerico.setvStrCode(objUtils.getObjProperties()
                .getProperty("fStrWARNING"));
        } else {

```

```

        objMsjGenerico.setvStrMessage(objUtils.getMessage("",
            "fStrMessageOK"));
        objMsjGenerico.setvStrCode(objUtils.getObjProperties()
            .getProperty("fStrSUCCESS"));
    }

}catch (Exception e){
    vStrResultado = objUtils.getMessage(vStrIdSolicitud,
        "fStrMessageErrorDatabase");
    e.printStackTrace();
    objMsjGenerico.setvStrMessage(objUtils.getMessage(vStrIdSolicitud,
        "fStrMessageErrorDatabase")
        + "----"
        + e.getLocalizedMessage());
    objMsjGenerico.setvStrCode(objUtils.getObjProperties().getProperty(
        "fStrERROR"));
}finally{
    objSession.close();
}
vStrResultado = new Gson().toJson(objMsjGenerico);
return vStrResultado;
}

/*****
 * Método           : procesoSiguiente.
 * Acceso           : público.
 * Descripción      : Consulta a la base y obtiene el proceso siguiente del tramite seleccionado, y los
retorna.
 * Autor           : Alvaro Vizueta
 * Creación        : 10/05/2015
 * Modificación    : 19/06/2015
 * *****/
public String procesoSiguiente(String vStrIdRegistro){
    String vStrResultado = "";

    ProcesoSiguienteEAO objProcesoEAO = new ProcesoSiguienteEAO();
    objProcesoEAO.setearConexion("sistemaacademico");
    Session objSession = objProcesoEAO.getSession();

    Utils objUtils = new Utils();
    objUtils.setProperties("/resources/mensajes.properties");
    MensajesGenericos objMsjGenerico = new MensajesGenericos("", "");

    try
    {
        ProcesoSiguiente objProcesoSiguiente = null;
        objProcesoSiguiente = objProcesoEAO.getProcesosiguiente(vStrIdRegistro);
        if (objProcesoSiguiente == null) {

            objMsjGenerico.setvStrMessage(objUtils.getMessage("", "fStrMessageNotFound"));

            objMsjGenerico.setvStrCode(objUtils.getObjProperties().getProperty("fStrWARNING"));

        } else {

            objMsjGenerico.setvStrMessage(objUtils.getMessage("",
                "fStrMessageOK"));
            objMsjGenerico.setvStrCode(objUtils.getObjProperties()
                .getProperty("fStrSUCCESS"));
            objMsjGenerico.addMensajeGenerico(new
MensajesGenericos("idProcesoSiguiente",objProcesoSiguiente.getProcesosByldProceso().getId()));

        }
    }catch (Exception e){
        vStrResultado = objUtils.getMessage(vStrIdRegistro,
            "fStrMessageErrorDatabase");
        e.printStackTrace();
        objMsjGenerico.setvStrMessage(objUtils.getMessage(vStrIdRegistro,

```

```

        "fStrMessageErrorDatabase")
        + "----"
        + e.getLocalizedMessage());
    objMsjGenerico.setvStrCode(objUtils.getObjProperties().getProperty(
        "fStrERROR"));
    }finally{
        objSession.close();
    }
    vStrResultado = new Gson().toJson(objMsjGenerico);
    return vStrResultado;
}

/*****
* Método           : tipoTramite.
* Acceso           : público.
* Descripción      : Consulta a la base y obtiene los tipo de tramite , y los retorna.
* Autor           : Alvaro Vizueta
* Creación        : 10/05/2015
* Modificación    : 19/06/2015
*****/
public String tipoTramite(String vStrIdRegistro)
{
    String vStrResultado = "";

    ProcesoSiguienteEAO objProcesoEAO = new ProcesoSiguienteEAO();
    objProcesoEAO.setearConexion("sistemaacademico");
    Session objSession = objProcesoEAO.getSession();

    Utils objUtils = new Utils();
    objUtils.setProperties("/resources/mensajes.properties");
    MensajesGenericos objMsjGenerico = new MensajesGenericos("", "");

    try
    {
        ProcesoSiguiente objProcesoSiguiente = null;
        objProcesoSiguiente = objProcesoEAO.getProcesosiguiente(vStrIdRegistro);
        if (objProcesoSiguiente == null) {

            objMsjGenerico.setvStrMessage(objUtils.getMessage("", "fStrMessageNotFound"));

            objMsjGenerico.setvStrCode(objUtils.getObjProperties().getProperty("fStrWARNING"));

        } else {

            objMsjGenerico.setvStrMessage(objUtils.getMessage("",
                "fStrMessageOK"));
            objMsjGenerico.setvStrCode(objUtils.getObjProperties()
                .getProperty("fStrSUCCESS"));
            objMsjGenerico.addMensajeGenerico(
                new
                MensajesGenericos("idProcesoSiguiente",objProcesoSiguiente.getId()));

        }
    }catch (Exception e){
        vStrResultado = objUtils.getMessage(vStrIdRegistro,
            "fStrMessageErrorDatabase");
        e.printStackTrace();
        objMsjGenerico.setvStrMessage(objUtils.getMessage(vStrIdRegistro,
            "fStrMessageErrorDatabase")
            + "----"
            + e.getLocalizedMessage());
        objMsjGenerico.setvStrCode(objUtils.getObjProperties().getProperty(
            "fStrERROR"));
    }finally{
        objSession.close();
    }
    vStrResultado = new Gson().toJson(objMsjGenerico);
    return vStrResultado;
}

```

```
}  
}
```

OpcionesBO.java

```
package ug.sistema.academico.bo.tramites;  
  
import java.util.ArrayList;  
  
import org.hibernate.Session;  
  
import ug.sistema.academico.eao.malla.CarreraEAO;  
import ug.sistema.academico.eao.seguridad.UsuariosEAO;  
import ug.sistema.academico.eao.tramites.EspecificacionTramiteEAO;  
import ug.sistema.academico.models.malla.Carrera;  
import ug.sistema.academico.models.seguridad.Usuarios;  
import ug.sistema.academico.models.tramites.EspecificacionTramite;  
import ug.sistema.academico.models.tramites.ProcesoPersona;  
import ug.sistema.academico.models.tramites.Procesos;  
import ug.sistema.academico.utils.ListaGenerica;  
import ug.sistema.academico.utils.Utils;  
  
import com.google.gson.Gson;  
  
import ec.edu.ug.academico.type.MensajesGenericos;  
  
public class OpcionesBO {  
  
    private Utils objUtils,objMensajes;  
    private String vStrResultado = "";  
    public OpcionesBO(){  
        objUtils = new Utils();  
        objMensajes = new Utils();  
        objUtils.setProperties("/resources/tramites.properties");  
        objMensajes.setProperties("/resources/mensajes.properties");  
    }  
  
    @SuppressWarnings("unused")  
    public String getTramite(String vStrQuery, String vStrParametros) {  
        EspecificacionTramiteEAO objTipoEAO = new EspecificacionTramiteEAO();  
        objTipoEAO.setearConexion("sistemaacademico");  
        Session objSessionR = objTipoEAO.getSession();  
  
        UsuariosEAO objUsuarioEAO =new UsuariosEAO();  
        objUsuarioEAO.setearConexion("seguridad");  
        Session objSessionUsuario = objUsuarioEAO.getSession();  
        Usuarios objUsuarios = null;  
  
        MensajesGenericos objMsjGenerico = new MensajesGenericos("", "");  
        Utils objUtils = new Utils();  
        objUtils.setProperties("/resources/constants.properties");  
  
        ArrayList<ArrayList<String>> objContenido = new ArrayList<ArrayList<String>>();  
        try {  
            ArrayList<Object> vArrayParameters =  
objUtils.getArrayListObject(vStrParametros,objUtils.getObjProperties().getProperty("fStrSeparadorParametros")  
);  
  
            ListaGenerica objListaGenerica =  
objTipoEAO.getResultListCustomAcademico(vStrQuery, vArrayParameters,false, "0", "0", "0");  
  
            ArrayList<String> arrayfilas = new ArrayList<String>();  
            ArrayList<Object> objArrayContenido =  
objListaGenerica.getArrayContenido();  
  
            for (Object objTemp : objArrayContenido) {
```

```

        arrayfilas = new ArrayList<String>();
        EspecificacionTramite objTipo =

(EspecificacionTramite) objTemp;

        arrayfilas.add(objTipo.getId());
        arrayfilas.add(objTipo.getNombre());

        objUsuarios =
objUsuarioEAO.getUsuarioByIdentificador(objTipo.getUsuarioCreacion(),"id");

        arrayfilas.add(objUsuarios.getPrimerNombre()+
"+objUsuarios.getApellidoPaterno()+" "+objUsuarios.getApellidoMaterno());

        arrayfilas.add(objTipo.getFechaCreacion().toString());

        arrayfilas.add(objTipo.getEstadoGeneral().getNombre());
        objContenido.add(arrayfilas);

        System.out.println(objTipo.getId()+objTipo.getNombre()+objTipo.getUsuarioCreacion()+objTipo.getFe
chaCreacion().toString()+objTipo.getEstadoGeneral().getNombre());
    }

    objUtils.setProperties("/resources/mensajes.properties");

    objMsjGenerico.setvStrMessage(objUtils.getMessage("", "fStrMessageOK"));

    objMsjGenerico.setvStrCode(objUtils.getObjProperties().getProperty("fStrSUCCESS"));
    } catch (Exception e) {
        System.out.println(e.toString());

    objMsjGenerico.setvStrMessage(objUtils.getMessage("", "fStrMessageErrorDatabase")+ "----"+
e.getLocalizedName());

    objMsjGenerico.setvStrCode(objUtils.getObjProperties().getProperty("fStrERROR"));
    System.out.println(e);
    } finally {
        objSessionR.close();

    }
    objMsjGenerico.setObjContenido(objContenido);
    vStrResultado = new Gson().toJson(objMsjGenerico);
    return vStrResultado;
}

public String getListPerfiles(){
    MensajesGenericos objMsjGenerico = new MensajesGenericos("", "");
    Utils objUtils = new Utils();
    objUtils.setProperties("/resources/tramites.properties");

    objMsjGenerico.setvStrMessage(objUtils.getObjProperties().getProperty("perfiles"));
    vStrResultado = new Gson().toJson(objMsjGenerico);
    System.out.println("perfiles: "+vStrResultado);
    return vStrResultado;
}

public String getProcesoPersona(String vStrQuery, String vStrParametro,
    boolean vBoolPaginador, String vIntMinimo, String vIntMaximo,
    String vIntPagina){

    String vStrResultado = "";

    ProcesoPersonaEAO objProceso = new ProcesoPersonaEAO();
    objProceso.setearConexion("sistemaacademico");
    Session objProcesosSession = objProceso.getSession();
    ProcesoPersona objProcesoPersona = null;

    CarreraEAO objCarreraEAO = new CarreraEAO();
    objCarreraEAO.setearConexion("sistemaacademico");
    Session objSessionMalla = objCarreraEAO.getSession();
    Carrera objCarrera = null;

```

```

UsuariosEAO objUsuarioEAO =new UsuariosEAO();
objUsuarioEAO.setearConexion("seguridad");
@SuppressWarnings("unused")
Session objSessionUsuario = objUsuarioEAO.getSession();
Usuarios objUsuarios = null;

MensajesGenericos objMsjGenerico = new MensajesGenericos("", "");
Utils objUtils = new Utils();
objUtils.setProperties("/resources/constants.properties");

ArrayList<ArrayList<String>> objContenido = new ArrayList<ArrayList<String>>();
try {
    ArrayList<Object> vArrayParameters = objUtils.getArrayListObject(
        vStrParametro,
        objUtils.getObjProperties().getProperty(
            "fStrSeparadorParametros"));

    ListaGenerica objListaGenerica = objProceso
        .getResultListCustomAcademico(vStrQuery,
vArrayParameters,vBoolPaginador, vIntMinimo, vIntMaximo, vIntPagina);

    ArrayList<String> arrayfilas = new ArrayList<String>();
    arrayfilas.add(String.valueOf(objListaGenerica.getvIntMinimo()));
    arrayfilas.add(String.valueOf(objListaGenerica.getvIntPagina()));
    arrayfilas.add(String.valueOf(objListaGenerica.getvIntMaximo()));
    arrayfilas.add(String.valueOf(objListaGenerica.getvNumPaginas()));
    objContenido.add(arrayfilas);

    ArrayList<Object> objArrayContenido =
objListaGenerica.getArrayContenido();

    if (!vBoolPaginador) {
        for (Object object : objArrayContenido) {
            objProcesoPersona = (ProcesoPersona) object;
            arrayfilas = new ArrayList<String>();
            arrayfilas.add(objProcesoPersona.getId());
            objUsuarios =
objUsuarioEAO.getUsuarioByIdentificador(objProcesoPersona.getIdPersona(),"id");
            arrayfilas.add(objUsuarios.getPrimerNombre()+
"+objUsuarios.getApellidoPaterno()
                "+"
"+objUsuarios.getApellidoMaterno());
            objCarrera =
objCarreraEAO.getRegistroIdentificador(objUsuarios.getCarreraDefault(),"id");

            arrayfilas.add(objCarrera.getFacultad().getNombre());

            arrayfilas.add(objProcesoPersona.getFechaCreacion().toString());

            arrayfilas.add(objProcesoPersona.getEstadoGeneral().getNombre());
            objContenido.add(arrayfilas);
        }
    } else {
        /*ArrayList<Object> arrayCertificados = objCertificadoEAO
            .quicksort(objArrayContenido);*/

        for (Object objTemp : objArrayContenido) {
            objProcesoPersona = (ProcesoPersona) objTemp;
            arrayfilas = new ArrayList<String>();
            arrayfilas.add(objProcesoPersona.getId());
            objUsuarios =
objUsuarioEAO.getUsuarioByIdentificador(objProcesoPersona.getIdPersona(),"id");
            arrayfilas.add(objUsuarios.getPrimerNombre()+
"+objUsuarios.getSegundoNombre()+ "+objUsuarios.getApellidoPaterno()
                "+"
"+objUsuarios.getApellidoMaterno());
            objCarrera =
objCarreraEAO.getRegistroIdentificador(objUsuarios.getCarreraDefault(),"id");

```

```

arrayfilas.add(objCarrera.getFacultad().getNombre());

arrayfilas.add(objProcesoPersona.getFechaCreacion().toString());

arrayfilas.add(objProcesoPersona.getEstadoGeneral().getNombre());
                                objContenido.add(arrayfilas);
                                }
                                }
                                objUtils.setProperties("/resources/mensajes.properties");
                                objMsjGenerico.setvStrMessage(objUtils.getMessage("",
                                "fStrMessageOK"));
                                objMsjGenerico.setvStrCode(objUtils.getObjProperties().getProperty(
                                "fStrSUCCESS"));
                                } catch (Exception e) {
                                System.out.println(e.toString());
                                objMsjGenerico.setvStrMessage(objUtils.getMessage("",
                                "fStrMessageErrorDatabase")
                                + "----"
                                + e.getLocalizedMessage());
                                objMsjGenerico.setvStrCode(objUtils.getObjProperties().getProperty(
                                "fStrERROR"));
                                System.out.println(e);
                                } finally {
                                objProcesosSession.close();
                                }
                                objMsjGenerico.setObjContenido(objContenido);
                                vStrResultado = new Gson().toJson(objMsjGenerico);
                                return vStrResultado;
                                }

public String creaCertificado(String vStrNombre,String vStrNemonico,
String vStrDescripcion, String vStrRuta, String vStrEstado, String
vStrProceso,
String tipoTramite, String vStrusuario){

EspecificacionTramiteEAO objTramEAO = new EspecificacionTramiteEAO();
objTramEAO.setearConexion("sistemaacademico");
Session objEspeSession = objTramEAO.getSession();

MensajesGenericos objMsjGenerico = new MensajesGenericos("", "");
Utils objUtils = new Utils();
objUtils.setProperties("/resources/mensajes.properties");
boolean flag = false;

try{
flag =
objTramEAO.insertaCertificado(vStrNombre,vStrNemonico,vStrDescripcion,vStrRuta,vStrEstado,vStrProceso,ti
poTramite,vStrusuario);

if(!flag){

objMsjGenerico.setvStrMessage(objUtils.getMessage("", "fStrMessageNotFound"));

objMsjGenerico.setvStrCode(objUtils.getObjProperties().getProperty("fStrWARNING"));
} else {
objMsjGenerico.setvStrMessage(objUtils.getMessage("", "fStrMessageOK"));

objMsjGenerico.setvStrCode(objUtils.getObjProperties().getProperty("fStrSUCCESS"));
}

} catch (Exception e){
vStrResultado = objUtils.getMessage(vStrusuario, "fStrMessageErrorDatabase");
e.printStackTrace();

objMsjGenerico.setvStrMessage(objUtils.getMessage(vStrusuario, "fStrMessageErrorDatabase")+ "----"
"+ e.getLocalizedMessage());
objMsjGenerico.setvStrCode(objUtils.getObjProperties().getProperty("fStrERROR"));
}finally{

```

```

        objEspeSession.close();
    }
    vStrResultado = new Gson().toJson(objMsjGenerico);
    return vStrResultado;
}

    public String creaProceso(String nombre, String estado, String usuario){

        ProcesosEAO objProcesoEAO = new ProcesosEAO();
        objProcesoEAO.setearConexion("sistemaacademico");
        Session objEspeSession = objProcesoEAO.getSession();

        MensajesGenericos objMsjGenerico = new MensajesGenericos("", "");
        Utils objUtils = new Utils();
        objUtils.setProperties("/resources/mensajes.properties");
        boolean flag = false;

        try{
            flag = objProcesoEAO.insertaProceso(nombre,"E8B0B0DD-A101-4C4B-975D-
2834B4808143",estado,usuario);

            if(!flag){

                objMsjGenerico.setvStrMessage(objUtils.getMessage("", "fStrMessageNotFound"));
                objMsjGenerico.setvStrCode(objUtils.getObjProperties().getProperty("fStrWARNING"));
            } else {

                objMsjGenerico.setvStrMessage(objUtils.getMessage("", "fStrMessageOK"));
                objMsjGenerico.setvStrCode(objUtils.getObjProperties().getProperty("fStrSUCCESS"));
            }

        } catch (Exception e){
            vStrResultado = objUtils.getMessage(nombre, "fStrMessageErrorDatabase");
            e.printStackTrace();

            objMsjGenerico.setvStrMessage(objUtils.getMessage(nombre, "fStrMessageErrorDatabase")+ "---"+
e.getLocalizedName());

            objMsjGenerico.setvStrCode(objUtils.getObjProperties().getProperty("fStrERROR"));
        } finally{
            objEspeSession.close();
        }
        vStrResultado = new Gson().toJson(objMsjGenerico);
        return vStrResultado;
    }

    public String creaSolicitud(String vStrNombre,String vStrNemonico,
        String vStrDescripcion, String vStrEstado, String vStrProceso,String
maxdias,
        String tipoTramite, String vStrusuario){

        EspecificacionTramiteEAO objTramEAO = new EspecificacionTramiteEAO();
        objTramEAO.setearConexion("sistemaacademico");
        Session objEspeSession = objTramEAO.getSession();

        MensajesGenericos objMsjGenerico = new MensajesGenericos("", "");
        Utils objUtils = new Utils();
        objUtils.setProperties("/resources/mensajes.properties");
        boolean flag = false;

        try{
            flag =
objTramEAO.insertaSolicitud(vStrNombre,vStrNemonico,vStrDescripcion,vStrEstado,vStrProceso,maxdias,tipo
Tramite,vStrusuario);

            if(!flag){

```

```

objMsjGenerico.setvStrMessage(objUtils.getMessage("", "fStrMessageNotFound"));

objMsjGenerico.setvStrCode(objUtils.getObjProperties().getProperty("fStrWARNING"));
    } else {
        objMsjGenerico.setvStrMessage(objUtils.getMessage("", "fStrMessageOK"));

objMsjGenerico.setvStrCode(objUtils.getObjProperties().getProperty("fStrSUCCESS"));
    }

} catch (Exception e){
    vStrResultado = objUtils.getMessage(vStrusuario, "fStrMessageErrorDatabase");
    e.printStackTrace();

    objMsjGenerico.setvStrMessage(objUtils.getMessage(vStrusuario, "fStrMessageErrorDatabase")+ "----
"+ e.getLocalizedMessage());
    objMsjGenerico.setvStrCode(objUtils.getObjProperties().getProperty("fStrERROR"));
} finally{
    objEspeSession.close();
}
vStrResultado = new Gson().toJson(objMsjGenerico);
return vStrResultado;
}

public String getProcesos(String vStrQuery, String vStrParametro,
    boolean vBoolPaginador, String vIntMinimo, String vIntMaximo,
    String vIntPagina){

    String vStrResultado = "";

    ProcesosEAO objProcesoEAO= new ProcesosEAO();
    objProcesoEAO.setearConexion("sistemaacademico");
    Session objProcesosSession = objProcesoEAO.getSession();
    Procesos objProcesos = null;

    UsuariosEAO objUsuarioEAO =new UsuariosEAO();
    objUsuarioEAO.setearConexion("seguridad");
    @SuppressWarnings("unused")
    Session objSessionUsuario = objUsuarioEAO.getSession();
    Usuarios objUsuarios = null;

    MensajesGenericos objMsjGenerico = new MensajesGenericos("", "");
    Utils objUtils = new Utils();
    objUtils.setProperties("/resources/constants.properties");

    ArrayList<ArrayList<String>> objContenido = new ArrayList<ArrayList<String>>();
    try {
        ArrayList<Object> vArrayParameters = objUtils.getArrayListObject(
            vStrParametro,
            objUtils.getObjProperties().getProperty(
                "fStrSeparadorParametros"));

        ListaGenerica objListaGenerica = objProcesoEAO
            .getResultListCustomAcademico(vStrQuery,
vArrayParameters,vBoolPaginador, vIntMinimo, vIntMaximo, vIntPagina);

        ArrayList<String> arrayfilas = new ArrayList<String>();
        arrayfilas.add(String.valueOf(objListaGenerica.getvIntMinimo()));
        arrayfilas.add(String.valueOf(objListaGenerica.getvIntPagina()));
        arrayfilas.add(String.valueOf(objListaGenerica.getvIntMaximo()));
        arrayfilas.add(String.valueOf(objListaGenerica.getvNumPaginas()));
        objContenido.add(arrayfilas);

        ArrayList<Object> objArrayContenido =
objListaGenerica.getArrayContenido();

        if (!vBoolPaginador) {
            for (Object object : objArrayContenido) {
                objProcesos = (Procesos) object;
            }
        }
    }
}

```

```

        arrayfilas = new ArrayList<String>();
        arrayfilas.add(objProcesos.getId());
        arrayfilas.add(objProcesos.getDescripcion());
        objUsuarios =
objUsuarioEAO.getUsuarioByIdentificador(objProcesos.getProcesoPersona().getIdPersona(), "id");
        arrayfilas.add(objUsuarios.getPrimerNombre()+
"+objUsuarios.getSegundoNombre()+ " "+objUsuarios.getApellidoPaterno()
        +"
"+objUsuarios.getApellidoMaterno());

        arrayfilas.add(objProcesos.getEstadoGeneral().getNombre());

        arrayfilas.add(objProcesos.getFechaCreacion().toString());
        objContenido.add(arrayfilas);
    }
} else {
    /*ArrayList<Object> arrayCertificados = objCertificadoEAO
        .quicksort(objArrayContenido);*/

    for (Object objTemp : objArrayContenido) {
        objProcesos = (Procesos) objTemp;
        arrayfilas = new ArrayList<String>();
        arrayfilas.add(objProcesos.getId());
        arrayfilas.add(objProcesos.getDescripcion());
        objUsuarios =
objUsuarioEAO.getUsuarioByIdentificador(objProcesos.getProcesoPersona().getIdPersona(), "id");
        arrayfilas.add(objUsuarios.getPrimerNombre()+
"+objUsuarios.getSegundoNombre()+ " "+objUsuarios.getApellidoPaterno()
        +"
"+objUsuarios.getApellidoMaterno());

        arrayfilas.add(objProcesos.getEstadoGeneral().getNombre());

        arrayfilas.add(objProcesos.getFechaCreacion().toString());
        objContenido.add(arrayfilas);
    }
}
objUtils.setProperties("/resources/mensajes.properties");
objMsjGenerico.setvStrMessage(objUtils.getMessage("",
    "fStrMessageOK"));
objMsjGenerico.setvStrCode(objUtils.getObjProperties().getProperty(
    "fStrSUCCESS"));
} catch (Exception e) {
    System.out.println(e.toString());
    objMsjGenerico.setvStrMessage(objUtils.getMessage("",
        "fStrMessageErrorDatabase")
        + "----"
        + e.getLocalisedMessage());
    objMsjGenerico.setvStrCode(objUtils.getObjProperties().getProperty(
        "fStrERROR"));
    System.out.println(e);
} finally {
    objProcesosSession.close();
}
objMsjGenerico.setObjContenido(objContenido);
vStrResultado = new Gson().toJson(objMsjGenerico);
return vStrResultado;
}
}

```

EnviarCorreo.java

```
package ug.sistema.academico.bo.tramites;

import java.util.*;

import javax.mail.*;
import javax.mail.internet.*;
import javax.activation.*;

import com.google.gson.Gson;

import ec.edu.ug.academico.type.MensajesGenericos;
import ug.sistema.academico.utils.Utils;

public class EnviarCorreo {
    private Utils objUtils,objMensajes;
    private MensajesGenericos objMsjGenericos;
    private String vStrResultado;

    public EnviarCorreo(){
        objUtils = new Utils();
        objMensajes = new Utils();
        objUtils.setProperties("/resources/tramites.properties");
        objMensajes.setProperties("/resources/mensajes.properties");
        objMsjGenericos = new MensajesGenericos("", "");
    }

    public String enviarMail(String to){
    try
    {
        // se obtiene el objeto Session. La configuración es para
        // una cuenta de gmail.
        Properties props = this.objUtils.getObjProperties();
        Session session = Session.getDefaultInstance(props);
        // session.setDebug(true);

        // Se compone la parte del texto
        BodyPart texto = new MimeBodyPart();
        texto.setText("Texto del mensaje");

        // Se compone el adjunto con la imagen
        BodyPart adjunto = new MimeBodyPart();
        adjunto.setDataHandler(
            new DataHandler(new FileDataSource("B:/pdf/prueba.pdf"));
        adjunto.setFileName("prueba.pdf");

        // Una MultiParte para agrupar texto e imagen.
        MimeMultipart multiParte = new MimeMultipart();
        multiParte.addBodyPart(texto);
        multiParte.addBodyPart(adjunto);

        // Se compone el correo, dando to, from, subject y el
        // contenido.
        MimeMessage message = new MimeMessage(session);
        message.setFrom(new
        InternetAddress(String.valueOf(this.objUtils.getObjProperties().get("mail.smtp.user"))));
        message.addRecipient(Message.RecipientType.TO, new InternetAddress(to));
        message.setSubject("Hola");
        message.setContent(multiParte);

        // Se envia el correo.
        Transport t = session.getTransport("smtp");
        t.connect(String.valueOf(this.objUtils.getObjProperties().get("mail.smtp.host")),
```

```

        String.valueOf(this.objUtils.getObjProperties().get("mail.smtp.user")),
        String.valueOf(this.objUtils.getObjProperties().get("mail.smtp.password"))));
t.sendMessage(message, message.getAllRecipients());
t.close();
System.out.print("OK");
objMsjGenericos.setvStrMessage(objMensajes.getMessage("", "fStrMessageOK"));

objMsjGenericos.setvStrCode(objMensajes.getObjProperties().getProperty("fStrSUCCESS"));
    }
    catch (Exception e)
    {
        e.printStackTrace();
        objMsjGenericos.setvStrCode(objMensajes.getObjProperties().getProperty("fStrERROR"+"
+e));
    }
    vStrResultado = new Gson().toJson(objMsjGenericos);
    return vStrResultado;
}
}

```

ProcesoPersonaEAO.java

```

package ug.sistema.academico.bo.tramites;

import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;

import ug.sistema.academico.eao.general.ConexionDefault;
import ug.sistema.academico.models.tramites.ProcesoPersona;

public class ProcesoPersonaEAO extends ConexionDefault<ProcesoPersona,Number>{

    private ProcesoPersona ObjProPersona;
    private Session objSession;

    public ProcesoPersona getProcesoPersona(String estado){
        objSession = getSession();
        ObjProPersona = (ProcesoPersona) objSession
            .createCriteria(ProcesoPersona.class)
            .add(Restrictions.eqOrIsNull("estadoGeneral.nombre", estado))
            .uniqueResult();

        return ObjProPersona;
    }
}

```

ProcesoEAO.java

```

package ug.sistema.academico.eao.tramites;

import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;

import ug.sistema.academico.eao.general.ConexionDefault;
import ug.sistema.academico.models.tramites.Procesos;

public class ProcesoEAO extends ConexionDefault<Procesos,Number>{

    Session objSession;
    public ProcesoEAO() {
        // TODO Auto-generated constructor stub
    }
}

```

```

        public Procesos getProcesoSiguiente(String idProceso){
            objSession = getSession();
            Procesos objProcesos= (Procesos)
            objSession.createCriteria(Procesos.class).add(Restrictions.eq("Procesos",
idProceso)).uniqueResult();
            System.out.println("valor idSiguiente:" + objProcesos.toString());
            return objProcesos;
        }
    }
}

```

ProcesosEAO.java

```

package ug.sistema.academico.bo.tramites;

```

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

```

```

import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;

```

```

import ug.sistema.academico.eao.general.ConexionDefault;
import ug.sistema.academico.models.tramites.EstadoGeneral;
import ug.sistema.academico.models.tramites.ProcesoPersona;
import ug.sistema.academico.models.tramites.Procesos;
import ug.sistema.academico.utils.Utilis;

```

```

public class ProcesosEAO extends ConexionDefault<Procesos,Number>{

```

```

    private ProcesoPersona objProceso;
    private EstadoGeneral objEstadoGeneral;
    private boolean flag;
    private Utilis objUtilis,objMensajes;
    private Connection objConnection;
    private Statement stmt = null;
    private Session objSession;

```

```

    public ProcesosEAO(){
        objUtilis = new Utilis();
        objMensajes = new Utilis();
        objUtilis.setProperties("/resources/tramites.properties");
        objMensajes.setProperties("/resources/mensajes.properties");
    }

```

```

    public boolean insertaProceso(String vStrNombre,String idUserProceso,String vStrEstado,String
usuario) throws ClassNotFoundException, SQLException{

```

```

        objSession = getSession();
        objProceso = (ProcesoPersona)
objSession.createCriteria(ProcesoPersona.class)
.add(Restrictions.eq("idPersona",idUserProceso)).uniqueResult();

```

```

objEstadoGeneral = (EstadoGeneral)objSession.createCriteria(EstadoGeneral.class)
.add(Restrictions.eq("nombre", vStrEstado)).uniqueResult();

```

```

        flag = false;
        objConnection = conexionBase();
        stmt = objConnection.createStatement();
        String

```

```

sql = "INSERT INTO Tramites.Procesos
(Id,Descripcion,IdProcesoPersona,UsuarioCreacion,UsuarioModificacion,FechaCreacion,FechaModificacion,Est
ado)"

```

```

        + "values(NEWID())"
        + "," + vStrNombre
        + "," + objProceso.getId()

```

```

        +""+usuario
        +",null,GETDATE(),null,"+objEstadoGeneral.getId()
        +""";
        System.out.println(sql);
        stmt.executeUpdate(sql);
        flag = true;
        return flag;
    }

    public Connection conexionBase() throws ClassNotFoundException, SQLException{
        Class.forName(String.valueOf(this.objUtils.getObjProperties().get("forname")));
        return objConnection =
        DriverManager.getConnection(String.valueOf(this.objUtils.getObjProperties().get("url")));
    }
}

```

DetalleRegistroEAO.java

```

package ug.sistema.academico.eao.tramites;

import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;

import ug.sistema.academico.eao.general.ConexionDefault;
import ug.sistema.academico.models.tramites.DetalleRegistro;

public class DetalleRegistroEAO extends ConexionDefault<DetalleRegistro,Number>{

    public DetalleRegistro getRegistroIdentificador(String vStrIdRegistro, String vStrCampo ){

        Session objSession = getSession();
        DetalleRegistro objRegistroTramites = (DetalleRegistro) objSession
        .createCriteria(DetalleRegistro.class)
        .add(Restrictions.eq(vStrCampo, vStrIdRegistro))
        .uniqueResult();

        return objRegistroTramites;

    }

}

```

EspecificacionTramiteEAO.java

```

package ug.sistema.academico.eao.tramites;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;

import ug.sistema.academico.eao.general.ConexionDefault;
import ug.sistema.academico.models.tramites.EspecificacionTramite;
import ug.sistema.academico.models.tramites.EstadoGeneral;
import ug.sistema.academico.models.tramites.TipoTramite;
import ug.sistema.academico.utils.Utilis;

public class EspecificacionTramiteEAO extends ConexionDefault<EspecificacionTramite,Number> {

    private TipoTramite objTipoTramite;
    private EstadoGeneral objEstadoGeneral;
    private boolean flag;
    private Utilis objUtils,objMensajes;
    private Connection objConnection;
}

```

```

private Statement stmt = null;
private Session objSession;

public EspecificacionTramiteEAO(){
    objUtils = new Utils();
    objMensajes = new Utils();
    objUtils.setProperties("/resources/tramites.properties");
    objMensajes.setProperties("/resources/mensajes.properties");
}

public boolean insertaCertificado(String vStrNombre,String vStrNemonico,String
vStrDescripcion,String vStrRuta,
String vStrEstado,String vStrProceso,String tipoTramite,String vStrusuario)
throws ClassNotFoundException, SQLException{

    objSession = getSession();
    objTipoTramite = (TipoTramite) objSession.createCriteria(TipoTramite.class)
        .add(Restrictions.eq("nombre", tipoTramite)).uniqueResult();

    objEstadoGeneral = (EstadoGeneral)
objSession.createCriteria(EstadoGeneral.class)
        .add(Restrictions.eq("nombre", vStrEstado)).uniqueResult();

    flag = false;
    objConnection = conexionBase();
    stmt = objConnection.createStatement();
    String sql = "Insert into Tramites.EspecificacionTramite "
+
"(Id,Nombre,IdTipoTramite,Nemonico,UsuarioCreacion,UsuarioModificacion,FechaCreacion,FechaModificacion,
Estado,RutaArchivo,Texto,IdProceso)"
+ "values(NEWID()"
+ ","+ vStrNombre
+ ","+objTipoTramite.getId()
+ ","+vStrNemonico
+ ","+vStrusuario
+ ",null,GETDATE(),null,"+objEstadoGeneral.getId()
+ ","+vStrRuta
+ ","+vStrDescripcion
+ ","+"EACC7BC7-937E-4888-AD56-C3E40D7DB9EC"+");"

    System.out.println(sql);
    stmt.executeUpdate(sql);
    flag = true;
    return flag;
}

public boolean insertaSolicitud(String vStrNombre,String vStrNemonico,String vStrDescripcion,
String vStrEstado,String vStrProceso,String maxDias,String tipoTramite,String
vStrusuario) throws ClassNotFoundException, SQLException{

    objSession = getSession();
    objTipoTramite = (TipoTramite) objSession.createCriteria(TipoTramite.class)
        .add(Restrictions.eq("nombre", tipoTramite)).uniqueResult();

    objEstadoGeneral = (EstadoGeneral)
objSession.createCriteria(EstadoGeneral.class)
        .add(Restrictions.eq("nombre", vStrEstado)).uniqueResult();

    flag = false;
    objConnection = conexionBase();
    stmt = objConnection.createStatement();
    String sql = "Insert into Tramites.EspecificacionTramite "
+
"(Id,Nombre,IdTipoTramite,Nemonico,UsuarioCreacion,UsuarioModificacion,FechaCreacion,FechaModificacion,
Estado,Texto,IdProceso,MaxDias)"
+ "values(NEWID()"

```

```

        +";"+ vStrNombre
        +";"+objTipoTramite.getId()
        +";"+vStrNemonico
        +";"+vStrusuario
        +";,null,GETDATE(),null,"+objEstadoGeneral.getId()
        +";"+vStrDescripcion
        +";"+"EACC7BC7-937E-4888-AD56-C3E40D7DB9EC"
        +";"+maxDias+");";
        System.out.println(sql);
        stmt.executeUpdate(sql);
        flag = true;
        return flag;
    }

    public Connection conexionBase() throws ClassNotFoundException, SQLException{
        Class.forName(String.valueOf(this.objUtils.getObjProperties().get("forname")));
        return objConnection =
        DriverManager.getConnection(String.valueOf(this.objUtils.getObjProperties().get("url")));
    }
}

```

ProcesoEAO.java

```

package ug.sistema.academico.eao.tramites;

import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;

import ug.sistema.academico.eao.general.ConexionDefault;
import ug.sistema.academico.models.tramites.Procesos;

public class ProcesoEAO extends ConexionDefault<Procesos,Number>{

    Session objSession;
    public ProcesoEAO() {
        // TODO Auto-generated constructor stub
    }

    public Procesos getProcesoSiguiente(String idProceso){

        objSession = getSession();
        Procesos objProcesos= (Procesos)
        objSession.createCriteria(Procesos.class).add(Restrictions.eq("Procesos",
idProceso)).uniqueResult();
        System.out.println("valor idSiguiente:" + objProcesos.toString());
        return objProcesos;
    }
}

```

ProcesoSiguienteEAO.java

```

package ug.sistema.academico.eao.tramites;

import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;

import ug.sistema.academico.models.tramites.ProcesoSiguiente;
import ug.sistema.academico.models.tramites.RegistroTramites;
import ug.sistema.academico.eao.general.ConexionDefault;

public class ProcesoSiguienteEAO extends ConexionDefault<ProcesoSiguiente,Number>{
    private Session objSession = getSession();
}

```

```

public ProcesoSiguiente getProcesosiguiente(String vStrIdRegistro){
    objSession = getSession();

    RegistroTramites objRegistroTramites = (RegistroTramites) objSession
        .createCriteria(RegistroTramites.class)
        .add(Restrictions.eq("id", vStrIdRegistro))
        .uniqueResult();
    System.out.println("IdProceso:"+objRegistroTramites.getProcesos().getId());

    ProcesoSiguiente objProcesoSiguiente = (ProcesoSiguiente) objSession
        .createCriteria(ProcesoSiguiente.class)

    .add(Restrictions.eq("procesosByldProcesoOrigen.id",objRegistroTramites.getProcesos().getId()))
        .uniqueResult();

    System.out.println("IdProcesoSiguiente:"+objProcesoSiguiente.getProcesosByldProceso().getId());
    return objProcesoSiguiente;
}

public RegistroTramites getProcesoActual(String vStrIdRegistro){
    objSession = getSession();

    RegistroTramites objRegistroTramites = (RegistroTramites) objSession
        .createCriteria(RegistroTramites.class)
        .add(Restrictions.eq("id", vStrIdRegistro))
        .uniqueResult();
    return objRegistroTramites;
}
}
}

```

RegistroTramitesEAO.java

```

package ug.sistema.academico.eao.tramites;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;

import ug.sistema.academico.eao.general.ConexionDefault;
import ug.sistema.academico.models.tramites.DetalleRegistro;
import ug.sistema.academico.models.tramites.RegistroTramites;
import ug.sistema.academico.utils.Utilis;

public class RegistroTramitesEAO extends ConexionDefault<RegistroTramites,Number>{

    private RegistroTramites objRegistroTramites;
    private boolean flag;
    private Utilis objUtilis,objMensajes;
    private Connection objConnection;
    private Statement stmt = null;
    private Session objSession;

    public RegistroTramitesEAO(){
        objUtilis = new Utilis();
        objMensajes = new Utilis();
        objUtilis.setProperties("/resources/tramites.properties");
        objMensajes.setProperties("/resources/mensajes.properties");
    }
}

```

```

    public RegistroTramites getRegistroIdentificador(String vStrIdRegistro, String vStrCampo ){
        //System.out.println("Id: "+vStrIdRegistro+" Campo: "+vStrCampo);
        objSession = getSession();
        objRegistroTramites = (RegistroTramites) objSession
            .createCriteria(RegistroTramites.class)
            .add(Restrictions.eq(vStrCampo, vStrIdRegistro))
            .uniqueResult();
        return objRegistroTramites;
    }

    public boolean actualizarRegistro(String vStrIdSolicitud,String vStrProcesoSiguiente,String
    vStrIdPersona, String vStrObservaciones) throws ClassNotFoundException, SQLException{
        objSession = getSession();

        DetalleRegistro registro = (DetalleRegistro)
        objSession.createCriteria(DetalleRegistro.class)
            .add(Restrictions.eq("registroTramites.id",
vStrIdSolicitud)).uniqueResult();

        flag = false;
        objConnection = conexionBase();
        stmt = objConnection.createStatement();
        String sql = "Insert into Tramites.LogTramites values(NEWID()
            +",""+ vStrIdSolicitud
            +",""+registro.getRegistroTramites().getProcesos().getId()
            +",""+vStrIdPersona
            +",""+vStrObservaciones
            +","GETDATE())";

        System.out.println(sql);
        stmt.executeUpdate(sql);

        String update = "Update Tramites.RegistroTramites set IdProceso =
        "+""+vStrProcesoSiguiente+"""+
            "where id = "+""+vStrIdSolicitud+""";

        System.out.println(update);
        stmt.executeUpdate(update);
        flag = true;
        return flag;
    }

    public boolean controlLog(String IdRegistro,String IdProceso) throws ClassNotFoundException,
    SQLException{
        objSession = getSession();
        flag = false;
        int contador=0;
        String query = "select * from Tramites.LogTramites "
            + "where IdRegistroTramite="+IdRegistro+""
            + "and IdProceso="+IdProceso+"";

        objConnection = conexionBase();
        stmt = objConnection.createStatement();

        ResultSet rs = stmt.executeQuery(query);
        if(!rs.isNull()){
            flag = true;
            contador++;
            System.out.println(query + " "+Integer.valueOf(contador));}
        return flag;
    }

    public Connection conexionBase() throws ClassNotFoundException, SQLException{
        Class.forName(String.valueOf(this.objUtils.getObjProperties().get("forname")));
        return objConnection =
        DriverManager.getConnection(String.valueOf(this.objUtils.getObjProperties().get("url")));
    }
}

```

TipoTramiteEAO.java

```
package ug.sistema.academico.eao.tramites;

import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;

import ug.sistema.academico.eao.general.ConexionDefault;
import ug.sistema.academico.models.tramites.TipoTramite;
import ug.sistema.academico.utils.Utilis;

public class TipoTramiteEAO extends ConexionDefault<TipoTramite,Number>{

    private TipoTramite objTipotramite;
    private Utilis objUtilis,objMensajes;
    private Session objSession;

    public TipoTramiteEAO(){
        objUtilis = new Utilis();
        objMensajes = new Utilis();
        objUtilis.setProperties("/resources/tramites.properties");
        objMensajes.setProperties("/resources/mensajes.properties");
    }

    public TipoTramite getTipoTramite(String nombre, String campo){
        objSession = getSession();
        objTipotramite = (TipoTramite) objSession
            .createCriteria(TipoTramite.class)
            .add(Restrictions.eq(campo, nombre))
            .uniqueResult();
        return objTipotramite;
    }
}
```



UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMATICAS Y FISICAS
CARRERA DE INGENIERIA EN SISTEMAS
COMPUTACIONALES

“DISEÑO Y DESARROLLO DE LA BANDEJA DE TRABAJO QUE PROCESARA SOLICITUDES Y CERTIFICADOS PARA EL PROTOTIPO DE SISTEMA ACADEMICO DE LA UNIVERSIDAD DE GUAYAQUIL.”.

MANUAL TECNICO

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS COMPUTACIONALES

AUTOR: ALVARO LUIS VIZUETA LOJAN

TUTOR: ING. MARY CABANILLA

GUAYAQUIL – ECUADOR

2015

MANUAL TECNICO

Contenido

MANUAL TECNICO.....	2
CONFIGUACION DE HERRAMIENTAS DE DESARROLLO	8
DESCARGA Y CONFIGURACION DE ECLIPSE Y ZK	8
CONFIGURACION ZK	11
INSTALACION DE ZK.....	12
DESCARGAR Y CONFIGURAR APACHE CXF.....	15
DESCARGA DE APACHE TOMCAT Y CONFIGURACION DE SERVIDORES WS Y WEB EN ECLIPSE	19
CREACION Y CONFIGURACION DE SERVERWEB	22
CREACION Y CONFIGURACION DE SERVERWS.....	25
INSTALACION DE HIBERNATE PARA MAPEO DE BASE DE DATOS	27
ABRIR PERSPECTIVA EN HIBERNATE	28
INSTALACIÓN DE BASE DE DATOS	29
RESTAURACIÓN DE BASE DE DATOS	40
CONFIGURACION DE ARCHIVOS PARA MAPEO DE BASE SQL-SERVER44	

Indicé de Gráficos

Gráfico 1.- Pagina descarga Eclipse	8
Gráfico 2.- Proceso descarga Eclipse	8
Gráfico 3.- Proceso de descargar y almacenaje de archivo eclipse	9
Gráfico 4.- Carpeta contenedora eclipse.....	9
Gráfico 5.- Ejecución de Eclipse	10
Gráfico 6.- Pantalla principal Eclipse.....	10
Gráfico 7.- Pagina web ZK.....	11
Gráfico 8.- registro Zk	11
Gráfico 9.- Ingreso a Eclipse Market Place	12
Gráfico 10.- Búsqueda de Zk Studio	12
Gráfico 11.- Confirmación de Instalación	13
Gráfico 12.- Página de Bienvenida de Zk.....	13
Gráfico 13.- Activación de Zk Studio	14
Gráfico 14.- Ingreso de usuario y contraseña	14
Gráfico 15.- Pagina Apache CXF	15
Gráfico 16.- Pagina descarga Apache CXF	15
Gráfico 17.- Descarga y almacenamiento Apache CXF	16
Gráfico 18.- Creación de Carpeta Apache CXF	16
Gráfico 19.- Conuiguracion de Apache CXF	17
Gráfico 20.- Pasos de seleccion carpeta Apache CXF.....	17
Gráfico 21.- Selección de opción Apache CXF	18
Gráfico 22.- Pagina descargar Apache Tomcat.....	19
Gráfico 23.- Archivo y carpeta descomprimida apache tomcat.....	19
Gráfico 24.- Creación de carpetas para servidores.	20

Gráfico 25.- Carpetas de Servidores WEB y WS	20
Gráfico 26.- cambio de perspectiva a servidores	20
Gráfico 27.- Seleccion de Perspectiva Servers	21
Gráfico 28.- Creación de Servidores	22
Gráfico 29.- Selección del tipo de Servidor	22
Gráfico 30.- Creación y selección de SERVERWEB	23
Gráfico 31.- Configuración de servidor creado.....	23
<i>Gráfico 32.- Configuración y parametrización serverweb</i>	<i>24</i>
Gráfico 33.- Pasos de Creacion y configuracion SERVERWS	25
Gráfico 34.-Configuración y parametrización SERVERWS	26
Gráfico 35.- Instalación mediante MarketPlace	27
Gráfico 36.- Búsqueda de HIBERNATE	27
Gráfico 37.- Confirmación de Instalación	28
Gráfico 38.- Abrir perspectiva HIBERNATE	28
Gráfico 39.- Selección de perspectiva HIBERNATE	29
Gráfico 40.- Inicio de Instalación	29
Gráfico 41.- Aceptación de Instalación.....	30
Gráfico 42.- Centro de instalación.....	30
Gráfico 43.- Selección de Nueva instalación.....	31
Gráfico 44.- Reglas Auxiliares.....	31
Gráfico 45.- Ingreso de clave de producto	32
Gráfico 46.- Aceptación de términos de licencia	32
Gráfico 47.- Verificación de Actualizaciones	33
Gráfico 48.- Archivos de configuración	33
Gráfico 49.- Reglas Auxiliares.....	34

Gráfico 50.- Rol de instalación	34
Gráfico 51.- Selección de Características	35
Gráfico 52.- Selección de Características	35
Gráfico 53.- Reglas de Instalación	36
Gráfico 54.- Configuración de Instancia	36
Gráfico 55.-Requisitos	37
Gráfico 56.- Configuración del Servidor	37
Gráfico 57.- Configuración motor base de Datos	38
Gráfico 58.- Informe de Errores	38
Gráfico 59.- Reglas de Instalación	39
Gráfico 60.-Aceptación de instalación	39
Gráfico 61.- Operación Completa.....	40
Gráfico 62.- Archivo de Restauración	40
Gráfico 63.- Proceso de Restauración	41
Gráfico 64.- Selección de archivo BACK.....	41
Gráfico 65.- Ventana de Selección.....	42
Gráfico 66.- Selección archivo de respaldo.....	42
Gráfico 67.- Archivo seleccionado.....	43
Gráfico 68.- Restauración de BD	43
Gráfico 69.- Selección de archivo de configuración	44
Gráfico 70.- Selección de la ruta para guardar el archivo de configuración.....	45
Gráfico 71.- Configuración de conexión a la Base de Datos	45
Gráfico 72.- configuración de Console Configuration	46
Gráfico 73.- Console Configuration	46
Gráfico 74.-Pre visualización de conexión a la Base.....	47

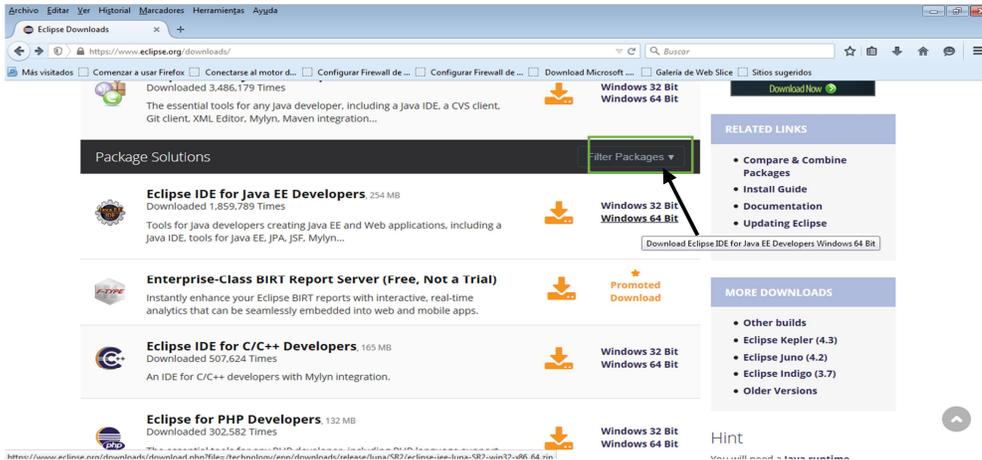
Gráfico 75.- Configuración de Ingeniería Inversa.....	47
Gráfico 76.- Selección de ruta y nombre de archivo Reverse Engineering.....	48
Gráfico 77.- Selección de las tablas para el Mapeo	48
Gráfico 78.- Generación de código Hibernate	49
Gráfico 79.- Configuración para la generación de código.....	49
Gráfico 80.- Selección de archivo reveng.xml	50
Gráfico 81.- Configuración de Exporters	50
Gráfico 82.- Generación de código Hibernate	51
Gráfico 83.- Verificación de Clases generadas	51

CONFIGUACION DE HERRAMIENTAS DE DESARROLLO

DESCARGA Y CONFIGURACION DE ECLIPSE Y ZK

Vamos a la página de descarga de Eclipse:
<https://www.eclipse.org/downloads/>

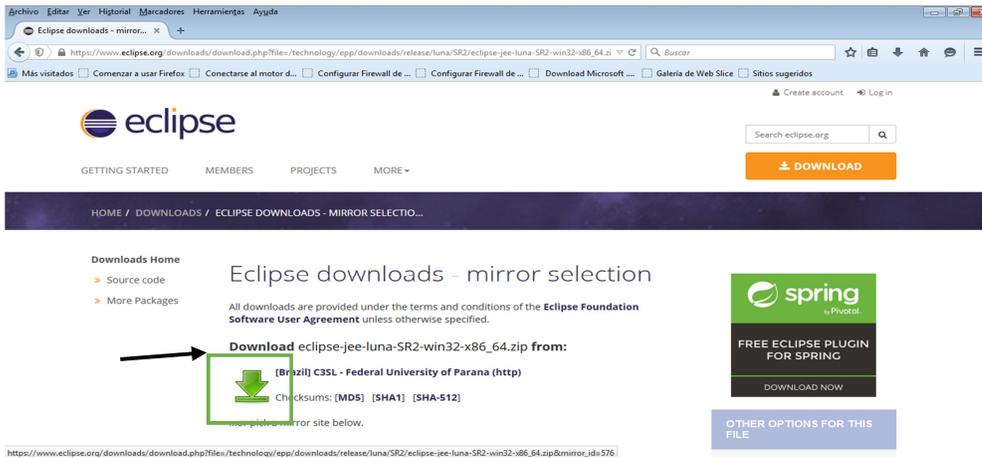
Gráfico 1.- Pagina descarga Eclipse



Elaboración: Álvaro Vizueta
Fuente: [ww.eclipse.org](http://www.eclipse.org)

Seleccionamos la versión que necesitamos según la versión de Windows que tengamos (32bit – 64bit).

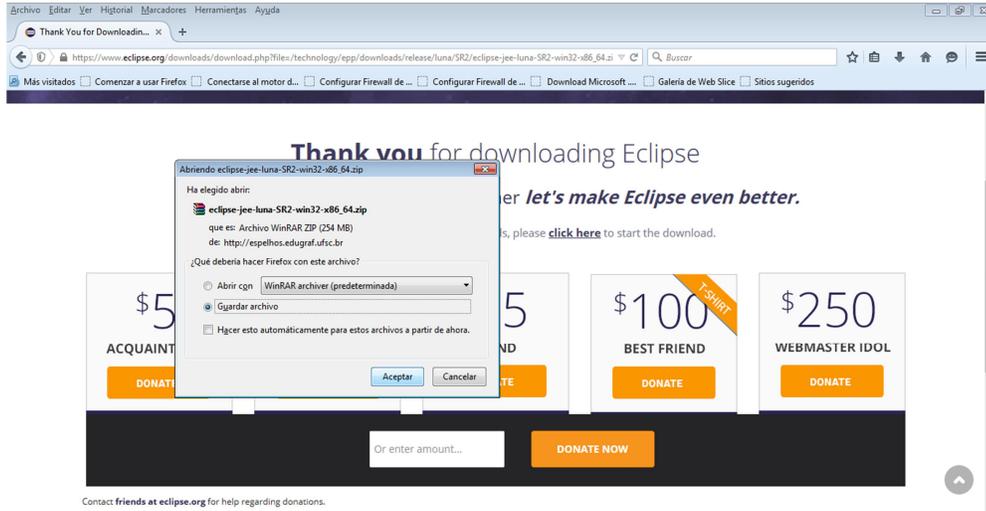
Gráfico 2.- Proceso descarga Eclipse



Elaboración: Álvaro Vizueta
Fuente: ww.eclipse.org

Damos clic justo donde muestra la imagen.

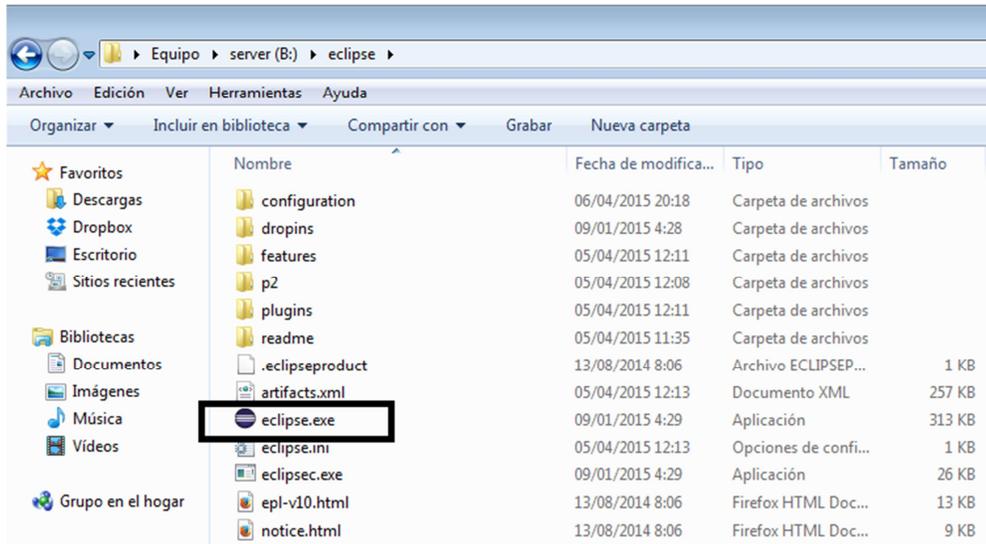
Gráfico 3.- Proceso de descargar y almacenaje de archivo eclipse



Elaboración: Álvaro Vizueta
Fuente: ww.eclipse.org

Escogemos la opción guardar y damos click en aceptar.

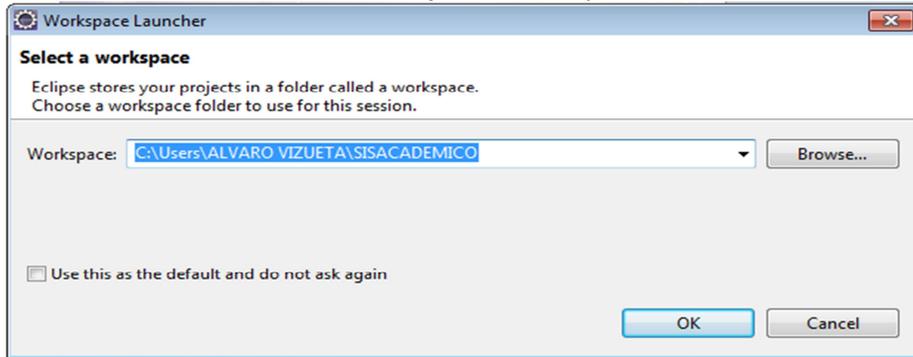
Gráfico 4.- Carpeta contenedora eclipse



Elaboración: Álvaro Vizueta
Fuente: ww.eclipse.org

Una vez que este descargado completamente, creamos una carpeta llamada **eclipse**, en la carpeta que creamos descomprimos los que hemos descargado y ejecutamos el archivo que dice **eclipse.exe** (para que el programa se ejecute es necesario tener instalado el JDK-7)

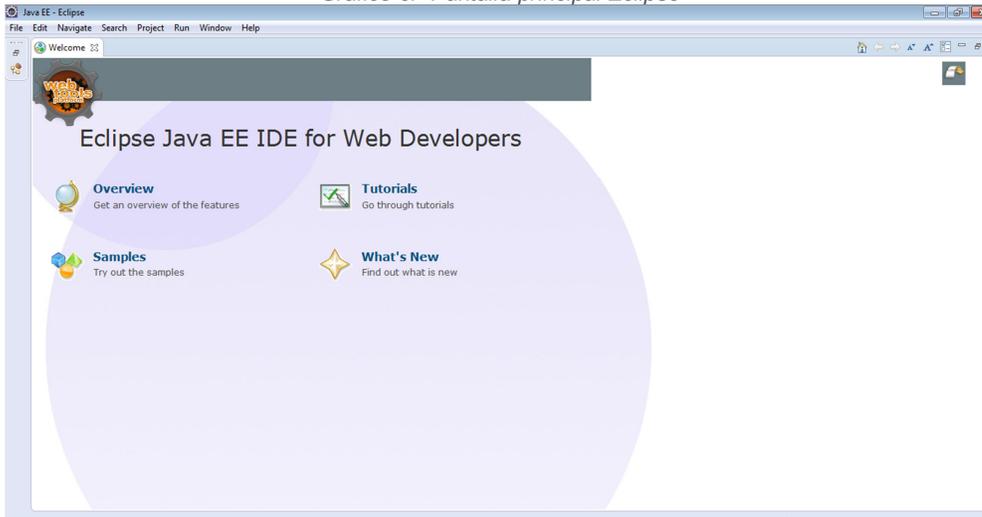
Gráfico 5.- Ejecución de Eclipse



Elaboración: Álvaro Vizuela
Fuente: ww.eclipse.org

Al ejecutar la aplicación nos indicara en que carpeta se inicia la ejecución de la aplicación, de ser el caso y así desearlo podemos cambiar la ruta de la carpeta dando clic en el botón **Browse**.

Gráfico 6.- Pantalla principal Eclipse



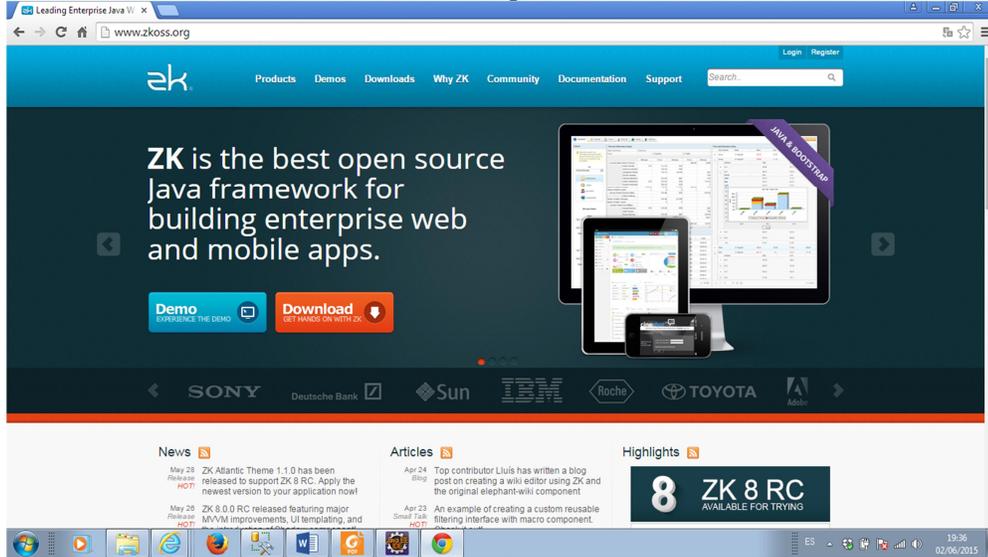
Elaboración: Álvaro Vizuela
Fuente: ww.eclipse.org

Esta es la página principal de eclipse la primera vez que se inicia.

CONFIGURACION ZK

Ingresamos a la página de Zk: <http://www.zkoss.org/> y damos clic en **Register**.

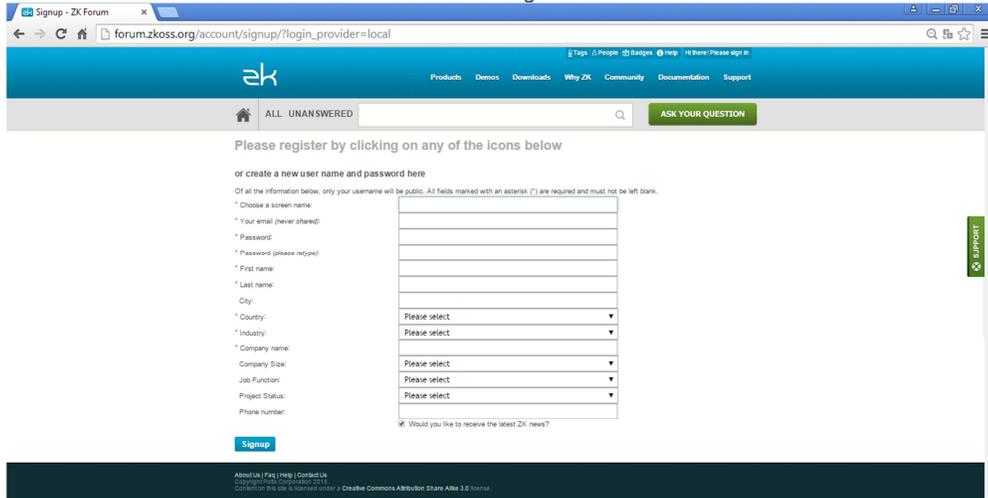
Gráfico 7.- Pagina web ZK



Elaboración: Álvaro Vizueta
Fuente: ww.zkoss.org

Para realizar las descargar necesarias y poder utilizar librerías en nuestro proyecto debemos crear una cuenta y registrarnos.

Gráfico 8.- registro Zk

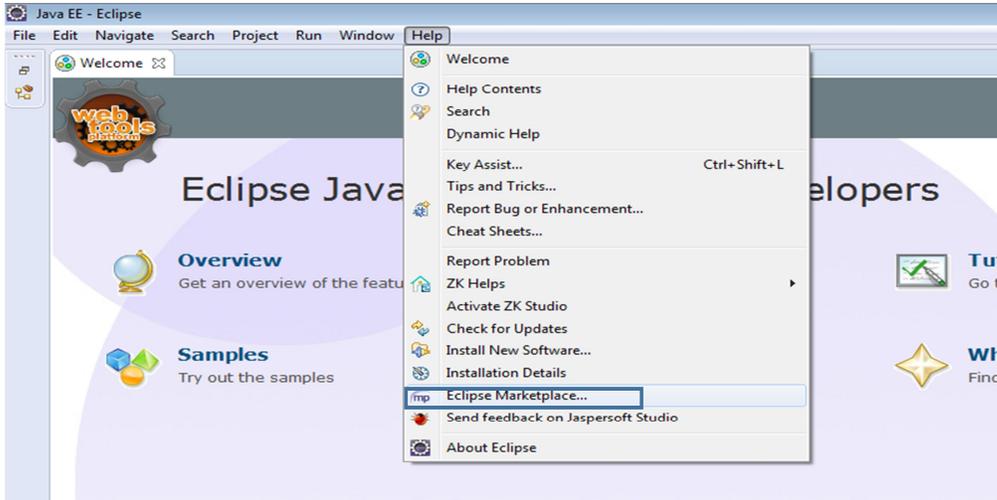


Elaboración: Álvaro Vizueta
Fuente: ww.zkoss.org

Una vez creado y registrado el usuario ejecutamos Eclipse.exe

INSTALACION DE ZK

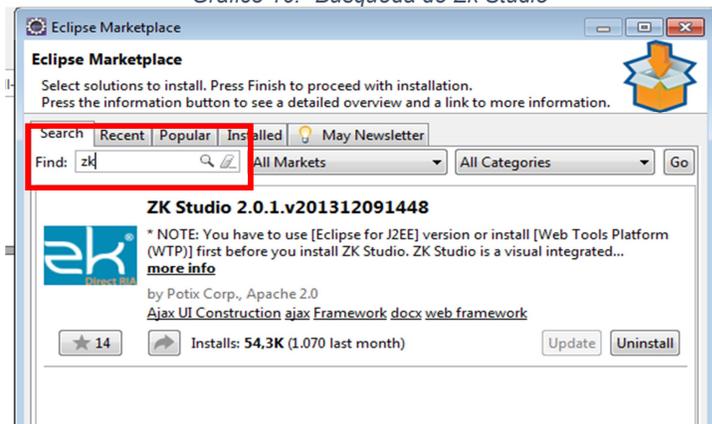
Gráfico 9.- Ingreso a Eclipse Market Place



Elaboración: Álvaro Vizuela
Fuente: www.zkoss.org

PASO 1: En la pantalla principal de eclipse seleccionamos: **Help -> Eclipse Marketplace**

Gráfico 10.- Búsqueda de Zk Studio



Elaboración: Álvaro Vizuela
Fuente: www.zkoss.org

PASO 2: Se abrirá una ventana de dialogo, en esta ventana tal como muestra la imagen, en el cuadro de búsqueda (**Find**), escribimos la palabra Zk. A

continuación se presentara la opción de **Zk Estudio**, daremos clic en **Install** para proceder a instalar este plugin.

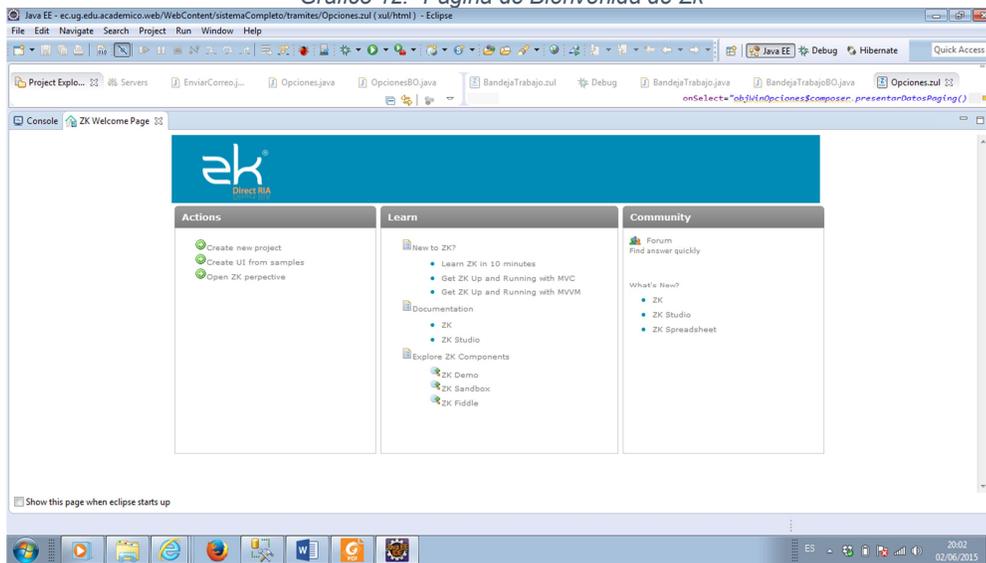
Gráfico 11.- Confirmación de Instalación



Elaboración: Álvaro Vizueta
Fuente: ww.zkoss.org

PASO 3: Confirmamos la instalación, después de haberse realizado con éxito la instalación, eclipse procederá a reiniciarse.

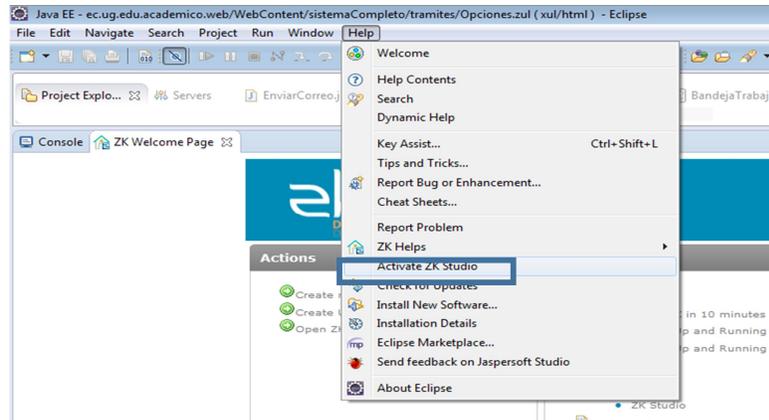
Gráfico 12.- Página de Bienvenida de Zk



Elaboración: Álvaro Vizueta
Fuente: ww.zkoss.org

Después que Eclipse se haya reiniciado, se mostrara la pantalla de bienvenida de Zk.

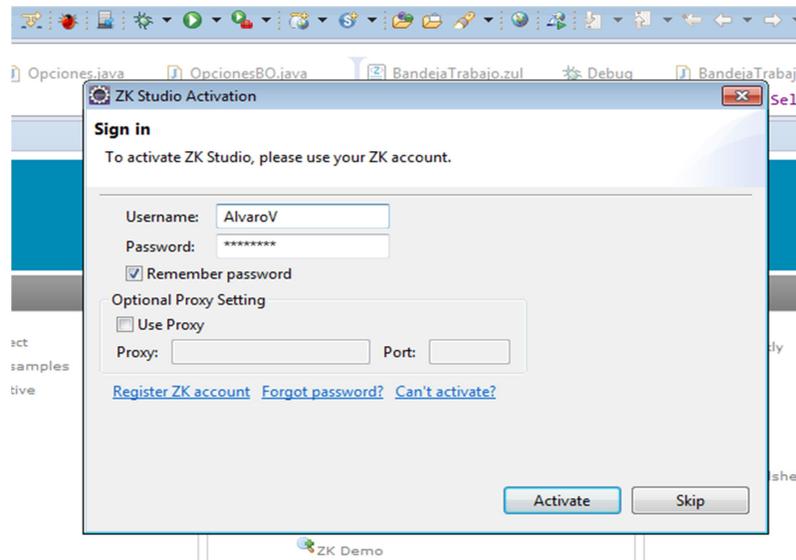
Gráfico 13.- Activación de Zk Studio



Elaboración: Álvaro Vizuela
Fuente: ww.zkoss.org

PASO 4: En la pantalla principal de Eclipse damos clic en: **Help -> Activate ZK Studio.**

Gráfico 14.- Ingreso de usuario y contraseña



Elaboración: Álvaro Vizuela
Fuente: ww.zkoss.org

PASO 5: En este cuadro de dialogo, ingresamos el usuario y contraseña con el cual nos registramos en <http://www.zkoss.org/>, damos clic en **Activate**, después de activar podremos acceder de manera más fácil a contenido de Zk.

DESCARGAR Y CONFIGURAR APACHE CXF

Ingresamos a la página de Apache CXF: <http://cxf.apache.org/>

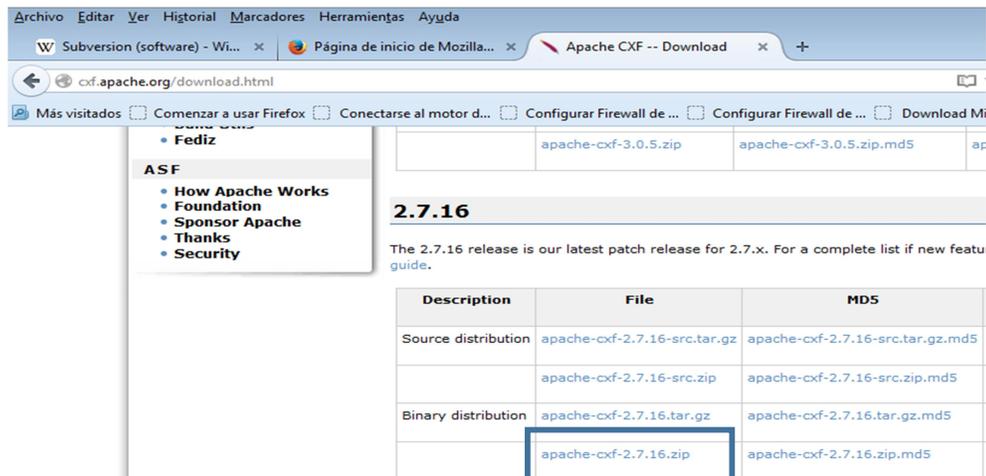
Gráfico 15.- Pagina Apache CXF



Elaboración: Álvaro Vizueta
Fuente: www.cxf.apache.org

Realizamos la descarga de Apache CXF, para eso damos clic en Download.

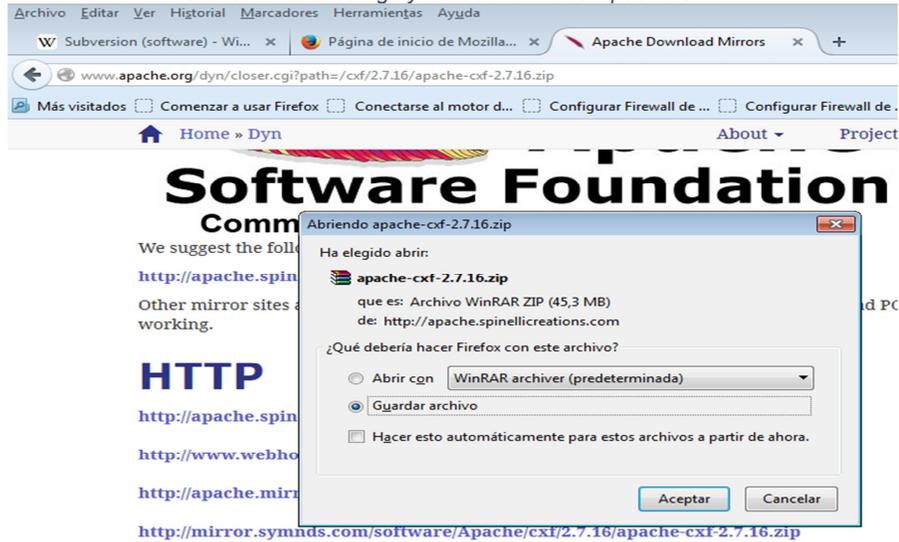
Gráfico 16.- Pagina descarga Apache CXF



Elaboración: Álvaro Vizueta
Fuente: www.cxf.apache.org

En esta página seleccionamos el paquete apache-cxf-2.7.16.zip.

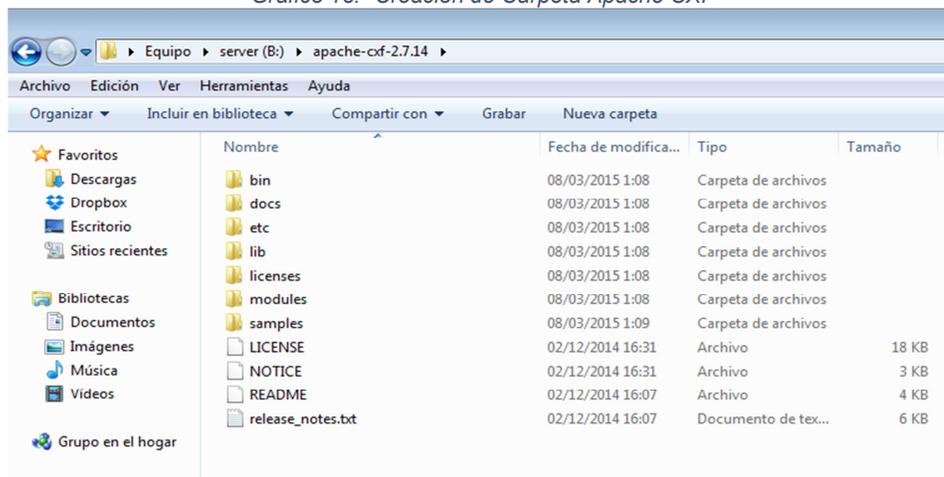
Gráfico 17.- Descarga y almacenamiento Apache CXF



Elaboración: Álvaro Vizuela
Fuente: www.cxf.apache.org

En esta página, seleccionamos uno de los enlaces de descarga, posteriormente aparecerá un cuadro de dialogo en donde seleccionamos **Guardar** y damos clic en **Aceptar**.

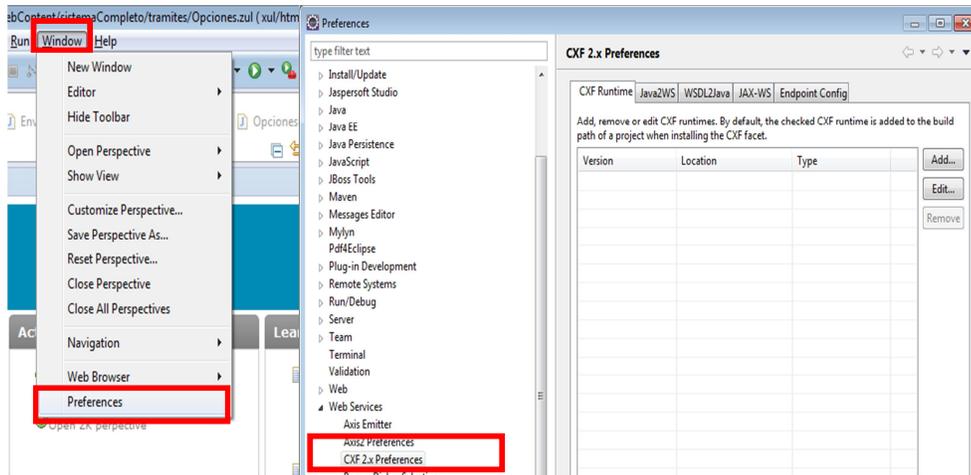
Gráfico 18.- Creación de Carpeta Apache CXF



Elaboración: Álvaro Vizuela
Fuente: www.cxf.apache.org

Una vez descargado, descomprimos el archivo donde encontraremos una carpeta similar a la de la imagen, esto nos servirá para compilar los WebServices.

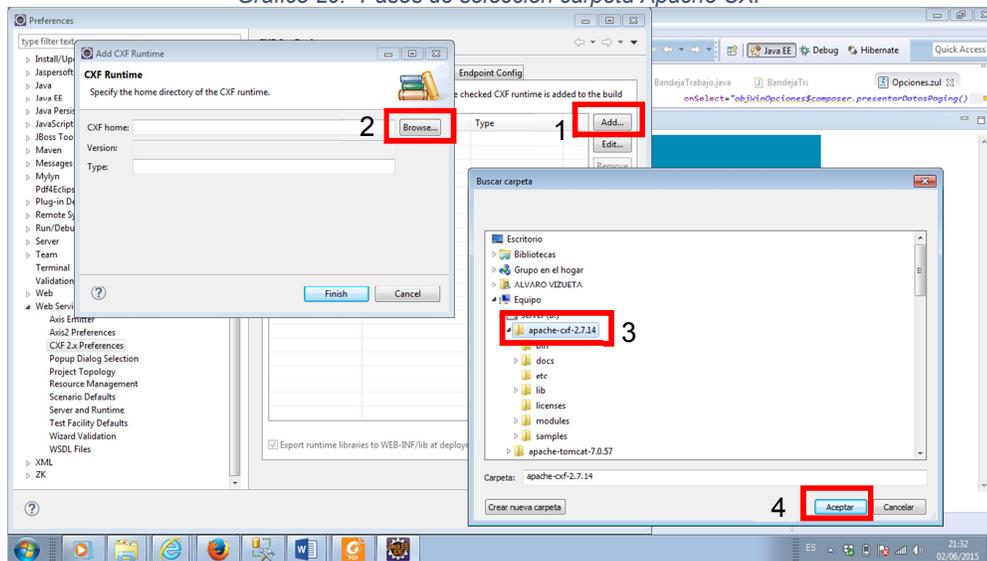
Gráfico 19.- Configuración de Apache CXF



Elaboración: Álvaro Vizueta
Fuente: www.cxf.apache.org

Damos clic en Window -> Preferences, aparecerá un cuadro de dialogo en el cual iremos a **Web Service -> CXF 2.x Preferences**.

Gráfico 20.- Pasos de selección carpeta Apache CXF

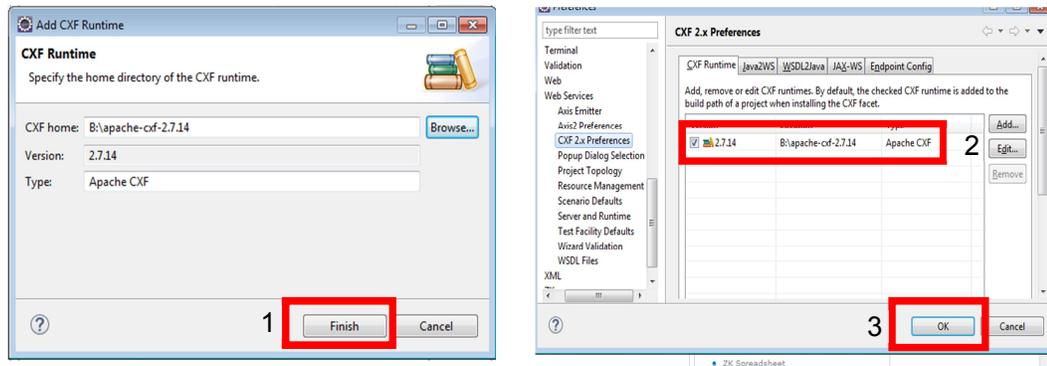


Elaboración: Álvaro Vizueta
Fuente: www.cxf.apache.org

En el cuadro de dialogo de **CXF 2.x Preferences** damos clic en:

1. –Add
2. Browser
3. Buscamos la carpeta **apache-cxf-2.7.16** que descomprimimos en un paso anterior.
4. Seleccionamos la carpeta di damos clic en **Aceptar**.

Gráfico 21.- Selección de opción Apache CXF



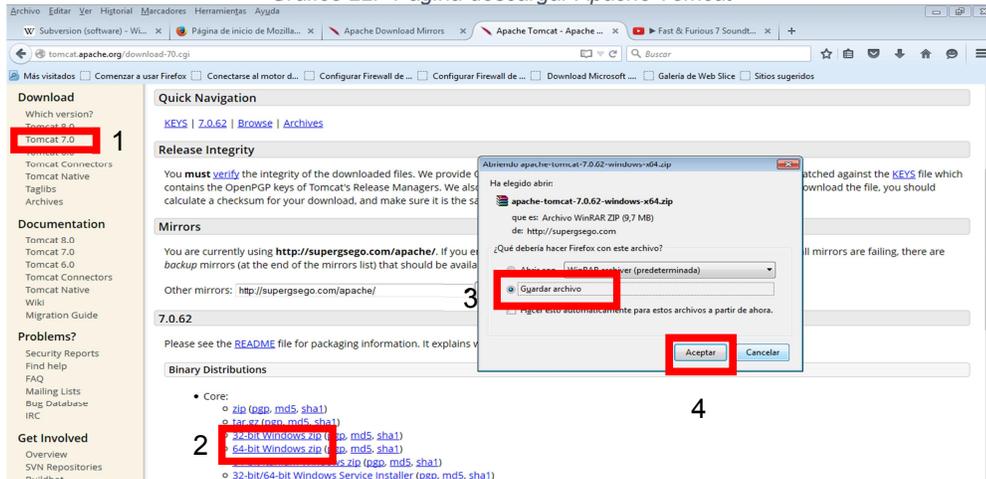
Elaboración: Álvaro Vizuela
Fuente: www.cxf.apache.org

En este cuadro damos clic en **Finish**, en el cuadro de **Preference**, seleccionamos la instalación de CXF que recién configuramos y damos clic en **OK**.

DESCARGA DE APACHE TOMCAT Y CONFIGURACION DE SERVIDORES WS Y WEB EN ECLIPSE

Ingresamos a la página de Apache Tomcat: <http://tomcat.apache.org/>

Gráfico 22.- Pagina descargar Apache Tomcat

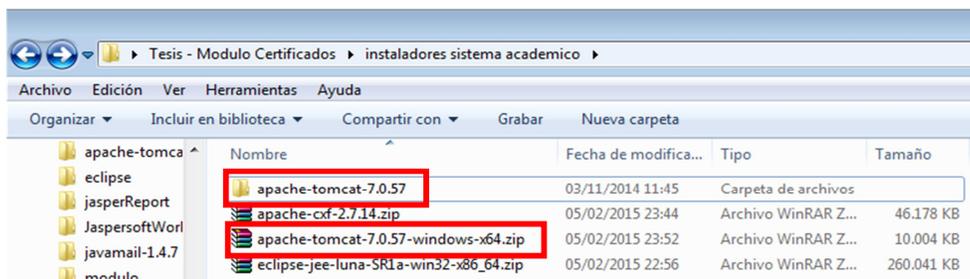


Elaboración: Álvaro Vizueta
Fuente: www.tomcat.apache.org

En la parte de descarga seleccionamos Tomcat 7.0, posteriormente seleccionaremos la versión acorde a nuestro sistema operativo (32bit – 64bit), aparecerá un cuadro de dialogo en donde escogeremos **Guardar Archivo** y después **Aceptar**.

Descomprimos el archivo que descargamos quedando una carpeta similar a la que se muestra en la imagen.

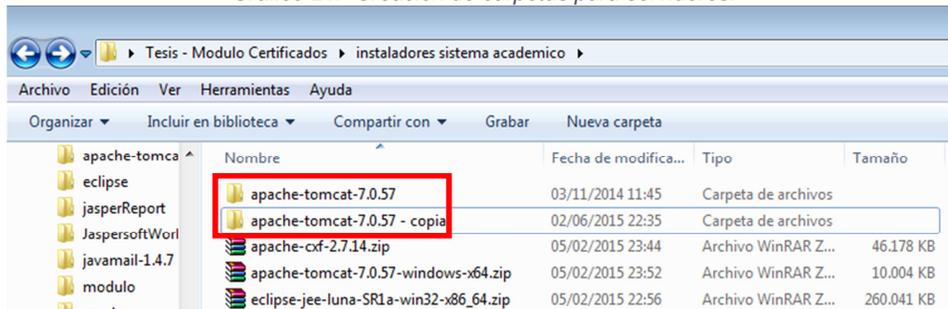
Gráfico 23.- Archivo y carpeta descomprimida apache tomcat



Elaboración: Álvaro Vizueta
Fuente: www.cxf.apache.org

Después procedemos a copiar la carpeta que recién ha sido extraída, y pegarla en la misma ubicación que la original.

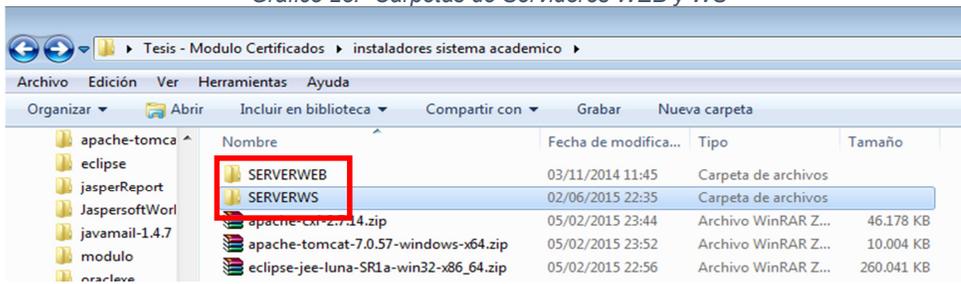
Gráfico 24.- Creación de carpetas para servidores.



Elaboración: Álvaro Vizuela
Fuente: www.cxf.apache.org

Después de realizar esto procedemos a cambiar de nombre a las dos carpetas que hemos creado.

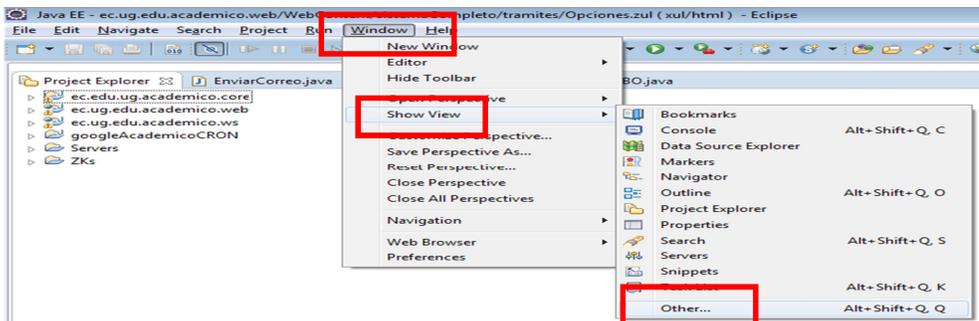
Gráfico 25.- Carpetas de Servidores WEB y WS



Elaboración: Álvaro Vizuela
Fuente: www.cxf.apache.org

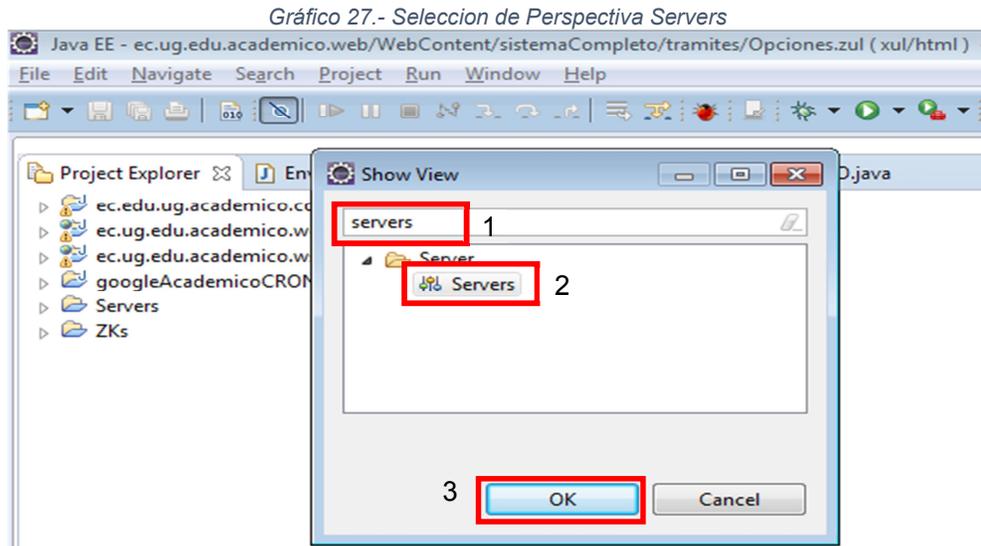
Una carpeta será renombrada como **SERVERWEB** y la otra como **SERVERWS**
Ahora procederemos a realizar lo siguiente:

Gráfico 26.- cambio de perspectiva a servidores



Elaboración: Álvaro Vizuela
Fuente: www.cxf.apache.org

En la pantalla principal de Eclipse damos clic en: **Window -> Show view -> Other**



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

En esta parte en el panel de búsqueda escribiremos **servers**, ahí aparecerá la opción para visualizar los servidores que hay configurados, seleccionamos y damos clic en **Ok**.

CREACION Y CONFIGURACION DE SERVERWEB

CREACION DE SERVERWEB

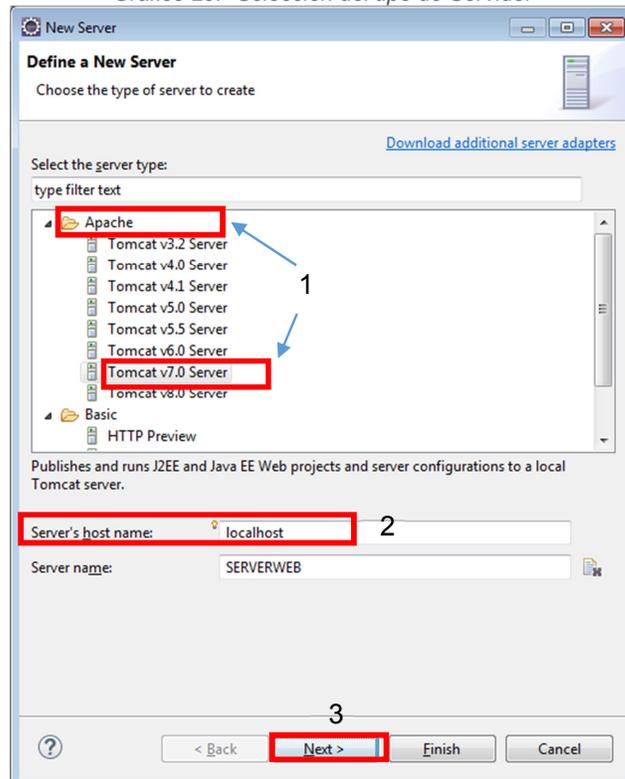
Gráfico 28.- Creación de Servidores



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

PASO 1: Seleccionamos la pestaña **Servers**, en un espacio en blanco de esta pestaña damos clic derecho, aparecerá un menú donde escogeremos **New -> Servers**

Gráfico 29.- Selección del tipo de Servidor

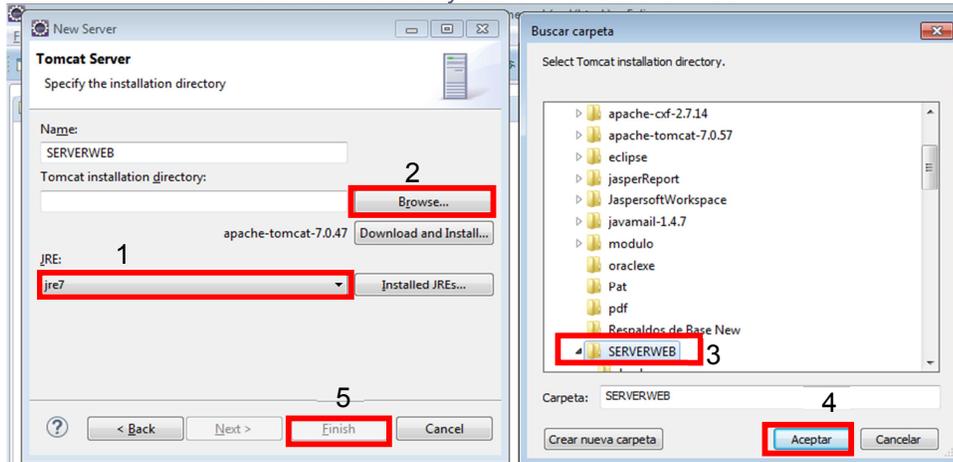


Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

PASO 2: Aparecerá este cuadro de diálogo donde escogeremos:

1. Apache-> Tomcat v7.0 Server
2. Colocamos como **Server Name** : SERVERWEB
3. Damos clic en **Next**.

Gráfico 30.- Creación y selección de SERVERWEB

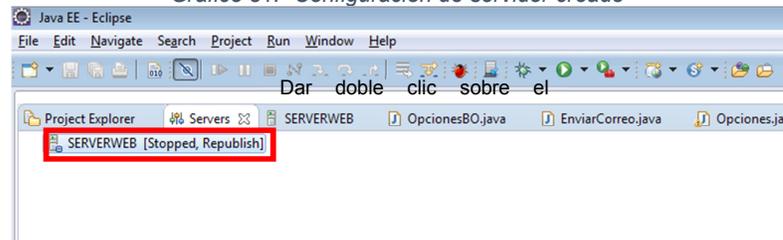


Elaboración: Álvaro Vizueta
Fuente: Álvaro Vizueta

PASO 3: En esta pantalla realizamos lo siguiente:

1. Seleccionamos el JRE (jre7)
2. Damos clic en **Browse**
3. Buscamos en el directorio la carpeta que renombramos como **SERVERWEB**, la seleccionamos dándole clic.
4. Damos clic en **Aceptar**.
5. Damos clic en **Finish**.

Gráfico 31.- Configuración de servidor creado

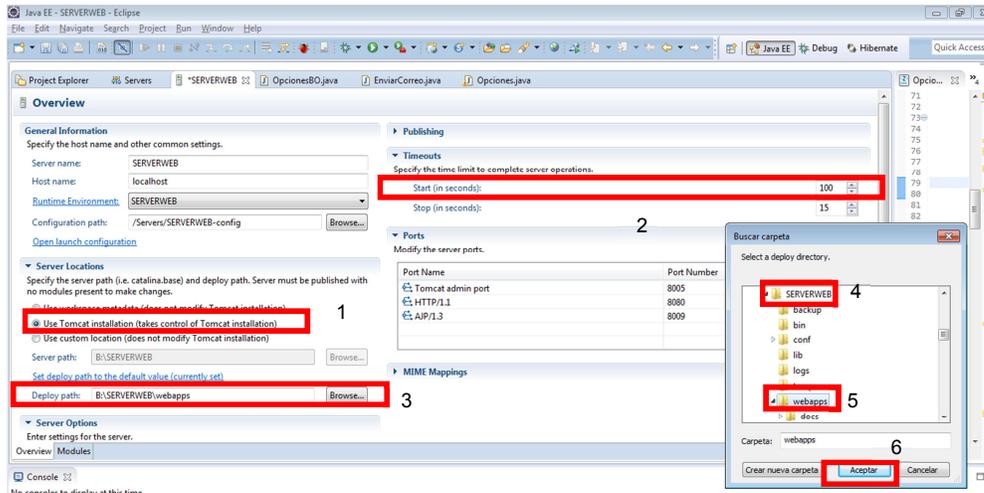


Elaboración: Álvaro Vizueta
Fuente: Álvaro Vizueta

Ya está creado nuestro servidor WEB, dar doble clic sobre el servidor creado para configurar las opciones

CONFIGURACIÓN DE SERVERWEB

Gráfico 32.- Configuración y parametrización serverweb



Elaboración: Álvaro Vizueta
Fuente: Álvaro Vizueta

En esta pantalla realizaremos lo siguiente:

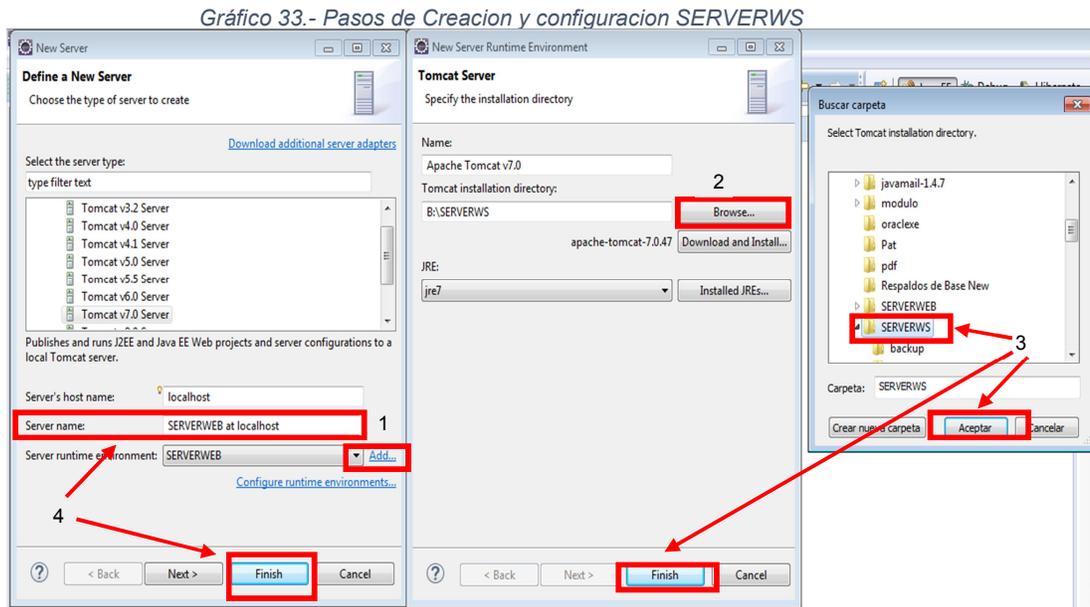
1. Seleccionar **Use Tomcat installation**
2. En la pestaña **Timeouts**, cambiar el tiempo de **Start** a 100.
3. En **Deploy path** cambiar el directorio dando clic en **Browse**.
4. Se abrirá una un cuadro de dialogo donde tendrá que buscar en el directorio la carpeta **SERVERWEB** que creamos.
5. Dentro de la carpeta **SERVERWEB** buscamos y seleccionamos con un clic la carpeta **webapps**.
6. Y por último damos clic en **Aceptar**.

Para guardar los cambios que hemos realizado basta con presionar **Ctrl + S**, o en su defecto dar clic en la opción guardar que es el icono de un diskette ubicado en la parte superior izquierda de Eclipse.

CREACION Y CONFIGURACION DE SERVERWS

CREACION DE SERVERWS

Haremos el mismo **Paso 1** para la creación de **SERVERWEB**



Elaboración: Álvaro Vizueta
Fuente: Álvaro Vizueta

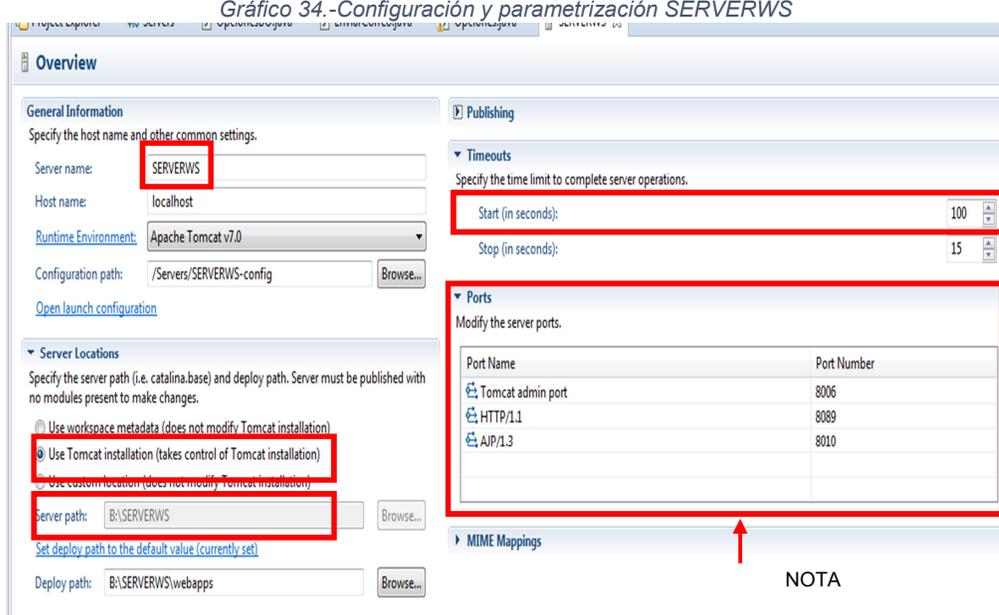
Después realizamos lo siguiente:

1. Damos clic en **Add**.
2. En la ventana que se abre damos clic en **Browse**.
3. Buscamos y seleccionamos la carpeta **SERVERWS** y luego damos clic en **Aceptar** y en **Finish**
4. Cambiamos el nombre de Server Name a **SERVERWS** y damos clic en **Finish**

CONFIGURACIÓN DE SERVERWS

Se sigue el mismo orden con el que se configuro el **SERVERWEB**, pero con la diferencia que todo lo que se hizo se trabaje en base a la carpeta **SERVERWS** que se había creado anteriormente.

Gráfico 34.-Configuración y parametrización SERVERWS

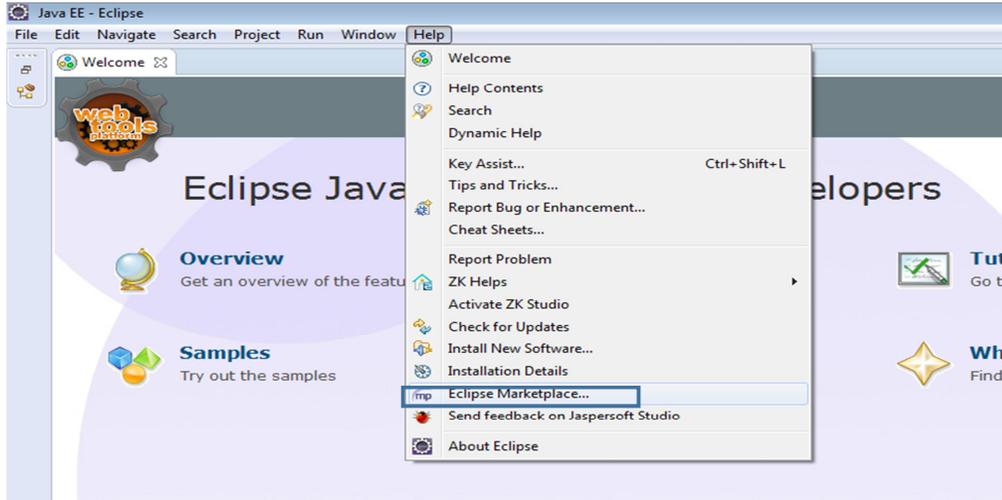


Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

Nota: Al momento de realizar la configuración, tomar en cuenta que los puertos de **SERVERWEB** y **SERVERWS** no se repitan, ya que esto ocasionaría problemas al momento de ejecutar los servidores.

INSTALACION DE HIBERNATE PARA MAPEO DE BASE DE DATOS

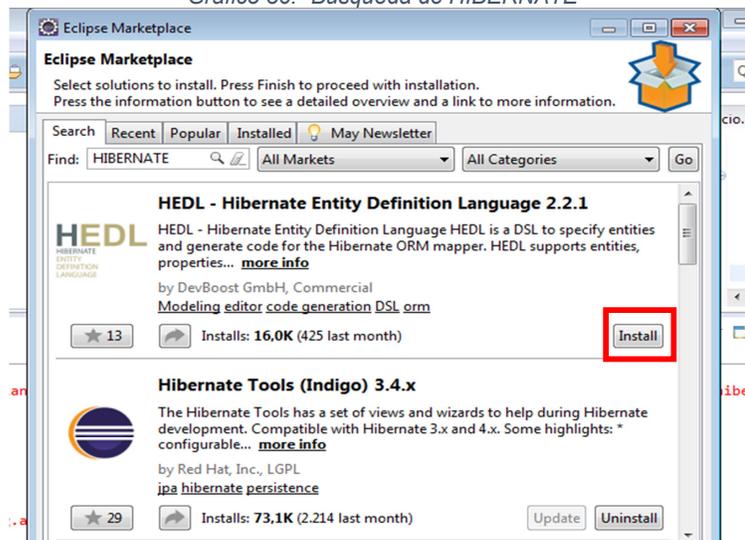
Gráfico 35.- Instalación mediante MarketPlace



Elaboración: Álvaro Vizueta
Fuente: Álvaro Vizueta

PASO 1: En la pantalla principal de eclipse seleccionamos: **Help -> Eclipse Marketplace**

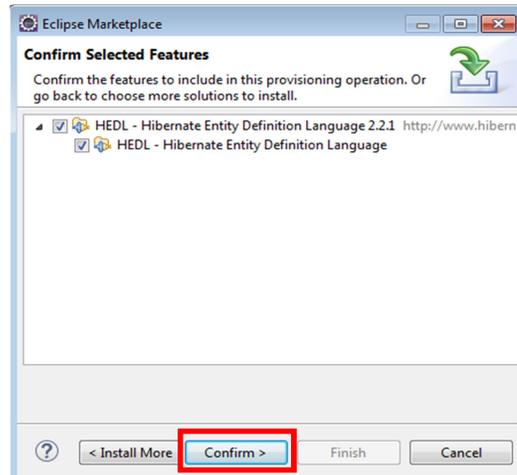
Gráfico 36.- Búsqueda de HIBERNATE



Elaboración: Álvaro Vizueta
Fuente: Álvaro Vizueta

PASO 2: Se abrirá una ventana de dialogo, en esta ventana tal como muestra la imagen, en el cuadro de búsqueda (**Find**), escribimos la palabra **HIBERNATE**. A continuación se presentara la opción de **HEDL – HIBERNATE DEFINITION LANGUAGE 2.2.1**, daremos clic en **Install** para proceder a instalar este plugin.

Gráfico 37.- Confirmación de Instalación

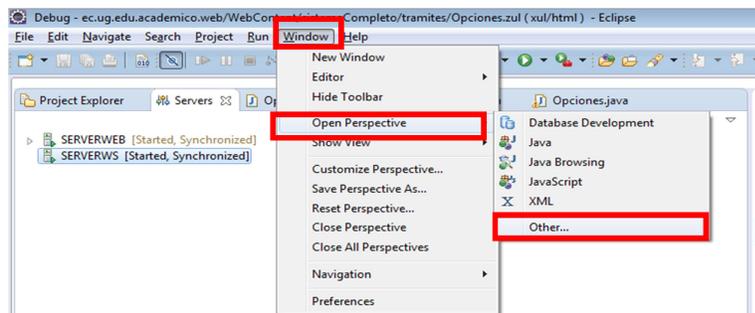


Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

PASO 3: En esta pantalla daremos clic en **Confirmar** para continuar con la instalación. Después de que se haya instalado la aplicación Eclipse se reiniciara para completar la actualización.

ABRIR PERSPECTIVA EN HIBERNATE

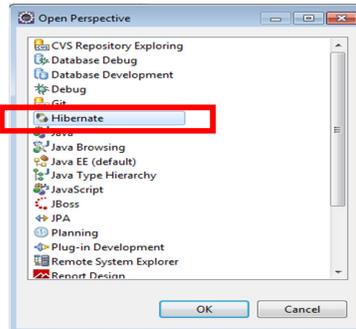
Gráfico 38.- Abrir perspectiva HIBERNATE



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

Para abrir la perspectiva realizamos lo siguiente: damos clic en **Window -> Open Perspective -> Other**

Gráfico 39.- Selección de perspectiva HIBERNATE



Elaboración: Álvaro Vizueta
Fuente: Álvaro Vizueta

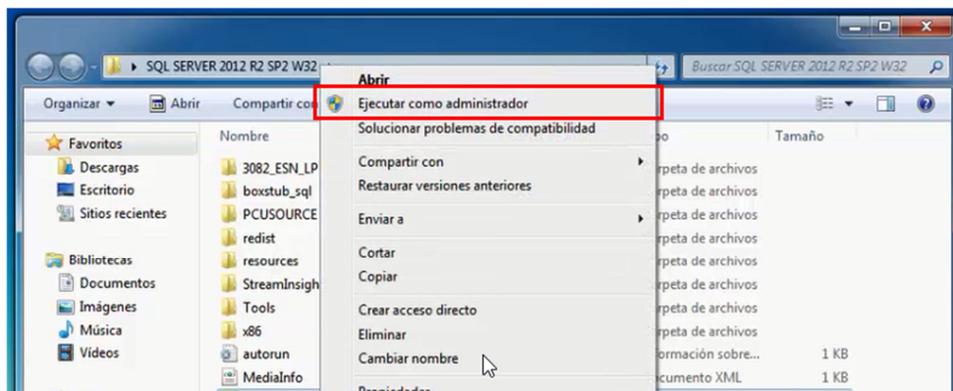
En la ventana que nos aparece seleccionamos **Hibernate**.

INSTALACIÓN DE BASE DE DATOS

Para levantar la base de datos del sistema académico, necesitaremos del instalador del SQL Server 2012 R2 SP2.

1. Nos dirigimos a la ruta donde se encuentra el instalador de Sql Server 2012 R2 SP2 y damos click derecho en la aplicación Setup y elegimos “Ejecutar como Administrador”.

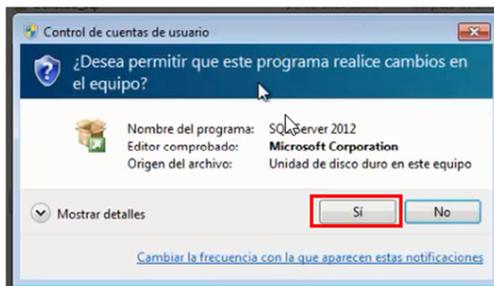
Gráfico 40.- Inicio de Instalación



Elaboración: Álvaro Vizueta
Fuente: Álvaro Vizueta

2. En caso de abrirse el control de cuentas de usuario de Windows, debemos darle la opción "SI" para permitir instalar el programa.

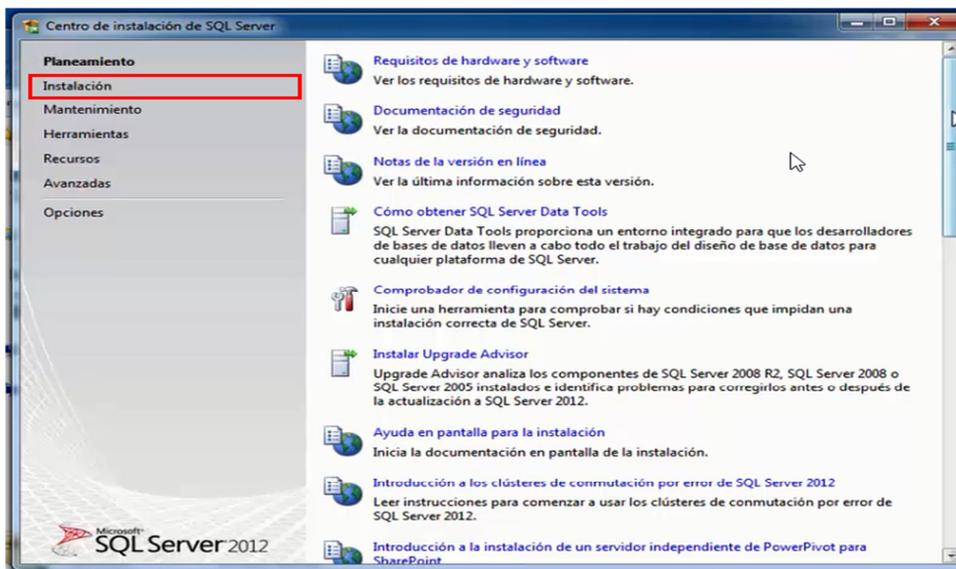
Gráfico 41.- Aceptación de Instalación



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

3. Esperamos a que se procese la operación.
4. Luego nos aparecerá la siguiente ventana de instalación de Sql y damos click en "Instalación".

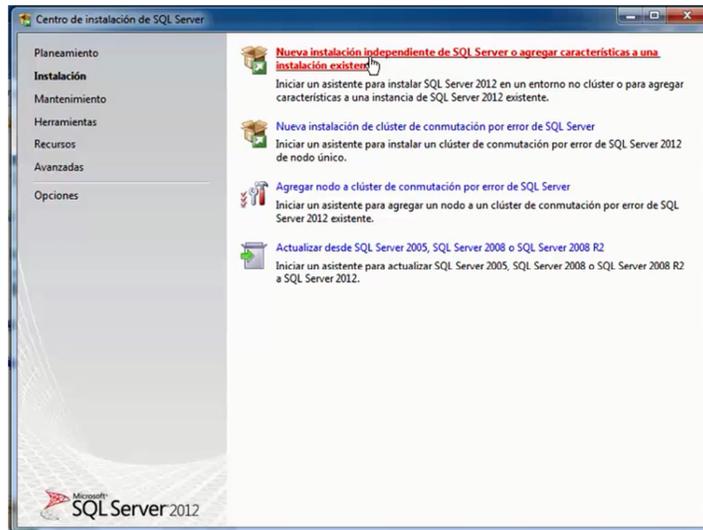
Gráfico 42.- Centro de instalación



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

5. Como vamos a realizar una nueva instalación, elegimos “Nueva Instalación Independiente de SQL”.

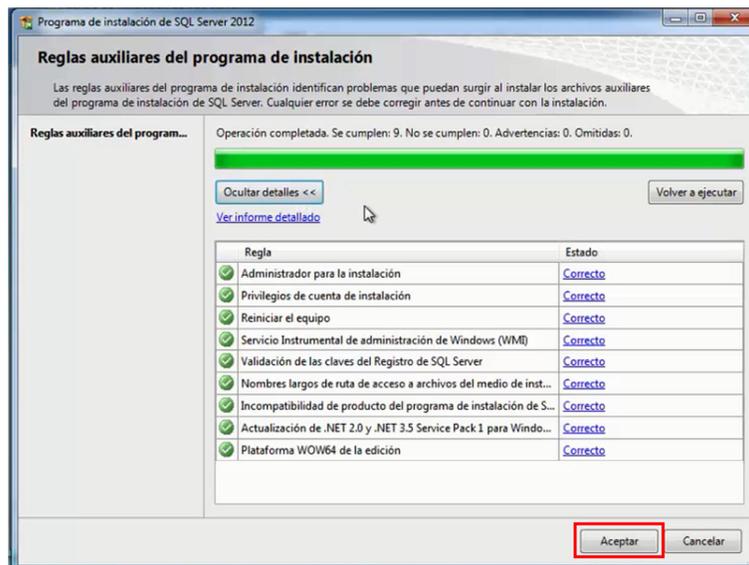
Gráfico 43.- Selección de Nueva instalación



Elaboración: Álvaro Vizueta
Fuente: Álvaro Vizueta

6. Esperamos a que se procese la operación.
7. Aparece la siguiente ventana y si cumplimos con todas las reglas requeridas damos click en “Aceptar”.

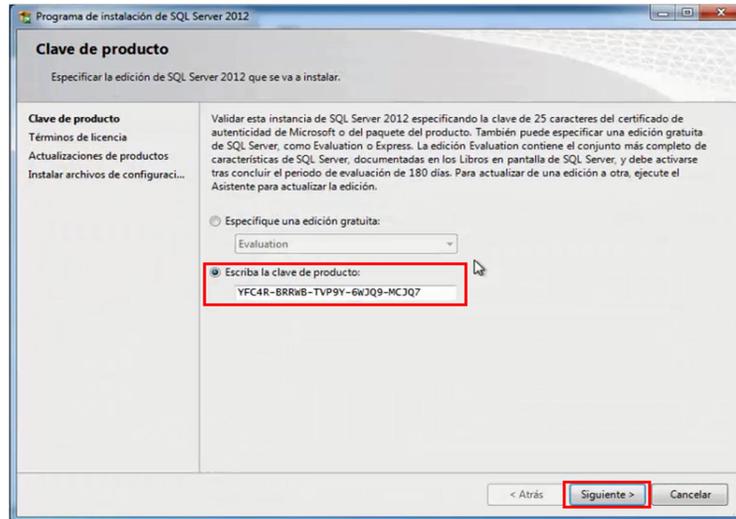
Gráfico 44.- Reglas Auxiliares



Elaboración: Álvaro Vizueta
Fuente: Álvaro Vizueta

8. Esperamos a que se procese la operación.
9. Por defecto nos muestra la clave del producto y damos click en “Siguiente”.

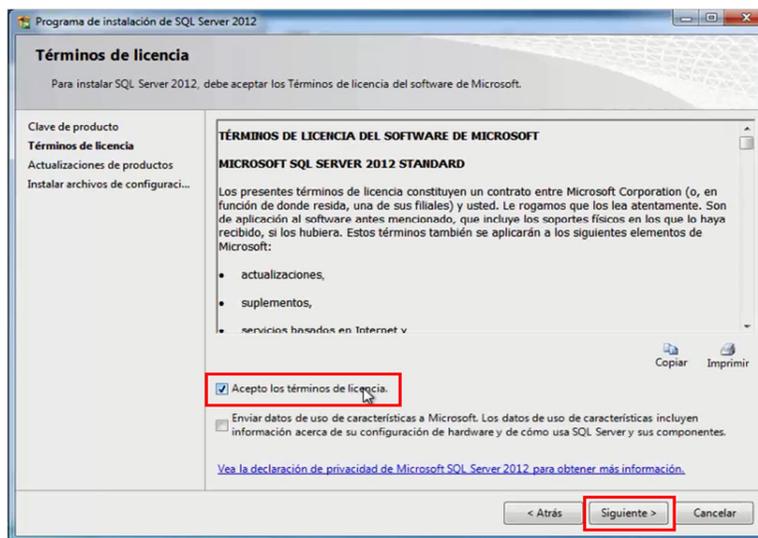
Gráfico 45.- Ingreso de clave de producto



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

10. Aceptamos los términos de la licencia y damos click en “Siguiente”.

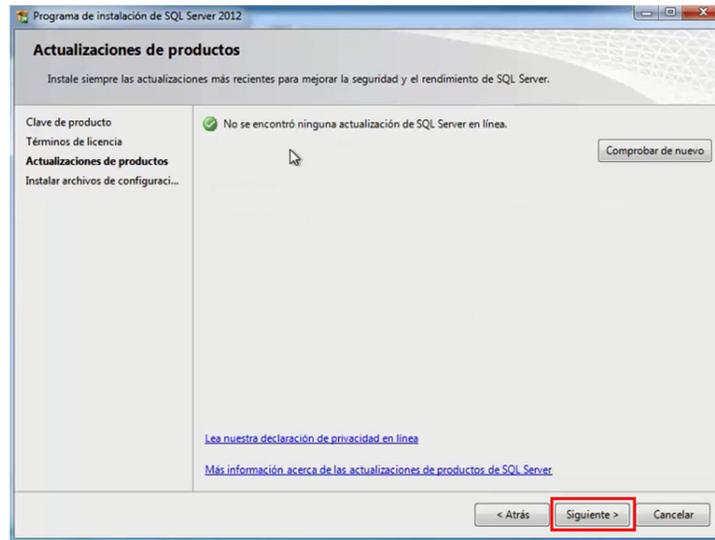
Gráfico 46.- Aceptación de términos de licencia



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

11. Después de la búsqueda e instalación de actualizaciones damos click en “Siguiente”

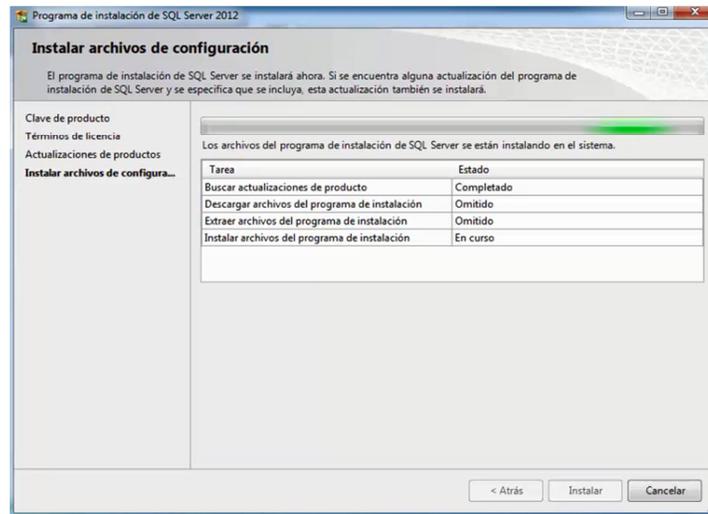
Gráfico 47.- Verificación de Actualizaciones



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

12. Esperamos unos minutos hasta pasar a la siguiente ventana.

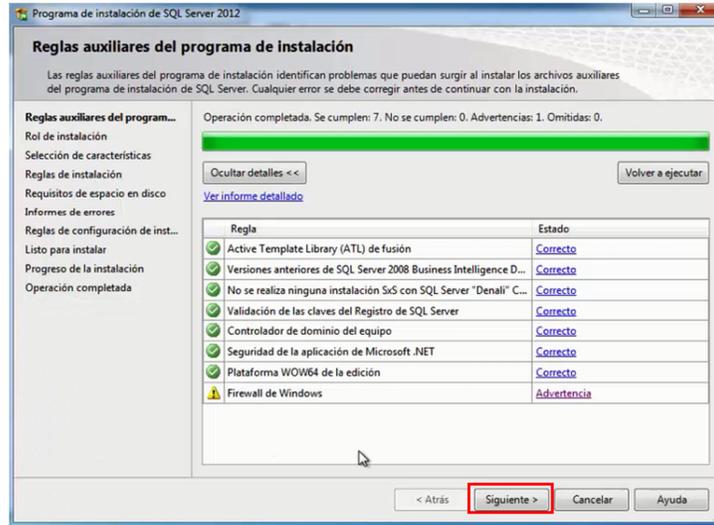
Gráfico 48.- Archivos de configuración



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

13. Nos aparece esta ventana en la cual nos indica una advertencia sobre el Firewall que está habilitado, si cumplimos todas las reglas damos click en “Siguiente”.

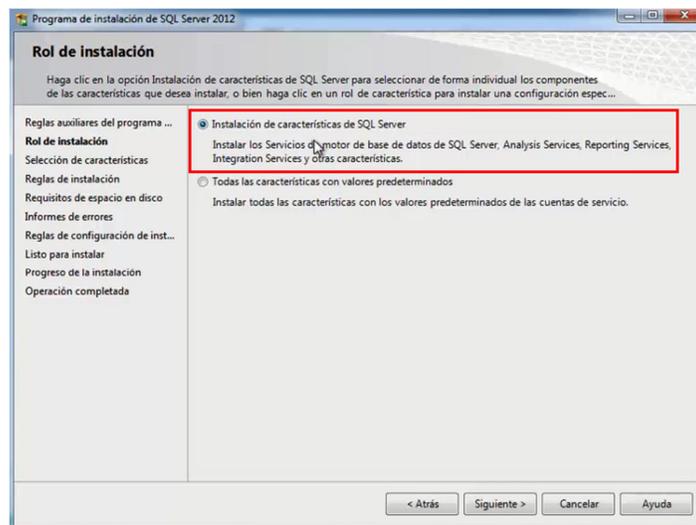
Gráfico 49.- Reglas Auxiliares



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

14. En esta ventana elegimos la opción “Instalación de características de SQL Server”.

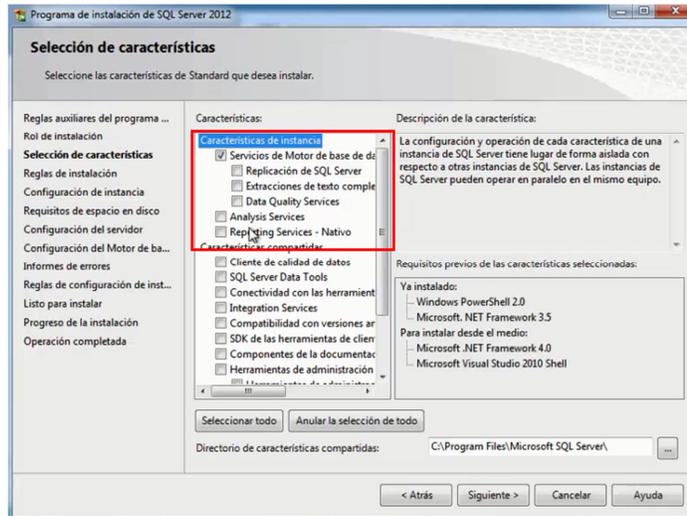
Gráfico 50.- Rol de instalación



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

15. En características de instancia solo seleccionamos “Servicios de motor de base de datos”.

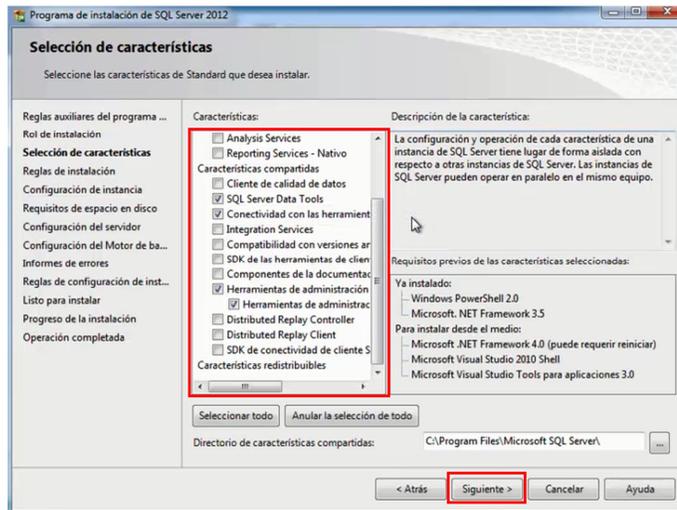
Gráfico 51.- Selección de Características



Elaboración: Álvaro Vizueta
Fuente: Álvaro Vizueta

16. En características compartidas seleccionamos las opciones que podemos visualizar en la imagen, luego damos click en “Siguiente”.

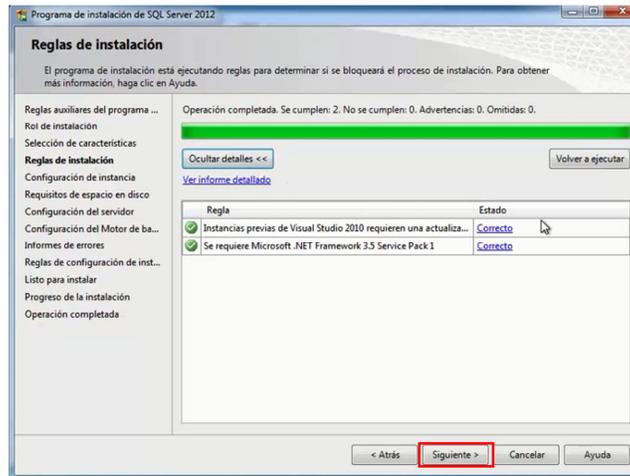
Gráfico 52.- Selección de Características



Elaboración: Álvaro Vizueta
Fuente: Álvaro Vizueta

17. Se verifican automáticamente reglas de instalación y damos click en “Siguiente”.

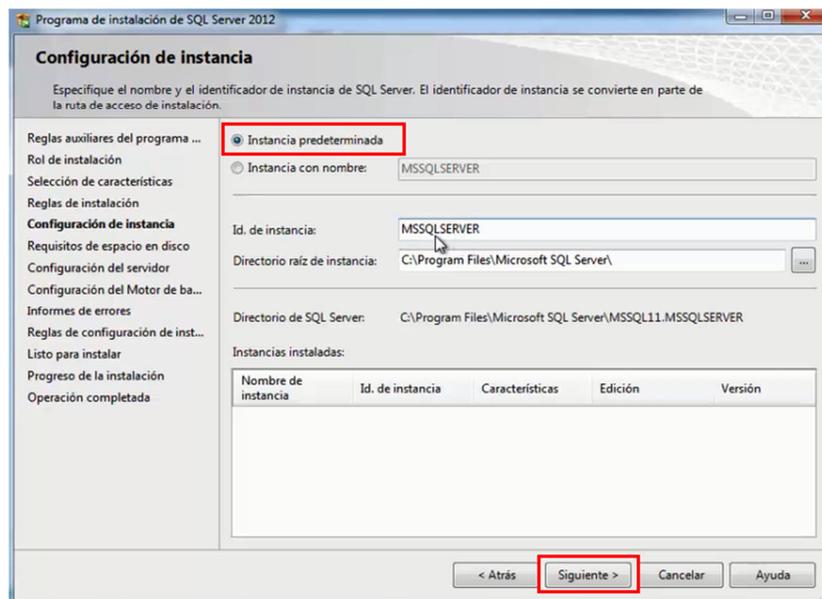
Gráfico 53.- Reglas de Instalación



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

18. Elegimos la instalación de instancia predeterminada y damos click en “Siguiente”.

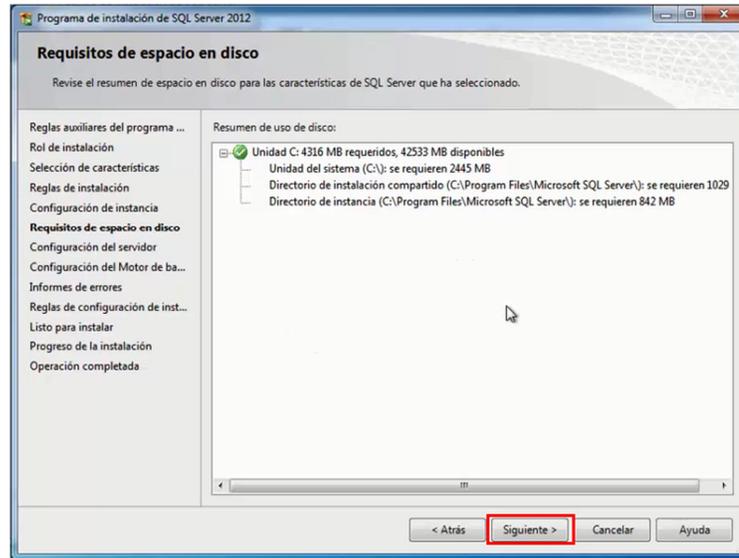
Gráfico 54.- Configuración de Instancia



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

19. En esta ventana nos muestra información de espacio requerido en disco, si el requisito se cumple damos click en “Siguiente”.

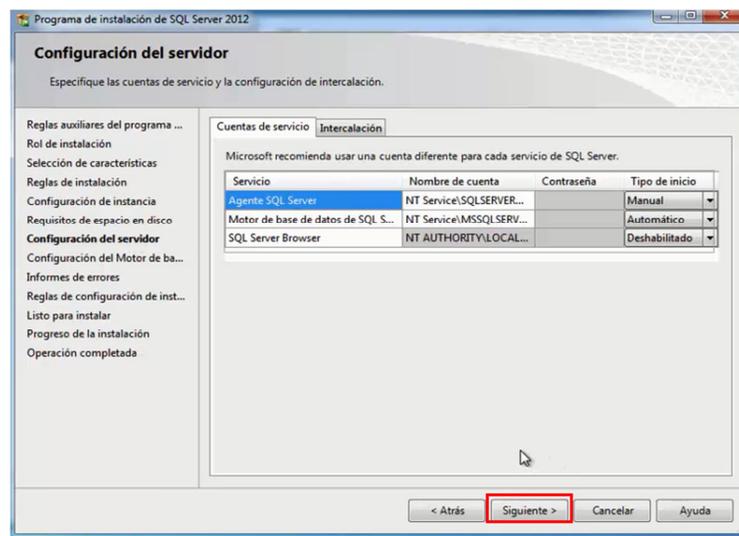
Gráfico 55.-Requisitos



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

20. En esta ventana dejamos los valores por defecto y damos click en “Siguiente”.

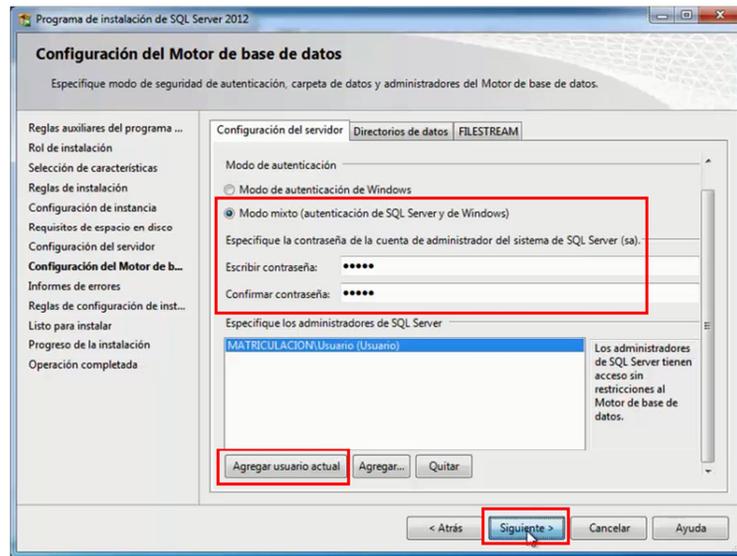
Gráfico 56.- Configuración del Servidor



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

21. Configuramos el motor de base de datos en “Modo de Autenticación Mixto” y asignamos una clave cualquiera, agregamos el usuario administrador dando click en “Agregar usuario Actual”, luego damos click en “Siguiente”.

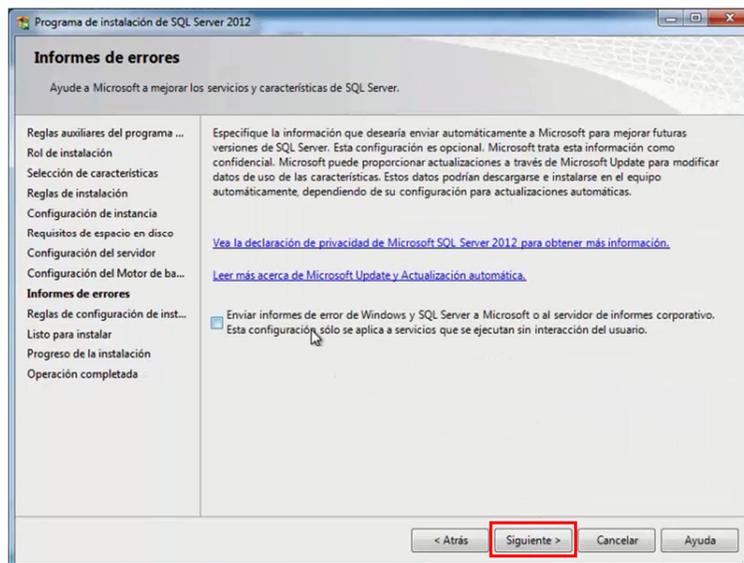
Gráfico 57.- Configuración motor base de Datos



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

22. En esta ventana damos click en “Siguiente”.

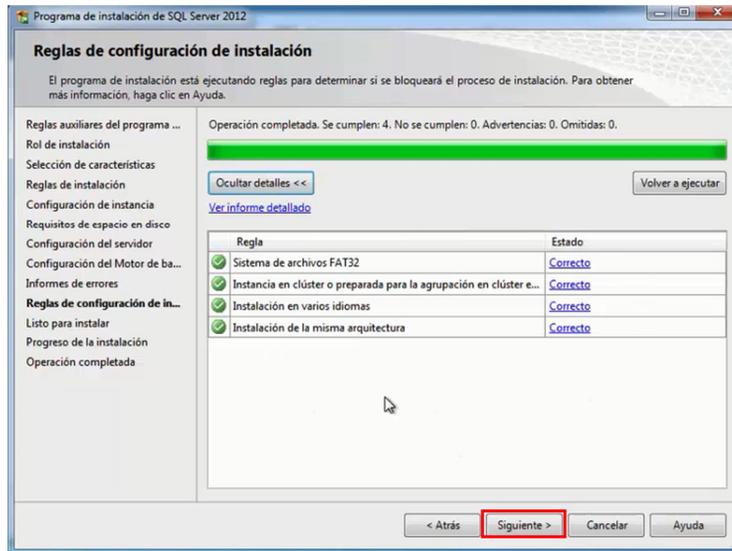
Gráfico 58.- Informe de Errores



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

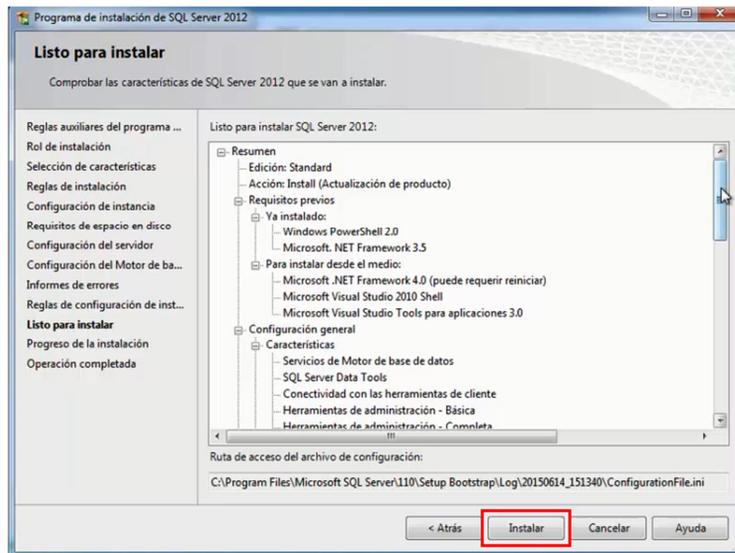
23. En esta ventana si cumplimos con todas las reglas damos click en “Siguiente”.

Gráfico 59.- Reglas de Instalación



24. En esta ventana nos muestra un resumen de las características que van a ser instaladas, damos click en “Instalar”.

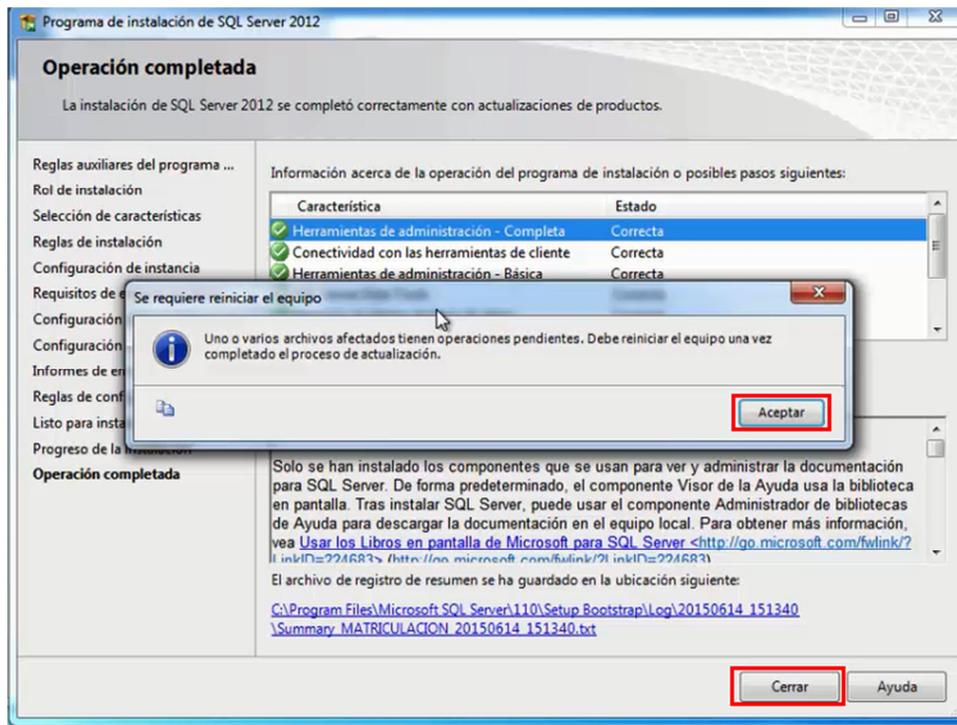
Gráfico 60.-Aceptación de instalación



Elaboración: Álvaro Vizueta
Fuente: Álvaro Vizueta

25. En ocasiones es necesario realizar el reinicio del sistema después de finalizada la actualización

Gráfico 61.- Operación Completa

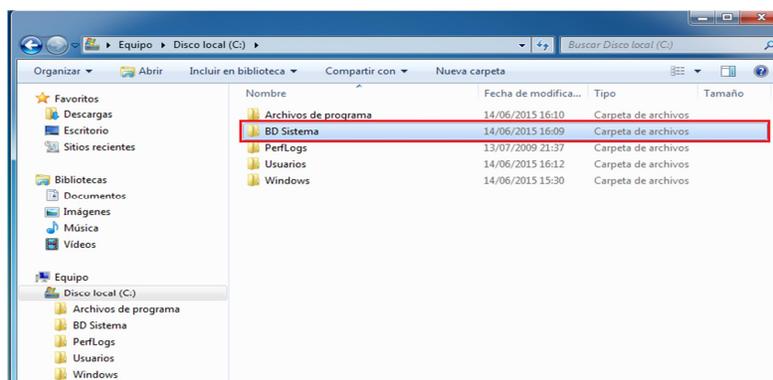


Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

RESTAURACIÓN DE BASE DE DATOS

1. De preferencia debemos copiar la carpeta que contiene el respaldo de la base en la raíz del disco C.

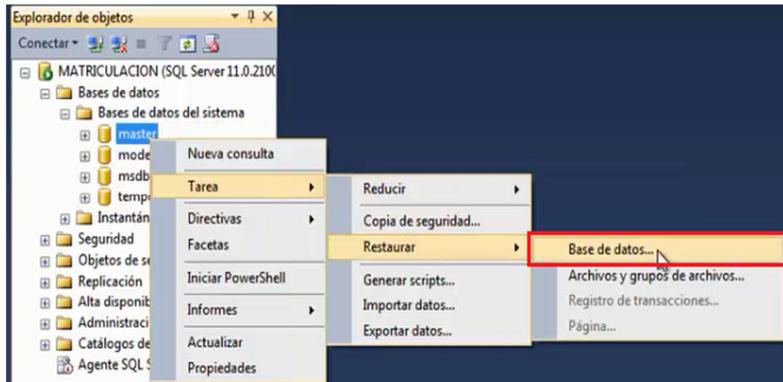
Gráfico 62.- Archivo de Restauración



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

2. Para importar la base de datos desplegamos del lado izquierdo la instancia -> Base de Datos-> Base de Datos del Sistema y damos click derecho en cualquiera de las base que nos muestra y luego navegamos -> Tareas-> Restaurar-> y damos click en Base de Datos...

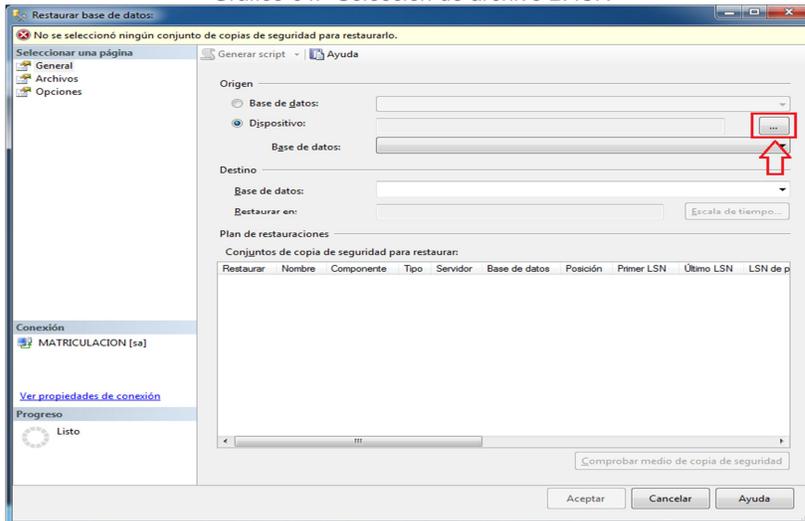
Gráfico 63.- Proceso de Restauración



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

3. En la siguiente ventana elegimos origen desde dispositivo y damos click en el botón que contiene puntos suspensivos para buscar la ruta.

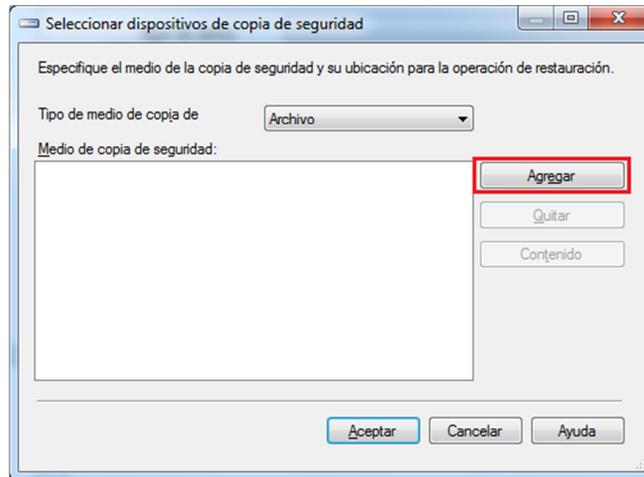
Gráfico 64.- Selección de archivo BACK



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

- Para seleccionar el archivo de copia de seguridad damos click en “Agregar”.

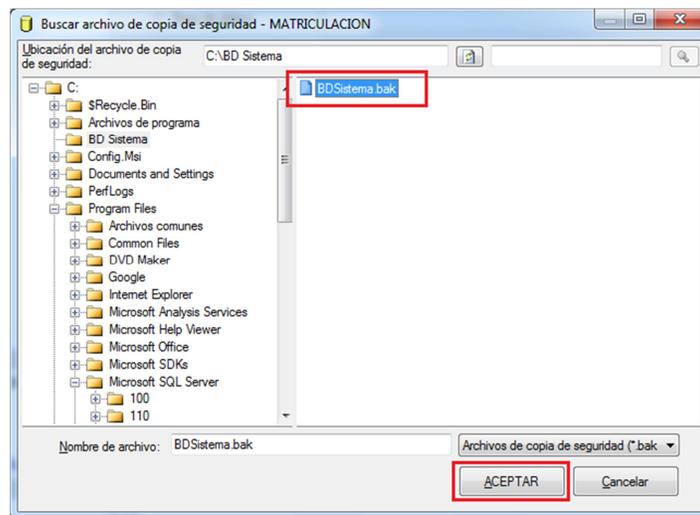
Gráfico 65.- Ventana de Selección



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

- Buscamos la carpeta donde se encuentra el archivo BDSistema.bak, lo seleccionamos y damos click en “Aceptar”.

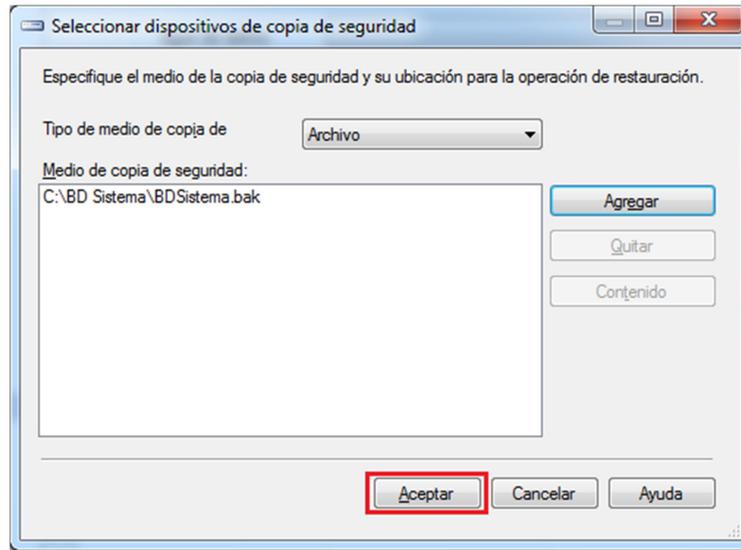
Gráfico 66.- Selección archivo de respaldo



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

- En esta ventana damos click en “Aceptar”.

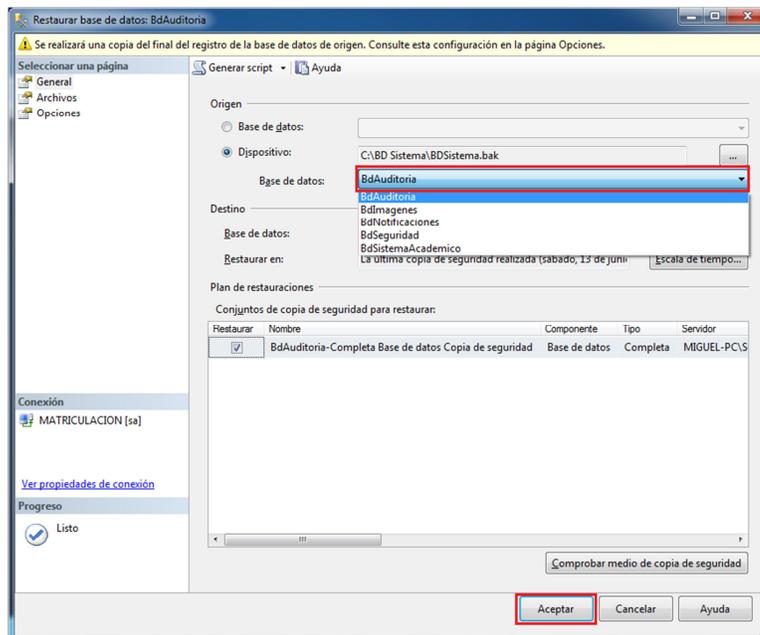
Gráfico 67.- Archivo seleccionado



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

7. En esta ventana elegimos la base que queremos importar, iremos importando una por una las bases, en este caso elegimos la base BDSistema.bak y damos click en "Aceptar".

Gráfico 68.- Restauración de BD



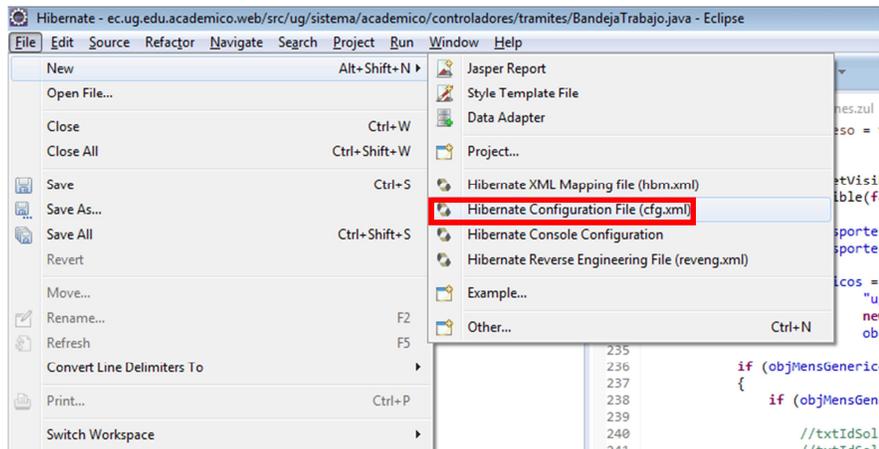
Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

8. Nos indica que la base se restauró correctamente y damos click en “Aceptar”.
9. Visualizamos del lado del panel izquierdo que la base Restaurada ya está disponible.

CONFIGURACION DE ARCHIVOS PARA MAPEO DE BASE SQL-SERVER

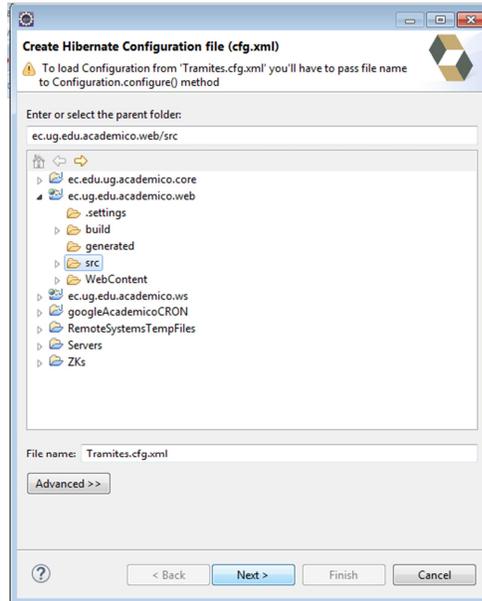
Una vez ya instalado y seleccionada la perspectiva de Hibernate procedemos a configurar los archivos necesarios para el mapeo de la base de datos. Se procede a crear el archivo de configuración (**cfg.xml**), esto los podemos realizar seleccionando en el menú de opciones: **File ->New -> Hibernate Configuration File (cfg.xml)**. Esto debe ser realizado en la perspectiva Hibernate

Gráfico 69.- Selección de archivo de configuración



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

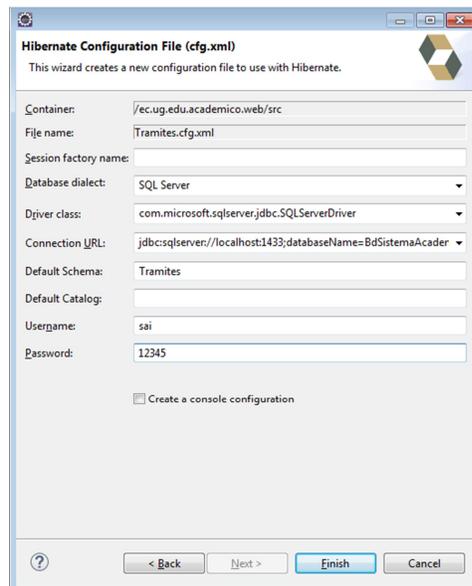
Gráfico 70.- Selección de la ruta para guardar el archivo de configuración



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

En esta parte escogemos la ruta donde guardaremos el archivo de configuración, y en esta misma pantalla le daremos el nombre con el que lo identificaremos para nuestro proyecto. Después de eso daremos clic en **Next**

Gráfico 71.- Configuración de conexión a la Base de Datos

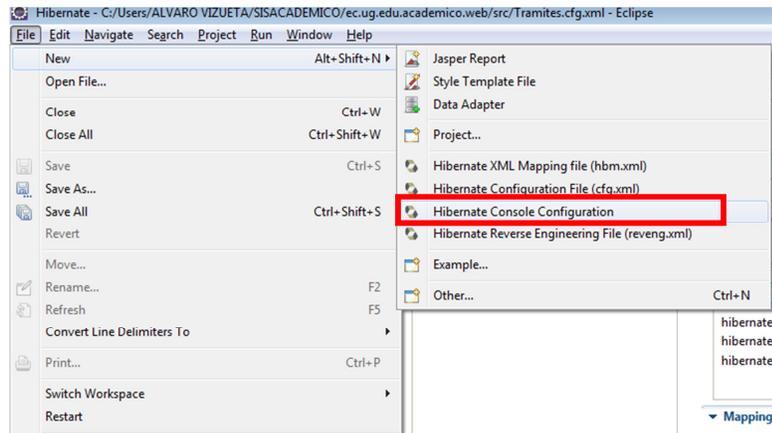


Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

En esta parte seleccionaremos el tipo de Base de Datos, Driver Class, Connection Url, Default Schema, Username y Password para establecer la conexión con la Base.

Ahora procederemos a crear el archivo de Consola, esto lo haremos desde el menú: **File -> New -> Hibernate Console configuration**.

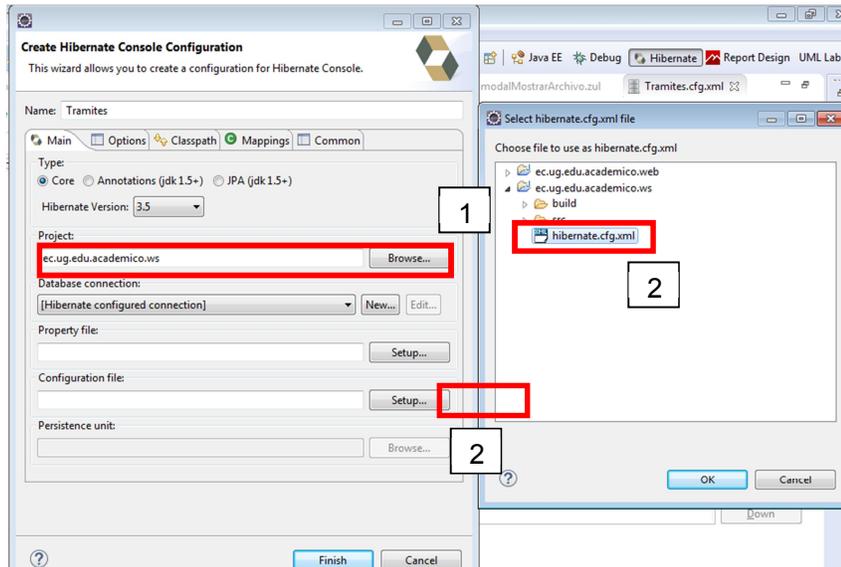
Gráfico 72.- configuración de Console Configuration



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

En esta parte utilizaremos el archivo que creamos anteriormente.

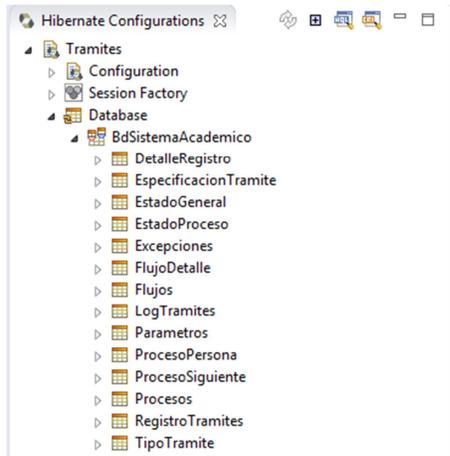
Gráfico 73.- Console Configuration



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

Aquí seleccionamos en que paquete del proyecto guardaremos la configuración (1), y en donde también seleccionaremos el archivo de configuración (cfg.xml) que creamos en primera instancia (2).

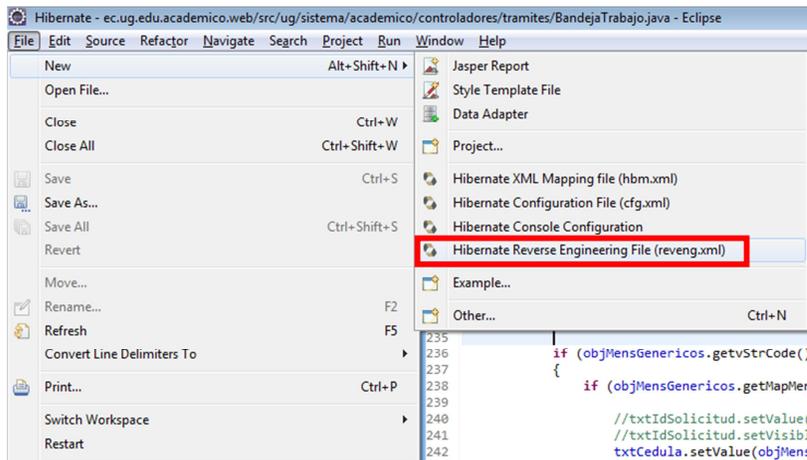
Gráfico 74.-Pre visualización de conexión a la Base



Elaboración: Álvaro Vizueta
Fuente: Álvaro Vizueta

En la perspectiva de Hibernate podremos verificar la conexión establecida con la base, y así probar que esta haya sido realizada correctamente. **Fig. 46**

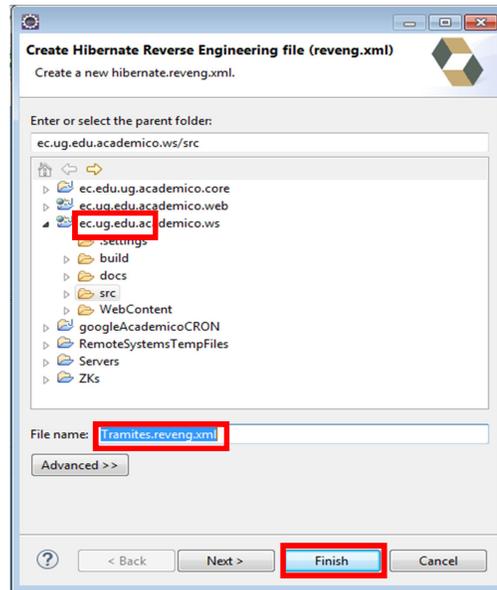
Gráfico 75.- Configuración de Ingeniería Inversa



Elaboración: Álvaro Vizueta
Fuente: Álvaro Vizueta

Después de realizar esto, crearemos el archivo de ingeniería inversa, esto lo podremos realizar desde el menú: **File-> New-> Hibernate Reverse Engineering File.**

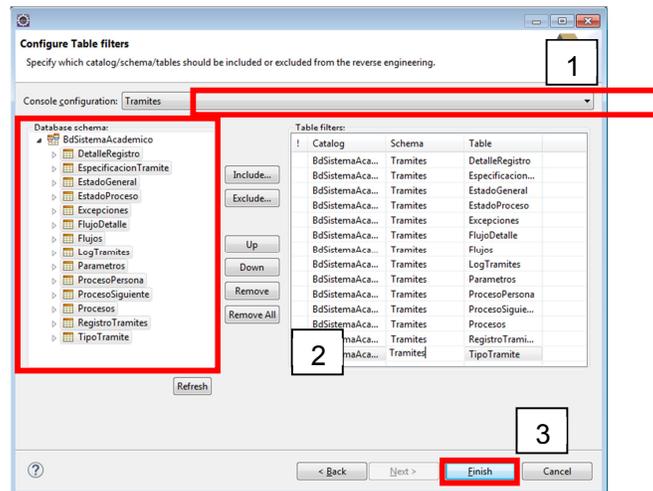
Gráfico 76.- Selección de ruta y nombre de archivo Reverse Engineering



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

Después de realizar esto daremos clic en **Next.**

Gráfico 77.- Selección de las tablas para el Mapeo



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

En esta pantalla seleccionaremos la configuración de consola que creamos anteriormente **(1)**.

Para enlistar las tablas que deseamos mapear damos clic en **Refresh**, después de eso se mostraran las tablas de la Base de datos, según el esquema que hayamos escogido, en esta parte podremos seleccionar las tablas que vamos a mapear sombreando la tabla respectiva y dando clic en **Include (2)**.

Después de seleccionar las tablas que necesitaremos daremos clic en **Finish (3)**.

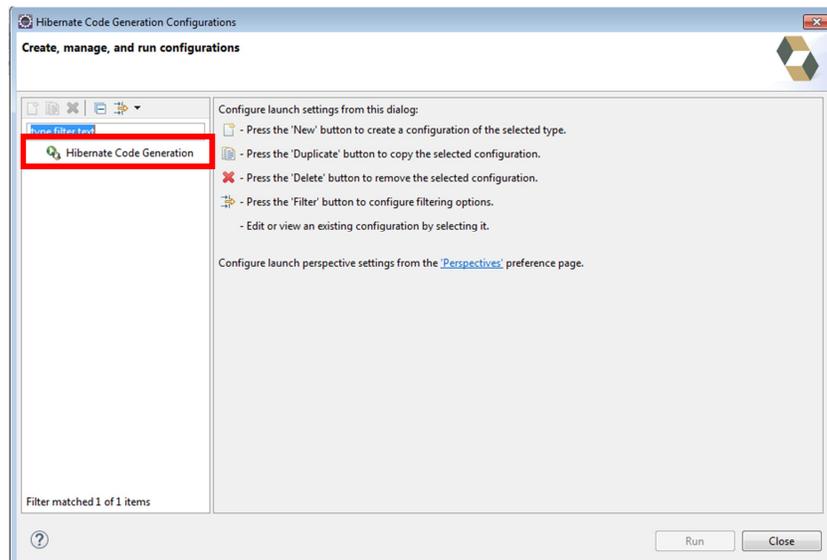
Gráfico 78.- Generación de código Hibernate



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

Ya con el archivo de ingeniería inversa creado, procederemos a generar el código respectivo de las tablas mapeadas.

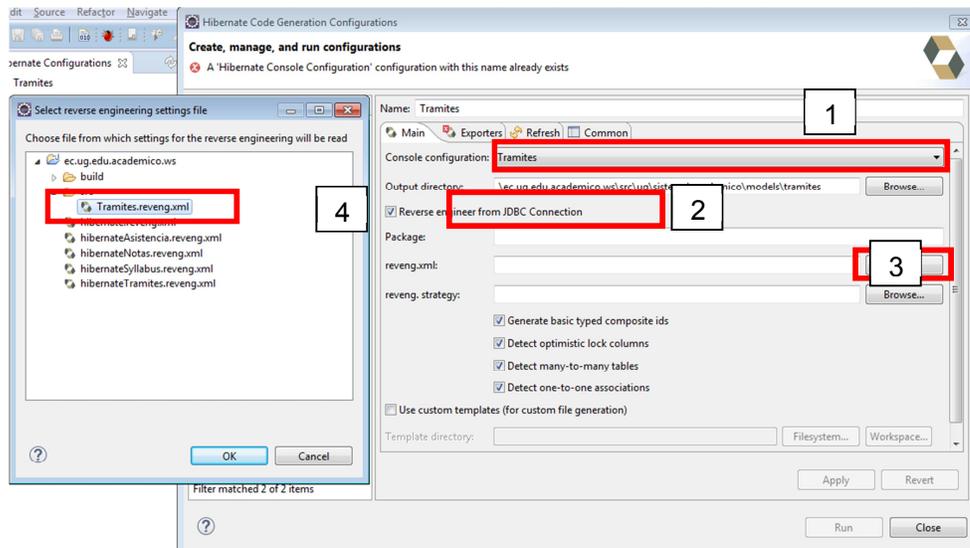
Gráfico 79.- Configuración para la generación de código.



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

En esta pantalla daremos doble clic en **Hibernate Code Generation**, para habilitar la configuración de la generación de código.

Gráfico 80.- Selección de archivo reveng.xml

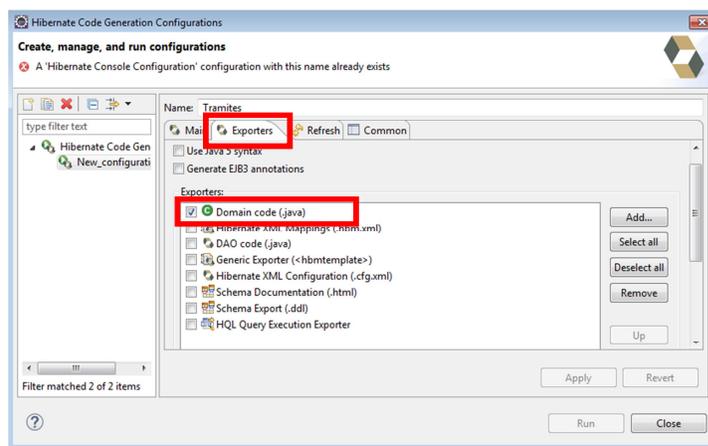


Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

En esta pantalla realizaremos lo siguiente:

1. Configuración de Consola ya creada.
2. Seleccionamos Reverse engineer from JDBC Connection.
3. Presionamos en Setup para seleccionar el archivo reveng correspondiente.
4. En esta pantalla seleccionaremos el archivo reveng y damos clic en ok.

Gráfico 81.- Configuración de Exporters

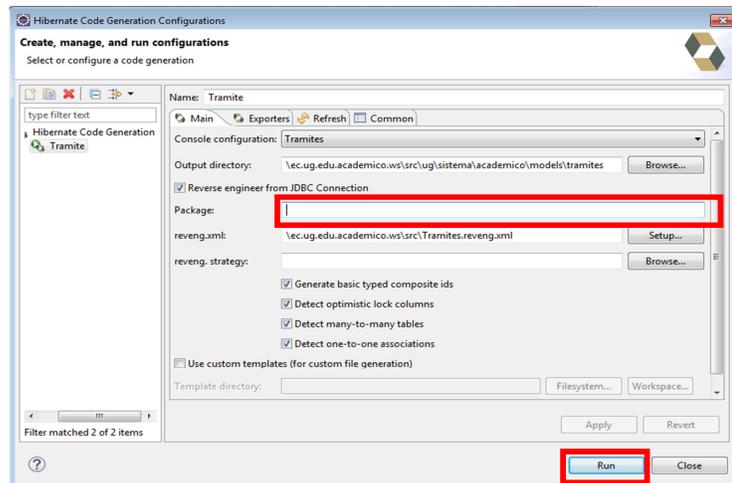


Elaboración: Álvaro Vizuela

Fuente: Álvaro Vizuela

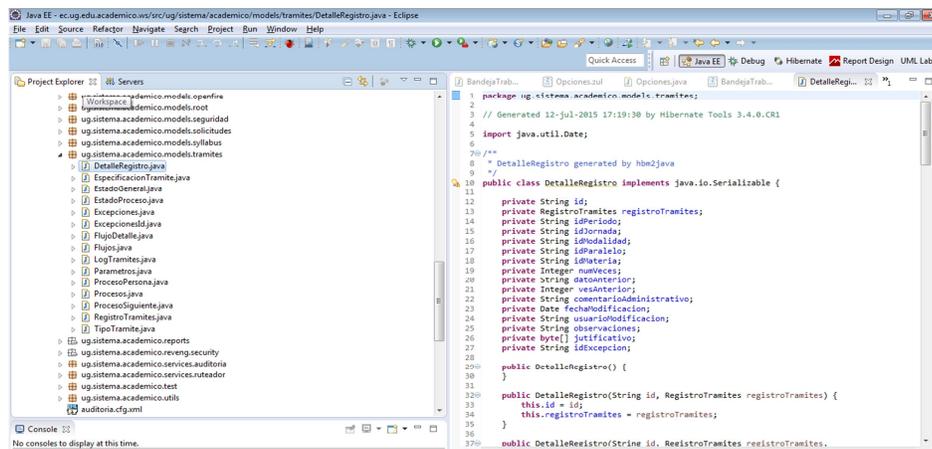
En la pestaña **Exporters**, seleccionaremos **Domain code (.java)**, Después de eso ingresaremos el nombre del paquete que deseamos para el proyecto, aplicamos los cambios y daremos clic en **Run** para generar los archivos de mapeo de la base de datos.

Gráfico 82.- Generación de código Hibernate



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

Gráfico 83.- Verificación de Clases generadas



Elaboración: Álvaro Vizuela
Fuente: Álvaro Vizuela

Cambiando de perspectiva y abriendo el explorador de proyecto, podremos verificar en la ruta que seleccionamos los archivos de mapeo generados.