



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMATICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN SISTEMAS

ESTUDIO DEL MODELO DE LÓGICA DIFUSA APLICADA EN UN MÓDULO EN
EL SISTEMA DE EVALUACIÓN INSTITUCIONAL.

TESIS DE GRADO

Previa la obtención del título de:
INGENIERO EN SISTEMAS INFORMÁTICOS

Presentado por:
ELVIA MARIELA CABAY HUEBLA

RIOBAMBA – ECUADOR

2010

AGRADECIMIENTO

Ante todo quiero expresar mi profundo agradecimiento a Dios por darme la perseverancia y la fuerza de voluntad necesaria para culminar esta etapa de mi vida.

Al Ing. Jorge Huilca por asesorarme y compartir conmigo sus conocimientos y sobre todo por brindarme su tiempo y paciencia.

A todas las personas, amigos, docentes, compañeros, que permitieron que este trabajo sea hoy una realidad.

DEDICATORIA

A mis padres por ser un pilar fundamental e incondicional en mi vida y a mis hermanos por ser la inspiración que me impulsó a continuar adelante y a no desmayar en los momentos difíciles, a ellos mi profunda gratitud por toda la comprensión y apoyo.

FIRMAS DE RESPONSABILIDAD

Ing. Jorge Huilca.

DIRECTOR DE TESIS

Dr. Alonso Alvares.

MIEBRO

Dr. Romeo Rodríguez.

MIEMBRO

DERECHOS DE AUTOR

Yo Elvia Mariela Cabay Huebla soy responsable de las ideas, doctrinas y resultados expuestos en esta Tesis, el patrimonio intelectual de la misma le pertenece a la Escuela Superior Politécnica de Chimborazo.

Elvia Mariela Cabay Huebla

INDICE DE ABREVIATURAS

INDICE GENERAL

PORTADA

AGRADECIMIENTO

DEDICATORIA

FIRMAS DE RESPONSABILIDAD

DERECHOS DE AUTOR

INDICE DE ABREVIATURAS

INDICE GENERAL

INDICE DE FIGURAS

INDICE DE TABLAS

INTRODUCCIÓN

CAPITULO I MARCO REFERENCIAL

<i>1.1 ANTECEDENTES</i>	- 21 -
1.1.1 Planteamiento del Problema.....	- 21 -
1.1.2 Análisis.....	- 24 -
1.1.3 Delimitación	- 25 -
1.1.4 Formulación del Problema	- 25 -
<i>1.2 JUSTIFICACIÓN</i>	- 26 -
1.2.1 Justificación Teórica	- 26 -
1.2.2 Justificación Aplicativa	- 27 -
<i>1.3 OBJETIVOS</i>	- 28 -
1.3.1 Objetivo General	- 28 -
1.3.2 Objetivos Específicos.....	- 28 -
<i>1.4 HIPÓTESIS</i>	- 28 -

CAPITULO II MARCO TEÓRICO

2.1 INTRODUCCION A LA LÓGICA DIFUSA - 29 -

 2.1.1 Historia De La Lógica Difusa - 30 -

 2.1.2 La Lógica Difusa - 32 -

 2.1.3 Conjuntos Difusos..... - 33 -

 2.1.4 Variables Lingüísticas. - 38 -

 2.1.4.1 Utilidad de las Variables Lingüísticas - 39 -

 2.1.4.2 Definición formal de las Variables Lingüísticas..... - 39 -

 2.1.5 Funciones de Pertenencia de Conjuntos Difusos. - 41 -

 2.1.6 Operaciones Con Conjuntos Difusos - 44 -

 2.1.7 Inferencia Difusa - 48 -

 2.1.7.1 Principio de Extensión - 49 -

 2.1.7.2 Relación Difusa - 49 -

 2.1.7.3 Métodos de Inferencia - 51 -

 2.1.7.4 Reglas Difusas - 52 -

 2.1.7.5 Dispositivos de inferencia difusa - 54 -

 2.1.8 Fuzzificador o Borrosificador..... - 55 -

 2.1.9 Defuzzificador o Desborrosificador - 56 -

 2.1.10 La Lógica Difusa Y La Probabilidad..... - 58 -

 2.1.11 Sistemas de Control Difuso - 58 -

 2.1.12 Ciclo de Análisis y Diseño usando Lógica Difusa..... - 61 -

 2.1.13 ¿Por qué usar los Sistemas de inferencia difusa? - 62 -

 2.1.14 Ventajas de un Sistema de Control Difuso - 63 -

 2.1.15 Áreas de Aplicación de la lógica Difusa - 64 -

 2.1.16 Inconvenientes de La Lógica Difusa..... - 65 -

SERVLETS..... - 66 -

 2.2.1 Estructura De Un Servlet - 67 -

2.2.2 Ciclo De Vida Un Servlet	- 70 -
2.2.3 Características De Los Servlets	- 71 -
2.2.4 Ventajas De Los Servlet	- 72 -
2.3 PÁGINAS JSP	- 73 -
2.3.1 Diferencias Entre JSP Y SERVLETS	- 74 -
2.3.2 Ventajas De JSP	- 75 -
2.3.3 Características De Las Páginas JSP	- 76 -
2.3.4 Elementos JSP	- 77 -
2.3.4.1 Código JAVA	- 77 -
2.3.4.2 Directivas	- 81 -
2.3.4.3 Acciones	- 82 -
2.3.5 OBJETOS IMPLÍCITOS DE JSP	- 88 -
2.3.5.1 request	- 89 -
2.3.5.2 response	- 90 -
2.3.5.3 out	- 90 -
2.3.5.4 Session	- 91 -
2.3.5.5 application	- 91 -
2.3.5.6 config	- 91 -
2.3.5.7 pageContext	- 92 -
2.3.5.8 page	- 92 -
2.3.6 FORMULARIOS EN PÁGINAS JSP	- 93 -
2.3.6.1. Conceptos básicos	- 93 -
2.2.6.2. GET y POST	- 93 -
2.2.6.3. Elementos de formulario	- 94 -
2.2.6.4. Botones radio.	- 95 -
2.2.6.5. Selecciones. Etiquetas <select> y <option>.	- 95 -
1.2.6.6. Codificación directa en el URL.	- 96 -

2.3.7 ACCESO A DATOS EN JSP	- 97 -
2.3.7.1 JDBC.....	- 97 -
2.3.7.2 Conectividad JDBC.....	- 98 -
2.3.7.3 Acceso De JDBC A Bases De Datos.....	- 99 -
2.3.7.4 Interfaces JDBC.....	- 101 -

CAPITULO III MARCO APLICATIVO

3.1 CASO DE ESTUDIO	- 114 -
3.1.1 Aspectos para la evaluación de un docente.....	- 115 -
3.1.2 Grupos definidos para la evaluación	- 116 -
3.1.3 Cuestionarios definidos para la evaluación.....	- 118 -
3.1.4 Informes De Evaluación.....	- 119 -
3.2 DISEÑO Y APLICACIÓN DEL SISTEMA DE LÓGICA DIFUSA.....	- 120 -
3.2.1 Análisis.....	- 120 -
3.2.1.1 Definición de Objetivos, Restricciones y Comportamiento del Sistema.....	- 120 -
3.2.1.2 Identificar las variables de Entrada y Salida	- 122 -
3.2.1.3 Definir procedimientos y criterios para la prueba y validación.....	- 124 -
3.2.2 Diseño.....	- 125 -
3.2.2.1 Definir los conjuntos difusos de cada etiqueta lingüística.	- 125 -
3.2.2.2 Definir el marco de representación de la solución	- 127 -
3.2.2.3 Definir el marco de los procedimientos empleados (Inferencia).	- 130 -
3.2.2.4 Especificar la forma requerida de la salida del sistema (decodificación).	- 130 -
3.2.2.5 Ejecutar pruebas para validar el sistema.....	- 130 -
3.2.2.6 Verificar si la solución es compatible con el paso 3.2.1.1:	- 131 -

CAPITULO IV DESARROLLO DEL MÓDULO DE ESTADÍSTICAS DEL SISTEMA DE EVALUACION DOCENTE DE LA ESPOCH

INICIACIÓN	- 134 -
4.1 INGENIERÍA DE LA INFORMACIÓN.....	- 134 -
4.1.1 Definición del Ámbito del problema	- 134 -
4.2. DEFINICIÓN TÉCNICA DE REQUERIMIENTOS (SRS)	- 136 -
4.2.1 INTRODUCCION	- 137 -
4.2.1.1 Objetivo	- 137 -
4.2.1.2 Ámbito	- 137 -
4.2.1.3 Definiciones, Acrónimos, Abreviaturas	- 139 -
4.2.1.4 Referencias.	- 140 -
4.2.1.5 Visión General	- 140 -
4.2.2 DESCRIPCION GENERAL	- 141 -
4.2.2.1 Perspectiva del Producto	- 141 -
4.2.2.2 Funciones del Producto	- 142 -
4.2.2.3 Características de los usuarios.....	- 143 -
4.2.2.4 Limitaciones Generales.....	- 143 -
4.2.2.5 Supuestos y dependencias	- 144 -
4.2.3 REQUERIMIENTOS ESPECIFICOS.....	- 145 -
4.2.3.1 Requisitos Funcionales	- 145 -
4.2.3.2 Requisitos de Comportamiento.....	- 161 -
4.2.3.3 Limitaciones de Diseño.....	- 162 -
4.2.3.4 Atributos.....	- 162 -
4.2.3.5 Requisitos de Interfaces Externas	- 162 -
4.2.3.6 Otros Requerimientos.	- 164 -
4.3 PLANIFICACIÓN DEL PROYECTO	- 164 -
4.3.1. Definir y Estimar recursos	- 164 -
4.3.1.1. Identificar una técnica de Estimación.....	- 164 -

4.3.1.2. Aplicar la técnica.....	- 165 -
4.3.1.3. Evaluar los resultados.....	- 169 -
4.3.2. Planificación Temporal.....	- 170 -
4.3.2.1 Cronograma de trabajo	- 170 -
4.3.2.2 Diagramas Gantt.....	- 170 -
4.3.3. Análisis y Gestión de Riesgos.....	172
4.3.3.1. Identificación de Riesgos	172
4.3.3.2. Categorización de Riesgos	174
4.3.3.3. Plan de Contingencia	175
4.3.4 Estudio de la Factibilidad.....	183
4.3.5.1 Factibilidad Operativa.....	183
4.3.5.2 Factibilidad Técnica	183
4.3.5.3 Factibilidad Económica	184
4.3.5.4 Factibilidad Legal	184
4.4 <i>DEFINIR CASOS DE USO DE ALTO NIVEL</i>	184
4.5 <i>MODELO CONCEPTUAL</i>	187
4.6 <i>DEFINIR BORRADOR DE LA ARQUITECTURA DEL SISTEMA</i>	187
ELABORACIÓN	188
4.7 <i>DEFINIR CASOS DE USO ESENCIALES EN FORMATO EXPANDIDO</i>	188
4.8 <i>REFINAR DIAGRAMAS DE CASOS DE USO</i>	189
4.9 <i>REFINAR EL MODELO CONCEPTUAL</i>	193
4.10 <i>DEFINIR CONTRATOS DE OPERACIÓN</i>	194
4.11 <i>DEFINIR LOS DIAGRAMAS DE ESTADO</i>	195
4.12 <i>DEFINIR LOS DIAGRAMAS DE ACTIVIDAD</i>	197

CONSTRUCCION.....	200
4.13 <i>DEFINIR CASOS DE USO REALES</i>	200
4.14 <i>DEFINIR INFORMES DE INTERFÁZ DE USUARIO</i>	205
4.15 <i>DEFINIR DIAGRAMAS DE INTERACCIÓN</i>	205
4.15.1 Definir Diagramas De Secuencia.....	205
4.15.2 Definir Diagramas de Colaboración.....	207
4.17 <i>DEFINIR EL ESQUEMA DE LA BASE DE DATOS</i>	208
4.17.1 DICCIONARIO DE DATOS	- 209 -
<i>Fuente: Base de datos SIEDD</i>	- 213 -
4.18 <i>MODELO FÍSICO Y ARQUITECTURA DEL SISTEMA</i>	- 214 -
4.18.1 Diagrama de Componentes.....	- 214 -
4.18.2 Diagrama de Despliegue.....	- 214 -
4.19 <i>IMPLEMENTACIÓN Y PRUEBAS</i>	- 215 -
4.19.1 Definición De Estándares De Programación.....	- 215 -
4.19.2 Pruebas Unitarias	- 216 -
4.19.3 Pruebas De Caja Negra.....	- 216 -
TRANSICIÓN	- 222 -
4.20 <i>IMPLANTACIÓN DEL SISTEMA</i>	- 222 -
 CAPITULO V ANÁLISIS DE RESULTADOS	
5.1 <i>OPERACIONALIZACIÓN DE VARIABLES</i>	- 224 -
5.1.2. OPERACIONALIZACIÓN CONCEPTUAL	- 224 -
5.1.3. OPERACIONALIZACIÓN METODOLÓGICA.....	- 224 -
5.2 <i>ESTADÍSTICA INFERENCIAL</i>	- 225 -
5.2.1 SELECCIÓN Y DETERMINACIÓN DE LA MUESTRA	- 226 -

5.2.2 OBTENCIÓN DE LOS DATOS.....	- 227 -
5.2.3 CLASIFICACIÓN Y ORGANIZACIÓN DE LOS DATOS.....	- 228 -
5.2.4 ANÁLISIS DE LOS DATOS.....	- 230 -
5.2.6 VALIDACIÓN DE LA HIPÓTESIS.....	- 231 -
5.2.7 CONCLUSIONES	- 231 -

CONCLUSIONES

RECOMENDACIONES

RESUMEN

SUMARY

GLOSARIO

ANEXOS

BIBLIOGRAFÍA

INDICE DE FIGURAS

Figura. II.1 Universo Discreto de un conjunto difuso.....	- 35 -
Figura. II.2 Universo Continuo de un conjunto difuso.	- 35 -
Figura. II.3 Términos de un conjunto difuso.	- 37 -
Figura. II.4 Granularidad fina.	- 40 -
Figura. II.5 Granularidad gruesa.....	- 40 -
Figura. II.6 Función tipo Trapezoidal	- 41 -
Figura. II.7 Función tipo singleton	- 42 -
Figura. II.8 Función tipo T.....	- 43 -
Figura. II.9 Función Tipo π	- 44 -
Figura. II.10 t-norma y t-conorma	- 47 -
Figura. II.11 Resultado de la agregación	- 56 -
Figura. II.12 Resultado de la Defuzzificación.....	- 56 -
Figura. II.13 Ejemplo método de fuzzificación del máximo.....	- 57 -
Figura. II.14 Ejemplo método de fuzzificación del Centroide	- 57 -
Figura. II.15 Estructura de un Sistema Difuso	- 59 -
Figura. II.16 Clases Relevantes de Servlets e Interfaces.....	- 67 -
Figura. II.17. Componentes Java.	- 99 -
Figura. II.18. Modelo de dos Capas.....	- 100 -
Figura. II.19 Modelo en tres capas.....	- 101 -
Figura. II.20 Clases y Objetos del JDBC	- 102 -
Figura. II.21 Principales Clases de JDBC.....	- 104 -
Figura. III.1 Variable lingüística pedagogía y conocimiento.....	- 125 -
Figura. III.2 Variable lingüística Evaluación.....	- 126 -
Figura. III.3 Variable lingüística Desempeño	- 126 -
Figura. III.4 Variable lingüística Relaciones Humanas y Atención al Estudiante.	- 126 -
Figura. III.5 Variable Lingüística Promedio Total.....	- 127 -
Figura. III.6 Variable Lingüística promedio Institucional	- 127 -

Figura. III.7 Representación del conocimiento con las variables de entrada.	- 128 -
Figura.III.8 Representación de la variable de Salida	- 129 -
Figura.III.9 Representación Variable de Salida Institucional	- 129 -
Figura IV.1 Funciones del producto	- 142 -
Figura IV.2 Resultados COCOMO.....	- 168 -
Figura IV.3 Resultados COCOMO costos.....	- 169 -
Figura IV.4 Cronograma de Trabajo.....	- 170 -
Figura IV.5 Diagrama Gantt	171
Figura IV.6 Modelo conceptual del módulo de estadísticas SIEDD.....	187
Figura IV.7 Arquitectura del Sistema SIEDD	187
Figura IV.8 Caso de Uso General Resultados de la evaluación a un Docente.....	188
Figura IV.9 Caso de Uso General Resultados de la evaluación a un nivel Institucional.	189
Figura IV.10 Resultados De La Evaluación Estudiante Docente.....	190
Figura IV.11 Resultados De La Autoevaluación Docente	190
Figura IV.12 Caso de uso Resultado por Facultad.....	191
Figura IV.13 Caso de uso resultados por Escuela.....	191
Figura IV.14 Caso de uso resultados por Carrera.	192
Figura IV.15 Caso de uso resultado por preguntas de una Carrera.....	192
Figura IV.16 Caso de uso resultados por pregunta de un carrera.....	193
Figura IV.18 Diagrama de Estado usuario Docente.....	195
Figura IV.19 Diagrama de Estado Usuario General	196
Figura IV. 20 Diagrama De Actividades Evaluación Estudiantes	197
Figura IV.21 Diagrama de Actividades Autoevaluaciones.....	198
Figura IV.22 Diagrama de actividades Usuario General.	199
Figura IV.23 Diagrama de secuencia Usuario Docente Evaluacion estudiantes.....	205
Figura IV.24 Diagrama de Secuencia Usuario Docente Autoevaluación	206
Figura IV.25 Diagrama de Secuencia Resultados institucionales.....	206
Figura IV.26 Diagrama de Colaboración Resultados evaluación estudiante	207
Figura IV.27 Diagrama de Colaboración Resultados Autoevaluación	207

Figura IV.28 Diagrama de Colaboración Resultados institucionales.....	207
Figura IV. 29 Esquema de la Base de Datos.....	208
Figura IV.30 Diagrama de Componentes del Sistema	- 214 -
Figura IV.31 Diagrama de Despliegue del sistema.....	- 214 -
Figura V.1 Distribución Z.....	- 231 -

INDICE DE TABLAS

TABLA II.1 OPERACIONES DE CONJUNTOS DIFUSOS	- 48 -
TABLA II.2 AREAS DE APLICACIÓN SE SISTEMAS DIFUSOS.....	- 65 -
TABLA II.3 CLASES E INTERFACES DEL DRIVER JDBC	- 103 -
TABLA III.4 TABLA DE VALORACION DE RESULTADOS DE LA EVALUACIÓN DOCENTE.....	- 119 -
TABLA III.5 TABLA DE VALORACIÓN DE RESULTADOS DE LA EVALUACIÓN INSTITUCIONAL.....	- 120 -
TABLA IV.6 DEFINICIONES	- 139 -
TABLA IV.7 ACRÓNIMOS	- 139 -
TABLA IV.8 PUNTOS DE FUNCION ILF	- 165 -
TABLA IV.9 PUNTOS DE FUNCION EI.....	- 165 -
TABLA IV.11 PUNTOS DE FUNCION EO Entrada Externa	- 166 -
TABLA IV.12 PUNTOS DE FUNCION EO Salida Externa.....	- 166 -
TABLA IV.13 PUNTOS DE FUNCION EQ Entrada Externa	- 167 -
TABLA IV.14 PUNTOS DE FUNCION EQ Entrada Externa DET y FTR.....	- 167 -
TABLA IV.15 PUNTOS DE FUNCION EQ Entrada Externa Complejidad	- 167 -
TABLA IV.16 PUNTOS DE FUNCION RESUMEN.....	- 168 -
TABLA IV.17 RIESGOS IDENTIFICADOS	172
TABLA IV.18 CATEGORIZACIÓN DE RIESGOS	174
TABLA IV.19 HOJA DE GESTIÓN DE RIESGOS R2	175
TABLA IV.20 HOJA DE GESTIÓN DE RIESGOS R15	176
TABLA IV.21 HOJA DE GESTIÓN DE RIESGOS R10.....	177
TABLA IV.22 HOJA DE GESTIÓN DE RIESGOS R3	178
TABLA IV.23 HOJA DE GESTIÓN DE RIESGOS R7	179
TABLA IV.24 HOJA DE GESTIÓN DE RIESGOS R1	180

TABLA IV.25 HOJA DE GESTIÓN DE RIESGOS R17	181
TABLA IV.26 HOJA DE GESTIÓN DE RIESGOS R16	182
TABLA IV.27 CASO DE USO GENERAL RESULTADOS DE LA EVALUACIÓN A UN DOCENTE	185
TABLA IV.28 CASO DE USO GENERAL RESULTADOS DE LA EVALUACIÓN A UN NIVEL INSTITUCIONAL.....	186
TABLA IV.29 CONTRATO DE OPERACIÓN VALIDARSE.....	194
TABLA IV.30 CONTRATO DE OPERACIÓN CONSULTAR RESULTADOS.....	194
TABLA IV.31 CONTRATO DE OPERACIÓN CONSULTAR RESULTADOS INSTITUCIONALES	195
TABLA IV.32 RESULTADOS DE LA EVALUACIÓN ESTUDIANTE DOCENTE	200
TABLA IV.33 RESULTADOS DE LA AUTOEVALUACIÓN DOCENTE.....	201
TABLA IV.34 RESULTADOS INSTITUCIONALES POR FACULTAD	202
TABLA IV.35 RESULTADOS INSTITUCIONALES POR ESCUELA	202
TABLA IV.36 RESULTADOS INSTITUCIONALES POR CARRERA	203
TABLA IV.37 RESULTADOS INSTITUCIONALES POR PREGUNTA DE UNA CARRERA	204
TABLA IV.38 DICCIONARIO DE DATOS	- 209 -
TABLA IV.38 ORGANIZACIÓN DE RESULTADOS	- 229 -

INTRODUCCIÓN

El desempeño docente en la universidad ecuatoriana es el pilar fundamental en la formación de nuevos profesionales, es por ello que la Escuela Superior Politécnica de Chimborazo promueve a través de la comisión de evaluación institucional la evaluación docente y de acuerdo con la tecnología promueve la creación del nuevo sistema integrado de evaluación al desempeño docente.

El presente trabajo presenta un estudio de la lógica difusa con el objetivo de proporcionar información al docente que sea fiable y que le permita tomar una decisión sobre su desempeño, es por ello que en el capítulo II se realiza un estudio de las principales características que forman parte de la lógica difusa, así como su estructura y principales componentes. De la misma forma es necesario conocer las herramientas con las que se desarrollará la aplicación por lo que se realiza una descripción de lo que es JSP con la herramienta Netbeans sus componentes y directivas. El capítulo III muestra la lógica difusa aplicada al módulo de resultados de la el nuevo sistema de evaluación SIEDD mientras que el capítulo IV se realiza la ingeniería de la aplicación basado en la metodología RUP. El capítulo V está dedicado a la comprobación de la hipótesis

CAPÍTULO I

MARCO REFERENCIAL

1.1 ANTECEDENTES

1.1.1 Planteamiento del Problema

La Escuela Superior Politécnica de Chimborazo, a través de la Comisión De Evaluación Institucional ha realizado evaluaciones de: funciones en Docencia, Investigación, Vinculación con la colectividad; y, Gestión Administrativa, a fin de asegurar niveles de calidad en todos los procesos, proporcionando a la persona una valoración que según una escala puede ser considerada para un análisis por parte de la comisión. Es necesario recalcar, que el sistema de Evaluación Institucional está basado en promedios y ponderaciones.

La Comisión de Evaluación Institucional es la encargada de presentar a las autoridades respectivas los resultados de las evaluaciones efectuadas; así como también, asesorar a las autoridades y organismos institucionales sobre los mismos.

La Docencia es una de las funciones trascendentales de la Universidad Ecuatoriana , la Escuela Superior Politécnica de Chimborazo se sustenta en la adquisición apropiada, la actualización ágil y el uso oportuno de los conocimientos y preceptos; por estas razones, la Comisión de Evaluación Institucional, promovió el desarrollo del Sistema de Evaluación del Desempeño del Ejercicio Docente de la Escuela Superior Politécnica de Chimborazo, el mismo que busca: analizar el trabajo y desempeño académico del docente en el continuo lógico, ámbitos, características, estándares, indicadores, y con ello construir información relevante, eficiente y de calidad, además de facilitar al nivel directivo académico información oportuna para la toma de decisiones.

Una de las grandes ventajas que proporciona la evaluación en cualquier aspecto, es conocer las falencias y fortalezas del objeto o persona evaluada. Algo que serviría de gran ayuda sería poder determinar y proporcionar información de dichas falencias y como se podría mejorar. Enfocados en este aspecto, una de las alternativas que se puede tomar en cuenta, es la aplicación de un modelo de representación y procesamiento del conocimiento; como lo es la **Lógica Difusa** que es un método de razonamiento de maquina similar al pensamiento humano, que puede procesar información incompleta o incierta. En Inteligencia artificial, la Lógica Difusa o Lógica Borrosa, se utiliza para la resolución de una variedad de problemas, principalmente los relacionados con control de procesos industriales complejos y sistemas de decisión en general, la resolución y la comprensión de datos. Los sistemas basados en lógica difusa imitan la forma en que toman decisiones los humanos, con la ventaja de ser mucho más rápidos. Estos sistemas son generalmente robustos y tolerantes a imprecisiones y ruidos en los datos de entrada. Consiste en la aplicación de la lógica difusa con la intención de imitar el razonamiento humano en la programación de computadoras. Con la lógica convencional, las

computadoras pueden manipular valores estrictamente duales como verdadero/falso, sí/no. En la lógica difusa, se usan modelos matemáticos para; representar nociones subjetivas como: *caliente, tibio, frío*, para valores concretos que puedan ser manipuladas por los ordenadores.

A continuación presentamos temas de tesis similares u orientadas a la misma rama de desarrollo que la propuesta mencionada anteriormente:

“ESTUDIO DE LA INTELIGENCIA ARTIFICIAL Y DE LA INGENIERIA DE SOFTWARE APLICADO A LA CONSTRUCCIÓN DE UN SISTEMA EXPERTO PARA EL DIAGNOSTICO DE ENFERMEDADES, MEJORAMIENTO GENETICO Y MANEJO DE GANADO LECHERO”

“APLICACIÓN DE LAS TÉCNICAS DE LA INTELIGENCIA ARTIFICIAL Y LA METODOLOGÍA UML PARA LA CONSTRUCCIÓN DE UN CONSULTORIO MEDICO ABSTRACTO”

“DESARROLLO DE UN SISTEMA INFORMATICO DE CONTROL CLINICO PARA PACIENTES UTILIZANDO INTELIGENCIA ARTIFICIAL COMO UNA HERRAMIENTA DE AYUDA EN LA MEDICINA CON PUNTOS DE -CONSULTA MEDICA REMOTOS”

“ESTUDIO DE DESARROLLO DE SISTEMAS EXPERTOS EN PROLOG USANDO XPCE EN EL DESARROLLO DEL SISTEMA EXPERTO PARA EL DIAGNOSTICO DE LABORATORIO Y TRATAMIENTO DE ENFERMEDADES VETERINARIAS”

Los temas mencionados hacen referencia a la misma rama del tema propuesto con la diferencia que se tratará directamente con la lógica difusa como un modelo de procesamiento de la información y que se encuentra dentro de lo que son los sistemas expertos y la inteligencia artificial, además que su campo de aplicación es en el sistema de evaluación institucional.

1.1.2 Análisis

El tema está haciendo referencia al Estudio del modelo de representación y procesamiento del conocimiento de la lógica difusa, aplicada al desarrollo de un módulo en el Sistema de Evaluación del Desempeño del Ejercicio Docente de la Escuela Superior Politécnica de Chimborazo; cuyo objetivo es, estudiar este modelo como parte de la inteligencia Artificial y poder aplicarlo a las evaluaciones docentes que se realizan continuamente en la Escuela Superior Politécnica de Chimborazo proporcionando así información apegada a la realidad.

El Sistema de Evaluaciones de la ESPOCH, proporciona un resultado a cada persona evaluada, siendo este considerado como bueno o malo, debido a que está basado en promedios y ponderaciones. El objetivo del desarrollo de un módulo para el Sistema de Evaluación del Desempeño del Ejercicio Docente de la Escuela Superior Politécnica de Chimborazo basado en la lógica difusa, es proporcionar a la persona evaluada además de su calificación, información que el docente pueda comprender y entender sobre los resultados obtenidos en dicha evaluación.

La Comisión de Evaluación Institucional servirá como base funcional para el desarrollo de la parte aplicativa de nuestra propuesta de investigación.

1.1.3 Delimitación

La tesis a desarrollar se limita específicamente a: La comisión de Evaluación Institucional, puesto que son quienes manejan los resultados de las evaluaciones realizadas y poseen la información referente a esto, así como también, el modulo se desarrollará en el Sistema de Evaluación del Desempeño del Ejercicio Docente de la Escuela Superior Politécnica de Chimborazo.

1.1.4 Formulación del Problema

¿Permitirá el modelo de la lógica difusa obtener mejores conclusiones a la hora de la toma de dediciones basados en los resultados obtenidos del Sistema de Evaluación del Desempeño del Ejercicio Docente de la ESPOCH?

1.1.5 Sistematización del Problema

- ¿Cuáles son los principales problemas que presenta el Sistema de Evaluación del Desempeño del Ejercicio Docente de la Escuela Superior Politécnica de Chimborazo?
- ¿Cuáles son los principales problemas que se presentan en el proceso de toma de

decisiones basados en los resultados obtenidos en el Sistema de Evaluación del Desempeño del Ejercicio Docente?

- ¿Qué es el modelo de la Lógica Difusa?
- ¿Cuáles son las ventajas de utilizar el modelo de Lógica Difusa?
- ¿Se permitirá obtener información acerca del proceso de evaluación institucional?

1.2 JUSTIFICACIÓN

1.2.1 Justificación Teórica

Las evaluaciones realizadas en la Escuela Superior Politécnica de Chimborazo si bien poseen un grado de aceptación amplia de la parte evaluadora; la parte evaluada requiere que también se proporcione información de lo que se podría realizar según las calificaciones obtenidas ya sean estas consideradas como muy buenas o malas, es por eso que se ha tomado en cuenta el modelo de procesamiento y representación de conocimiento de lógica difusa; puesto que, su razonamiento se basa en la aproximación a la percepción humana, por lo que se podría dar soluciones o sugerencias para las preguntas con menor puntaje mas apegadas a la realidad de lo evaluados.

La Lógica Difusa ha sido probada para ser particularmente útil en sistemas expertos y otras aplicaciones de inteligencia artificial, y se usan modelos matemáticos para representar nociones subjetivas como: caliente, tibio, frío, para valores concretos que puedan ser manipuladas por los ordenadores permitiendo así obtener valores que se acercan a lo verdadero.

Una de las principales ventajas de la Lógica difusa es la velocidad en obtener una salida con una gran fiabilidad. Nos permiten solucionar gran parte de los problemas de control de una manera sencilla, sin necesidad de conocer un modelo matemático que lo pueda controlar.

Los sistemas de lógica difusa reúnen los conocimientos de humanos expertos en un área específica, de esta forma los datos son recopilados y representados de una manera mas precisa.

A través de nuestra propuesta de investigación, se pretende obtener que las evaluaciones institucionales del ejercicio docente no solo arrojen datos numéricos o estadísticos sino también valores que el docente puede entender y comprender en lenguaje natural.

1.2.2 Justificación Aplicativa

Las evaluaciones realizadas en la ESPOCH son de gran utilidad pues, mediante este sistema se obtiene información de cómo las personas evaluadas están rindiendo en sus labores, a pesar de todo ello, los resultados que se obtiene son solo porcentajes numéricos, es por eso que el desarrollo de un módulo basado en la lógica difusa implementado en el sistema de evaluación institucional, ayudará a las personas evaluadas a observar los aspectos que se tomaron en cuenta para su valoración y como estas influyen en su valoración total.

1.3 OBJETIVOS

1.3.1 Objetivo General

- Realizar un estudio del modelo de lógica difusa y aplicarla en un módulo en el sistema de evaluaciones de la Espoch.

1.3.2 Objetivos Específicos

- Determinar y analizar el proceso de evaluación que sigue de la ESPOCH.
- Determinar los principales problemas en el proceso de toma de decisiones realizados en base a los resultados obtenidos en el sistema de Evaluación del Desempeño del Ejercicio Docente de la ESPOCH.
- Establecer la forma de mejorar los resultados obtenidos en las evaluaciones docentes realizadas.
- Investigar el modelo de representación y procesamiento del conocimiento conocido como lógica difusa.
- Desarrollar un módulo basado en la lógica difusa que permita mejorar el proceso de toma de decisiones de la Comisión de Evaluación institucional.

1.4 HIPÓTESIS

El uso de la lógica difusa en un módulo del Sistema de Evaluación del Desempeño del Ejercicio Docente aumentará el número de parámetros analizados para la toma de decisiones.

CAPÍTULO II

MARCO TEÓRICO

2.1 INTRODUCCION A LA LÓGICA DIFUSA

Uno de los desafíos a los que se enfrentan las ciencias y las matemáticas en el último siglo es el manejo del concepto de incertidumbre. En las ciencias, en general, en los últimos años esta preocupación se refleja en el hecho de que un concepto en principio indeseable, que se debería evitar a toda costa, se está convirtiendo en un concepto admisible con el que se puede trabajar y además, se quiera o no, no se puede evitar, pues el mundo que nos rodea es en gran medida incierto en muchos de sus aspectos.

El comportamiento de muchos de los sistemas que nos rodean dista mucho de lo exacto y lo preciso. Es por lo tanto aquí donde surge la idea de la incertidumbre. Las cosas no son ni blancas ni negras, si no que pueden ser de color gris, y debemos descartar la idea de solución única. La solución no es «lo blanco» o lo «negro», si no que una solución

aceptable en función de un determinado criterio puede ser un ligero tono «gris claro». La incertidumbre considerada aisladamente puede resultar indeseable, sin embargo en determinados sistemas puede ayudar a reducir la complejidad y a aumentar la credibilidad del sistema que pretendemos simular. Finalmente podemos hablar de imprecisión en la descripción de algo y entonces podremos usar adjetivos del tipo: vago, difuso, posible. Es aquí donde aparece lo que se denomina lógica difusa pues esta puede emular matemáticamente un problema que puede tener no solo una solución determinada sino que se puede determinar distintos valores que pueden dar la solución a dicho problema.

2.1.1 Historia De La Lógica Difusa

En 1965 en la Universidad de California en Berkeley, **Lotfi A. Zadeh**, considerado ahora el padre de la Lógica Difusa, introdujo el concepto de la lógica difusa guiado por el principio de que las matemáticas pueden ser usadas para encadenar el lenguaje con la inteligencia humana; pero la idea de la lógica difusa o borrosa no es nueva, sus orígenes se remontan hasta 2.500 años. Aún Aristóteles consideraba que existían ciertos grados de veracidad y falsedad. Platón había considerado ya grados de pertenencia.

Particularmente la escuela americana de la filosofía llamada pragmatismo fundada a principios de siglo por Charles Sanders Peirce, fue el primero en considerar "vaguedades", más que falso o verdadero, como forma de acercamiento al mundo y a la forma en que la gente funciona.

La idea de que la lógica produce contradicciones fue popularizada por el filósofo y matemático británico Bertrand Russell, a principios del siglo XX. Estudio las

vaguedades del lenguaje, concluyendo con precisión que la vaguedad es un grado. La primera lógica de vaguedades fue desarrollada en 1920 por el filósofo Jan Lukasiewicz, visualizó los conjuntos con un posible grado de pertenencia con valores de 0 y 1, después los extendió a un número infinito de valores entre 0 y 1.

Uno de los países donde más éxito ha tenido los sistemas borrosos ha sido en Japón. Empresas como Fuji Elec.& TIT han desarrollado aplicaciones de control fuzzy para el proceso de purificación del agua, Hitachi con una aplicación de control fuzzy para el Metro en Sendai City o Matsushita con una aplicación de control fuzzy para la unidad de suministro de agua caliente para uso doméstico.

Merece una mención especial la creación de LIFE (Laboratory for International Fuzzy Engineering Research) en marzo de 1989, creado por el Ministerio de Comercio Internacional e Industria en Japón (MITI). El capital de LIFE es al 50% de compañías privadas japonesas y del propio Ministerio, y su presidente es miembro del Instituto de Tecnología de Tokio (TIT). En su sede trabajan en la actualidad alrededor de 30 investigadores a tiempo completo.

Pero también en USA las empresas han comenzado a aplicar la lógica borrosa a sus desarrollos y proyectos. Entre otras encontramos a la NASA, Boeing, Rochwell, Bell o a Ford Motor Co., que experimenta con un sistema de aparcamiento automático para camiones con remolque.

2.1.2 La Lógica Difusa

La lógica difusa o borrosa (Fuzzy logic) descansa en la idea que en un instante dado, no es posible precisar el valor de una variable X , sino tan solo conocer el grado de pertenencia a cada uno de los conjuntos en que se ha participado el rango de variación de la variable. Es aquí donde aparece el término conjunto difuso¹ mismo que está asociado a un determinado valor lingüístico definido por un adjetivo o etiqueta lingüística, estos conjuntos parten de la teoría clásica de conjuntos, añadiendo una función de pertenencia definida como un número real entre 0 y 1.

Entonces la *Lógica Difusa* es una extensión de la Lógica Multivaluada, que además está relacionada y fundamentada en la teoría de conjuntos difusos. Según esta teoría, será una función de pertenencia (que tomará cualquiera de los valores reales comprendidos en el intervalo $[0,1]$) la que determine el grado de pertenencia de un elemento a un conjunto.

La lógica difusa, al igual que la lógica clásica se basa en el hecho de analizar proposiciones las que se les puede otorgar un valor de verdad. La diferencia fundamental entre las proposiciones clásicas y las proposiciones difusas, está en el rango en que se les otorga sus valores de verdad. Mientras que cada proposición clásica, requiere que se le otorgue un valor de verdad de: o estrictamente verdadero o estrictamente falso (o 1, o 0); la verdad o falsedad, en las proposiciones difusas, es un valor en el intervalo $[0,1]$ y está determinada por el grado de pertenencia al conjunto difuso.

¹ **Bonifacio Martín del Brío.** Redes Neuronales y sistemas borrosos. 2ª Ed.

2.1.3 Conjuntos Difusos

De manera intuitiva se tiene el concepto de conjunto como una colección bien definida de elementos, en la que es posible determinar para un objeto cualquiera, en un universo dado, si acaso este pertenece o no al conjunto. La decisión, naturalmente, es “si pertenece” o bien “no pertenece”.

Es así que los conjuntos difusos surgieron como una nueva forma de representar la imprecisión y la incertidumbre, debido a que hay conceptos que no tienen límites claros como por ejemplo:

- ¿La temperatura 25°C es “alta”?

Si definimos, por ejemplo: Alta (30)=1, Alta (10)=0, Alta (25)=**0.75**

Definición 1: “Llamaremos conjunto difuso a un valor lingüístico junto a una función de pertenencia. El valor lingüístico es el “nombre” del conjunto, y la función de pertenencia se define como aquella aplicación que asocia a cada elemento del universo de discurso el grado con que pertenece al conjunto difuso. Decimos que un conjunto es *nítido* si su función de pertenencia toma valores en $\{0,1\}$, y *difuso* si toma valores en $[0,1]$ ”².

² **Bonifacio Martín del Brío.** Redes Neuronales y sistemas borrosos. 2ª Ed.

Definición 2: “Un conjunto difuso A se define como una Función de Pertenencia que enlaza o empareja los elementos de un dominio o Universo de discurso U con elementos del intervalo $[0,1]$ ”³:

$$A: U \rightarrow [0,1]$$

Llamaremos universo de discurso al rango de valores que pueden tomar los elementos que poseen la propiedad expresada por la variable lingüística.

Cuanto más cerca esté $A(x)$ del valor 1, mayor será la pertenencia del objeto x al conjunto A .

Los valores de pertenencia varían entre 0 (no pertenece en absoluto) y 1 (pertenencia total).

Representación: La notación habitual para los conjuntos difusos es la definida por Lofti Zadeh, que es la siguiente:

Sea A un conjunto difuso definido sobre el universo U :

$$A = \{ x, A(x) / x \in U \}$$

Que indica que A está formado por todos los pares ordenados x y el resultado de la función de pertenencia para todo elemento x dentro del universo de discurso U .

Para denotar el conjunto difuso A :

³ **ZADEH L.A.** Fuzzy sets: Information and control, 2a Ed. California. Center. 1965. pp. 338-353.

Si el universo es discreto: $\sum_U A(x)/x$

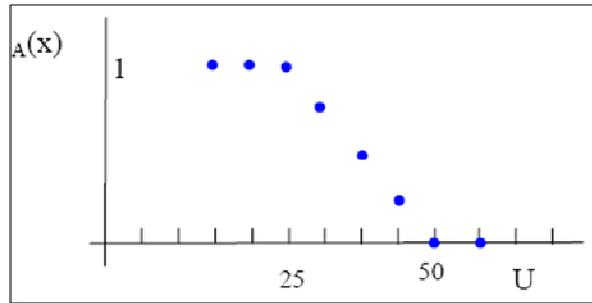


Figura. II.1 Universo Discreto de un conjunto difuso

Si el universo es continuo: $\int_U A(x)/x$

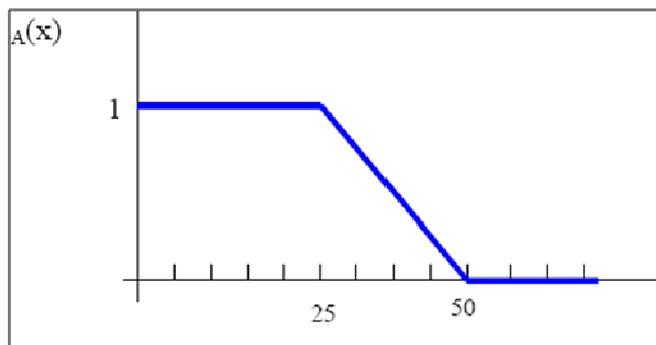


Figura. II.2 Universo Continuo de un conjunto difuso.

El sumatorio o la integral pierden su significado habitual, en lógica difusa quieren simbolizar una mera enumeración de tuplas. La barra tampoco indica una fracción sino que simplemente separa los dos elementos de la tupla. Así por ejemplo el conjunto difuso discreto "Tirada alta del dado" podría definirse como:

$$F = \{ 0/1 + 0/2 + 0.3/3 + 0.6/4 + 0.9/5 + 1/6 \}$$

La parte derecha de la tupla indica el elemento y la parte izquierda el grado de pertenencia.

En los apuntes publicados en la página oficial de la Universidad de Málaga (<http://www.lcc.uma.es>) se definen algunos términos dado un conjunto difuso con un ejemplo explicativo que se mencionan a continuación:

- Dado un conjunto difuso A , se define como **alfa-corte** de A , al conjunto de elementos que pertenecen al conjunto difuso A con grado mayor o igual que α , es decir:

$$A_{\alpha} = \{x \in X / A(x) \geq \alpha\}$$

- Se define como **alfa corte estricto** al conjunto de elementos con grado de pertenencia estrictamente mayor que α , es decir:

$$A_{\alpha} = \{x \in X / A(x) > \alpha\}$$

- Se define como **sopORTE** de un conjunto difuso A , al conjunto nítido de elementos que tienen grado de pertenencia estrictamente mayor que 0, o sea, al alfa-corte estricto de nivel 0.

$$\text{Soporte}(A) = \{x \in X / A(x) > 0\}$$

- Se define como **núcleo** de un conjunto difuso A , al conjunto nítido de elementos que tienen grado de pertenencia 1. (alfa-corte de nivel 1)

$$\text{Núcleo}(A) = \{x \in X / A(x) = 1\}$$

- Se define la **altura** de un conjunto difuso A como el valor más grande de su función de pertenencia.
- Se dice que un conjunto difuso está **normalizado** si y solo si su núcleo contiene algún elemento (o alternativamente, si su altura es 1), es decir:

$$\exists x \in X \quad A(x) = 1$$

- El elemento x de U para el cual $A(x) = 0.5$ se llama el **punto de cruce**.
- Un conjunto difuso cuyo soporte es un único punto x de U y tal que la función de pertenencia de x es 1 (es decir, el soporte coincide con el núcleo y tienen un único punto) se llama un **conjunto difuso unitario (singleton)**.

Ejemplo 1

Consideremos la variable lingüística “Altura de los seres humanos”, que toma valores en el universo de discurso $U = [1.4, 2.50]$. Vamos a hacer una clasificación difusa de los seres humanos en tres conjuntos difusos (o valores lingüísticos): bajos, medianos y altos.

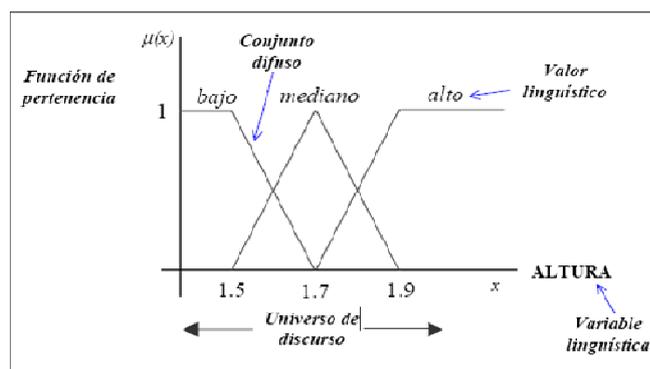


Figura. II.3 Términos de un conjunto difuso.

En esta ilustración se ha dibujado 3 conjuntos difusos sobre la variable lingüística altura, cuyos valores lingüísticos asociados son *bajo*, *mediano* y *alto* respectivamente. Las funciones de pertenencia son de tipo L para *bajo*, Lambda o Triángulo para el *mediano* y Gamma para el *alto*. De este modo si Luis mide 1.80 metros, la lógica difusa nos dice que es un 0.2 mediano y un 0.8 alto.

De este modo expresamos que mientras un elemento puede estar dentro de un determinado conjunto, puede no cumplir las especificaciones de dicho conjunto al cien por cien (por ejemplo, en el caso de Luis, a la vista del resultado podríamos afirmar que es *poco mediano* y *más bien alto*).

En este ejemplo, dado el conjunto difuso mediano tenemos que:

- El alfa-corte 0.5 es el intervalo [1.6,1.8]
- El alfa corte estricto 0.5 es el intervalo (1.6, 1.8)
- El soporte es (1.5, 1.9)
- El núcleo es 1.7
- Es un conjunto difuso normalizado
- Tiene dos puntos de cruce: 1.6 y 1.8

2.1.4 Variables Lingüísticas.

Se denomina Variable Lingüística⁴ a aquella que puede tomar por valor términos del lenguaje natural, como mucho, poco, positivo, etc., que son las palabras que desempeñan el papel de etiquetas en un conjunto difuso.

⁴ **Bonifacio Martín del Brío.** Redes Neuronales y sistemas borrosos. 2ª Ed.

Se define a una variable lingüística a aquella noción o concepto que vamos a calificar de forma difusa. Por ejemplo: la altura, la edad, el error, la variación del error. Le aplicamos el adjetivo "lingüística" porque definiremos sus características mediante el lenguaje hablado.⁵

2.1.4.1 Utilidad de las Variables Lingüísticas

1. Es una forma de comprimir información llamada granulación (*granulation*)
2. Una etiqueta incluye muchos valores posibles.
3. Ayuda a caracterizar fenómenos que o están mal definidos o son complejos de definir o ambas cosas.
4. Es un medio de trasladar conceptos o descripciones lingüísticas a descripciones numéricas que pueden ser tratadas automáticamente (Relaciona o traduce el proceso simbólico a proceso numérico).
5. Usando el principio de extensión, muchas herramientas ya existentes pueden ser extendidas para manejar variables lingüísticas, obteniendo las ventajas de la lógica difusa en gran cantidad de aplicaciones.

2.1.4.2 Definición formal de las Variables Lingüísticas

Según Zadeh L.A (1995) de manera más formal la Variable lingüística se define como una quintupla:

⁵ <http://www.lcc.uma.es/~eva/aic/apuntes/fuzzy.pdf>

(H, T (H), U, G, M)

H es el nombre de la variable

T(H) denota un conjunto de términos de H, esto es, la colección de sus valores lingüísticos.

U representa el universo de discurso.

G es una regla sintáctica que asocia a cada valor lingüístico X,

M es una regla semántica para asociar un significado a cada valor M: $T(H) \rightarrow F(U)$.

2.3.4.3 Granularidad

Es el número de valores que se definen para una variable lingüística. Normalmente se usa un conjunto pequeño de valores para una variable lingüística.

Granularidad fina (fine)

Define un gran número de valores para una variable lingüística.

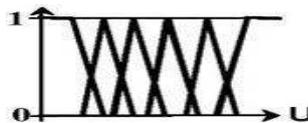


Figura. II.4 Granularidad fina.

Granularidad gruesa (coarse)

Define un pequeño número de valores.

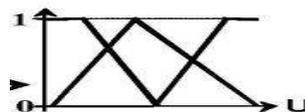


Figura. II.5 Granularidad gruesa.

2.1.5 Funciones de Pertenencia de Conjuntos Difusos.

Según la Revista de la Facultad de Ingeniería de la Universidad de Tarapaca en Chile menciona en su página web facing@uta.cl que Una función de pertenencia se puede definir como una curva donde se relaciona términos lingüísticos, no fácilmente mensurables, con un valor numérico.

Cualquier función podría representar un conjunto difuso, pero se ha visto mucho más recomendable hacer uso de funciones simples que permitan el desarrollo de las operaciones con sencillez.

Según Bonifacio Martin del Brio (2002) las funciones más frecuentes son:

- Función de tipo trapezoidal.
- Función Singleton.
- Función Triangular,
- Función Tipo S
- Función Exponencial.
- Función Tipo π .

Función Tipo Trapezoidal

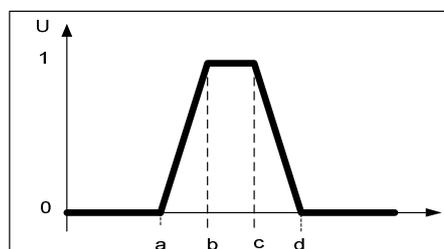


Figura. II.6 Función tipo Trapezoidal

Se define por cuatro puntos a, b, c, d. Esta función es cero para valores menores de a y mayores de d, vale 1 entre b y c, y toma valores entre [0,1] entre a y b, y entre c y d. Se utiliza habitualmente en sistemas borrosos sencillos, pues permite definir un conjunto borroso con datos, y calcular su valor de pertenencia con pocos cálculos.

Se define con:

$$S(u; a, b, c, d) = \begin{cases} 0 & u < a \\ \left(\frac{u-a}{b-a}\right) & a \leq u \leq b \\ 1 & b \leq u \leq c \\ \left(\frac{d-u}{d-c}\right) & c \leq u \leq d \\ 0 & u > d \end{cases}$$

Esta función resulta adecuada para modelar propiedades que comprenden un rango de valores (adulto, normal, adecuada...). Para modelar una función triangular se hace b=c, para una función de tipo S se hace c=d=max (U), y para una función de tipo Singleton a=b=c=d.

Función Tipo Singleton

Tiene valor de 1 solo para el punto a y 0 para el resto. Se define con:

$$S(u; a) = \begin{cases} 1 & u = a \\ 0 & u \neq a \end{cases}$$

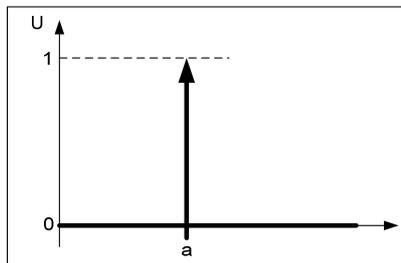


Figura. II.7 Función tipo singleton

Función Tipo T (Triangular)

Esta función es adecuada para modelar propiedades con un valor de inclusión distinto de cero para un rango de valores estrecho en torno a un punto b.

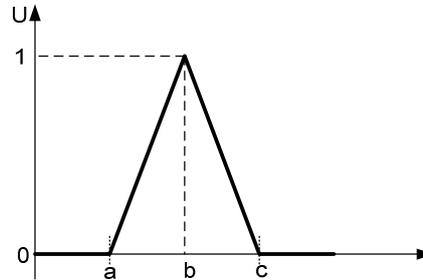


Figura. II.8 Función tipo T

Puede definirse como:

$$T(u; a, b, c) = \begin{cases} 0 & u < a \\ \frac{u-a}{b-a} & a \leq u \leq b \\ \frac{c-u}{c-b} & b \leq u \leq c \\ 0 & u > c \end{cases}$$

Función Tipo S

Esta función resulta adecuada para modelar propiedades como grande, mucho, positivo. Se caracteriza por tener un valor de inclusión distinto de 0 para un rango de valores por encima de cierto punto a, siendo 0 por debajo de a y 1 para valores mayores de c. Su punto de cruce (valor 0.5) es $b=(a+c)/2$; y entre los puntos a y c es de tipo cuadrático. Se puede definir como⁶:

⁶ **Bonifacio Martín del Brío.** Redes Neuronales y sistemas borrosos. 2ª Ed.

$$T(u; a, b, c) = \begin{cases} 0 & u < a \\ 2 \left(\frac{u-a}{b-a} \right)^2 & a \leq u \leq b \\ 1 - 2 \left(\frac{c-u}{c-b} \right)^2 & b \leq u \leq c \\ 0 & u > c \end{cases}$$

Función Tipo π

Esta función tiene forma de campana, y resulta adecuada para los conjuntos definidos en torno a un valor c , como medio, normal, cero. Puede definirse también utilizando expresiones analíticas exponenciales o cuadráticas, como la bien conocida campana de Gauss. Se define como:

$$S(u; b, c) = \begin{cases} S(u; c-b, c-b/2, c) & u \leq c \\ 1-S(u; c-b, c-b/2, c) & u \geq c \end{cases}$$

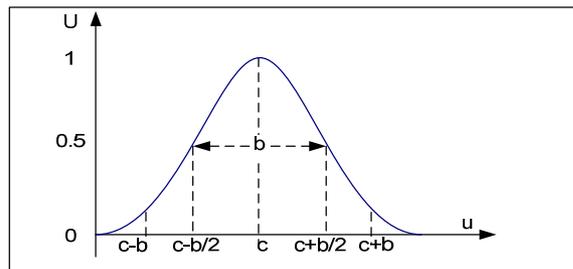


Figura. II.9 Función Tipo π

2.1.6 Operaciones Con Conjuntos Difusos

Según Bonifacio Matin del Brio (2002) a los subconjuntos borrosos se les puede aplicar determinados operadores, o bien pueden realizarse operaciones entre ellos. Al aplicar un operador sobre un solo conjunto difuso se obtiene otro conjunto difuso; así también al combinar dos o más subconjuntos difusos mediante alguna operación, se obtendrá otro subconjunto difuso.

Cabe resaltar que el funcionamiento de las operaciones básicas coincide con las de la teoría clásica de conjuntos.

Como se ha visto, la función de pertenencia es la componente fundamental de un conjunto difuso. De ahí que las operaciones con tales conjuntos se definan a través de dicha función. Veremos, en primer lugar, las operaciones elementales paralelas a las realizadas con conjuntos comunes (Zadeh, 1965).

Diremos que un conjunto A es un subconjunto difuso de B , es decir, $A \subseteq B$, si:

$$\forall x \in U: m_A(x) \leq m_B(x)$$

Y si existe, al menos, un punto $x \in U$ tal que $m_A(x) < m_B(x)$, entonces escribiremos que $A \subset B$.

Intersección (min-intersección):

La intersección de dos conjuntos difusos, A y B , viene dada, punto a punto, por:

$$m_{A \cap B}(x) = \min \{m_A(x), m_B(x)\}$$

Unión (max-intersección):

La unión de dos conjuntos difusos, A y B , viene dada, punto a punto, por:

$$m_{A \cup B}(x) = \max \{m_A(x), m_B(x)\}$$

Complementación:

El complemento, A^c , de un conjunto difuso A , viene dado, punto a punto, por:

$$m_{A^c}(x) = 1 - m_A(x)$$

Estas tres operaciones definidas para conjuntos difusos cumplen, al igual que en la teoría clásica de conjuntos, asociatividad, conmutatividad y distributividad así como las leyes de Morgan.

Sin embargo, también hay que destacar que existen dos leyes fundamentales de la teoría clásica de conjuntos como son el Principio de contradicción: $A \cup \overline{A} = U$, y el Principio de exclusión: $A \cap \overline{A} = F$ que no se cumplen en la teoría de conjuntos difusos; de hecho una de las formas para describir en qué se diferencia la teoría clásica de conjuntos de la teoría difusa es explicar que estas dos leyes en términos de fuzzy logic no se cumplen. En consecuencia, algunas de las teorías derivadas de la teoría de conjuntos como por ejemplo la de la probabilidad será muy diferente planteada en términos difusos.

Las funciones correspondientes a la unión e intersección de conjuntos difusos pueden generalizarse⁷, y pertenecen a una clase general de operadores llamados **normas triangulares** (*t-normas*), y **conormas triangulares** (*t-conormas*) respectivamente.

- **t-norma del mínimo:** La función \min (\wedge) es una t-norma, que corresponde a la operación de **intersección** en conjuntos clásicos cuyos grados de pertenencia están en $\{0,1\}$. Por eso, esta función es la extensión natural de la intersección en conjuntos difusos. Algunas de las más usadas son:

⁷ **ALEGRE, M.C.** Inteligencia Artificial en el control de procesos controladores borrosos. California. Mundo Electrónico. 1991. pp 42-49

$$\text{MIN}(a, b)$$

$$(ab)$$

$$a * b = \text{MAX}(0, a+b-1)$$

- **t-conorma o s-norma del máximo:** La función $\text{máx}(\vee)$ es una s-norma, que corresponde a la operación de **unión** en conjuntos clásicos cuyos grados de pertenencia están en $\{0,1\}$. Por eso, esta función es la extensión natural de la unión en conjuntos difusos. Las más usadas son:

$$\text{MAX}(a,b)$$

$$(a+b-ab)$$

$$a+b = \text{MIN}(1, a+b)$$

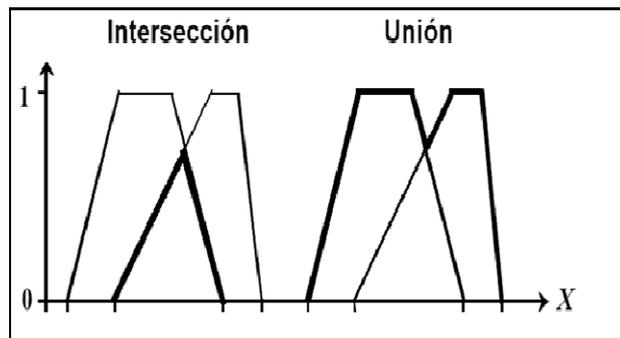


Figura. II.10 t-norma y t-conorma

El concepto de norma triangular se usa en estadística para datos que pertenecen al intervalo $[0, 1]$. Este es el motivo de que en el desarrollo de la matemática difusa sea de interés, especialmente en el campo de la lógica difusa y de los sistemas expertos.

A continuación se detallan las operaciones principales y más conocidas de conjuntos difusos.

TABLA II.1 OPERACIONES DE CONJUNTOS DIFUSOS

	OPERACIÓN	RANGO
Igualdad	$m_A(x) = m_B(x)$	$x \in U$
Unión	$m_{A \cup B}(x) = \max \{m_A(x), m_B(x)\}$	$\forall x \in U$
Intersección	$m_{A \cap B}(x) = \min \{m_A(x), m_B(x)\}$	$\forall x \in U$
Complemento	$m_A^c(x) = 1 - m_A(x)$	$x \in U$
Norma	$m_{Norma(A)}(x) = \frac{\min \{m_A(x)\}}{\max \{m_A(x)\}}$	$x \in U$
Concentración	$m_{Conc(A)}(x) = (m_A(x))^2$	$x \in U$
Dilatación	$m_{Dilat(A)}(x) = (m_A(x))^{0.5}$	$x \in U$

Fuente: Redes Neuronales y sistemas borrosos.

Elaborado Por: Bonifacio Martin del Brio.

Los últimas 3 operaciones son mejor conocidas como modificadores, ya que permiten formalizar el tipo de modificadores aplicados sobre el mismo término en el lenguaje común, como por ejemplo muy o más o menos. En la fórmula de concentración al elevar al cuadrado un valor entre 0 y 1 se obtiene un valor más pequeño por lo que puede representar en términos lingüísticos términos como muy frío y por otra parte términos como más o menos frío puede definirse con la formula de dilatación puesto que al sacar la raíz cuadrada de un número entre 0 y 1 se obtiene un valor más grande.

2.1.7 Inferencia Difusa

Como en la lógica clásica, la lógica difusa se ocupa del razonamiento formal con proposiciones, pero a diferencia de esta, los valores de las proposiciones pueden tomar valores intermedios entre verdadero y falso.

Uno de los objetivos de la lógica difusa es proporcionar fundamentos para el razonamiento o inferencia aproximada con proposiciones imprecisas.⁸

2.1.7.1 Principio de Extensión

El principio de extensión permite convertir conceptos no difusos en difusos, siendo además la base de la inferencia en los sistemas difusos (Bonifacio Martín del Brio, 2002).

Según los apuntes publicados por la Universidad de Málaga por *J. Galindo Gómez* el principio de extensión es usado para transformar conjuntos difusos, que tengan iguales o distintos universos, según una función de transformación en esos universos.

Sean X e Y dos conjuntos y f una función de transformación de uno en otro: $f: X \rightarrow Y$.

Sea A un conjunto difuso en X .

El principio de extensión sostiene que la “imagen” de A en Y , bajo la función f es un conjunto difuso $B=f(A)$, Definido como:

$$B(y) = \sup\{A(x) \mid x \in X, y \in f(x)\}$$

La función \sup se aplica si existen dos o más valores de x que tengan igual valor $f(x)$.

2.1.7.2 Relación Difusa

En los apuntes publicados por la Universidad de Catalunya⁹ explica de forma clara que una relación difusa representa el grado de presencia o ausencia de asociación,

⁸ <http://www.dma.fi.upm.es/java/fuzzy/tutfuzzy/contenido3.html>.

interacción o interconexión entre elementos de dos o más conjuntos difusos, por ejemplo: “x es mayor que y”. Supongamos U y V dos universos de discurso, la relación difusa R (U, V) es un conjunto difuso en el espacio producto $U \times V$ que se caracteriza por la función de pertenencia $\mu_R(x, y)$ donde x pertenece a U e y pertenece a V, es decir:

$$R(U, V) = \{((x, y), R(x, y)) \mid (x, y) \in U \times V\}.$$

En el caso de las relaciones difusas $\mu_R(x, y) \in [0, 1]$ y en caso de las relaciones clásicas

$$\mu_R(x, y) = 0 \text{ ó } 1.$$

Como las relaciones difusas son en sí mismas un conjunto difuso en el espacio producto, las operaciones entre conjuntos y los operadores definidos anteriormente también pueden ser aplicadas a ellas. Supongamos $R(x, y)$ y $S(x, y)$ dos relaciones en el mismo espacio producto $U \times V$. La intersección o unión entre R y S, que son composiciones entre las dos relaciones, se definen como:

$$\mu_{R \cap S}(x, y) = \mu_R(x, y) * \mu_S(x, y)$$

$$\mu_{R \cup S}(x, y) = \mu_R(x, y) \oplus \mu_S(x, y)$$

Donde * es cualquier t-norma, y \oplus es cualquier t-conorma.

Si consideramos las relaciones difusas R y S que pertenecen a diferentes espacios producto $R(U, V)$ y $S(V, W)$, por ejemplo “x es mayor que y” y “y es cercano a z”, su composición difusa se define de forma análoga a la composición clásica teniendo en cuenta que en el caso difuso la relación difusa R tiene asociada una función

⁹ http://www.tesisexarxa.net/TESIS_UPC/AVAILABLE/TDX-0207105-105056//04Rpp04de11.pdf

característica $\mu_R(x, y)$ que toma valores en el intervalo $[0,1]$ y la relación difusa S también tiene asociada una función característica $\mu_S(y,z)$ que de igual forma toma valores en el intervalo $[0,1]$. Entonces la composición difusa entre R y S , es decir $\underline{R} \circ S$, cuando R y S pertenecen a universos discretos de discurso, se define como una relación difusa en $U \times W$ cuya función de pertenencia viene dada por:

$$\mu_{R \circ S}(x,z) = \sup[\mu_R(x, y) * \mu_S(y,z)$$

Dónde el operador \sup es el máximo y el operador $*$ puede ser cualquier t-norma. En función de la t-norma elegida podemos obtener distintas composiciones.

2.1.7.3 Métodos de Inferencia

El conocimiento se representa por reglas, por lo general esta se expresa de modo IF-THEN (Si-Entonces) y estas representan una relación o proposición borrosa.

El Centro Politécnico Superior en una publicación de la Universidad de Zaragoza¹⁰ dice que existen dos tipos para realizar la inferencia difusa entre las reglas, estas son: Modus Ponens Generalizado y el Modus Tollens Generalizado que son extensiones o generalizaciones del razonamiento clásico.

- **Modus Ponens Generalizado (GMP)**

Se fundamenta en el hecho intuitivo de que, si se tiene una regla de implicación del tipo $p \rightarrow q$ junto con una proposición p' parecida (aunque no idéntica) a la proposición p ,

¹⁰ http://www.cps.unizar.es/~te/Docencia_archivos/rnlb_archivos/teruel98.pdf

debería inferirse del conjunto una proposición q' distinta de q aunque no muy diferente. Por su evidente parecido con el *modus ponens* clásico este esquema de razonamiento recibe el nombre de *modus ponens generalizado*, en donde A , A' , B y B' son conjuntos difusos, con $A \stackrel{\mu}{=} A'$ y $B \stackrel{\mu}{=} B'$.

Conocido también como razonamiento directo, puede expresarse de la siguiente forma:

(Conocimiento): Si x es A entonces y es B

(Hecho): x es A'

(Conclusión): y es B'

$B' = A' \circ R$

Si $A' = A$ y $B' = B$ equivalente Modus Ponens

El Modus Ponens está asociado a la implicación "A implica B" ($A \rightarrow B$).

- **Modus Tollens Generalizado (GMT)**

Conocido también como razonamiento inverso y puede resumirse de la siguiente forma:

(Conocimiento): Si x es A entonces y es B

(Hecho): y es B'

(Conclusión): x es A'

$A' = R \circ B'$

Si $A' = \text{no } A$ y $B' = \text{no } B$ equivalente Modus Tollens

2.1.7.4 Reglas Difusas

Las reglas difusas combinan uno o más conjuntos difusos de entradas, llamadas antecedentes o premisas, y les asocian un conjunto borroso de salida, llamado

consecuente o consecuencia. Los conjuntos difusos de la premisa se asocia mediante conjuntivas lógicas como y, o, etc.

Las reglas difusas permiten expresar el conocimiento que se dispone sobre la relación entre antecedentes y consecuentes. Para expresar este conocimiento de forma completa se precisan de varias reglas, que se agrupan formando lo que se conoce como una base de reglas, es decir, el conjunto de reglas que expresan las relaciones conocidas entre antecedentes y consecuentes.

Una regla difusa base es un conjunto de reglas SI-ENTONCES que pueden ser expresadas de la siguiente forma:

R^m : SI u_1 es A_1^m y u_2 es A_2^m y ... u_p es A_p^m , ENTONCES v es B^m

Con $m=1,2,\dots, M$

Y donde A_i^m y B^m son conjuntos difusos en $U_i \subset \mathbb{R}$ (números reales) y $V \subset \mathbb{R}$ respectivamente, $u = (u_1, u_2, \dots, u_n) \in U_1 \times U_2 \times \dots \times U_n$ y $v \in V$, son los valores numéricos concretos de u y v , también respectivamente.

Este formato de reglas se conoce como **borroso puro o de tipo Mandani** por ser quien primero lo propuso en 1974.

Vemos que esta regla tiene además la particularidad de que es un regla multi antecedente; este tipo de reglas, que combina varias variables en el antecedente, es el más utilizado en el diseño de sistemas difusos.

Un sistema difuso estará formado por varias reglas difusas base con diferentes consecuentes, ya que una regla con multi antecedente y multi consecuente siempre podrá ser descompuesta en un conjunto de reglas base con multi antecedente pero un solo consecuente.

Los sistemas difusos descritos con u_i entradas y una sola salida v , se conoce como **MISO** (Multiple Input Single Output), mientras que los que tienen varias salidas se conoce como **MIMO** (Multiple Input Multiple Output).

2.1.7.5 Dispositivos de inferencia difusa

Se llaman dispositivos de inferencia borrosa a los sistemas que interpretan las reglas de tipo SI-ENTONCES (IF-THEN) de una base de reglas, con el fin de obtener los valores de salida a partir de los actuales valores de las variables lingüísticas de entrada al sistema. En un sistema borroso las reglas que se conocen como de tipo Mandani se interpretan como una implicación borrosa o implicación difusa.

Existen varios operadores de implicación difusa pero las más utilizadas según Bonifacio Matin del Brio (2002) se expresan a continuación:

- **Implicación borrosa por la regla del mínimo o implicación de Madani:**

$$\mu_{A \rightarrow B}(x, y) = \min [\mu_A(x), \mu_B(y)]$$

- **Implicación borrosa por la regla del producto:**

$$\mu_{A \rightarrow B}(x, y) = \mu_A(x) \mu_B(y)$$

- **Implicación borrosa por la regla aritmética:**

$$\mu_{A \rightarrow B}(x, y) = \min [1, 1 - \mu_A(x) + \mu_B(y)]$$

- **Implicación borrosa por la regla Max-Min:**

$$\mu_{A \rightarrow B}(x, y) = \max \{ \min [\mu_A(x), \mu_B(y)], 1 - \mu_A(x) \}$$

- **Implicación Borrosa por la regla Booleana:**

$$\mu_{A \rightarrow B}(x, y) = \max \{1 - \mu_A(x), \mu_B(y)\}$$

- **Implicación Borrosa por la regla de Goguen:**

$$\mu_{A \rightarrow B}(x, y) = \begin{cases} 1 & \mu_A(x) \leq \mu_B(y) \\ \mu_A(x) / \mu_B(y) & \mu_A(x) > \mu_B(y) \end{cases}$$

2.1.8 Fuzzificador o Borrosificador

El Fuzzificador es el encargado de establecer la relación entre los datos de entrada al sistema y sus correspondientes conjuntos difusos, las variables procedentes del exterior por lo general serán no difusas y habrá que fuzzificarlas previamente, para realizar el cálculo. Entre las estrategias de fuzzificación mencionamos las 2 más importantes¹¹:

- **Fuzzificador singleton:** es el método de fuzzificación más utilizado y consiste en considerar los propios valores discretos como conjuntos borrosos, es decir para cada valor de entrada x se define un conjunto A' que lo soporta, con función de pertenencia $\mu_{A'}(x')$, de modo que $\mu_{A'}(x) = 1, (x = x')$, y $\mu_{A'}(x') = 0$, para todos los otros $x' \in U$ en los que $x' \neq x$.
- **Fuzzificador no singleton:** En este método de borrosificación se utiliza una función exponencial de tipo Campana, centrada en el valor x de entrada, anchura σ y amplitud a .

$$\mu_{A'}(x') = a \cdot \exp \left[- \left(\frac{x' - x}{\sigma} \right)^2 \right]$$

¹¹ MARTÍN DEL BRÍO BONIFACIO. Redes Neuronales y sistemas borrosos. 2ª Ed. Madrid España. Ra-Ma. 2002. pp. 245-273

2.1.9 Defuzzificador o Desborrosificador

El bloque defuzzificador realiza la función contraria al Fuzzificador. El Fuzzificador tiene como entradas valores concretos de las variables de entrada y como salidas grados de pertenencia a conjuntos difusos (entre 0 y 1). La entrada al bloque defuzzificador es el conjunto difuso de salida, resultado del bloque de inferencia y la salida es un valor concreto de la variable de salida. Para obtener, a partir del conjunto difuso de salida que resulta de la agregación de todas las reglas, un resultado escalar, se aplican métodos matemáticos.

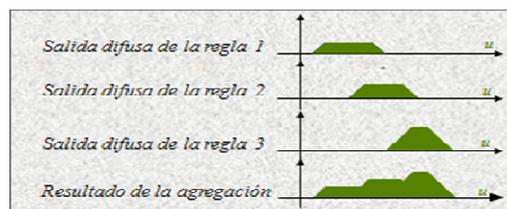


Figura. II.11 Resultado de la agregación

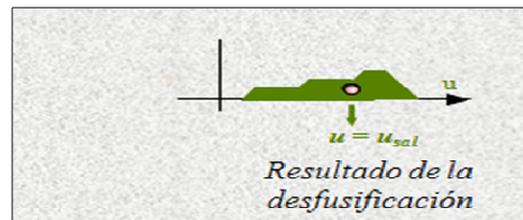


Figura. II.12 Resultado de la Defuzzificación

Algunos de los métodos más utilizados se mencionan a continuación:

- **Método del Máximo**

Se elige como valor para la variable de salida aquel para el cual la función característica del conjunto difuso de salida es máxima. En general no es un método óptimo, ya que este valor máximo puede ser alcanzado por varias salidas.

Si se producen empates puede seleccionarse el primer valor encontrado o la media (en cuyo caso el método se denomina **media de máximos** (MOM en FuzzyCLIPS)).

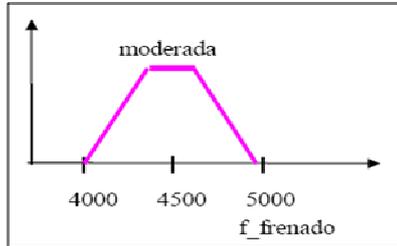


Figura. II.13 Ejemplo método de fuzzificación del máximo

Considerando el gráfico nos encontramos que la función de pertenencia tiene varios máximos: todos los valores entre 4250 y 4750. La estrategia MOM cogería el valor medio, esto es, 4500, y la estrategia del primer valor máximo cogería 4250.

- **Método por media de centros (Centroide)**

En el cual actúa las variables de entrada en su forma normal, así como la función de pertenencia correspondiente a dicha variable.

Definido como:

$$y_{centroide} = \frac{\sum_{x \in X} x \mu_A(x)}{\sum_{x \in X} \mu_A(x)}$$

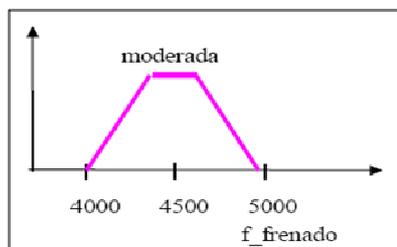


Figura. II.14 Ejemplo método de fuzzificación del Centroide

En nuestro ejemplo sería: $\frac{4000*0+4250*1+4750*1+5000*0}{1+1} = 4500$

2.1.10 La Lógica Difusa Y La Probabilidad

Es importante conocer la diferencia existente entre una función de pertenencia de un conjunto difuso y una función de densidad de probabilidad puesto que los dos contienen valores entre 0 y 1.

- Se debe tener en claro que la función de pertenencia de la lógica difusa indica hasta qué punto cierto valor está incluido en dicho conjunto, mientras que la probabilidad indica la frecuencia con que los diversos valores de una magnitud se presentan.
- Las funciones de pertenencia usada por la lógica difusa son fijadas arbitrariamente por un observador, por el contrario la probabilidad se determina por la observación de la ocurrencia de ciertos eventos, en algunos casos se mide esta probabilidad.
- La difusividad aplicada en la lógica difusa describe eventos ambiguos, la probabilidad describe los eventos que ocurren. Si un evento ocurre es aleatorio. El grado con el cual ocurre es difuso.

2.1.11 Sistemas de Control Difuso

Al estudiar algunos aspectos de lo que son los conjuntos difusos así como la lógica difusa es posible conocer y entender la estructura de un sistema difuso (SLD's) o también conocidos como FLC (Fuzzy Logic Controllers) o sistemas de inferencia

borrosa FIS (Fuzzy Interface Systems). Para un mayor de profundización en el tema, se recomienda consultar a Duarte (1998).

La estructura de un sistema difuso se tiene cuatro componentes principales:

- Fuzzificador
- Base de Reglas
- Motor de Inferencia
- Defuzzificador

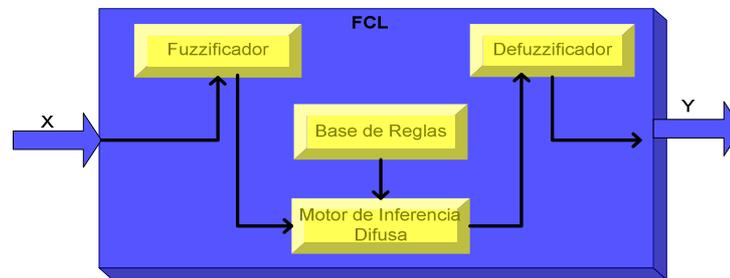


Figura. II.15 Estructura de un Sistema Difuso

Todo sistema de lógica difusa está constituido por cuatro módulos básicos denominados Fuzzificador, Base de Reglas, Motor de Inferencia y Defuzzificador (Fig. 2.18). El papel de estos módulos, es el de recibir entradas o datos discretos, procesarlos analíticamente y generar salidas concretas, que corresponderían al resultado de la interacción de las variables lingüísticas.

- **Módulo Fuzzificador**

El módulo Fuzzificador recibe las múltiples entradas concretas que llegan al sistema de lógica difusa, y produce un conjunto difuso por cada una de ellas, determinado por la

funcion de pertenencia respectiva. Cada conjunto difuso producido por el fuzzificador, esta definido sobre el universo de discurso de la variable lingüística respectiva.

- **Base de Reglas**

Sintetizan la heurística o experiencia acumulada de los expertos a través de un conjunto de proposiciones condicionales difusas, obteniéndose de esa forma lo que se denomina un modelo lingüístico.

La base de reglas es un conjunto de m proposiciones, cada una de las cuales es de la forma:

Si (entrada 1 es conjunto i_1 Y entrada 2 es conjunto i_2 Y...Y entrada p es conjunto i_p), entonces (salida 1 es conjunto i_1 Y salida 2 es conjunto i_2 Y...Y salida q es conjunto i_q).

En donde el conjunto i_j , es uno de los valores lingüísticos que puede tomar la variable de entrada o de salida j . Si las m reglas cubren todas las posibles combinaciones se dice que la base de reglas es completa. En ningún caso puede permitirse que existan dos reglas con el mismo antecedente y diferente consecuente.

Tiene dos objetivos principales:

1. Proporciona las definiciones necesarias para determinar las reglas lingüísticas de control y manipulación de los datos difusos del controlador.
2. Almacena los objetivos y la política de control (como un experto en el dominio).

- **Motor de Inferencia**

Simula el razonamiento humano haciendo inferencia sobre las entradas y las reglas IF-THEN.

De acuerdo con Duarte 1998, el motor de inferencia recibe los p conjuntos difusos producidos por el Fuzzificador, y los aplica a cada una de las m reglas de la Base de Reglas, para producir $m*q$ conjuntos difusos (un conjunto difuso por cada variable de salida en cada una de las reglas) definidos sobre los universos de discurso de las variables lingüísticas de salida.

- **Módulo Defuzzificador**

Este módulo recibe los $m*q$ conjuntos difusos generados por el motor de inferencia, y produce q valores concretos correspondientes a cada una de las variables de salida de sistema de lógica difusa. El módulo defuzzificador por lo general efectúa la unión de los $m*q$ conjuntos difusos de salida para cada una de las respectivas variables, y mediante algún algoritmo de defuzzificación produce un valor concreto que es la salida final del SLD.

2.1.12 Ciclo de Análisis y Diseño usando Lógica Difusa

Al conocer como se encuentra estructurado un sistema basado en un control difuso es necesario definir los pasos a realizar al momento de comenzar con el desarrollo del sistema.

A continuación se presentan las principales fases a realizar en un sistema difuso:

– **Análisis:**

1. Definir los Objetivos, las Restricciones y el Comportamiento del Sistema.
2. Identificar las variables de Entrada y Salida: Variables lingüísticas y sus etiquetas lingüísticas.
3. Definir procedimientos y criterios para la prueba y validación.

– **Diseño:**

4. Definir los conjuntos difusos de cada etiqueta lingüística.
5. Definir el marco de representación de la solución (representación del conocimiento, reglas...).
6. Definir el marco de los procedimientos empleados (Inferencia).
7. Especificar la forma requerida de la salida del sistema (decodificación).
8. Ejecutar pruebas para validar el sistema.
9. Verificar si la solución es compatible con el paso 1:
 - Refinar el Diseño: Ir al paso 4.
 - Necesidad de Mejora: Ir al paso 1.

2.1.13 ¿Por qué usar los Sistemas de inferencia difusa?

- La lógica borrosa no resuelve problemas nuevos, sino que utiliza nuevos métodos para resolver los problemas de siempre.
- Los conceptos matemáticos dentro del razonamiento borroso son muy simples.
- La lógica borrosa es flexible: es fácil transformar un FIS añadiendo o eliminando reglas sin tener que empezar desde cero.

- La lógica borrosa admite datos imprecisos (NO estudia la incertidumbre): maneja elementos de un conjunto borroso, es decir, valores de una función de pertenencia. Por ejemplo, en lugar de manejar el dato "Mide 180 cm", maneja "Es alto con una precisión de 0.8".
- La lógica borrosa se construye sobre la experiencia de los expertos: confía en la experiencia de quien ya conoce el sistema.
- La lógica borrosa puede mezclarse con otras técnicas clásicas de control.
- Permite tomar una decisión completa en base a información conocida para el ser humano.

2.1.14 Ventajas de un Sistema de Control Difuso

La lógica difusa es muy simple a implementar y practicar comparando con las técnicas tradicionales de control basado en modelos matemáticos. Este tipo de sistemas utilizan un conjunto de ecuaciones diferenciales que permiten calcular la respuesta del sistema a partir de los señales de entrada. Además, no existe siempre un modelo matemático para todas las situaciones, y aunque existiría, nos llevaría mucho tiempo y sería muy costoso.

- Pueden ser evaluadas mayor cantidad de variables.
- Variables lingüísticas no numéricas, son usadas simulando el conocimiento humano.
- Un FLC puede enlazar entradas y salidas sin tener que entender todas las variables.
- Simplifica asignación de soluciones previas a problemas sin resolver.

- Es posible obtener prototipos rápidamente, ya que no requiere conocer todas las variables acerca del sistema antes de empezar a trabajar.
- El desarrollo de los FLC's es más económico que para los controladores convencionales, porque son más fáciles de diseñar.
- Simplifican la adquisición y representación del conocimiento.
- Por medio de los FLC, es cómodo designar rápidamente un prototipo que sistemas convencionales.
- Unas pocas reglas abarcan gran cantidad de complejidades.
- Los Sistemas difusos requieren mayor simulación, una excelente depuración y prueba antes de que sean operacionales.
- No se necesita conocer el modelo matemático que rige su funcionamiento.

2.1.15 Áreas de Aplicación de la lógica Difusa

En esta investigación realizada se pudo conocer que los sistemas basados en la lógica difusa posee varios campos de aplicación que según los requerimientos pueden resolver distintos problemas; de formas diferentes, por ello se realiza una especificación de algunas de áreas más comunes donde se aplicado un sistema difuso.

TABLA II.2 AREAS DE APLICACIÓN SE SISTEMAS DIFUSOS

	Problemas de interfaces Hombre/máquina	Problemas no lineales variantes en el tiempo	Clasificación de problemas
Problemas de los métodos convencionales	<ul style="list-style-type: none"> • Dificultad para expresar numéricamente los objetivos del control • Evaluación del control por interpretación humana 	<ul style="list-style-type: none"> • La dinámica de la planta varía con el tiempo. • Plantas no lineales <ul style="list-style-type: none"> • Sobre flujo oscilación 	<ul style="list-style-type: none"> • La acción a tomar no es clara • No es posible describir todas las trayectorias de solución • Limitaciones hardware/velocidad
Aplicaciones	<ul style="list-style-type: none"> • Control de suspensión • Transmisiones automáticas • Metro de Sendai 	<ul style="list-style-type: none"> • Control de temperatura • Control de posición de las cabezas de un disco duro • Pilotos automáticos 	<ul style="list-style-type: none"> • Auto ZOOM • Reconocimiento de patrones escritos a mano • Transmisiones automáticas

Fuente: Universidad de las América

Elaborado Por: Universidad de las América – Puebla

2.1.16 Inconvenientes de La Lógica Difusa

Estabilidad

No hay garantía teórica que un sistema difuso no tenga un comportamiento caótico y no siga siendo estable, aunque tal posibilidad parece ser baja debido a los resultados obtenidos hasta ahora.

Es difícil llegar a una función de membrecía o pertenencia y a una regla confiable sin la participación de un experto humano.

Capacidad de aprender

Son sistemas sin memoria, no poseen la capacidad de aprender. La determinación de las funciones de membrecía y las reglas no siempre son sencillas.

INTRODUCCION A JSP

Java Server Pages (JSP) es una tecnología basada en el lenguaje Java que permite incorporar contenido dinámico a las páginas web.

Con JSP podemos crear aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Se puede decir que la tecnología JSP es una extensión de la tecnología ofrecida por los servlets creada para ofrecer soporte a la creación de páginas HTML y XML.

SERVLETS

Los servlets no son más que una clase de Java. La clase Servlet fue diseñada para permitir desarrollar de una forma sencilla aplicaciones cliente-servidor¹².

Se ejecuta en un servidor de aplicación para contestar a las peticiones realizadas por los usuarios en HTTP, siendo el formato más común de salida una página HTML o un archivo XML.

La palabra servlet deriva de otra anterior, applet, que se refería a pequeños programas escritos en Java que se ejecutan en el contexto de un navegador web. Por contraposición, un servlet es un programa que se ejecuta en un servidor.

El uso más común de los servlets es generar páginas web de forma dinámica a partir de los parámetros de la petición que envía el navegador web.

¹² **ANGEL ESTEBAN.** Tecnologías De Servidor con Java. Servlets, JavaBeans, JSP.

De una forma más formal, podemos decir que un servlet es un objeto Java que implementa la interfaz `javax.servlet.Servlet` o que hereda de alguna de sus subclases.

Para poder trabajar con Servlets Java será necesario :

- **JDK Java:** El kit de desarrollo de Java para poder compilar nuestros servlets. Este Kit proporciona el conjunto de herramientas necesarias para el desarrollo de *servlets*.
- **Apache Tomcat:** Un contenedor de servlets, esto es, el lugar donde colocaremos nuestro servlets compilados y donde se ejecutarán. Será el servidor de servlets.
- **Navegador web:** Para poder acceder al contenedor de Servlets, poder utilizarlos e interactuar con ellos. Será el cliente y nos servirá cualquiera: Firefox, Internet Explorer, etc.
- **Una base de datos:** donde podremos almacenar nuestros datos y que será utilizada por nuestros servlets.

2.2.1 Estructura De Un Servlet

La clase `GenericServlet` proporciona una implementación básica del interfaz `Servlet`. Para escribir un servlet específicamente para el protocolo HTTP, se usa la clase `HttpServlet`, que extiende a `GenericServlet`.

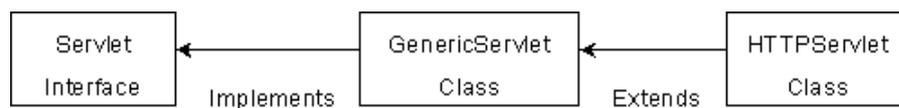


Figura. II.16 Clases Relevantes de Servlets e Interfaces.

Todo servlet debe directa o indirectamente implementar este interfaz donde se debe realizar una colección de declaraciones de métodos.

Los siguientes métodos¹³ están declarados en el interfaz Servlet.

1. `Public abstract void init (ServletConfig config) throws ServletException.`

El método *init* se usa para inicializar los parámetros proporcionados por el objeto ServletConfig. Se invoca sólo una vez, a menos que el servlet sea reiniciado si se destruye y se vuelve a cargar. El método *init* es el lugar en el que inicializar los parámetros de configuración como la conexión con una base de datos, la inicialización de archivos y otros valores de entorno. Ninguno método del servlet puede ser invocado a no ser que el servlet esté inicializado mediante el uso del método *init()*.

2. `Public abstract ServletConfig getServletConfig ()`

Este método proporciona el objeto ServletConfig para la inicialización de los parámetros del servlet. Se pueden crear parámetros adicionales especificándolos en el archivo `servlet.properties`. Una vez que hayan sido especificados en este archivo, se puede acceder a ellos usando el objeto ServletConfig.

3. `Public abstract void service (ServletRequest req, ServletResponse res) throws ServletException, IOException.`

El método *service* es el punto esencial del modelo petición respuesta del protocolo HTTP. Recibe una petición del cliente en forma de objeto ServletRequest. Los parámetros del cliente son pasados junto al objeto de petición (aunque existen otras formas de enviar parámetros desde el cliente al servlet, por ejemplo, usando cookies o

¹³ **ANGEL ESTEBAN.** Tecnologías De Servidor con Java. Servlets, JavaBeans, JSP.

por medio de una reescritura del URL). La respuesta resultante se envía al cliente usando el objeto `ServletResponse`.

4. `Public abstract String getServletInfo()`

Este método se usa para la extracción de metadatos del servlet, como por ejemplo el autor, la versión del servlet, y otra información concerniente al copyright. El método tendrá que ser redefinido dentro del servlet para que devuelva esta información.

5. `Public abstract void destroy ()`

El método *destroy* se invoca para liberar todos los recursos solicitados como la base de datos, y otros recursos del servidor. También se encarga de la sincronización de cualquier hilo pendiente. Este método se llama una sola vez, automáticamente, como el método *init*.

Además de los métodos que heredan de la clase `GenericServlet`, la clase `HttpServlet` tiene métodos adicionales para cada uno de los métodos de respuesta HTTP tratados antes.

1. `doDelete (HttpServletRequest, HttpServletResponse)`
2. `doGet (HttpServletRequest, HttpServletResponse)`
3. `doOptions (HttpServletRequest, HttpServletResponse)`
4. `doPost (HttpServletRequest, HttpServletResponse)`
5. `doPut (HttpServletRequest, HttpServletResponse)`
6. `doTrace (HttpServletRequest, HttpServletResponse)`

Por ejemplo, la clase Servlet tiene que redefinir `doGet` o `doPost` (o ambos métodos, dependiendo de si los datos serán enviados por un GET o por métodos de petición POST HTTP). Estos métodos toman dos parámetros: un objeto `HttpServletRequest` y un objeto `HttpServletResponse`. Estos dos objetos dan acceso total a toda la información sobre la petición del cliente, y ayudan a controlar la salida enviada al cliente como respuesta a dicha petición. Los métodos `doGet` y `doPost` son invocados por el método `service`. El método `service` puede usarse directamente al redefinirlo.

2.2.2 Ciclo De Vida Un Servlet

El ciclo de vida de un Servlet consta de los siguientes puntos:

2. El cliente solicita una petición a un servidor vía [URL](#).
3. El servidor recibe la petición.
 - Si es la primera petición, se utiliza el motor de Servlets para cargarlo y se llama al método `init()`.
 - Si ya está iniciado, cualquier petición se convierte en un nuevo hilo. Un Servlet puede manejar múltiples peticiones de clientes.
4. Se llama al método `service()` para procesar la petición devolviendo el resultado al cliente.
5. Cuando se apaga el motor de un Servlet se llama al método `destroy()`, que lo destruye y libera los recursos abiertos.

2.2.3 Características De Los Servlets

Los *servlets* poseen las siguientes características:

1. Son independientes del servidor utilizado y de su sistema operativo, lo que quiere decir que a pesar de estar escritos en **Java**, el servidor puede estar escrito en cualquier lenguaje de programación, obteniéndose exactamente el mismo resultado que si lo estuviera en **Java**.
2. Los *servlets* pueden llamar a otros *servlets*, e incluso a métodos concretos de otros *servlets*. De esta forma se puede distribuir de forma más eficiente el trabajo a realizar. Por ejemplo, se podría tener un *servlet* encargado de la interacción con los clientes y que llamara a otro *servlet* para que a su vez se encargara de la comunicación con una base de datos. De igual forma, los *servlets* permiten *redireccionar* peticiones de servicios a otros *servlets* (en la misma máquina o en una máquina remota).
3. Los *servlets* pueden obtener fácilmente información acerca del *cliente* (la permitida por el protocolo **HTTP**), tal como su dirección **IP**, el **puerto** que se utiliza en la llamada, el método utilizado (**GET**, **POST**, ..), etc.
4. Permiten además la utilización de *cookies* y *sesiones*, de forma que se puede guardar información específica acerca de un usuario determinado,

personalizando de esta forma la interacción cliente-servidor. Una clara aplicación es *mantener la sesión* con un cliente.

5. Los *servlets* pueden actuar como enlace entre el cliente y una o varias *bases de datos* en arquitecturas *cliente-servidor de 3 capas* (si la base de datos está en un servidor distinto).
6. Asimismo, pueden realizar tareas de *proxy* para un *applet*. Debido a las restricciones de seguridad, un *applet* no puede acceder directamente por ejemplo a un servidor de datos localizado en cualquier máquina remota, pero el *servlet* sí puede hacerlo de su parte.
7. Los *servlets* permiten la generación dinámica de código *HTML* dentro de una propia página *HTML*. Así, pueden emplearse *servlets* para la creación de contadores, banners, etc.

2.2.4 Ventajas De Los Servlet

- La tecnología *Servlet* proporciona las mismas ventajas del lenguaje *Java* en cuanto a *portabilidad* (“*write once, run anywhere*”) y *seguridad*, ya que un *servlet* es una *clase* de *Java* igual que cualquier otra, y por tanto tiene en ese sentido todas las características del lenguaje.

- Otra de las principales ventajas de los *servlets* es la del rendimiento, y esto a pesar de que *Java* no es un lenguaje particularmente rápido, pues los *servlets*, una vez que son llamados por primera vez, *quedan activos en la memoria del servidor hasta que el programa que controla el servidor los desactiva*. De esta manera se minimiza en gran medida el tiempo de respuesta. Además maneja threads que son más rápidos y ligeros que los procesos.
- Además, los *servlets* se benefician de la gran capacidad de Java para ejecutar métodos en ordenadores remotos, para conectar con bases de datos, para la seguridad en la información, etc. Se podría decir que las clases estándar de Java ofrecen resueltos muchos problemas que con otros lenguajes tiene que resolver el programador.

2.2 PÁGINAS JSP

Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java y que tiene la extensión jsp.

Las páginas JSP son en realidad *servlets*¹⁴: un JSP se compila a un programa en Java la primera vez que se invoca, y del programa en Java se crea una clase que se empieza a ejecutar en el servidor como un *servlet*.

¹⁴ ANGEL ESTEBAN. Tecnologías De Servidor con Java. Servlets, JavaBeans, JSP.

Los JSPs y servlets se ejecutan en una máquina virtual Java, lo cual permite que, en principio, se puedan usar en cualquier tipo de ordenador, siempre que exista una máquina virtual Java para él. Cada servlet o JSP, se ejecuta en su propia hebra, es decir, en su propio contexto; pero no se comienza a ejecutar cada vez que recibe una petición, sino que persiste de una petición a la siguiente, de forma que no se pierde tiempo en invocarlo (cargar programa + intérprete). Su persistencia le permite también hacer una serie de cosas de forma más eficiente: conexión a bases de datos y manejo de sesiones. Realmente una página JSP se convierte en un servlet que es quien se encarga de la comunicación con el cliente

2.3.1 Diferencias Entre JSP Y SERVLETS

Como se menciona anteriormente una página JSP no es más que un Servlet pero difiere de este en algunos aspectos que se mencionan a continuación:

- La principal diferencia entre los servlets y los JSPs es el enfoque de la programación: un JSP es una página Web con etiquetas especiales y código Java incrustado, mientras que un servlet es un programa que recibe peticiones y genera a partir de ellas una página web.
- La página JSP se puede definir como una presentación gráfica de una aplicación java y el servlet es una aplicación java.

- Además una Página JSP interactúa directamente con HTML mientras que el servlet lo hace imprimiendo código HTML mediante sentencias como el `out.println()`.
- En una página JSP ya se incluyen una serie de objetos (`request`, `response`, `session`, `out`, etc) que no hacen falta ser declarados pues el ya los tiene por defecto, a diferencia de un servlet en el que se los debe declarar.
- Una página JSP permite mejor comunicación con el usuario pues permite mostrar la interfaz al mismo en tanto que un servlet está dedicado al proceso de datos.

2.3.2 Ventajas De JSP

- Por ser considerado como un servlet las páginas JSP contendrán todas las ventajas de estos como son: Altas Prestaciones, Mayor escalabilidad, independencia de plataforma entre otras.

- Mayor encapsulamiento:

Una de las principales ventajas del JSP, es que permite al programador integrar los scripts con clases de Java (en estos casos llamados servlets), lo que permite tener por separado los módulos que se encargan de hacer los procesos de datos (por lo general los archivos `.class`), de los que se encargan de presentar visualmente dichos datos (los documentos JSP).

Permite el uso de clases

Permite el uso de *JavaBeans*

- Uso de *tags*: similitud con HTML
- Las páginas JSP permiten mayor portabilidad a otros sistemas operativos y servidores Web.

2.3.3 Características De Las Páginas JSP

1. Fusiona el poder de Java en el servidor y la flexibilidad de HTML (HyperText Markup Language) en el browser.
2. No sólo se puede utilizar HTML, sino también XML (eXtensible Markup Language) o WML (Wireless Markup Language).
3. Hace más fácil reutilizar componentes como JavaBeans y Enterprise JavaBeans los cuales realizan tareas más específicas.
4. Forma parte integral de Java 2 Enterprise Edition (J2EE).
5. Su función es saber cómo procesar una solicitud para crear una respuesta.
6. Soporta contenido dinámico que refleja las condiciones del mundo real.
7. Existe independencia entre la parte del diseño (interfaz) y la lógica (programa).
8. Ante un cambio se compila automáticamente. Ya que se realizó un precompilado en el servlet antes de ser usado.

2.3.4 Elementos JSP

Como ya se menciona Java Server Pages (JSP) es una tecnología basada en el lenguaje Java que permite incorporar contenido dinámico a las páginas web. Los archivos JSP combinan HTML con etiquetas especiales y fragmentos de código Java.

El código fuente de una página JSP puede contener¹⁵:

1. **Scriptlets:** Código Java embebido.
2. **Directivas:** Indican información general de la página, como puede ser importación de clases, página a invocar ante errores, si la página forma parte de una sesión, etc.
3. **Declaraciones:** Sirven para declarar métodos o variables.
4. **Expresiones:** Expresiones Java que se evalúan y se envían a la salida.
5. **Tags JSP:** Etiquetas especiales que interpreta el servidor.

2.3.4.1 Código JAVA

Podemos insertar código Java dentro de JSP de tres formas: Expresiones, Scriptlets, Comentarios y Declaraciones.

¹⁵ <http://java.sun.com/jsp>

2.3.4.1.1 Expresiones

Son fragmentos de código Java, con la forma `<%= expresión %>` que se evalúan y se muestran en la salida del navegador. En general, dentro de una expresión podemos usar cualquier cosa que usaríamos dentro de un `System.out.print(expr);`

Ejemplos:

```
<%= "Tamaño de cadena: "+cadena.length() %>
```

```
<%= new java.util.Date() %>
```

```
<%= Math.PI*2 %>
```

Los ejemplos muestran el tamaño de una cadena, la fecha actual y la multiplicación de Pi por 2 respectivamente.

2.3.4.1.2 Scriptlets

Son fragmentos de código Java con la forma `<% código %>`, en general, podemos insertar cualquier código que pudiéramos usar dentro de una función Java. Para acceder a la salida del navegador, usamos el objeto implícito `out`.

Ejemplo:

```
<%  
    if(ubProceso.mostrarVigente()==0)  
        out.println(ubProceso.getStrProDetalle());  
    else  
        out.println("Ninguno");  
%>
```

Los dos ejemplos anteriores realizan las mismas acciones y se podría decir que el segundo ejemplo es el más viable por tener menos líneas de código y parecer mucho más concreto, pero en general hay que evitar el uso de `out.println()` para elementos

HTML. En un proyecto en el que trabajen programadores y diseñadores conjuntamente, hay que separar presentación y código tanto como sea posible.

Dentro de un scriptlet podemos usar cualquier librería de Java, incluyendo las propias, lo cual hace que resulte muy sencillo construir interfaces *web* de entrada y salida para nuestras clases.

```
<%  
  
    ubDocente.setStrDocCedula(request.getParameter("id"));  
    if(ubDocente.mostrarDatos()==0)  
        out.print(ubDocente.getStrDocNombreCompleto());  
%>
```

2.3.4.1.3 Comentarios

Para introducir comentarios en JSP, usaremos las marcas `<%-- comentario --%>`, dentro de un scriptlet o declaración podemos usar comentarios siguiendo la sintaxis de Java.

```
<%-- Comentario JSP --%>  
<!-- Comentario HTML -->  
<%  
// Comentario  
/* Comentario */  
%>
```

2.3.4.1.4 Declaraciones

Contienen declaraciones de variables o métodos, con la forma `<%! declaración %>`. Estas variables o métodos serán accesibles desde cualquier lugar de la página JSP. Hay que tener en cuenta que el servidor transforma la página JSP en un servlet, y éste es usado por múltiples peticiones, lo que provoca que las variables conserven su valor entre sucesivas ejecuciones.

Ejemplos:

```
<%! int numeroAccesos=0; %>
<html>
  <body>
    <%=
      "La página ha sido accedida "+(++numeroAccesos)+
      " veces desde el arranque del servidor"
    %>
  </body>
</html>
```

Muestra el número de accesos al incrementar una variable.

```
<%! java.util.Date primerAcceso=new java.util.Date(); %>
<html>
  <body>
    El primer acceso a la página se realizó en:
    <%= primerAcceso %>
  </body>
</html>
```

En el ejemplo el método guarda la fecha del primer acceso y lo muestra al ser invocado.

```
<%!
  private String ahora()
  {
    return ""+new java.util.Date();
  }
%>
<html>
  <body>
    <%= ahora() %>
  </body>
</html>
```

En el ejemplo el método retorna la fecha actual y lo devuelve al ser invocado.

2.3.4.2 Directivas

Las directivas¹⁶ son elementos que proporcionan información al motor JSP, e influirán en la estructura del servlet generado. Hay tres tipos de directivas: page, taglib e include.

2.3.4.2.1 Page

Se indica con la forma `<% @ page atributo="valor">`. Tiene diversos usos, entre los cuales destacaremos:

- Importar clases. Importar código, de la misma forma que se realiza en un programa en Java, se indica con el atributo import.

Ejemplo:

```
<% @page import="java.io.*, miPackage.miClase"%>
```

- Indicar si la página tendrá acceso a la sesión. Se especifica con el atributo session. El uso de sesiones se verá con más detalle en el apartado de objetos implícitos.

Ejemplo:

```
<% @page session="true" import="java.util.ArrayList"%>
```

- Gestión de errores. Permite redireccionar a una página cuando se produzca un error, se indica con los atributos errorPage y isErrorPage.

Ejemplos:

¹⁶ ANGEL ESTEBAN. Tecnologías De Servidor con Java. Servlets, JavaBeans, JSP.

```
<% @page errorPage="error.jsp">
```

2.3.4.2.2 Include

Permite incluir un archivo en el lugar donde se especifique, al contrario que con la acción `<jsp:include>` que veremos más adelante, la directiva `include` simplemente copia el contenido del archivo byte a byte, siendo el resultado similar a si copiáramos el texto del archivo incluido y lo pegáramos en el JSP.

Ejemplo:

```
<html>
  <head>
    <% @ include file="titulo.txt"%>
  </head>
  <body>
    <% @ include file="cuerpoPagina.jsp"%>
  </body>
</html>
```

2.3.4.2.2 Taglib

Se emplea para indicar que se van a emplear librerías de etiquetas

Ejemplo:

```
<% @ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

2.3.4.3 Acciones

Las acciones tienen la forma `<jsp:accion [parámetros]/>`, y tienen diversos usos, entre los que destacan la inclusión de páginas y transferencia de control.

Existen siete acciones estándar dentro de la especificación JSP. Estas etiquetas afectan el comportamiento de la ejecución de las páginas JSP y afectan también a la respuesta devuelta al cliente.

Las acciones estándar ofrecen al desarrollador o autor de páginas JSP una funcionalidad básica.

Las acciones estándar son:

- `<jsp:include>`
- `<jsp:forward>`
- `<jsp:useBean>`
- `<jsp:plugin>`
- `<jsp:param>`
- `<jsp:setProperty>`
- `<jsp:getProperty>`

- **`<jsp:include>` Inclusión de páginas**

Se realiza con la acción `<jsp:include page="pagina.jsp">`. Incluye la salida de otra página JSP en la actual, al contrario que con la directiva `<%@include file="fichero.ext"%>` la página incluida se *ejecuta* y su salida se inserta en la página que la incluye, con la directiva se incluye el contenido del archivo (no su salida) y se ejecuta conjuntamente con la página principal.

La página incluida tiene acceso a los parámetros enviados a la principal, y podemos enviarle nuevos parámetros con la subetiqueta `<jsp:param name="nombre" value="valor"/>`.

Ejemplo:

```
<html>
  <head>
    <jsp:include page="cabecera.jsp"/>
  </head>
  <body>
    <jsp:include page="cuerpo.jsp">
      <jsp:param name="tipo" value="paginaPrincipal"/>
    </jsp:include>
  </body>
</html>
```

- **<jsp:forward>Transferencia de control**

Se realiza con la acción `<jsp:forward page="pagina.jsp"/>`. La petición es redirigida a otra página, y la salida de la actual se descarta. Al igual que con la inclusión, la página a la que se redirige tiene acceso a los parámetros pasados a la actual, y es posible el envío de nuevos parámetros.

Ejemplo:

```
<jsp:forward page="login.jsp">
<jsp:param name="validar" value="1"/>
</jsp:forward>
```

- **<jsp:useBean> Declaración de Componentes Java**

Esta acción se utiliza para poder utilizar dentro de una página JSP un componente JavaBean en un ámbito determinado. El componente JavaBean podrá ser utilizado dentro de la página JSP haciendo referencia al nombre indicado dentro de la acción

<jsp:useBean>, teniendo siempre en cuenta el ámbito al que pertenece el Bean, y que se indica también en la acción <jsp:useBean>.

Si tiene cuerpo se realiza con la acción:

```
<jsp:useBean id="ubProceso" scope="request" class="AccesoDatos.Procesos"/>
```

El cuerpo suele contener etiquetas de tipo jsp:setProperty que se empleara para modificar el objeto que se acaba de crear

En caso de no tener cuerpo su sintaxis es:

```
<jsp:useBean id="nom" scope="sco" class="NombreClase"
```

➤ **Atributos**

- **Id:** nombre utilizado para identificar la instancia del objeto así como la variable declarada e inicializada con referencia al objeto durante todo su ciclo de vida.
- **Scope:** Ámbito de dicha referencia; puede ser:
 - Page:** accesible desde un objeto PageContext.
 - Request:** accesible desde un objeto servletRequest.
 - Sesión:** accesible desde un objeto HttpSession.
 - Application:** accesible desde un ServletContext.
- **Class:** en este atributo se indica el nombre de la clase del Bean, a cada Bean le va a corresponder una clase para organizar los componentes JavaBeans estas clases suelen encontrarse en paquetes, y si no los hemos importado en la directiva page, deberemos utilizar el nombre completo de la clase del Bean, indicando paquetes y subpaquetes.
- **type:** este atributo no suele ser utilizado, es opcional e indica el tipo de la clase del Bean, este tipo suele ser la clase padre, un interfaz o la propia

clase, se utilizará para casos muy concretos cuando queremos realizar una casting de clases. Por defecto tiene el mismo valor que el indicado atributo class.

- **beanName**: es el nombre del Bean que se le pasa al método instantiate() de la clase java.beans.Beans, es posible indicar el atributo type y el atributo beanName y omitir el atributo class.

- **<jsp:plugin>**

Esta acción permite generar código HTML para asegurar que el navegador utiliza el plug-in de Java. Donde el servidor JSP transforma automáticamente en una etiqueta que incluye un applet para indicar al navegador que para un applet determinado debe cargar el plug-in de Java para que el applet pueda ser ejecutado satisfactoriamente por el navegador.

- **<jsp:param>**

Esta acción se utiliza en colaboración con cualquiera de las siguientes acciones:

<jsp:forward>, <jsp:include> o <jsp:plugin>. La sintaxis general de esta acción es la siguiente:

```
<jsp:param name="nombreParametro" value="valorParametro"/>
```

La acción <jsp:param> permite ofrecer información adicional a otra acción.

```
<html>  
  <head>  
    <title>Param e include</title>  
  </head>  
  <body>
```

```
<h3>Página que incluye a otra</h3>
<jsp:include page="incluida.jsp" flush="true">
<jsp:param name="ciudad" value="Riobamba"/>
<jsp:param name="domicilio" value="Santa Rosa"/>
</jsp:include>
<h3>Se vuelve a la página de origen</h3>
</body>
</html>
```

- **<jsp:getProperty>**

Esta acción nos permiten utilizar componentes JavaBeans dentro de nuestras páginas JSP, en este caso la acción `<jsp:getProperty>` nos va a permitir obtener el valor de la propiedad de un JavaBean creado en la página.

La sintaxis de esta acción no posee cuerpo y presenta dos atributos o propiedades:

```
<jsp:getProperty name="nombreBean"
property="nombrePropiedad" />
```

La propiedad **name** indica el identificador del Bean que hemos creado con la acción `<jsp:useBean>`, y cuyo valor de la propiedad queremos obtener. Se corresponderá con el valor del atributo **id** de la acción `<jsp:useBean>` correspondiente.

El atributo **property** indica el nombre de la propiedad del Bean cuyo valor se desea obtener. El valor de la propiedad se mostrará como código HTML, reemplazando en tiempo de ejecución a la acción `<jsp:getProperty>` correspondiente.

Esta acción accede al valor de la propiedad especificada del Bean correspondiente, la convierte en un objeto de la clase String y la imprime en el flujo de salida de la página JSP.

- **<jsp:setProperty>**

Esta acción permite modificar las propiedades de los Beans a los que hacemos referencia en nuestras páginas JSP a través de la acción `<jsp:useBean>`, es la acción complementaria a la acción `<jsp.getProperty>`. Su sintaxis general es la que se muestra a continuación:

```
<jsp:setProperty name="nombreBean" detallesPropiedad/>
```

El atributo **name** tiene el mismo significado que en la acción vista anteriormente, es decir, es el identificador del componente JavaBean al que se hace referencia en la página.

Los **detalles** de la propiedad son una serie de atributos que combinados entre sí permiten asignar el valor a la propiedad del Bean de distinta forma. Así por ejemplo la forma de establecer el valor de la propiedad de un Bean puede ser cualquiera de las que aparecen a continuación:

- `property="*"`
- `property="nombrePropiedad"`
- `property="nombrePropiedad" param="nombreParámetro"`
- `property="nombrePropiedad" value="valorPropiedad"`

2.3.5 OBJETOS IMPLÍCITOS DE JSP

En JSP disponemos de algunos objetos implícitos¹⁷, que nos permitirán acceder a diferente información y realizar diversas acciones. En JSP tenemos los siguientes

¹⁷ http://www.programacionfacil.com/java_jsp/start.

objetos implícitos: request, response, out, session, application, config, pageContext, y page. A continuación describiremos las funciones de estos objetos.

2.3.5.1 request

Es un objeto de la clase `HttpServletRequest`, su uso principal es el acceso a los parámetros de la petición. Destacaremos las siguientes funciones:

- `String getParameter(String name)`
Devuelve el valor de un parámetro.
- `Enumeration getParameterNames()`
Devuelve una enumeración con los nombres de todos los parámetros de la petición.
- `String [] getParameterValues(String name)`
Los parámetros pueden tener valor múltiple, con esta función recuperamos un array con todos los valores para un nombre dado.
- `String getRemoteAddr()`
Devuelve la IP del *host* desde el que se realiza la petición.
- `String getRemoteHost()`

Devuelve el nombre del *host* desde el que se realiza la petición.

Ejemplo:

```
<%
//PROMEDIO TOTAL DOCENTE ESTUDIANTE
ubEstudianteDocente.setStrDocCedula(request.getParameter("id"));
ubEstudianteDocente.setStrProCodigo(ubProceso.getStrProCodigo());
ResultSet          rsEstudianteDocenteTotal          =
ubEstudianteDocente.recuperarPromedioTotalDocente();
    if (rsEstudianteDocenteTotal!=null)
    {

        while(rsEstudianteDocenteTotal.next())
        {
            out.print(rsEstudianteDocenteTotal.getString("PROMEDIO"));
        }
    }
%>
```

2.3.5.2 response

Es un objeto de la clase `HttpServletResponse`, que asiste al servlet en su generación de la respuesta para el cliente, contiene funciones para manejo de cabeceras, códigos de estado, *cookies* y transferencia de control.

2.3.5.3 out

Es un objeto de la clase `JspWriter`, es el que nos permite acceder a la salida del navegador desde los *scriptlet*.

Ejemplo:

```
<%
    out.print("promedio");
```

```
        out.println("fuzzi");  
%>
```

2.3.5.4 Session

Es un objeto de la clase HttpSession. Nos permite acceder a la sesión asociada a la petición. A través de este objeto podemos, entre otras cosas, guardar objetos que serán accesibles desde cualquier JSP de la sesión o invalidarla.

Para guardar y recuperar información usaremos:

```
Object session.getAttribute("clave");  
void session.setAttribute("clave", Object objeto);
```

Y para invalidar la sesión:

```
void session.invalidate();
```

2.3.5.5 application

Es un objeto de la clase ServletContext. Este objeto es común para toda la aplicación web y, entre otras cosas, nos permite almacenar información que será accesible desde todas las páginas de la aplicación web, independientemente de la sesión.

Para guardar y recuperar valores:

```
Object application.getAttribute("clave");  
void application.setAttribute("clave", Object objeto);
```

2.3.5.6 config

Es un objeto de la clase ServletConfig. Permite acceder a parámetros de inicialización del servlet y a su contexto.

2.3.5.7 pageContext

Es un objeto de la clase PageContext. Entre otras cosas, nos permite almacenar información localmente a la página.

Para guardar y recuperar valores:

```
Object pageContext.getAttribute("clave");  
void pageContext.setAttribute("clave", Object objeto);
```

También podemos usar PageContext para almacenar y recuperar información en sesión y en aplicación:

Almacenar en contexto de página:

```
PageContext.setAttribute("clave", obj, PageContext.PAGE_SCOPE);  
PageContext.setAttribute("clave", obj);
```

Almacenar en contexto de sesión:

```
PageContext.setAttribute("clave",obj, PageContext.SESSION_SCOPE);  
session.setAttribute("clave", objeto);
```

Almacenar en contexto de aplicación:

```
PageContext.setAttribute("clave",obj,PageContext.APPLICATION_SCOPE);  
application.setAttribute("clave",objeto);
```

2.3.5.8 page

Es un sinónimo de this, no tiene utilidad en el estado actual de la especificación.

2.3.6 FORMULARIOS EN PÁGINAS JSP

Es necesario conocer las distintas formas de enviar parámetros desde un formulario y de recibirlos desde un JSP. En este apartado daremos a conocer alguna de las formas que pueden ser usadas.

2.3.6.1. Conceptos básicos

Un formulario HTML tiene la forma:

```
<form action="destino" method="método">
```

Elementos de formulario

```
</form>
```

En *destino* especificaremos la página que recibe los datos del formulario en el atributo `method` podemos indicar dos valores diferentes GET y POST. Si no se especifica el valor de los atributos, los valores por defecto son la página actual para `action` y GET para `method`.

2.2.6.2. GET y POST

Cuando usamos GET, la información se codifica directamente en la URL, con la forma:

```
http://url?param1=valor1&param2=valor2...&paramN=valorN
```

Con GET no podemos manejar grandes cantidades de información, y existe la desventaja de que el servidor o el navegador guarden en caché la página llamada. Hay que tener en cuenta que los *logs* del servidor y el historial del navegador guardarán el

acceso incluyendo los parámetros, lo cual hace desaconsejable el uso del GET para el envío de información privada. Por otro lado, al visualizarse en la URL los parámetros, facilita el desarrollo y depurado de la aplicación web, y en algunos casos, es imprescindible para realizar estadísticas basadas en los *logs* del servidor.

Con POST la información se envía directamente al servidor, no se codifica en la URL, y además permite el envío de grandes cantidades de información, como podrían ser archivos.

2.2.6.3. Elementos de formulario

Se indican con las etiquetas HTML:

```
<input type="tipo" name="nombre" value="valor"/>
<textarea name="nombre"/>Contenido por defecto</textarea>
<select name="nombre">
    <option value="valorOpcion">Texto opcion</option>
    [...]
</select>
```

Para enviar los datos usamos el tipo submit.

```
<input type="submit"/>
```

2.2.6.3.1. Campos de texto.

Los tipos que se envían como texto simple son text y password para <input>, y el elemento <textarea>.

```
<input name="textfield" type="password" value="" />
```

2.2.6.3.2. Selectores ON/OFF. Checkbox.

Se indica con el tipo checkbox.

```
<input type="checkbox" name="nombreCheckbox"/>
```

Si el checkbox está marcado, se envía un parámetro con el nombre especificado con el valor on. Si no está marcado, no se envía el parámetro. Así que podemos recibirlo en JSP con la directiva request.

```
String checkbox= request.getParameter("nombreCheckbox");
```

2.2.6.3.3 Parámetros ocultos. Tipo hidden.

Se indican con el tipo hidden, los pares clave valor indicados se enviarán siempre junto con el resto de información del formulario. Se reciben en el JSP de la misma forma que los campos de texto.

2.2.6.4. Botones radio.

Son grupos de valores ON/OFF, sólo puede haber uno seleccionado dentro del grupo con el mismo nombre en el atributo name. Sólo se envía un parámetro para el botón seleccionado, con el valor indicado en la etiqueta value. El valor se recibe en el JSP de forma similar a los campos de texto.

2.2.6.5. Selecciones. Etiquetas <select> y <option>.

Se usan para despletables y listas. Se especifica un nombre para el parámetro y se envía como valor el contenido de la etiqueta <option>. Si especificamos el atributo value en la etiqueta option, Se mostrará en el despletable el texto en el cuerpo de la etiqueta, pero se enviará el valor especificado en el atributo value en caso de ser la opción seleccionada.

Ejemplo:

```
<form action="pagina.jsp">
  <select name="selectSimple">
    <option value="1">Uno</option>
    <option>Dos</option>
    <option>Tres</option>
    <option>Cuatro</option>
  </select>
  <input type="submit"/>
</form>
```

También podemos crear listas que nos permitan la selección de múltiples valores, esto se indica añadiendo el atributo MULTIPLE al <select> en este caso se generan parámetros con el mismo nombre para cada opción seleccionada, lo que nos obliga a recuperar estos valores con la función:

1.2.6.6. Codificación directa en el URL.

En algunos casos resulta útil enviar directamente información a JSP, codificada en la URL.

Ejemplo:

```
<a href="noticias.jsp?param=nacional">Actualidad</a> <br>
<a href="noticias.jsp?param=deportes">Deportes</a> <br>
<a href="noticias.jsp?param=sociedad">Sociedad</a> <br>
```

Debemos tener cuidado cuando enviemos caracteres especiales, para codificar cadenas podemos usar la función estática `String java.net.URLEncoder.encode(String s);`

Al pulsar un enlace se puede comprobar los parámetros en la barra de direcciones.

```
<br>
```

```
<a href="<%=url%>">URL sin codificar</a>
```


<a href="<%=urlCodificada%>" />URL codificada

2.3.7 ACCESO A DATOS EN JSP

Las páginas JSP utilizan el API de acceso a bases de datos que ofrece el lenguaje Java, y este API es común a cualquier tecnología Java, ya sea a aplicaciones convencionales, applets, servlets o páginas JSP.

El API que ofrece el lenguaje Java para el acceso a bases de datos es el API JDBC.

2.3.7.1 JDBC

JDBC (*Java DataBase Connectivity*)¹⁸ es un API de Java que permite al programador ejecutar instrucciones en lenguaje estándar de acceso a Bases de Datos, SQL (*Structured Query Language*, lenguaje estructurado de consultas), que es un lenguaje de muy alto nivel que permite crear, examinar, manipular y gestionar Bases de Datos relacionales. Para que una aplicación pueda hacer operaciones en una Base de Datos, ha de tener una conexión con ella, que se establece a través de un *driver*, que convierte el lenguaje de alto nivel a sentencias de Base de Datos. Es decir, las tres acciones principales que realizará JDBC son las de establecer la conexión a una base de datos, ya

1. ¹⁸ <http://www.scribd.com/doc/3321228/JDBC?page=37>

sea remota o no; enviar sentencias SQL a esa base de datos y, en tercer lugar, procesar los resultados obtenidos de la base de datos.

2.3.7.2 Conectividad JDBC

Para la gente del mundo Windows, JDBC es para Java lo que ODBC es para Windows. Windows en general no sabe nada acerca de las bases de datos, pero define el estándar ODBC consistente en un conjunto de primitivas que cualquier *driver* o fuente ODBC debe ser capaz de entender y manipular. Los programadores que a su vez deseen escribir programas para manejar bases de datos genéricas en Windows utilizan las llamadas ODBC.

Con JDBC ocurre exactamente lo mismo: JDBC es una especificación de un conjunto de clases y métodos de operación que permiten a cualquier programa Java acceder a sistemas de bases de datos de forma homogénea. Lógicamente, al igual que ODBC, la aplicación de Java debe tener acceso a un driver JDBC adecuado. Este driver es el que implementa la funcionalidad de todas las clases de acceso a datos y proporciona la comunicación entre el API JDBC y la base de datos real.

La necesidad de JDBC, a pesar de la existencia de ODBC, viene dada porque ODBC es un interfaz escrito en lenguaje C, que al no ser un lenguaje portable, haría que las aplicaciones Java también perdiesen la portabilidad. Y además, ODBC tiene el inconveniente de que se ha de instalar manualmente en cada máquina; al contrario que los drivers JDBC, que al estar escritos en Java son automáticamente instalables, portables y seguros.

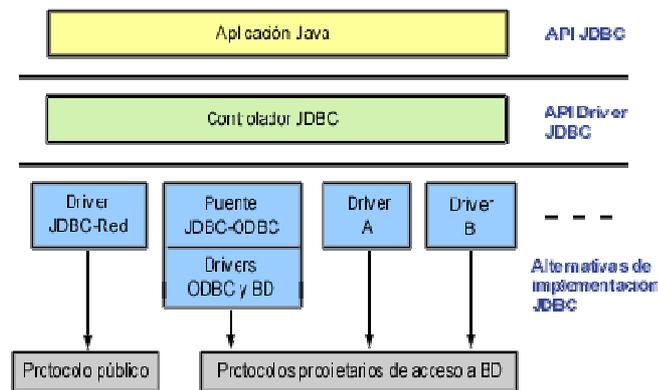


Figura. II.17. Componentes Java.

Toda la conectividad de bases de datos de Java se basa en sentencias SQL, por lo que se hace imprescindible un conocimiento adecuado de SQL para realizar cualquier clase de operación de bases de datos. Aunque, afortunadamente, casi todos los entornos de desarrollo Java ofrecen componentes visuales que proporcionan una funcionalidad suficientemente potente sin necesidad de que sea necesario utilizar SQL, aunque para usar directamente el JDK se haga imprescindible.

2.3.7.3 Acceso De JDBC A Bases De Datos

El API JDBC soporta dos modelos diferentes de acceso a Bases de Datos, los modelos de dos y tres capas.

- **Modelo de dos capas**

Este modelo se basa en que la conexión entre la aplicación Java, el applet o aplicación jsp que se ejecuta en el navegador, se conectan directamente a la base de datos.

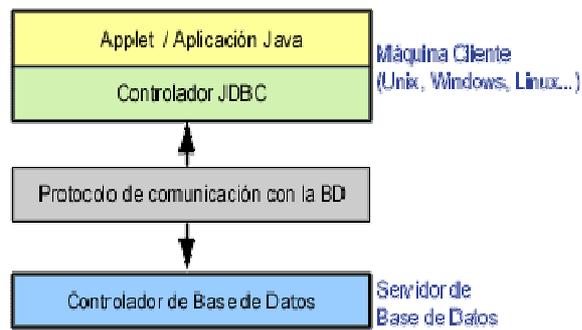


Figura. II.18. Modelo de dos Capas.

Esto significa que el driver JDBC específico para conectarse con la base de datos, debe residir en el sistema local. La base de datos puede estar en cualquier otra máquina y se accede a ella mediante la red. Esta es la configuración de típica *Cliente/Servidor*: el programa cliente envía instrucciones SQL a la base de datos, ésta las procesa y envía los resultados de vuelta a la aplicación.

- **Modelo de tres capas**

En este modelo de acceso a las bases de datos, las instrucciones son enviadas a una capa intermedia entre Cliente y Servidor, que es la que se encarga de enviar las sentencias SQL a la base de datos y recoger el resultado desde la base de datos. En este caso el usuario no tiene contacto directo, ni a través de la red, con la máquina donde reside la base de datos.

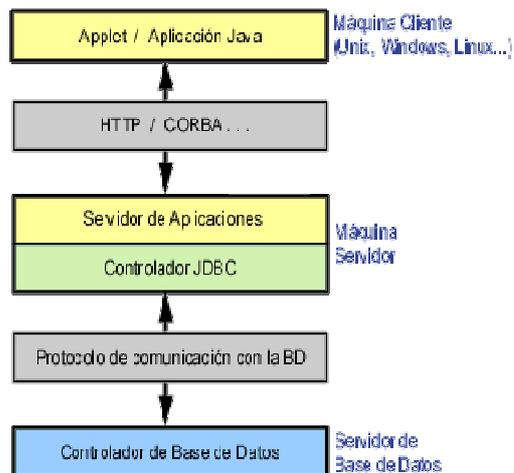


Figura. II.19 Modelo en tres capas.

Este modelo presenta la ventaja de que el nivel intermedio mantiene en todo momento el control del tipo de operaciones que se realizan contra la base de datos, y además, está la ventaja adicional de que los drivers JDBC no tienen que residir en la máquina cliente, lo cual libera al usuario de la instalación de cualquier tipo de driver.

2.3.7.4 Interfaces JDBC

JDBC define ocho interfaces¹⁹ para operaciones con bases de datos, de las que se derivan las clases correspondientes. La figura siguiente, en formato OMT, con nomenclatura UML, muestra la interrelación entre estas clases según el modelo de objetos de la especificación de JDBC.

¹⁹ www.programacion.net/java/tutorial

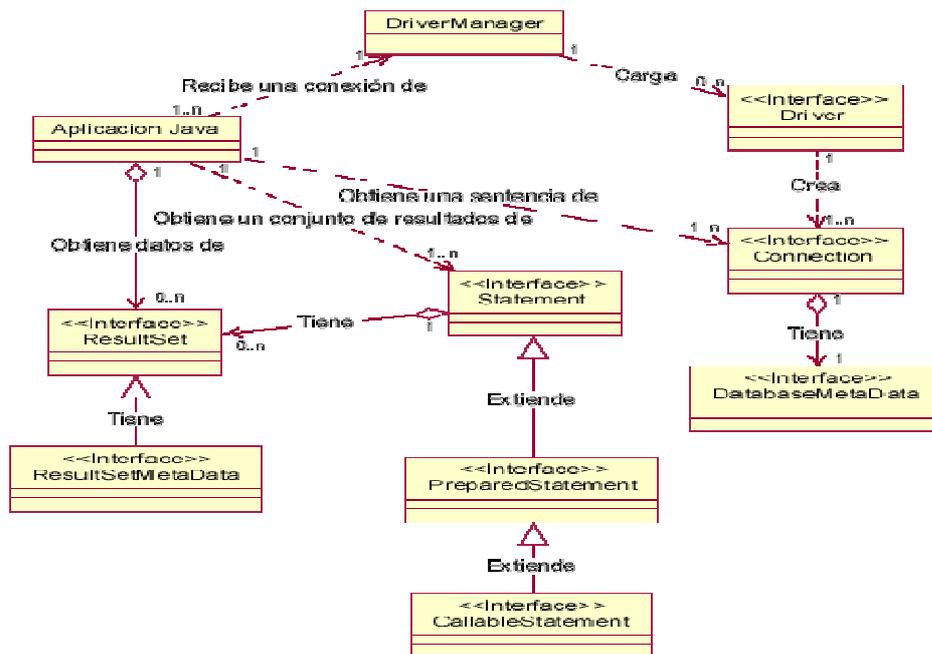


Figura. II.20 Clases y Objetos del JDBC

La clase que se encarga de cargar inicialmente todos los drivers JDBC disponibles es **DriverManager**. Una aplicación puede utilizar **DriverManager** para obtener un objeto de tipo conexión, **Connection**, con una base de datos. La conexión se especifica siguiendo una sintaxis basada en la especificación más amplia de los URL, de la forma *jdbc:subprotocolo//servidor:puerto/base de datos*

Por ejemplo, si se utiliza mSQL el nombre del subprotocolo será **msql**. En algunas ocasiones es necesario identificar aún más el protocolo. Por ejemplo, si se usa el puente JDBC-ODBC no es suficiente con **jdbc:odbc**, ya que pueden existir múltiples drivers ODBC, y en este caso, hay que especificar aún más, mediante **jdbc:odbc:fuentes de datos**. Una vez que se tiene un objeto de tipo **Connection**, se pueden crear sentencias, **statements**, ejecutables. Cada una de estas sentencias puede devolver cero o más resultados, que se devuelven como objetos de tipo **ResultSet**.

Y la tabla siguiente muestra la misma lista de clases e interfaces junto con una breve descripción.

TABLA II.3 CLASES E INTERFACES DEL DRIVER JDBC

<i>Clase/Interface</i>	<i>Descripción</i>
Driver	Permite conectarse a una base de datos: cada gestor de base de datos requiere un driver distinto
DriverManager	Permite gestionar todos los drivers instalados en el sistema
DriverPropertyInfo	Proporciona diversa información acerca de un driver
Connection	Representa una conexión con una base de datos. Una aplicación puede tener más de una conexión a más de una base de datos
DatabaseMetadata	Proporciona información acerca de una Base de Datos, como las tablas que contiene, etc.
Statement	Permite ejecutar sentencias SQL sin parámetros
PreparedStatement	Permite ejecutar sentencias SQL con parámetros de entrada
CallableStatement	Permite ejecutar sentencias SQL con parámetros de entrada y salida, típicamente procedimientos almacenados
ResultSet	Contiene las filas o registros obtenidos al ejecutar un SELECT
ResultSetMetadata	Permite obtener información sobre un ResultSet , como el número de columnas, sus nombres, etc.

Fuente: www.programacion.net/java/tutorial

Elaborado por: Java Sun

El siguiente diagrama relaciona las cuatro clases principales que va a usar cualquier programa Java con JDBC, y representa el esqueleto de cualquiera de los programas que se desarrollan para atacar a bases de datos.

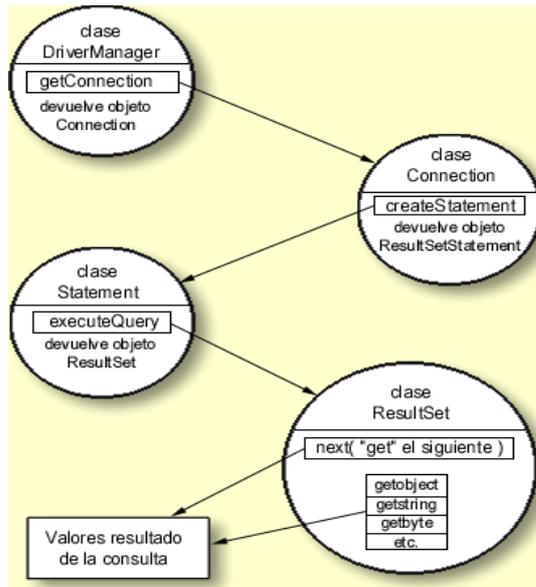


Figura. II.21 Principales Clases de JDBC.

La aplicación siguiente es un ejemplo en donde se aplica el esquema anterior, se trata de un paquete java que muestra los resultados al recorrer la tabla **escuelas**

```
import java.sql.*;
public class Conexion
{
    public final static String DRIVER="com.microsoft.sqlserver.jdbc.SQLServerDriver";
    public final static String CADENA_CONEXION="jdbc:sqlserver://localhost:1433;databaseName=SIEDD";
    public static String USER = "usuario";
    public static String PASSWORD= "usuario";
}

Public class escuela{

public Escuela()
{
    this.strCadenaConexion = Conexion.CADENA_CONEXION;
    this.strDriver = Conexion.DRIVER;
    this.strUsuario = Conexion.USER;
    this.strPassword = Conexion.PASSWORD;
}

public byte mostrarDatos()
{
    byte bytEstado=2;

    try
    {
        Class.forName(strDriver);
    }
    catch(ClassNotFoundException e)
    {

```

```
        this.strErrorDriver=e.getMessage().toString();
        return bytEstado=1;
    }
    try
    {
        //Establecer la conexion
        conexion= DriverManager.getConnection(strCadenaConexion,strUsuario,strPassword);
        //Creamos la sentencia
        sentencia=conexion.createStatement();
        /*Ejecutamos una consulta contra la base de datos. Obtenemos un ResultSet como
        resultado de la consulta*/
        resultado=sentencia.executeQuery("SELECT * FROM ESCUELA WHERE
        EscCodigo="+strEscCodigo+""");
        //Procesamos los resultados
        //Mientras siga habiendo registros, desplazamos una posición el cursor,
        if (resultado.next())
        {
            //Accedemos a los registros resultado mediante los métodos getXXXX
            //Indicando el nombre de la columna a la que queremos acceder

            strFacCodigo = resultado.getString("FacCodigo");
            strEscCodigo = resultado.getString("EscCodigo");
            strEscDetalle= resultado.getString("EscDetalle");

            bytEstado = 0;
        }
    }
    catch (Exception e)
    {
        //Asigna el mensaje de error
        this.strError = e.getMessage().toString();
        bytEstado = 1;
    }
    finally
    {
        try
        {
            if (sentencia!=null)
                sentencia.close();
            if (resultado!=null)
                resultado.close();
            if(conexion!=null)
                conexion.close();
        }
        catch(SQLException sqle ){
            this.strErrorDriver = sqle.getMessage().toString();
        }
    }
    return bytEstado;
}
}
```

Las partes más interesantes del código son las que se van a revisar a continuación, profundizando en cada uno de los pasos.

En primer lugar se importa toda la funcionalidad de JDBC, a través de la primera sentencia ejecutable del programa.

```
import java.sql.*;
```

Se define una clase conexión donde se puntualiza la cadena de conexión, el driver que permitirá realizar la carga del puente JDBC-ODBC además del usuario y password con el que se intentará de acceso a la base de datos.

```
public class Conexion
{
public final static String DRIVER="com.microsoft.sqlserver.jdbc.SQLServerDriver";
public final static String CADENA_CONEXION="jdbc:sqlserver://localhost:1433;databaseName=SIEDD";
public static String USER = "usuario";
public static String PASSWORD= "usuario";
}
```

Se define una nueva clase denominada escuela que hará uso de la clase conexión en el constructor de dicha clase.

```
public Escuela()
{
    this.strCadenaConexion = Conexion.CADENA_CONEXION;
    this.strDriver = Conexion.DRIVER;
    this.strUsuario = Conexion.USER;
    this.strPassword = Conexion.PASSWORD;
}
```

Luego en una clase denominada mostrar datos haremos uso de todos estos elementos primero el método `forName()` localiza, lee y enlaza dinámicamente una clase determinada. Para drivers JDBC, la sintaxis que JavaSoft recomienda de `forName()` es `nombreEmpresa.nombreBaseDatos.nombreDriver`, y el driver deberá estar ubicado en el directorio `nombreEmpresa\nombreBaseDatos\nombreDriver.class` a partir del directorio

indicado por la variable de entorno CLASSPATH. En este caso se indica que el puente JDBC-ODBC que se desea leer es precisamente el de Microsoft.

Si por cualquier motivo no es posible conseguir cargar JdbcOdbcDriver.class, se intercepta la excepción y se sale del programa. En este momento es la hora de echar mano de la información que puedan proporcionar las trazas ODBC.

A continuación, se solicita a DriverManager que proporcione una conexión para una fuente de datos ODBC. Donde strCadenaConexión es la dirección donde se encuentra la base de datos a la que queremos acceder.

```
conexion= DriverManager.getConnection(strCadenaConexion,strUsuario,strPassword);
```

El segundo y tercer parámetro son el nombre del usuario y la clave con la cual se intentará la conexión. Esta es la única línea que con seguridad habrá de cambiar el programador para probar sus aplicaciones. **getConnection** admite también una forma con un único parámetro (el URL de la base de datos), que debe proporcionar toda la información de conexión necesaria al driver JDBC correspondiente.

Para el resto de los drivers JDBC, habrá que consultar la documentación de cada driver en concreto.

Inmediatamente después de obtener la conexión, en la siguiente línea

```
sentencia=conexion.createStatement();
```

Se solicita que proporcione un objeto de tipo Statement para poder ejecutar sentencias a través de esa conexión. Para ello se dispone de los métodos execute(String sentencia) para ejecutar una petición SQL que no devuelve datos o executeQuery(String sentencia) para ejecutar una consulta SQL. Este último método devuelve un objeto de tipo ResultSet.

Una vez que se tiene el objeto Statement ya se pueden lanzar consultas y ejecutar sentencias contra el servidor. A partir de aquí el resto del programa realmente es SQL adornado: en la línea:

```
resultado=sentencia.executeQuery("SELECT * FROM ESCUELA WHERE  
EscCodigo="+strEscCodigo+"");
```

En resultado se guarda la sentencia ejecutada. Ahora se verán los pasos que hay que dar para obtener información a partir de una base de datos ya creada. Como se ha dicho anteriormente, se utilizará executeQuery en lugar de execute para obtener resultados.

```
if (resultado.next())  
{  
    //Accedemos a los registros resultado mediante los métodos getXXXX  
    //Indicando el nombre de la columna a la que queremos acceder  
  
    strFacCodigo = resultado.getString("FacCodigo");  
    strEscCodigo = resultado.getString("EscCodigo");  
    strEscDetalle= resultado.getString("EscDetalle");  
  
    bytEstado = 0;  
}
```

En caso de que exista más de un elemento de la variable resultado se debería usar un while en lugar de if.

Se utilizan los métodos getXXX() de resultado para tener acceso a las diferentes columnas. El acceso se puede hacer por el nombre de la columna, como en las dos primeras líneas, o bien mediante su ubicación relativa, como en la última línea. Además de getString() están disponibles getBoolean(), getByte(), getDouble(), getFloat(), getInt(), getLong(), getNumeric(), getObject(), getShort(), getDate(), getTime() y getUnicodeStream(), cada uno de los cuales devuelve la columna en el formato correspondiente, si es posible.

Después de haber trabajado con una sentencia o una conexión es recomendable cerrarla mediante sentencia.close() o conexión.close().

```
try
    {
        if (sentencia!=null)
            sentencia.close();
        if (resultado!=null)
            resultado.close();
        if(conexion!=null)
            conexion.close();
    }
catch(SQLException sqle )
    {
        this.strErrorDriver = sqle.getMessage().toString();
    }
}
```

De forma predeterminada los drivers JDBC deben hacer un COMMIT de cada sentencia. Este comportamiento se puede modificar mediante el método `Connection.setAutoCommit(boolean nuevovalor)`. En el caso de que se establezca `AutoCommit` a `false`, será necesario llamar de forma explícita a `Connection.commit()` para guardar los cambios realizados o `Connection.rollback()` para deshacerlos.

El objeto `ResultSet` devuelto por el método `executeQuery()`, permite recorrer las filas obtenidas, no proporciona información referente a la estructura de cada una de ellas; para ello se utiliza `ResultSetMetaData`, que permite obtener el tipo de cada campo o columna, su nombre, si es del tipo autoincremento, si es sensible a mayúsculas, si se puede escribir en dicha columna, si admite valores nulos, etc.

Para obtener un objeto de tipo `ResultSetMetaData` basta con llamar al método `getMetaData()` del objeto `ResultSet`.

En la lista siguiente aparecen algunos de los métodos más importantes de `ResultSetMetaData`, que permiten averiguar toda la información necesaria para formatear la información correspondiente a una columna, etc.

`getCatalogName()`

Nombre de la columna en el catálogo de la base de datos

`getColumnName()`

Nombre de la columna

`getColumnLabel()`

Nombre a utilizar a la hora de imprimir el nombre de la columna

`getColumnDisplaySize()`

Ancho máximo en caracteres necesario para mostrar el contenido de la columna

`getColumnCount()`

Número de columnas en el ResultSet

`getTableName()`

Nombre de la tabla a que pertenece la columna

`getPrecision()`

Número de dígitos de la columna

`getScale()`

Número de decimales para la columna

`getColumnType()`

Tipo de la columna (uno de los tipos SQL en `java.sql.Types`)

`getColumnTypeName()`

Nombre del tipo de la columna

`isSigned()`

Para números, indica si la columna corresponde a un número con signo

`isAutoIncrement()`

Indica si la columna es de tipo autoincremento

`isCurrency()`

Indica si la columna contiene un valor monetario

`isCaseSensitive()`

Indica si la columna contiene un texto sensible a mayúsculas

`isNullable()`

Indica si la columna puede contener un NULL SQL. Puede devolver los valores `columnNoNulls`, `columnNullable` o `columnNullableUnknown`, miembros finales estáticos de `ResultSetMetaData` (constantes)

`isReadOnly()`

Indica si la columna es de solo lectura

`isWritable()`

Indica si la columna puede modificarse, aunque no lo garantiza

`isDefinitelyWritable()`

Indica si es absolutamente seguro que la columna se puede modificar

isSearchable()

Indica si es posible utilizar la columna para determinar los criterios de búsqueda de un SELECT

getSchemaName()

Devuelve el texto correspondiente al esquema de la base de datos para esa columna

En general pues, los objetos que se van a poder encontrar en una aplicación que utilice JDBC, serán los que se indican a continuación.

Connection

Representa la conexión con la base de datos. Es el objeto que permite realizar las consultas SQL y obtener los resultados de dichas consultas. Es el objeto base para la creación de los objetos de acceso a la base de datos.

DriverManager

Encargado de mantener los drivers que están disponibles en una aplicación concreta. Es el objeto que mantiene las funciones de administración de las operaciones que se realizan con la base de datos.

Statement

Se utiliza para enviar las sentencias SQL simples, aquellas que no necesitan parámetros, a la base de datos.

PreparedStatement

Tiene una relación de herencia con el objeto Statement, añadiéndole la funcionalidad de poder utilizar parámetros de entrada. Además, tiene la particularidad de que la pregunta ya ha sido compilada antes de ser realizada, por lo que se denomina preparada. La principal ventaja, aparte de la utilización de parámetros, es la rapidez de ejecución de la pregunta.

CallableStatement

Tiene una relación de herencia con el objeto PreparedStatement. Permite utilizar funciones implementadas directamente sobre el sistema de gestión de la base de datos. Teniendo en cuenta que éste posee información adicional sobre el uso de las estructuras internas, índices, etc.; las funciones se realizarán de forma más eficiente. Este tipo de operaciones es muy utilizado en el caso de ser funciones muy complicadas o bien que vayan a ser ejecutadas varias veces a lo largo del tiempo de vida de la aplicación.

ResultSet

Contiene la tabla resultado de la pregunta SQL que se haya realizado. En párrafos anteriores se han comentado los métodos que proporciona este objeto para recorrer dicha tabla.

CAPÍTULO III

MARCO APLICATIVO

3.1 CASO DE ESTUDIO

La evaluación del desempeño del ejercicio docente politécnico, es la manera de valorar cuali-cuantitativamente al docente para lograr su mejoramiento y desarrollo personal y profesional. Este proceso cobra mucha importancia en el desarrollo de los procesos de formación, puesto que si se tiene docentes capaces de crear espacios adecuados en la construcción del conocimiento se garantizarán profesionales idóneos que aportarán con responsabilidad y ética al crecimiento de la sociedad.

En lo concerniente a la Evaluación del desempeño del ejercicio docente, la ESPOCH ejecuta esta disposición constitucional éstos procesos desde el año 2001. Sin embargo,

los preceptos filosóficos de calidad demandan, que pese a los esfuerzos realizados hasta ahora, es urgente enriquecer los criterios teóricos y técnicos para que la evaluación, como formas específicas de conocer y relacionarse con la realidad docente, catalice cambios optimizadores.

3.1.1 Aspectos para la evaluación de un docente

Le evaluación docente tiene como objeto la totalidad del trabajo académico de los profesores y sus resultados. Se tomarán en cuenta los siguientes ámbitos²⁰:

- a) Las actividades de docencia: trabajo en el aula, desarrollo de prácticas, programación macro y micro curricular, evaluación de logros del aprendizaje, preparación y utilización de materiales didácticos, dirección de tesis y trabajos de titulación y tutorías.
- b) Las actividades de investigación formativa y generativa: programación, ejecución y evaluación de investigaciones en los campos científicos, tecnológicos, artísticos y humanísticos.
- c) Las actividades de administración y gestión: dirección de unidades académicas, administración institucional, en lo que tiene relación con los procesos académicos.

²⁰ Propuesta de Evaluación Docente. Comisión de Evaluación Institucional.

- d) Las actividades de vinculación con la colectividad: programas de extensión cultural o técnica, servicios al sector externo, cooperación con programas de desarrollo de la comunidad, cooperación interuniversitaria y otros semejantes.
- e) El compromiso institucional: participación en los procesos de cohesión y desarrollo de la comunidad académica y de la institución; y,
- f) Perfil académico y competencias ético-profesionales: tiempo de ejercicio de la cátedra, títulos académicos de pregrado y posgrado, categoría escalafonaria, experiencia profesional, actualización de conocimientos, reconocimientos institucionales, práctica de valores éticos.

3.1.2 Grupos definidos para la evaluación

Para una mejor valoración de los aspectos tomados en cuenta para la evaluación de un docente en un proceso evaluativo se crearon grupos que contendrán los distintos ítems mencionados anteriormente.

Grupo1: Pedagogía y el conocimiento

Refiriéndonos a pedagogía y el conocimiento a la forma como el docente transmite experiencias, conocimientos, valores, con los recursos que tenemos a nuestro alcance, como son: experiencia, materiales, la misma naturaleza, los laboratorios, los avances tecnológicos, la escuela, el arte, el lenguaje hablado, escrito y corporal a los estudiantes.

Grupo 2: Evaluación

Al impartir sus conocimientos a un grupo de estudiantes el docente debe utilizar cualquier método sistemático para recabar información sobre el impacto y efectividad de su acción formativa.

Grupo 3: Relaciones humanas y atención al estudiante

El hombre es un ser social, por ende comparte y se relaciona con los demás, es por ello que es de gran importancia conocer al grado de aceptación por parte de los estudiantes y la atención que les ofrece según el reglamento académico.

Grupo 4: Desempeño

Con el fin de determinar el cumplimiento del conjunto de normas que establecen las funciones que deben desempeñar los docentes para dar cumplimiento a su importante labor como orientadores de las generaciones futuras la evaluación del desempeño se considera como un aspecto muy importante a ser tomado en cuenta.

Grupo 5: Otras

Otros aspectos que pueden ser tomados en cuenta son:

- Las actividades extracurriculares (asesorías de proyectos de tesis, comisiones encargadas, investigaciones, extensión, etc.).
- Conocer el Estatuto Politécnico, normativo de carga académica y demás normativo institucionales.
- Conocer las líneas y tipos de investigación institucional y/o de la Facultad.

- Participación en el diseño macrocurricular de la carrera donde realiza su trabajo académico.

3.1.3 Cuestionarios definidos para la evaluación

Cabe destacar que cada grupo corresponde a un cuestionario²¹, por lo que actualmente se cuenta con los siguientes cuestionarios:

Cuestionario general docente – autoevaluación

Este cuestionario está dirigido a los docentes con la finalidad de superar los obstáculos existentes y considerar los logros alcanzados, por ellos mismos.

En este cuestionario están considerados los grupos de: Pedagogía y conocimiento, Evaluación, Relaciones humanas y atención al estudiante y Otras

Cuestionario general estudiante – docente

Es el cuestionario dirigido a los estudiantes donde se evalúa a los docentes tomando en cuenta los grupos: Pedagogía y conocimientos, Evaluación, Relaciones humanas y atención al estudiante

Cuestionario vinculación estudiante – docente

Donde se toma en consideración grupos como: Pedagogía y conocimientos, Desempeño, Evaluación y Relaciones humanas y atención al estudiante

²¹ Sistema de Evaluación Docente. Comisión de evaluación Institucional

3.1.4 Informes De Evaluación

Al realizar la evaluación en la institución es necesario dar a conocer los resultados de la misma, mediante un informe se dará a conocer los resultados obtenidos, es por ello que la comisión de evaluación ha desarrollado la siguiente tabla donde se valora a cada uno de los docentes según el rango de valores obtenidos.

El análisis de la información se hará en base a la clasificación de la siguiente tabla:

**TABLA III.4 TABLA DE VALORACION DE RESULTADOS DE LA
EVALUACIÓN DOCENTE**

EVALUACIÓN CUALITATIVA	EVALUACIÓN CUANTITATIVA PRIMARIA %	RESULTADOS
A. Excelente desempeño docente	76-100	FORTALEZA: Excelente Docente
B. Buen desempeño docente	51-75	FORTALEZA: Buen Docente
C. Regular desempeño docente	26-50	DEBILIDAD: Docente con limitaciones
D. Insuficiente desempeño docente	0-25	DEBILIDAD: Docente con muchas limitaciones

Fuente: Departamento de Comisión de Evaluación ESPOCH

Elaborado Por: Comisión de Evaluación institucional

Así como a los docentes se les informa de sus resultados obtenidos es necesario conocer también resultados a nivel institucional, es por ello que también se tomará en cuenta una escala de valores donde se evalúa la calidad institucional.

**TABLA III.5 TABLA DE VALORACIÓN DE RESULTADOS DE LA
EVALUACIÓN INSTITUCIONAL**

EVALUACIÓN CUALITATIVA	EVALUACIÓN CUANTITATIVA PRIMARIA %	RESULTADOS
A. Muy Buena. Objetivo logrado (solución o resultado excelente que puede servir como modelo)	76-100	FORTALEZA
B. Buena. Avance significativo existe preocupación y mejoras sustanciales faltando aprovechar todo el potencial	51-75	FORTALEZA
C. Regular. Cierto Avance (Logros parciales que dan lugar a ciertas mejoras con resultados aislados)	26-50	DEBILIDAD
D. Insuficiente. Ningún avance. (Ninguna acción, quizá ciertas ideas buenas pero no concretadas)	0-25	DEBILIDAD

Fuente: Serie de documentos técnicos, Guía de autoevaluación con fines de acreditación

Elaborado Por: CONEA (Consejo Nacional de Evaluación y Acreditación de la educación Superior del Ecuador)

3.2 DISEÑO Y APLICACIÓN DEL SISTEMA DE LÓGICA DIFUSA

3.2.1 Análisis

3.2.1.1 Definición de Objetivos, Restricciones y Comportamiento del Sistema

Objetivos:

- Proporcionar información al docente acerca de los resultados obtenidos en el proceso de evaluación en lenguaje natural y con resultados difusos en el módulo de estadísticas del sistema de evaluación de la ESPOCH.

- Proporcionar información acerca de la evaluación a nivel institucional en lenguaje natural y con sus respectivos resultados difusos en el módulo de estadísticas del sistema de la ESPOCH.
- Obtener los resultados de la forma antes mencionada en base a cada cuestionario en el que cada docente ha sido evaluado.
- Determinar el grado de pertenencia que tiene el promedio del docente a su respectivo conjunto difuso.

Restricciones

- El sistema difuso no mostrará recomendaciones a un docente.
- El sistema difuso no mostrará valores probabilísticos de las evaluaciones realizadas, sino valores de pertenencia de las variables de entrada que en este caso son los grupos y los valores inferidos.

Comportamiento del sistema

El sistema tomará el valor de los grupos de cada cuestionario y los convertirá en valores difusos utilizando la función trapezoidal, luego se obtendrá un valor del cuestionario mismo que se determinará a través de la inferencia para luego por último tomar los valores de los cuestionarios existentes y definir una salida total difusa. Todo el proceso anterior se lo realizará de forma general.

Hay que mencionar que se mostrará también valores difusos por cada cuestionario indicando el grado de pertenencia de una variable lingüística a cualquiera de los conjuntos difusos.

A nivel institucional se presentará los resultados por facultad, carrera y escuela.

3.2.1.2 Identificar las variables de Entrada y Salida

VARIABLES LINGÜÍSTICAS Y SUS ETIQUETAS LINGÜÍSTICAS.

En el caso de estudio se definen los parámetros en el que cada docente será evaluado, es así como determinaremos las Variables lingüísticas que se utilizarán para realizar el sistema de lógica difusa.

Mediante el diseño y la aplicación del sistema de lógica difusa se podrá determinar el promedio que un docente ha obtenido según los resultados obtenidos en el proceso de evaluación en lenguaje natural.

Cabe resaltar que estos resultados se obtendrán por cada cuestionario aplicado a cada docente, al momento se cuenta con el cuestionario de autoevaluación docente, cuestionario de evaluación del estudiante al docente y el cuestionario de evaluación del directivo al docente.

VARIABLES DE ENTRADA

Como variable de entrada se han tomado en cuenta los grupos que se mencionan en el caso de estudio,

- Pedagogía y conocimiento
- Evaluación
- Desempeño

- Relaciones humanas y atención al estudiante
- Otras

Debido a que el resultado se mostrará por cada cuestionario se define a continuación los aspectos que se toma en cuenta en cada cuestionario como variable de entrada:

Cuestionario General Docente – Autoevaluación

- Pedagogía Y Conocimiento
- Evaluación
- Relaciones Humanas Y Atención Al Estudiante
- Otras

Cuestionario General Estudiante – Docente

- Pedagogía Y Conocimientos
- Evaluación
- Relaciones Humanas Y Atención Al Estudiante

Cuestionario Vinculación Estudiante - Docente

- Pedagogía y conocimientos
- Desempeño
- Evaluación
- Relaciones Humanas Y Atención Al Estudiante

VARIABLES DE SALIDA

La primera variable de salida será el promedio identificado por una variable lingüística obtenido en cada grupo en el que el docente ha sido evaluado, la denominaremos **Promedio de Grupo** identificándolo con su respectivo nombre.

Otra variable de salida será el promedio identificado por una variable lingüística que cada docente obtenga en cada cuestionario. La denominaremos **Promedio de Cuestionario**

Al final se obtendrá el **promedio total**.

A nivel institucional también se obtendrá una variable de salida denominada **Promedio por Facultad**.

3.2.1.3 Definir procedimientos y criterios para la prueba y validación.

Para el proceso de validación se realizará una pequeña prueba del sistema en cada una de las facultades de la esoch a una muestra de estudiantes de los siguientes cuestionarios:

- Autoevaluación docente
- La evaluación de Estudiantes a los docentes

Donde se obtendrán los datos necesarios para realizar el proceso de difuzzificación.

Se comprobará los resultados difusos con los resultados de los promedios obtenidos con el proceso normal.

3.2.2 Diseño

3.2.2.1 Definir los conjuntos difusos de cada etiqueta lingüística.

Comenzaremos definiendo los conjuntos difusos de las variables de entrada que en nuestro caso son los grupos en los que cada docente es evaluado.

Es necesario mencionar que la función de pertenencia que mejor se ha acoplado a la definición de las variables es la **función de tipo trapezoidal**.

Entonces iniciaremos definiendo la variable denominada Pedagogía y conocimiento.

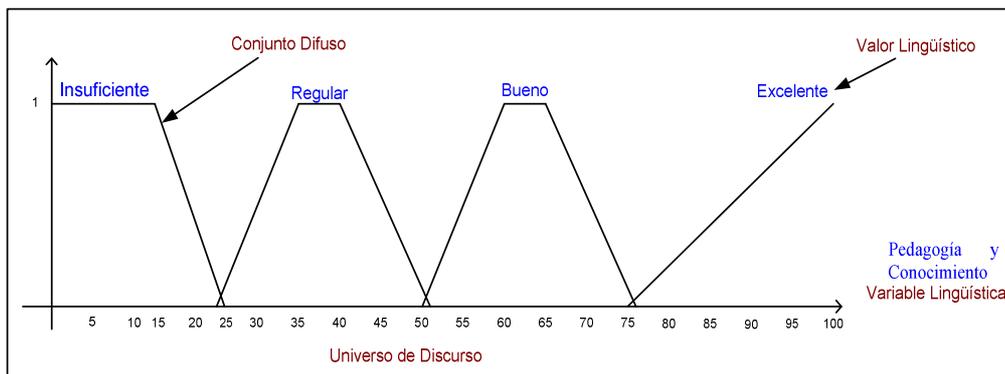


Figura. III.1 Variable lingüística pedagogía y conocimiento

Al mostrar los conjuntos difusos de la variable de pedagogía y conocimiento podemos darnos cuenta que los valores del universo de discurso están comprendidos entre 0 y 100.

Cabe mencionar que al realizar la evaluación docente todas las preguntas tendrán una valoración entre 0 y 100 por lo que los grupos que se tomarán en cuenta para el tratamiento difuso también estarán en el rango de 0 a 100; es decir todas las variables

tendrán el mismo universo de discurso así como también las mismas etiquetas lingüísticas, como se muestra en los gráfico que se muestran a continuación:

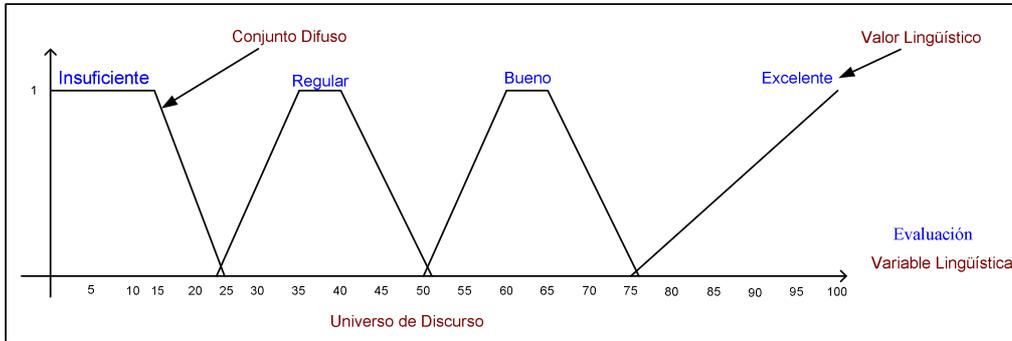


Figura. III.2 Variable lingüística Evaluación

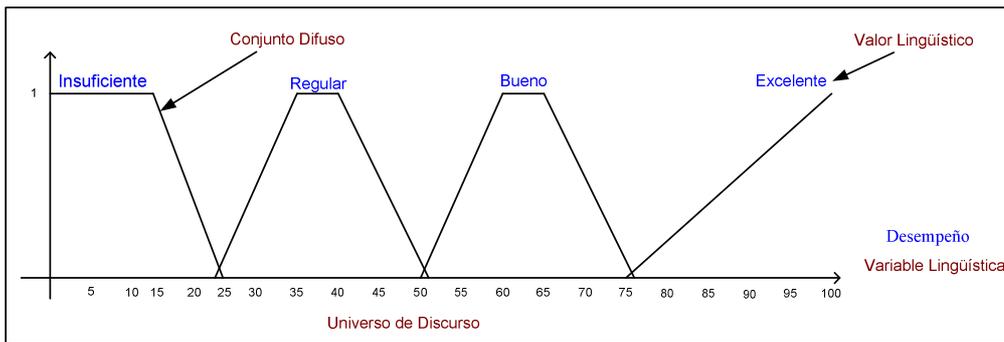


Figura. III.3 Variable lingüística Desempeño

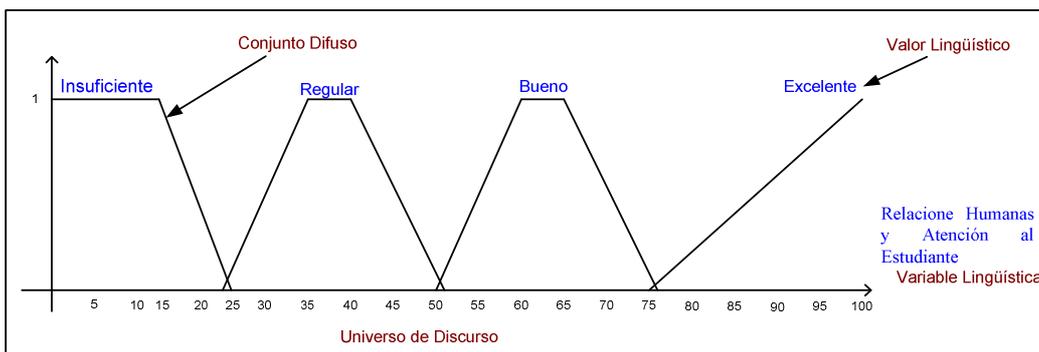
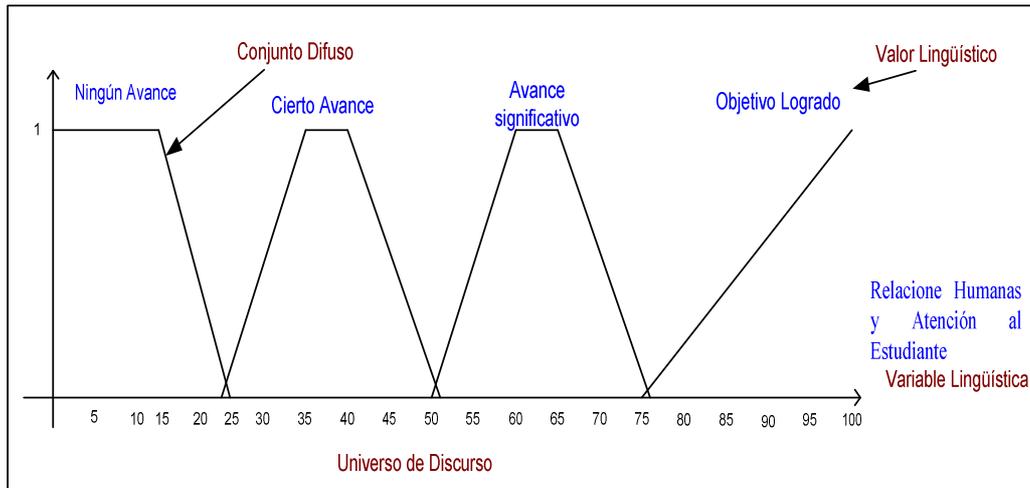


Figura. III.4 Variable lingüística Relaciones Humanas y Atención al Estudiante.

Ahora definiremos los conjuntos difusos para la variable de salida **Promedio Total**.



Figura

. III.5 Variable Lingüística Promedio Total

Para el caso de los resultados institucionales se tomará en cuenta la variable de salida **Promedio Total** con las siguientes etiquetas lingüísticas:

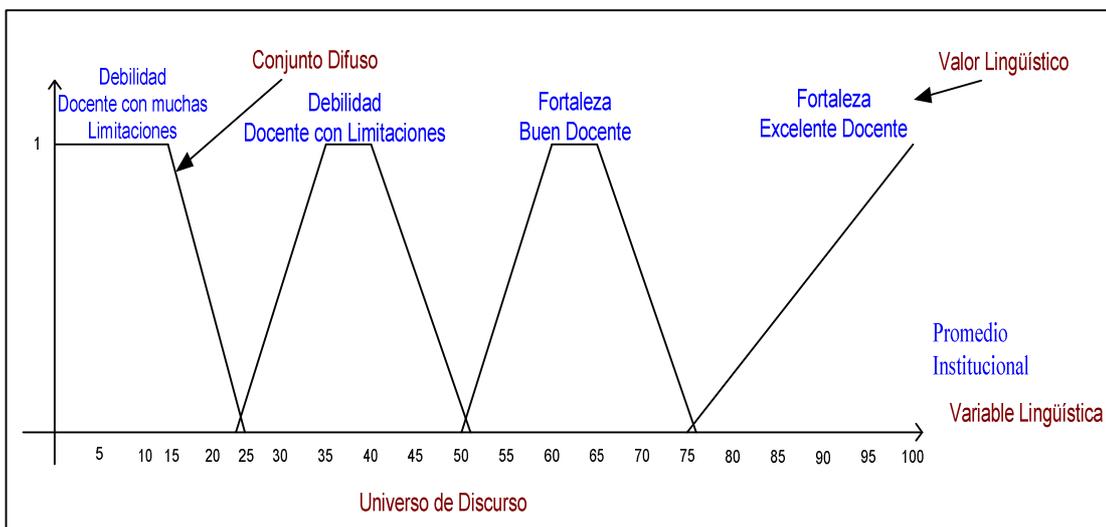


Figura. III.6 Variable Lingüística promedio Institucional

3.2.2.2 Definir el marco de representación de la solución

Para definir el conocimiento que se proporcionará al sistema se toma en cuenta las variables de entrada, para representar de mejor forma las variables se presenta el siguiente esquema:

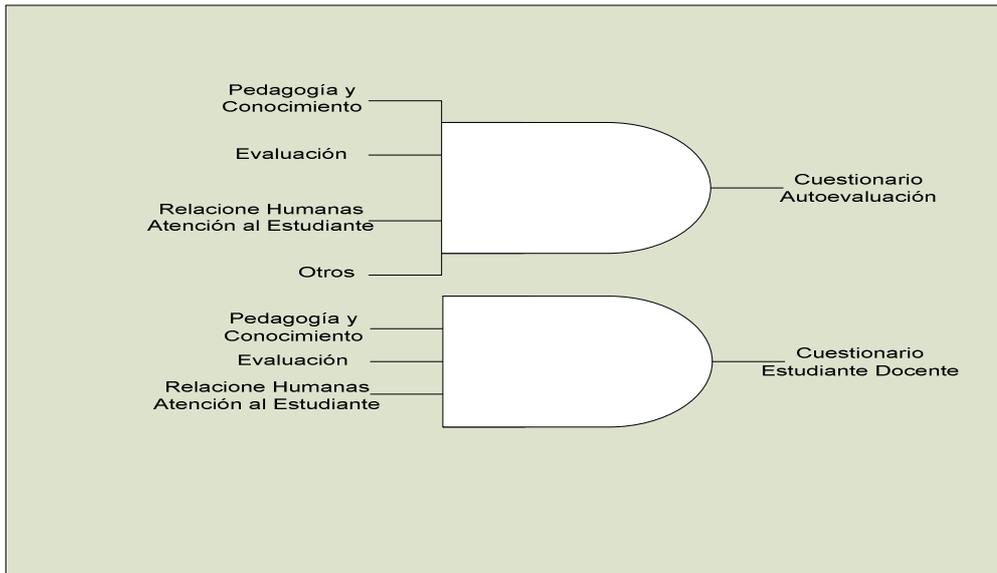


Figura. III.7 Representación del conocimiento con las variables de entrada.

El valor que el docente obtenga en el grupo de pedagogía, conocimiento y el valor que se obtenga en el grupo de evaluación y el valor que se obtenga en el grupo de Relaciones Humanas y Atención al Estudiante y el valor que se obtenga en el grupo Otros determinará el valor del cuestionario de Autoevaluación del docente.

El valor que se obtenga en pedagogía y conocimiento, el valor que se obtenga en el grupo de evaluación y el valor que se obtenga en el grupo relaciones humanas y atención al estudiante determinarán el valor del cuestionario estudiante docente.

Se identifica que existe el mismo nombre de variables de entrada para distintos cuestionarios pero es importante recalcar que los informantes para los cada uno de ellos son distintos, por lo que el valor de las variables es diferente.

Al momento de tener el valor de los distintos cuestionarios se obtendrá el valor del promedio total, y esto se representa en la siguiente figura.

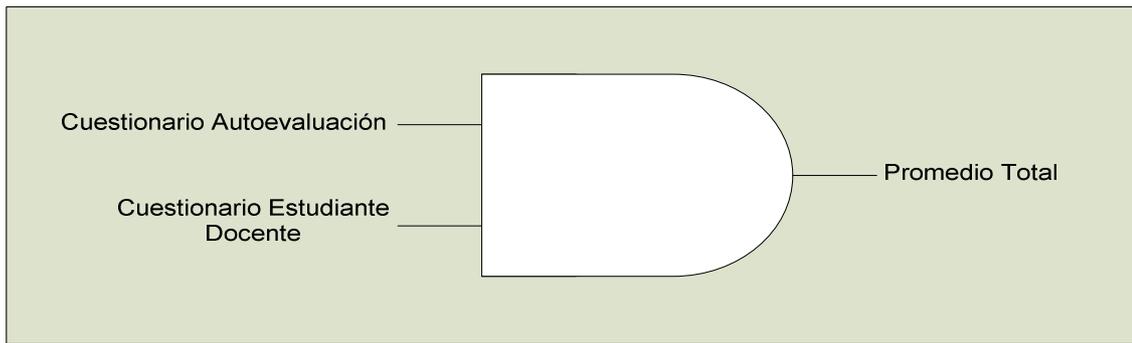


Figura.III.8 Representación de la variable de Salida

Se recalca además que por recomendación de la comisión de evaluación institucional las variables de salida seguirán las siguientes reglas:

R1: Si el valor esta en el rango 0- 25 entonces se tiene un promedio Insuficiente.

R2: Si el valor esta en el rango 26- 50 entonces se tiene un promedio Regular.

R3: Si el valor esta en el rango 51- 75 entonces se tiene un promedio Bueno.

R4: Si el valor esta en el rango 76- 100 entonces se tiene un promedio Excelente.

Siguiendo el mismo esquema para la obtención de resultados a nivel institucional se tendrá:

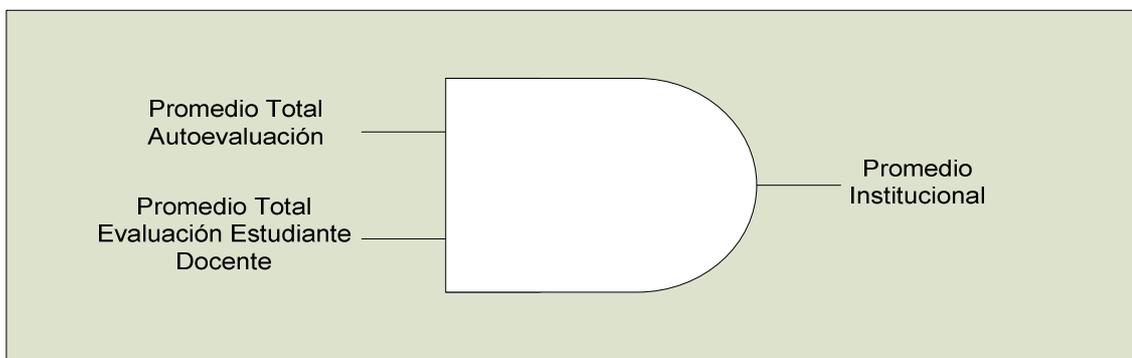


Figura.III.9 Representación Variable de Salida Institucional

3.2.2.3 Definir el marco de los procedimientos empleados (Inferencia).

Cabe destacar que para realizar el proceso de inferencia de el conocimiento que el sistema adquiera se tomará en cuenta la implicación borrosa por la regla del mínimo o implicación de Madani.

Además se utilizará las operaciones entre conjuntos borrosos como son la unión y la intersección con sus distintas fórmulas.

3.2.2.4 Especificar la forma requerida de la salida del sistema (decodificación).

Hay que tomar en cuenta que en este caso la decodificación o defusificación no será necesaria debido a que el resultado difuso será en gran manera lo que proporcione mayor información a los usuarios.

Pero en caso de ser obligatoria, para realizar la decodificación se tomará en cuenta el método de media de centros pues actúa tanto en las variables difusas así como también en las variables de tipo normal es decir, valores dentro del universo de discurso.

3.2.2.5 Ejecutar pruebas para validar el sistema.

Para realizar las pruebas del sistema se ha de realizar un periodo de evaluación a los docentes donde se podrán obtener datos con los cuales se realizarán las distintas pruebas de validación.

3.2.2.6 Verificar si la solución es compatible con el paso 3.2.1.1:

Al realizar el periodo de evaluación a los docentes se obtuvo los datos necesarios para con los que pudimos comprobar que el sistema funciona de acuerdo con los resultados requeridos por la comisión de evaluación institucional.

Cabe destacar que los datos obtenidos en las operaciones realizadas por la lógica difusa concuerdan en gran medida con los datos que se obtienen de forma normal, tanto en la evaluación docente como en la evaluación institucional, cumpliendo así con los objetivos requeridos por el sistema.

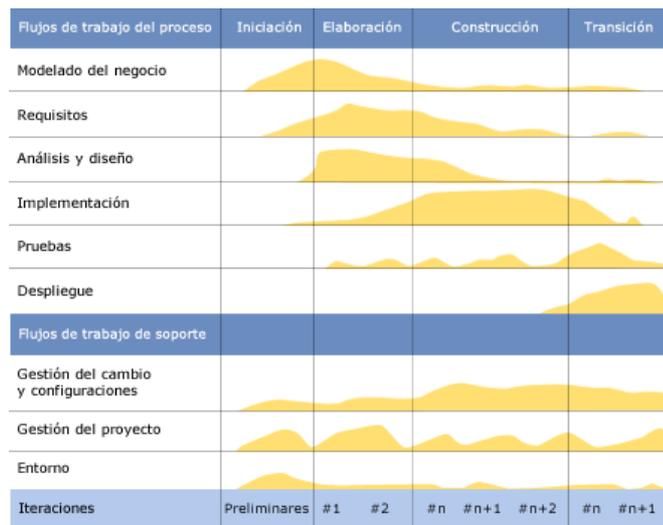
CAPÍTULO IV

DESARROLLO DEL MÓDULO DE ESTADÍSTICAS DEL SISTEMA DE EVALUACION DOCENTE DE LA ESPOCH

Para realizar la ingeniería de la aplicación se tomo en cuenta el **Proceso Unificado Racional** (*Rational Unified Process* en inglés, habitualmente resumido como **RUP**) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El ciclo de vida RUP es una implementación del [Desarrollo en espiral](#). Fue creado ensamblando los elementos en secuencias semi-ordenadas. El ciclo de vida organiza las tareas en fases e iteraciones.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. En la Figura muestra cómo varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto RUP.



Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una baseline (Linea Base) de la arquitectura.

Durante la fase de inicio las iteraciones hacen mayor énfasis en actividades de modelado del negocio y de requerimientos.

En la fase de elaboración, las iteraciones se orientan al desarrollo de la baseline de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la baseline de la arquitectura.

En la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones.

Para cada iteración se selecciona algunos Casos de Uso, se refina su análisis y diseño y se procede a su implementación y pruebas. Se realiza una pequeña cascada para cada ciclo. Se realizan tantas iteraciones hasta que se termine la implementación de la nueva versión del producto.

En la fase de transición se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios.

En cada fase participan todas las disciplinas, pero que dependiendo de la fase el esfuerzo dedicado a una disciplina varía.

INICIACIÓN

4.1 INGENIERÍA DE LA INFORMACIÓN

4.1.1 Definición del Ámbito del problema

La Escuela Superior Politécnica de Chimborazo “ESPOCH”, tiene su origen en el Instituto Técnico Superior de Chimborazo que fue creado mediante Ley # 6090, expedida por el Congreso Nacional, el 18 de abril de 1969.

Inicia sus actividades académicas el 2 de mayo de 1972 en las Escuelas de:

- Ingeniería Zootécnica
- Nutrición y Dietética
- Ingeniería Mecánica, se inaugura el 3 de abril de 1972

El 28 de septiembre de 1973 se anexa la Escuela de Ciencias Agrícolas de la PUCE, contando con todos los recursos académicos, adoptando la designación de Escuela de Ingeniería Agronómica.²²

Objetivo General

²² http://www.espoch.edu.ec/index.php?action=i_general

- Impartir enseñanza a nivel superior, de postgrado y continua en ciencias y tecnologías basadas en la investigación y producción de bienes y servicios.

Objetivos Específicos

- Coligar la formación científica y tecnológica con la configuración de un profesional humanista y solidario.
- Organizar la profesionalización de los estudiantes en relación a las necesidades científicas – técnicas del país, e integrarlos como factores de desarrollo.
- Participar en los sectores de la economía, interactuando en actividades de fomento, extensión educativa y tecnológica.
- Planificar y realizar la investigación científica.

La Docencia es una de las funciones trascendentales de la Universidad Ecuatoriana que se materializa en los procesos de formación científico-técnica y humanística de los profesionales que contribuyen al desarrollo, en su sentido más amplio, del país.

El interés de la Escuela Superior Politécnica de Chimborazo se sustenta en la adquisición apropiada, la actualización ágil y el uso oportuno de los conocimientos, de nuestros docente y para asegurar el buen nivel de los mismos la ESPOCH de conformidad con la Constitución Política del Ecuador, la Ley de Educación Superior, su Estatuto, y ante todo por propia voluntad y decisión política de las autoridades, el 9 de Febrero de 2001, conforma la Comisión de Evaluación Interna cuya misión es evaluar las funciones: Docencia, Investigación, Vinculación con la colectividad y Gestión Administrativa, a fin de asegurar niveles de calidad en todos los procesos y con el

objetivo de planificar y coordinar, así como asesorar al H. Consejo Politécnico sobre los procesos de Autoevaluación institucional, Evaluación Externa y Acreditación en concordancia con el mandato de la Ley de Educación Superior y las reglamentaciones que a nivel del Sistema Universitario Nacional se establezcan, así como en función de la reglamentación interna que la ESPOCH determine

Por esta razón se ha venido realizado el proceso de evaluación docente en línea, que permite valorarlos en distintas áreas consultando a quienes mejor los conocen, los estudiantes; y poniendo en consideración la autovaloración de sus actividades como docente.²³

En la actualidad la aplicación para la evaluación consta de accesos para personal administrativo, quienes no hacen uso del mismo, además al realizar el proceso para la obtención de resultados tarda varios días convirtiéndose en un sistema poco ágil. Además este no consta con información que permita entender de forma clara, no solo numérica, los resultados obtenidos.

4.2. DEFINICIÓN TÉCNICA DE REQUERIMIENTOS (SRS)

Para la definición de requerimientos nos basaremos en el estándar SRS definido por la IEEE. A continuación especificaremos cada uno de los requerimientos para el desarrollo del módulo de Estadísticas de la ESPOCH.

²³ <http://www.esPOCH.edu.ec/index.php?action=rectorado&idr=12#>

4.2.1 INTRODUCCION

El SRS (Especificación de requerimientos Software) es la base para realizar la validación del sistema, además de que ayuda a entender a los desarrolladores lo que el cliente desea. La Especificación de Requerimientos de Software es un acuerdo entre el cliente y los desarrolladores que es legalizado por cada una de las partes antes mencionadas. El SRS es aplicable a cualquier producto de software es por ello que se aplicara al desarrollo del módulo de estadísticas de la ESPOCH.

4.2.1.1 Objetivo

El SRS nos permitirá entender de manera específica los principales requerimientos que tiene el usuario para poder desarrollar un sistema de forma eficiente y amigable.

4.2.1.2 Ámbito

➤ **Identificación del producto software:**

Para referirnos al producto software a desarrollar lo haremos mediante el nombre de: **Módulo de Estadísticas de la ESPOCH.**

➤ **Objetivos del sistema.**

Objetivo general

Informar a la comunidad politécnica acerca de los resultados obtenidos en el proceso de evaluación realizado en la ESPOCH.

Objetivos específicos

- Proveer a los profesores la información sobre los resultados obtenidos en el proceso de evaluación.
- Proveer a la ESPOCH de instrumentos para la toma de decisiones relacionadas con la planificación de actividades de su personal académico.
- Dar a conocer a la comunidad en general los resultados obtenidos en el proceso de valuación realizado en la ESPOCH a nivel institucional con un enfoque general.

➤ **Beneficios del Sistema**

Beneficios Tangibles

- El docente puede recibir información exacta sobre la evaluación realizada.
- Reducción de costos para la institución en cuanto a la adquisición de licencias.

Beneficios Intangibles

- Mediante la información recibida el docente puede mejorar la calidad del trabajo y desempeño académico.
- Con el sistema el docente puede considerar o no estar en permanente actualización de conocimientos y acceder a la capacitación pedagógica.
- Además se puede promover la cultura de evaluación en un ambiente de interacción entre actores que ejercen la libertad académica con responsabilidad.
- La institución puede ser evaluada de acuerdo a las calificaciones obtenidas en el proceso de evaluación.

4.2.1.3 Definiciones, Acrónimos, Abreviaturas

TABLA IV.6 DEFINICIONES

Docente	Persona encargada de conducir la actividad lectiva, de investigación, de extensión o de proyección en la Universidad.
Estudiante	Persona que está formalmente matriculada en un programa de estudios. Hay distintos tipos de estudiantes, en función del modelo de enseñanza, de su dedicación temporal, del plan de estudios en el que se matricula o inscribe. ²⁴
Autoevaluación	Es el riguroso y sistemático examen que una Institución realiza, con amplia participación de sus integrantes a través de un análisis crítico y un diálogo reflexivo sobre la totalidad de las actividades institucionales o de un programa específico, a fin de superar los obstáculos existentes y considerar los logros alcanzados, para mejorar la eficiencia institucional y alcanzar la excelencia académica.
Promedio Total	Es el resultado final de algún proceso u operación.
Usuario General	Persona que consume los servicios del módulo de Estadísticas del sistema SIEDD sin autenticarse.

TABLA IV.7 ACRÓNIMOS

ESPOCH	Escuela Superior Politécnica de Chimborazo.
SRS	Especificación de Requisitos de Software
SIEDD	Sistema Integrado de Evaluación del desempeño docente

²⁴ Manual de Evaluación externa institucional para la acreditación de universidades y escuelas politécnicas

4.2.1.4 Referencias.

IEEE Recommended Practice for Software Requirements Specification. ANSI/IEEE std. 830, 1998.

4.2.1.5 Visión General

El SRS está organizado de la siguiente manera:

4.2.1 INTRODUCCION

4.2.1.1 Objetivos

4.2.1.2 Ámbito

4.2.1.3 Definiciones, Acrónimos, Abreviaturas

4.2.1.4 Visión General

4.2.2 DESCRIPCION GENERAL

4.2.2.1 Perspectiva del Producto

4.2.2.2 Funciones del Producto

4.2.2.3 Características de los usuarios

4.2.2.4 Limitaciones Generales

4.2.2.5 Supuestos y dependencias

4.2.3 REQUERIMIENTOS ESPECIFICOS

4.2.3.1 Requisitos Funcionales

- Introducción

- Entradas
- Proceso
- Salidas

4.2.3.2 Requisitos de Comportamiento

4.2.3.3 Limitaciones de Diseño

4.2.3.4 Atributos

4.2.3.5 Requisitos de Interfaces Externas

4.2.3.6 Otros Requerimientos.

4.2.2 DESCRIPCION GENERAL

4.2.2.1 Perspectiva del Producto

- El sistema de evaluación que se ha utilizado en la época para realizar el proceso de evaluación arroja solo datos numéricos, además este sistema se encuentra implementado en una plataforma que requiere la compra de licencias para su funcionamiento, es importante recalcar además que el sistema contiene accesos para personal administrativo que no han sido utilizados, por lo que el Módulo de Estadísticas contendrá solo los accesos necesarios para visualizar la información de resultados tanto a nivel de docente como a nivel institucional, es necesario mencionar también que el módulo se desarrollará en una plataforma libre como lo es Java lo que permitirá reducir costos en cuanto a la adquisición de licencias.
- El módulo a desarrollar es parte del sistema denominado SIEDD (Sistema Integrado de Evaluación al Desempeño Docente) dedicado únicamente al

procesamiento de los datos obtenidos en el proceso de evaluación realizado en la epoch.

4.2.2.2 Funciones del Producto

El módulo de estadísticas del sistema SIEDD se encuentra directamente relacionado con las estadísticas públicas y las estadísticas privadas que se mostrarán a los usuarios.

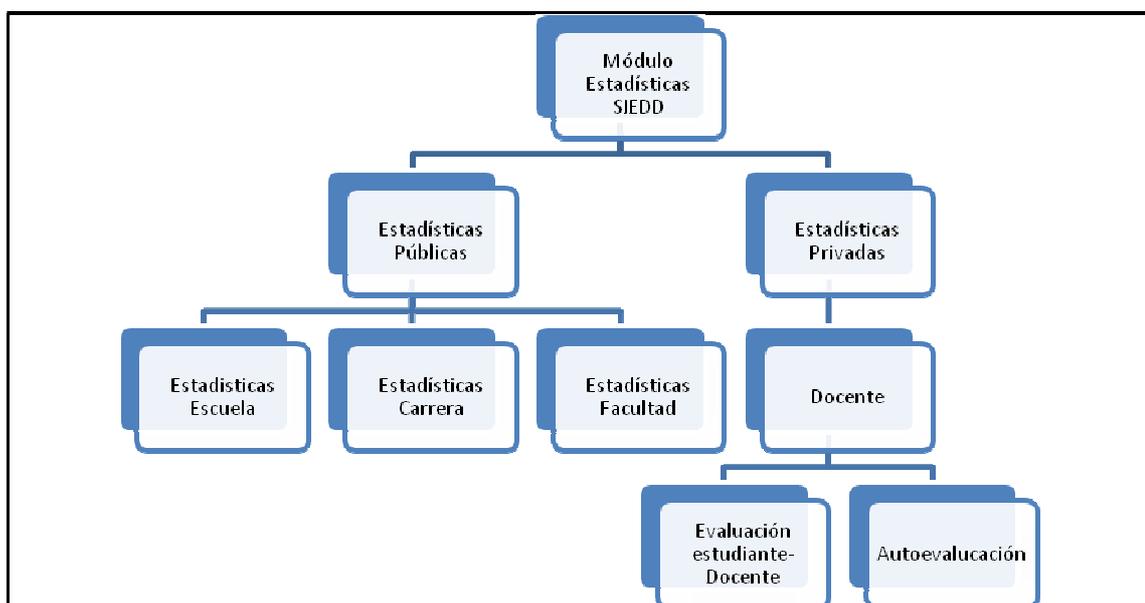


Figura IV.1 Funciones del producto

El módulo de estadísticas del sistema SIEDD constará de 2 componentes, mismos que se describen a continuación:

Estadísticas Públicas

Este componente se encuentra caracterizado por contener resultados a nivel institucional, es decir por Facultades, escuelas y carreras, identificando los valores obtenidos tanto a nivel de evaluación de estudiante como autoevaluación docente.

Estadísticas Privadas

Este componente presentará resultados obtenidos en el proceso de evaluación a nivel de estudiantes, indicando las carreras en las que imparte conocimiento así como también las materias incluyendo el cuestionario que los estudiantes contestaron.

Del mismo modo se mostrará los resultados obtenidos en la autoevaluación docente por cada carrera en la que el docente imparte clases así como el cuestionario que fue evaluado.

4.2.2.3 Características de los usuarios

Se ha identificado 2 usuarios bien definidos que harán uso del módulo de estadísticas del sistema SIEDD, y son:

Docentes Evaluados: quienes visualizarán su propia información mediante el ingreso autenticado al módulo.

Usuario General: Quien tendrá acceso directo a los resultados institucionales sin autenticación.

4.2.2.4 Limitaciones Generales

- Debido a que el proyecto de evaluación docente puede generar nuevos procesos de negocio es de esperar que futuros cambios se efectúen en el sistema general SIEDD del cual es parte el módulo de estadísticas provocando un impacto en el normal desenvolvimiento del módulo.

- Se deberá tomar en cuenta que se utilizará JSP con la herramienta Javabeans para el desarrollo de la aplicación por lo que debemos tomar en cuenta el número de usuarios que soporta el servidor Tomcat que es con él trabaja esta herramienta.
- Se deberá realizar una base de datos que permita la interacción de los distintos módulos que componen el sistema SIEDD.

4.2.2.5 Supuestos y dependencias

a. Supuestos

Se asume que los requisitos descritos en este documento son estables una vez que sea aprobado por los directivos de la institución. Cualquier petición de cambios e la especificación debe ser aprobada por todas las partes y gestionada por el grupo de desarrollo y planificación.

b. Dependencias

Debido a que el modulo de estadísticas pertenece al sistema SIEDD dependerá directamente de que en el módulo de administración de dicho sistema realice el proceso de cierre del periodo de evaluaciones, para así poder obtener los datos necesarios para este módulo.

La disponibilidad del sistema dependerá de la conexión a internet que exista, debido a que la misma será publicada en un sitio web dentro de la página de la espoch, y del las condiciones en las que se encuentre la máquina que asuma las funciones de servidor.

4.2.3 REQUERIMIENTOS ESPECIFICOS

4.2.3.1 Requisitos Funcionales

Debido a que el módulo consta de dos componentes, especificaremos los requerimientos según cada componente:

Estadísticas Privadas

R1.

Introducción

El módulo de estadísticas deberá mostrar el promedio total obtenido por un docente en la evaluación realizada por los estudiantes.

Entradas

Fuentes de las entradas:

- Cédula del docente
- Contraseña para autenticación

Rangos de entradas validas y requisitos de control

- La cédula para el ingreso contendrá 11 dígitos pues incluirá el guión.

Proceso

- Obtener la cédula del docente y su contraseña.
- Validar datos obtenidos para acceder al sistema.

- Calcular el promedio total del docente en las evaluaciones realizadas por los estudiantes.

Salidas

- Se muestra en la pantalla el promedio total del docente obtenido en la evaluación por parte de los estudiantes

R2.

Introducción

El módulo de estadísticas deberá mostrar el promedio total obtenido por un docente en la autoevaluación.

Entradas

Fuentes de las entradas:

- Cédula del docente
- Contraseña para autenticación

Rangos de entradas validas y requisitos de control

- La cédula para el ingreso contendrá 11 dígitos pues incluirá el guión.

Proceso

- Obtener la cédula del docente y su contraseña.
- Validar datos obtenidos para acceder al sistema.
- Calcular el promedio total del docente obtenida en la autoevaluación.

Salidas

- Se muestra en la pantalla el promedio total del docente obtenida en la autoevaluación.

R3.

Introducción

El módulo de estadísticas deberá mostrar el promedio total obtenido por un docente en cada carrera en la que este imparta clases.

Entradas

Fuentes de las entradas:

- Cédula del docente
- Contraseña para autenticación

Rangos de entradas validas y requisitos de control

- La cédula para el ingreso contendrá 11 dígitos pues incluirá el guión.

Proceso

- Obtener la cédula del docente y su contraseña.
- Validar datos obtenidos para acceder al sistema.
- Buscar carreras asignadas al docente.
- Calcular promedio total por carrera del docente.

Salidas

- Se muestra en la pantalla el promedio total que el docente obtuvo en la evaluación por los estudiantes desplegada por cada carrera en la que este imparte clases.

R4.

Introducción

El módulo de estadísticas deberá mostrar el promedio total obtenido por un docente en cada materia de una carrera en la que el docente imparta clases.

Entradas

Fuentes de las entradas:

- Cédula del docente
- Contraseña para autenticación
- Código de la asignatura o materia deseada.

Rangos de entradas validas y requisitos de control

- La cédula para el ingreso contendrá 11 dígitos pues incluirá el guión.
- El código de la materia contendrá hasta 10 caracteres incluyendo números y letras.

Proceso

- Obtener la cédula del docente y su contraseña.
- Validar datos obtenidos para acceder al sistema.
- Buscar carreras asignadas al docente.
- Seleccionar carrera.
- Calcular promedio por cada materia asignada a la carrera seleccionada.

Salidas

- Se muestra en la pantalla el promedio total que el docente obtuvo en la evaluación por los estudiantes desplegada por cada materia de una carrera en la que este imparte clases.

R5.

Introducción

El módulo de estadísticas deberá mostrar el promedio total obtenido por un docente en pregunta de una materia.

Entradas

Fuentes de las entradas:

- Cédula del docente
- Contraseña para autenticación
- Código de la carrera.
- Código de la asignatura o materia deseada.

Rangos de entradas validas y requisitos de control

- La cédula para el ingreso contendrá 11 dígitos pues incluirá el guión.
- El código de la carrera y materia contendrá hasta 10 caracteres incluyendo números y letras.

Proceso

- Obtener la cédula del docente y su contraseña.
- Validar datos obtenidos para acceder al sistema.
- Buscar carreras asignadas al docente.
- Seleccionar carrera.
- Seleccionar materia.
- Calcular promedio por pregunta de la materia seleccionada.

Salidas

- Se muestra en la pantalla el promedio total que el docente obtuvo en la evaluación por los estudiantes desplegada por cada pregunta de la materia seleccionada.

R6.

Introducción

El módulo de estadísticas deberá mostrar el número de estudiantes registrados en una materia.

Entradas

Fuentes de las entradas:

- Cédula del docente
- Contraseña para autenticación
- Código de la carrera.
- Código de la asignatura o materia deseada

Rangos de entradas validas y requisitos de control

- La cédula para el ingreso contendrá 11 dígitos pues incluirá el guión.
- El código de la carrera y materia contendrá hasta 10 caracteres incluyendo números y letras.

Proceso

- Obtener la cédula del docente y su contraseña.
- Validar datos obtenidos para acceder al sistema.
- Identificar la materia seleccionada por el docente.
- Buscar el número de estudiantes asignados a la materia seleccionada.

Salidas

- Número de estudiantes asignados a la materia seleccionada por el docente.

R7.

Introducción

El módulo de estadísticas deberá mostrar el número de estudiantes que evaluaron al docente en esa materia.

Entradas

Fuentes de las entradas:

- Cédula del docente
- Contraseña para autenticación
- Código de la carrera.
- Código de la asignatura o materia deseada

Rangos de entradas validas y requisitos de control

- La cédula para el ingreso contendrá 11 dígitos pues incluirá el guión.
- El código de la carrera y la materia contendrá hasta 10 caracteres incluyendo números y letras.

Proceso

- Obtener la cédula del docente y su contraseña.
- Validar datos obtenidos para acceder al sistema.
- Identificar la materia seleccionada por el docente.
- Buscar el número de estudiantes que respondieron a las preguntas de la evaluación en la materia seleccionada.

Salidas

- Número de estudiantes que evaluaron en la materia seleccionada por el docente.

R8.

Introducción

El módulo de estadísticas deberá mostrar los comentarios que los estudiantes escribieron al docente en una materia.

Entradas

Fuentes de las entradas:

- Cédula del docente
- Contraseña para autenticación
- Código de la carrera.
- Código de la asignatura o materia deseada

Rangos de entradas validas y requisitos de control

- La cédula para el ingreso contendrá 11 dígitos pues incluirá el guión.
- El código de la carrera y la materia contendrá hasta 10 caracteres incluyendo números y letras.

Proceso

- Obtener la cédula del docente y su contraseña.
- Validar datos obtenidos para acceder al sistema.
- Identificar la materia seleccionada por el docente.
- Recorrer buscando los comentarios asignados por estudiantes al docente en esa materia.

Salidas

- Se enumerará y mostrará los comentarios que los estudiantes escribieron a los docentes.

R9.

Introducción

El módulo de estadísticas deberá mostrar el promedio total obtenido por un docente en la autoevaluación clasificado por las carreras en las que imparte clases.

Entradas

Fuentes de las entradas:

- Cédula del docente
- Contraseña para autenticación

Rangos de entradas validas y requisitos de control

- La cédula para el ingreso contendrá 11 dígitos pues incluirá el guión.

Proceso

- Obtener la cédula del docente y su contraseña.
- Validar datos obtenidos para acceder al sistema.
- Buscar carreras asignadas al docente.
- Calcular promedio total por carreras en la autoevaluación.

Salidas

- Se mostrará en la pantalla las carreras en las que el docente participa y su respectivo promedio.

R10.

Introducción

El módulo de estadísticas deberá mostrar el promedio total obtenido por preguntas en una carrera en la que el docente imparte clases.

Entradas

Fuentes de las entradas:

- Cédula del docente
- Contraseña para autenticación

Rangos de entradas validas y requisitos de control

- La cédula para el ingreso contendrá 11 dígitos pues incluirá el guión.
- El código de la carrera contendrá hasta 10 caracteres incluyendo números y letras.

Proceso

- Obtener la cédula del docente y su contraseña.
- Validar datos obtenidos para acceder al sistema.
- Mostrar carreras asignadas al docente.
- Seleccionar carrera
- Calcular promedio total por preguntas de la carrera en la autoevaluación.

Salidas

Se mostrará el cuestionario de la autoevaluación con los promedios en cada pregunta.

R11.

Introducción

El módulo de estadísticas deberá mostrar el promedio total obtenido por un docente.

Entradas

Fuentes de las entradas:

- Cédula del docente
- Contraseña para autenticación

Rangos de entradas validas y requisitos de control

- La cédula para el ingreso contendrá 11 dígitos pues incluirá el guión.

Proceso

- Obtener la cédula del docente y su contraseña.
- Validar datos obtenidos para acceder al sistema.
- Obtener el promedio total en la evaluación por parte de los estudiantes.
- Obtener el promedio de la autoevaluación.
- Calcular el promedio total del docente.

Salidas

Se mostrará el promedio total del docente.

R12.

Introducción

El módulo de estadísticas deberá mostrar la desviación estándar de cada pregunta de un cuestionario.

Entradas

Fuentes de las entradas:

- Cédula del docente

- Contraseña para autenticación

Rangos de entradas validas y requisitos de control

- La cédula para el ingreso contendrá 11 dígitos pues incluirá el guión.

Proceso

- Obtener la cédula del docente y su contraseña.
- Validar datos obtenidos para acceder al sistema.
- Identificar la materia seleccionada por el docente.
- Calcular la desviación estándar por cada pregunta del cuestionario.

Salidas

- Se mostrará las preguntas de las materias con un campo adicional donde se indicará la desviación estándar de cada pregunta.

Estadísticas Públicas

R13.

Introducción

El módulo de estadísticas deberá mostrar el promedio total obtenido por Facultad.

Proceso

- Se tomará a nivel general los resultados obtenidos por los docentes y los resultados obtenidos en la autoevaluación.
- Con los datos mencionados anteriormente se realizará un cálculo para obtener el promedio de cada facultad.

Salidas

- Se mostrará los promedios obtenidos en la evaluación de los estudiantes por cada facultad.
- Se mostrará los promedios obtenido en la autoevaluación por cada facultad.
- Se mostrara los promedios calculado por facultad.

R14.

Introducción

El módulo de estadísticas deberá mostrar el promedio total obtenido por Escuela de una Facultad.

Entradas

Fuentes de las entradas:

- Código de la facultad

Rangos de entradas validas y requisitos de control

- El código de la carrera contendrá hasta 10 caracteres incluyendo números y letras.

Proceso

- Seleccionar la facultad.
- Calcular el promedio que se obtuvo en el proceso de evaluación en todas las carreras que pertenecen a esta facultad.

Salidas

- Mostrar los resultados de la evaluación de estudiantes de la carrera.

- Mostrar la autoevaluación docente realizada en esa carrera.
- Mostrar el promedio obtenido por la carrera.

R15.

Introducción

El módulo de estadísticas deberá mostrar el promedio total obtenido por Carrera de una Escuela.

Entradas

Fuentes de las entradas:

- Código de la facultad
- Código de la escuela

Rangos de entradas validas y requisitos de control

- El código de la facultad y de la carrera contendrá hasta 10 caracteres incluyendo números y letras.

Proceso

- Seleccionar la facultad en la que se encuentra la escuela.
- Calcular el promedio de todas la escuelas pertenecientes a esta facultad

Salidas

- Mostrar los resultados de la evaluación de estudiantes de la escuela.
- Mostrar la autoevaluación docente realizada en esa escuela.
- Mostrar el promedio obtenido por la escuela.

R16.

Introducción

El módulo de estadísticas deberá mostrar el promedio total obtenido por preguntas de una carrera en la evaluación de los estudiantes.

Entradas

Fuentes de las entradas:

- Código de la facultad
- Código de la escuela
- Código de la carrera

Rangos de entradas validas y requisitos de control

- El código de la facultad, carrera y escuela contendrá hasta 10 caracteres incluyendo números y letras.

Proceso

- Seleccionar la carrera de la cual deseamos tener información.
- Procesamos los datos y obtenemos el promedio de la evaluación de los estudiantes.
- Calcular el promedio de las preguntas del cuestionario asignado a la carrera.

Salidas

- Se muestra el cuestionario de la carrera en la evaluación de los estudiantes con los promedios en las respectivas preguntas.

R17.

Introducción

El módulo de estadísticas deberá mostrar el promedio total obtenido por preguntas de una carrera en la autoevaluación docente.

Entradas

Fuentes de las entradas:

- Código de la facultad
- Código de la escuela
- Código de la carrera

Rangos de entradas validas y requisitos de control

- El código de la facultad, carrera y escuela contendrá hasta 10 caracteres incluyendo números y letras.

Proceso

- Seleccionar la carrera de la cual deseamos tener información.
- Procesamos los datos y obtenemos el promedio de la autoevaluación docente.
- Calcular el promedio de las preguntas del cuestionario asignado a la carrera.

Salidas

- Se muestra el cuestionario de la carrera en la autoevaluación docente con los promedios en las respectivas preguntas.

4.2.3.2 Requisitos de Comportamiento

- Número de usuarios simultáneos: Varios
- Número de tablas.

Aproximadamente existen 22 tablas de las cuales depende el módulo de estadísticas de la epoch

- Número de registros por tabla.
 - La tabla Docente posee 3 registros
 - La tabla Estudiante posee 3 registros
 - La tabla Materia posee 8 registros
 - La tabla Carrera posee 4 registros
 - La tabla Escuela posee 3 registros
 - La tabla Facultad posee 2 registros
 - La tabla Cuestionario posee 4 registros
 - La tabla Pregunta posee 8 registros
 - La tabla grupo posee 4 registros
 - La tabla proceso posee 6 registro
 - La tabla proceso evaluación posee 5 registros
 - La tabla respuesta docente posee 6 registros
 - La tabla respuesta estudiante posee 6 registros
 - La tabla comentarios estudiantes posee 2 registros
 - La tabla comentarios docentes posee 2 registros
 - La tabla opciones posee 8 registros
 - La tabla estadísticas estudiante docente posee 12 registros

La tabla estadísticas docente docente posee 12 registros

4.2.3.3 Limitaciones de Diseño

Obediencia a Estándares

- Los resultados siempre mostrarán el nombre del docente evaluado.
- También se mostrará el periodo vigente de evaluación.

4.2.3.4 Atributos

1.-Fácil de Manejar

El módulo de estadísticas del sistema SIEDD será fácil de manejar ya que proporcionará una interfaz amigable e intuitiva en el manejo de la interfaz gráfica.

2.- Sistema seguro

Puesto que se realizará procesos para proteger la información de los docentes.

3.- Respaldo de Información

- El sistema permitirá realizar backups de la base de datos periódicamente.
- Se realizará la debida documentación sobre el funcionamiento del sistema.

4.2.3.5 Requisitos de Interfaces Externas

➤ INTERFACES DE USUARIO

ESTADISTICAS PÚBLICAS

- Pantalla con resultados por Facultad
- Pantalla con resultados por Escuela
- Pantalla con resultados por Carrera

- Pantalla con resultados por Cuestionario

ESTADÍSTICAS PRIVADAS

- Pantalla para ingreso de datos para la Validación
- Pantalla con resultados con promedios Totales
- Pantalla con resultados con promedio por Carrera
- Pantalla con resultados con promedio por Asignatura
- Pantalla con resultados con promedio por Cuestionario.

➤ INTERFACES HARDWARE

Se utilizaran las interfaces hardware instaladas en cada laboratorio de cómputo de la ESPOCH.

➤ INTERFACES SOFTWARE

Haremos uso de las herramientas con las que se cuentan en la institución. Como son: el motor de bases de datos SQL Server y la herramienta para el desarrollo de aplicaciones JavaBeans que es de uso libre.

➤ INTERFACES DE COMUNICACIÓN

La conexión a la red se establecerá por medio de una conexión directa usando cableado estructurado entre los distintos laboratorios de la ESPOCH.

4.2.3.6 Otros Requerimientos.

El modelo ciclo de vida para desarrollar el producto software será el Modelo RUP basándonos en la metodología de Craig Larman, por ser adecuado a nuestro estudio y se basa en la orientación a objetos.

4.3 PLANIFICACIÓN DEL PROYECTO

4.3.1. Definir y Estimar recursos

Para continuar con nuestro proyecto es necesario realizar estimaciones a cerca de esfuerzo humano, duración del proyecto y costo del mismo, por lo que a continuación se analizan los parámetros que se tomarán en cuenta.

4.3.1.1. Identificar una técnica de Estimación

Para realizar la estimación del módulo de estadísticas del sistema SIEDD se tomarán en cuenta las medidas indirectas del software que se centran en su funcionalidad basada en los **puntos de función**

4.3.1.2. Aplicar la técnica

a. Puntos de Función

ILF

TABLA IV.8 PUNTOS DE FUNCION ILF

Fichero lógico interno	Numero DET	Numero RET	Complejidad
Usuario	6	2	Media
Docente	3	1	Media
Materia	8	1	Baja
Carrera	4	1	Baja
Facultad	2	1	Baja
Cuestionario	4	1	Baja
Pregunta	8	1	Baja
Grupo de Pregunta	4	1	Baja
Comentarios	2	1	Baja
Estadísticas Estudiante Docente	12	1	Baja
Estadísticas Docente Docente	12	1	Baja

EIF

Cero para nuestro proyecto

EI

TABLA IV.9 PUNTOS DE FUNCION EI

Entrada Externa	Función	Número de Entradas
Datos Docente	Alta/Baja/Modificación	3

Entrada Externa	Numero de FTR	Numero de DET	Complejidad
Datos Docente	2	2	Baja

EO

TABLA IV.11 PUNTOS DE FUNCION EO Entrada Externa

Entrada Externa	Función	Número de Entradas
Validad Docente.	Pantalla	2
Promedio por Cuestionario del docente	Pantalla	2
Promedio por carrera del docente	Pantalla	2
Promedio por Asignatura del docente	Pantalla	3
Promedio por Pregunta del docente	Pantalla	4
Promedio por Facultad	Pantalla	0
Promedio por Escuela	Pantalla	1
Promedio por Carrera	Pantalla	2
Promedio por preguntas de una carrera	Pantalla	3

TABLA IV.12 PUNTOS DE FUNCION EO Salida Externa

Salida Externa	Numero de FTR	Numero de DET	Complejidad
Validad Usuario.	1	2	Baja
Promedio por Cuestionario del docente	1	2	Baja
Promedio por carrera del docente	1	4	Baja
Promedio por Asignatura del docente	1	4	Baja
Promedio por Pregunta del docente	1	4	Baja
Promedio por Facultad.	1	4	Baja
Promedio por Escuela	1	4	Baja
Promedio por Carrera	1	4	Baja
Promedio por preguntas de una carrera	1	3	Baja

EQ

TABLA IV.13 PUNTOS DE FUNCION EQ Entrada Externa

Entrada Externa	Función	Número de Entradas
Consultar Promedio por Cuestionario del docente.	Pantalla/Papel	2
Consultar Promedio por carrera del docente	Pantalla/Papel	2
Consultar Promedio por Asignatura del docente	Pantalla/Papel	2
Consultar Promedio por Pregunta del docente.	Pantalla/Papel	2
Consultar Promedio por Facultad	Pantalla/Papel	0
Consultar Promedio por Escuela.	Pantalla/Papel	1
Consultar Promedio por Carrera	Pantalla/Papel	2
Promedio por preguntas de una carrera	Pantalla/Papel	3

TABLA IV.14 PUNTOS DE FUNCION EQ Entrada Externa DET y FTR

Entrada Externa	Entrada		Salida	
	DET	FTR	DET	FTR
Consultar Promedio por Cuestionario del docente.	1	1	2	1
Consultar Promedio por carrera del docente	1	1	2	1
Consultar Promedio por Asignatura del docente	1	1	2	1
Consultar Promedio por Pregunta del docente.	1	1	4	1
Consultar Promedio por Facultad	1	1	3	1
Consultar Promedio por Escuela.	1	1	3	1
Consultar Promedio por Carrera	1	1	3	1
Consultar Promedio por preguntas de una carrera	1	1	3	1

TABLA IV.15 PUNTOS DE FUNCION EQ Entrada Externa Complejidad

Entrada Externa	C. Entrada	C. Salida	Complejidad
Consultar Promedio por Cuestionario del docente.	Baja	Baja	Baja
Consultar Promedio por carrera del docente	Baja	Baja	Baja
Consultar Promedio por Asignatura del docente	Baja	Baja	Baja
Consultar Promedio por Pregunta del docente.	Baja	Media	Media
Consultar Promedio por Facultad	Baja	Baja	Baja
Consultar Promedio por Escuela.	Baja	Baja	Baja
Consultar Promedio por Carrera	Baja	Baja	Baja
Consultar promedio por preguntas de una carrera	Baja	Media	Media

TABLA IV.16 PUNTOS DE FUNCION RESUMEN

Parámetro	Complejidad	No	Peso	Total
ILF	ALTA	0	15	0
	MEDIA	2	10	20
	BAJA	9	7	63
EIF	ALTA	0	10	0
	MEDIA	0	7	0
	BAJA	0	5	0
EI	ALTA	0	6	0
	MEDIA	0	4	0
	BAJA	1	3	3
EO	ALTA	0	7	0
	MEDIA	0	5	0
	BAJA	9	4	36
EQ	ALTA	0	6	0
	MEDIA	2	4	8
	BAJA	6	3	18
TOTAL				148

b. COCOMO II

SLOC Input Dialog - Modulo Estadísticas

Sizing Method:
 SLOC
 Function Points
 Adaptation

Breakage
 % of code thrown away due to requirements volatility
 BRAK 0.00

Module Size in Function Points
 Language C

Function Type	# of Function Points			SubTotal
	Low	Average	High	
Inputs	1	0	0	3
Outputs	9	0	0	36
Files	9	2	0	83
Interfaces	0	0	0	0
Queries	6	2	0	18
Total Unadjusted Function Points				140
Equivalent Total in SLOC				17920

OK Cancel Help

Figura IV.2 Resultados COCOMO

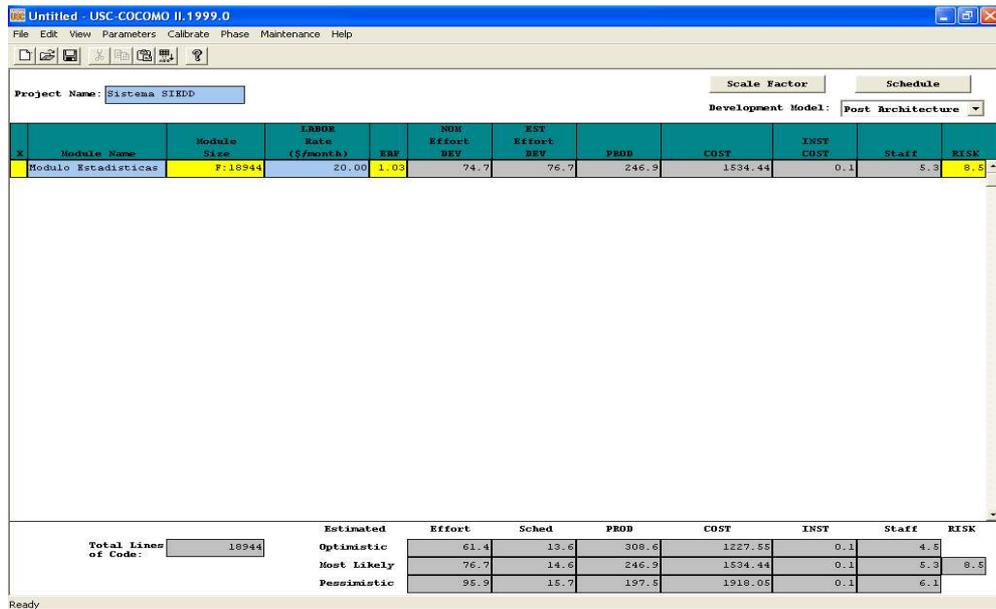


Figura IV.3 Resultados COCOMO costos

4.3.1.3. Evaluar los resultados

Según la herramienta el módulo de estadísticas del sistema SIEDD contendrá aproximadamente 18944 líneas de código.

- **Estimación Optimista**

Con un esfuerzo del 61.4 y con un staff de 4 personas el módulo tendrá un costo de 1227.55

- **Estimación Adecuada**

Con un esfuerzo del 76.7 y con un staff de 5 personas el módulo tendrá un costo de 1534.44

- **Estimación Pesimista**

Con un esfuerzo del 95.9 y con un staff de 6 personas el módulo tendrá un costo de 1918.05

4.3.2. Planificación Temporal

4.3.2.1 Cronograma de trabajo

	Nombre de tarea	Duración	Comienzo	Fin	Nombres de los recursos
1	INICIACION	8 días	lun 30/03/09	mié 08/04/09	
2	INGENIERIA DE LA INFORMACIÓN	2 días	lun 30/03/09	mar 31/03/09	Computador
3	DEFINICIÓN TÉCNICA DE REQUERIMIENTOS	2 días	mié 01/04/09	jue 02/04/09	Usuario
4	PLANIFICACIÓN Y ESTIMACION DE COSTOS DEI	2 días	vie 03/04/09	lun 06/04/09	Computador,Hojas
5	DEFINICION DE CASOS DE USO DE ALTO NIVEL	2 días	mar 07/04/09	mié 08/04/09	Computador,Hojas[1],Usuar
6	ELABORACIÓN	10 días	jue 09/04/09	mié 22/04/09	
7	DEFINIR CASOS DE USO ESENCIALES	2 días	jue 09/04/09	vie 10/04/09	Computador,Hojas[1]
8	REFINAR DIAGRAMAS DE CASOS DE USO	2 días	lun 13/04/09	mar 14/04/09	Analista,Computador,Hojas
9	DEFINIR DIAGRAMAS DE SECUENCIA	2 días	mié 15/04/09	jue 16/04/09	Analista,Programador,Com
10	REFINAR EL MODELO CONCEPTUAL	2 días	vie 17/04/09	lun 20/04/09	Analista,Programador
11	DEFINIR CONTRATOS DE OPERACIÓN	2 días	mar 21/04/09	mié 22/04/09	Analista
12	CONSTRUCCIÓN	80 días	jue 23/04/09	mié 12/08/09	
13	DEFINIR CASOS DE USO REALES	2 días	jue 23/04/09	vie 24/04/09	Analista
14	DEFINIR INFORMES DE INTERFÁZ DE USUARIO	2 días	lun 27/04/09	mar 28/04/09	Programador,Analista
15	DEFINIR DIAGRAMAS DE INTERACCIÓN	2 días	mié 29/04/09	jue 30/04/09	Analista,Programador
16	DEFINIR DIAGRAMAS DE CLASE	2 días	vie 01/05/09	lun 04/05/09	Analista,Programador
17	DEFINIR EL ESQUEMA DE LA BASE DE DATOS	2 días	mar 05/05/09	mié 06/05/09	Analista,Programador
18	MODELO FÍSICO Y ARQUITECTURA DEL SISTEM	5 días	jue 07/05/09	mié 13/05/09	Analista,Programador
19	IMPLEMENTACION	60 días	jue 14/05/09	mié 05/08/09	Programador,Computador
20	PRUEBAS	5 días	jue 06/08/09	mié 12/08/09	Programador,Analista,Usue
21	TRANSICIÓN	1 día	vie 14/08/09	vie 14/08/09	
22	IMPLANTACIÓN DEL SISTEMA	1 día	vie 14/08/09	vie 14/08/09	Usuario

Figura IV.4 Cronograma de Trabajo

4.3.2.2 Diagramas Gantt

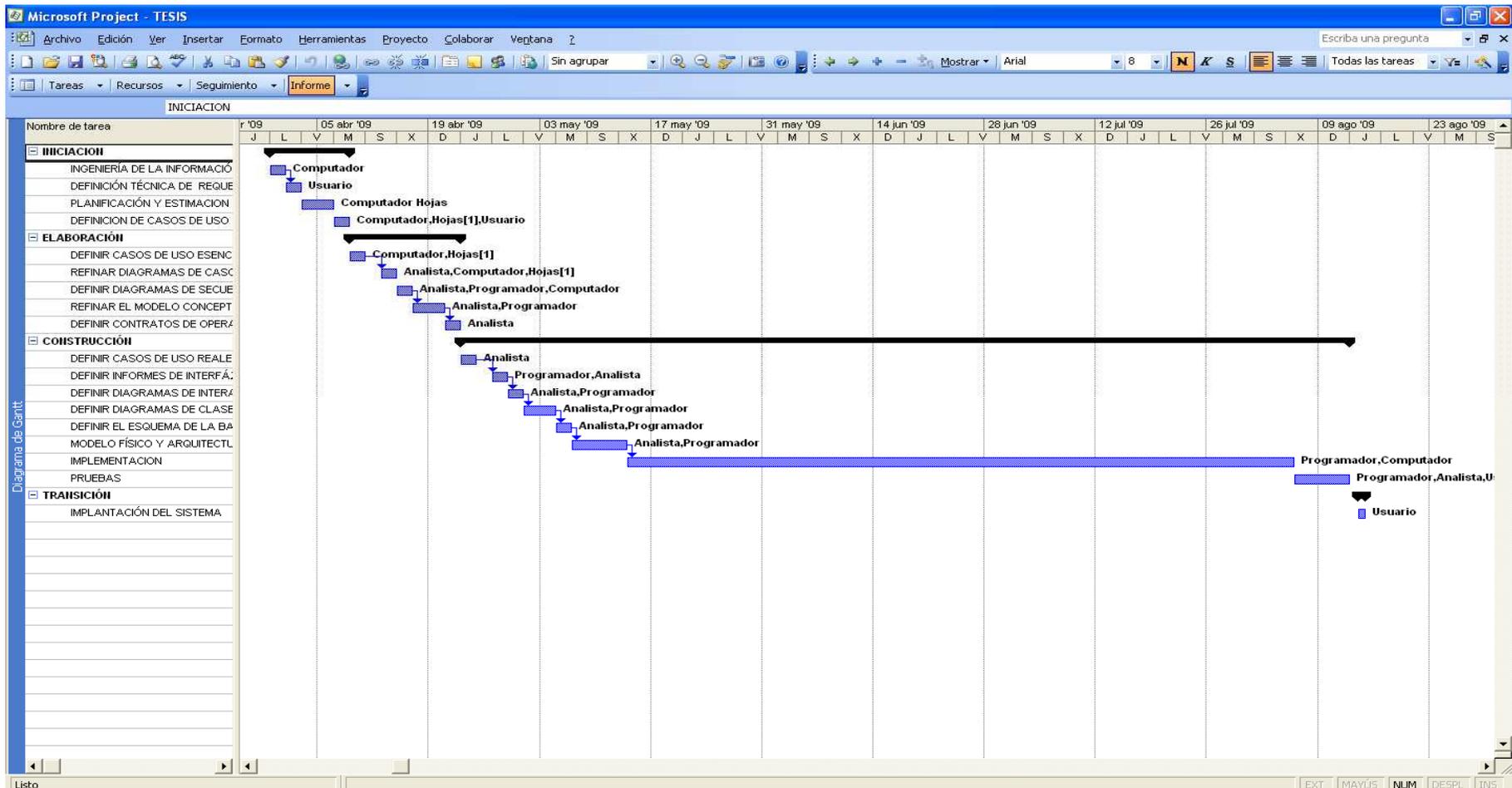


Figura IV.5 Diagrama Gantt

4.3.3. Análisis y Gestión de Riesgos

4.3.3.1. Identificación de Riesgos

A continuación vamos a identificar los riesgos relacionados al desarrollo del módulo de estadísticas del sistema SIEDD de la ESPOCH donde los describiremos, clasificaremos por categorías y daremos a conocer sus consecuencias.

Se utiliza además la siguiente nomenclatura:

R.P: riesgos de proyecto

R.T: riesgos técnicos

R.N: riesgos de negocios

TABLA IV.17 RIESGOS IDENTIFICADOS

IDENTIFICACION	DESCRIPCION DEL RIESGO	CATEGORIA	CONSECUENCIA
R1	Falta de tecnología actualizada en la empresa	R.T	Retraso en el proyecto por tecnología inadecuada
R2	El usuario cambia continuamente los requisitos	R.P	- Retraso en el proyecto - Rediseño del proyecto
R3	Falta de software necesario para el desarrollo del proyecto	R.T	- elevaciones de costos del proyecto pos la adquisición de nuevos SW
R4	Falta de personal en la empresa que proporcione información al equipo de desarrollo	R.P	- Retraso del proyecto - insuficiente información para el desarrollo del sw
R5	Usuarios finales se resisten a utilizar el sistema	R.N	- Proyecto desarrollado en vano - perdida de recursos
R6	No existe un software base para orientar el desarrollo del nuevo proyecto	R.P	El desarrollo del proyecto tomara mucho tiempo para su estudio e iniciación
R7	Interfaces difíciles de comprender para el usuario	R.N	Abandono de la utilización del SW por los usuarios
R8	Perder el apoyo del nivel estratégico	R.N	Suspensión del proyecto
R9	Sistema operativo no soporta el software que se encuentra desarrollando	R.T	-Elevaciones de costos -Retraso del proyecto

R10	Ambigüedad en las especificaciones de requerimientos proporcionados por el usuario	R.P	Construcción equivocada de las funciones del proyecto
R11	Retraso de las asignaciones económicas del proyecto	R.P	Retraso en la planificación
R12	Desacuerdos entre el equipo de desarrollo y los usuarios	R.P	Retraso en la planificación del proyecto
R13	Recursos hardware limitados	R.T	- Software desarrollado con menor calidad - retrasos en el proyecto
R14	No existen técnicos en la empresa que nos informen acerca de los recursos hardware y software existentes en la empresa.	R.P	Información incompleta o no veras para el desarrollo del proyecto
R15	No se cumple con la planificación prevista	R.P	Retraso en el proyecto
R16	No contar con la base de datos apropiada para el sistema a desarrollar	R.T	Software desarrollado con menor calidad
R17	Menos reutilización del software de la prevista	R.P	Demora en la programación.
R18	Falta de seguridad en el software desarrollado.	R.T	Software deficiente
R19	El software no cumple con lo requerimientos especificados al inicio del proyecto.	R.N	Software desarrollado con baja calidad
R20	Cambio de responsable del proyecto	R.N	Suspensión del proyecto

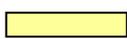
4.3.3.2. Categorización de Riesgos

TABLA IV.18 CATEGORIZACIÓN DE RIESGOS

IDENTIFICACION	PROBABILIDAD	IMPACTO	EXPOSICION	PRIORIDAD
R2	3	3	9	1
R15	3	3	9	1
R10	2	4	8	2
R3	2	4	8	2
R7	2	3	6	3
R1	2	3	6	3
R17	3	2	6	3
R16	3	2	6	3
R5	1	4	4	4
R8	1	4	4	4
R11	1	4	4	4
R18	1	4	4	4
R19	1	4	4	4
R12	1	4	4	4
R9	2	1	2	5
R13	1	2	2	5
R20	1	2	2	5
R4	1	1	1	6
R6	1	1	1	6
R14	1	1	1	6

A continuación categorizaremos los riesgos de acuerdo a su prioridad e incluiremos el código de colores y la línea de corte que va a ser tomado encuentra como principales a los riesgos de alta y media prioridad.

 RIESGOS DE MAYOR IMPACTO

 RIESGOS DE IMPACTO MEDIO

 RIESGOS DE MENOR IMPACTO

 LINEA DE CORTE, ES DECIR MUESTRA HASTA QUE RIESGOS SE VAN A TRATAR

4.3.3.3. Plan de Contingencia

TABLA IV.19 HOJA DE GESTIÓN DE RIESGOS R2
HOJA DE GESTIÓN DE RIESGOS

ID DEL RIESGO: R2				FECHA: 19 de Diciembre del 2005	
Probabilidad: Alta Valor: 2		Impacto: Alto Valor 4		Exposición: Alta Valor: 8	
Prioridad: Rojo Valor: 2					
DESCRIPCIÓN: El usuario cambia constantemente de requisitos.					
REFINAMIENTO: Causa: El usuario al momento de proporcionar la información requerida por el personal de desarrollo no toma en cuenta todos los aspectos que podrían necesitar lo que provoca que cambie constantemente de requerimientos. Consecuencia: Retraso del proyecto y hasta rediseño del mismo.					
REDUCCIÓN Y SUPERVISIÓN: Reducción: Realizar varias consultas a los usuarios antes de comenzar el proyecto. Tener comunicación permanente con los usuarios manteniéndolos al tanto de todos los avances del proyecto. Supervisión: Durante el desarrollo del proyecto se deberá tomar en cuenta todos los posibles criterios que puedan dar los usuarios sobre el proyecto.					
GESTIÓN: Tratar de adaptar lo mejor posible o lo más cerca que se pueda a los nuevos requerimientos de usuario para que el desarrollo de Software conseguido no deba replantearse nuevamente.					
ESTADO ACTUAL: Fase De Reducción Iniciada <input checked="" type="checkbox"/> Fase de Supervisión Iniciada <input type="checkbox"/> Gestionando El Riesgo <input type="checkbox"/>					
RESPONSABLES: Ing. Jorge Huilca Mariela Cabay.					

TABLA IV.20 HOJA DE GESTIÓN DE RIESGOS R15

HOJA DE GESTIÓN DE RIESGOS			
ID DEL RIESGO: R15		FECHA: 19 de Diciembre del 2005.	
Probabilidad: Media Valor: 3	Impacto: Alto Valor: 3	Exposición: Alto Valor: 9	Prioridad: Rojo Valor: 1
DESCRIPCIÓN: No se cumple con la planificación prevista.			
REFINAMIENTO: Causa: Problemas en la recolección de Información. Inconvenientes con la tecnología para el desarrollo del Software. Consecuencias: Retraso del proyecto.			
REDUCCIÓN Y SUPERVISIÓN: Reducción: Estudiar los factores que retrasan la planificación. Asignar más tiempo a cada recurso de desarrollo. Supervisión: Colocar hitos de control para revisar periódicamente los avances. Realizar reuniones continuamente con todos los desarrolladores para que todos expongan sus avances.			
GESTIÓN: Implantar técnicas para intentar recuperar el tiempo perdido como realizar horas extras en las áreas retrasadas o cambiar el cronograma establecido prolongando por mas tiempo las áreas con dificultades.			
ESTADO ACTUAL: Fase De Reducción Iniciada <input checked="" type="checkbox"/> Fase de Supervisión Iniciada <input type="checkbox"/> Gestionando El Riesgo <input type="checkbox"/>			
RESPONSABLES: Ing. Jorge Huilca Mariela Cabay.			

TABLA IV.21 HOJA DE GESTIÓN DE RIESGOS R10

HOJA DE GESTIÓN DE RIESGOS			
ID DEL RIESGO: R10		FECHA: 19 de Diciembre del 2005.	
Probabilidad: Media Valor: 2	Impacto: Crítico Valor: 4	Exposición: Alto Valor: 8	Prioridad: Rojo Valor: 2
DESCRIPCIÓN: Ambigüedad en las especificaciones de requerimientos proporcionados por los usuarios.			
REFINAMIENTO: Causa: Especificaciones de requerimientos recogidas a usuarios diferentes o por diferentes personas del equipo de desarrollo. Consecuencia: Construcción equivocada de las funciones del proyecto.			
REDUCCIÓN Y SUPERVISIÓN: Reducción: Encargar a una sola persona del equipo de desarrollo para que realice la recolección de información. Realizar más de una vez la recolección de especificaciones y requerimientos. Supervisión: Revisar la información verificando si se encuentra o no con ambigüedades o confusiones.			
GESTIÓN: Eliminar las ambigüedades procurando no volver al inicio del desarrollo del proyecto , tomando en cuenta los requerimientos principales de los usuarios.			
ESTADO ACTUAL: Fase De Reducción Iniciada <input checked="" type="checkbox"/> Fase de Supervisión Iniciada <input type="checkbox"/> Gestionando El Riesgo <input type="checkbox"/>			
RESPONSABLES: Ing. Jorge Huilca Mariela Cabay.			

TABLA IV.22 HOJA DE GESTIÓN DE RIESGOS R3

HOJA DE GESTIÓN DE RIESGOS			
ID DEL RIESGO: R3		FECHA: 19 de Diciembre del 2005	
Probabilidad: Media Valor: 2	Impacto: Crítico Valor: 4	Exposición: Alta Valor: 8	Prioridad: Roja Valor: 2
DESCRIPCIÓN: Falta de Software necesario para el desarrollo del proyecto			
REFINAMIENTO: Causa: Los componentes del sistema de la empresa es básico, solo contiene lo esencial para el trabajo de oficina no Software que permite el desarrollo del proyecto. Consecuencia: Elevaciones de costos por la adquisición de los Nuevos Software.			
REDUCCIÓN Y SUPERVISIÓN: Reducción: Tomar en cuenta los programas que fueron instalados inicialmente asegurándose de que existan o no programas adecuados para el desarrollo de Software. Supervisión: Peticionar lo Software necesario al inicio del proyecto revisando su funcionamiento y acoplamiento al Software en desarrollo.			
GESTIÓN: Reutilizar el código realizado para otros programas que permitan optimizar y realizar un Software de calidad.			
ESTADO ACTUAL: Fase De Reducción Iniciada <input checked="" type="checkbox"/> Fase de Supervisión Iniciada <input type="checkbox"/> Gestionando El Riesgo <input type="checkbox"/>			
RESPONSABLES: Ing. Jorge Huilca Mariela Cabay.			

TABLA IV.23 HOJA DE GESTIÓN DE RIESGOS R7

HOJA DE GESTIÓN DE RIESGOS			
ID DEL RIESGO: R7		FECHA:19 de Diciembre del 2005	
Probabilidad: Media Valor: 2	Impacto: Crítico Valor:3	Exposición: Alta Valor: 6	Prioridad: Roja Valor:3
DESCRIPCIÓN: Interfaces difíciles de comprender para el usuario			
REFINAMIENTO: Causa: Aplicación con demasiados accesorios o componentes no explicativos. Consecuencia: Rechazo del usuario a utilizar la aplicación propuesta.			
REDUCCIÓN Y SUPERVISIÓN: Reducción: Tomar en cuenta los principales requerimientos del usuario y desglosarlos. Supervisión: Avanzar con el usuario en la adquisición de requerimientos.			
GESTIÓN: Realizar prototipos para avanzar de acuerdo a los requerimientos del usuario..			
ESTADO ACTUAL: Fase De Reducción Iniciada <input checked="" type="checkbox"/> Fase de Supervisión Iniciada <input type="checkbox"/> Gestionando El Riesgo <input type="checkbox"/>			
RESPONSABLES: Ing. Jorge Huilca Mariela Cabay.			

TABLA IV.24 HOJA DE GESTIÓN DE RIESGOS R1

HOJA DE GESTIÓN DE RIESGOS			
ID DEL RIESGO: R1		FECHA: 19 de diciembre del 2005	
Probabilidad: Media Valor: 2	Impacto: Alta Valor:3	Exposición: Alta Valor: 6	Prioridad: Rojo Valor 3
DESCRIPCIÓN: Falta de Tecnología Actualizada			
REFINAMIENTO: Causa: la empresa no dispone de tecnología informática actualizada. Consecuencias: Se producirá un retraso en el proyecto por tecnología inadecuada.			
REDUCCIÓN Y SUPERVISIÓN: Reducción: Informar a los directivos de las últimas tendencias de la tecnología y dar a conocer las posibles mejoras que estas pueden proporcionar, incentivando a la adquisición de estos productos. Supervisión: Es aconsejable revisar el nivel de trabajo de los equipos tanto como su vida útil. Examinar el tipo de mantenimiento que se da a los equipos.			
GESTIÓN: Conseguir que los equipos existentes produzcan al máximo vigilando la calidad de la producción y obteniendo el máximo provecho de los mismos. Dar mantenimiento consecuentemente para evitar posibles daños.			
ESTADO ACTUAL: Fase De Reducción Iniciada <input checked="" type="checkbox"/> Fase de Supervisión Iniciada <input type="checkbox"/> Gestionando El Riesgo <input type="checkbox"/>			
RESPONSABLES: Ing. Jorge Huilca Mariela Cabay.			

TABLA IV.25 HOJA DE GESTIÓN DE RIESGOS R17

HOJA DE GESTIÓN DE RIESGOS			
ID DEL RIESGO: R17		FECHA: 19 de diciembre del 2005	
Probabilidad: Media Valor: 3	Impacto: Alta Valor:2	Exposición: Alta Valor: 6	Prioridad: Rojo Valor 3
DESCRIPCIÓN: Menos reutilización del software de la prevista			
REFINAMIENTO: Causa: No existen componentes que ayuden al desarrollo rápido de la aplicación Consecuencias: Se producirá un retraso en el proyecto por aumento en el tiempo de desarrollo.			
REDUCCIÓN Y SUPERVISIÓN: Reducción: Informar a los directivos sobre la inexistencia de componentes y recomendar una biblioteca de componentes. Supervisión: Es aconsejable revisar el nivel de trabajo de los equipos tanto como su vida útil.			
GESTIÓN: Crear de forma organizada y documentada una biblioteca de componentes que permitan el desarrollo rápido de aplicaciones futuras.			
ESTADO ACTUAL: Fase De Reducción Iniciada <input checked="" type="checkbox"/> Fase de Supervisión Iniciada <input type="checkbox"/> Gestionando El Riesgo <input type="checkbox"/>			
RESPONSABLES: Ing. Jorge Huilca Mariela Cabay.			

TABLA IV.26 HOJA DE GESTIÓN DE RIESGOS R16

HOJA DE GESTIÓN DE RIESGOS			
ID DEL RIESGO: R16		FECHA: 19 de diciembre del 2005	
Probabilidad: Media Valor: 3	Impacto: Alta Valor:2	Exposición: Alta Valor: 6	Prioridad: Rojo Valor 3
DESCRIPCIÓN: No contar con la base de datos apropiada para el sistema a desarrollar			
REFINAMIENTO: Causa: Inapropiada recolección de requerimientos, inadecuada modelación de componentes Consecuencias: Software desarrollado con menor calidad.			
REDUCCIÓN Y SUPERVISIÓN: Reducción: Refinar adecuadamente los requerimientos que actúan con la base de datos de forma directa. Supervisión: Es aconsejable refinar correctamente el modelado de casos de uso y de componentes			
GESTIÓN: Análisis y rediseño de la base de datos actual documentando todos los posibles cambio y errores.			
ESTADO ACTUAL: Fase De Reducción Iniciada <input checked="" type="checkbox"/> Fase de Supervisión Iniciada <input type="checkbox"/> Gestionando El Riesgo <input type="checkbox"/>			
RESPONSABLES: Ing. Jorge Huilca Mariela Cabay.			

4.3.4 Estudio de la Factibilidad

4.3.5.1 Factibilidad Operativa

Previo a una socialización del nuevo sistema de evaluación docente institucional los usuarios que tendrán acceso al mismo serán los docentes previa autenticación y usuarios en general para visualizar información de la institución.

4.3.5.2 Factibilidad Técnica

a. Recurso Humano

El recurso humano con el que se cuenta para el desarrollo del sistema es:

- Ing. Jorge Huilca Director del proyecto
- Dra. Yolanda Días Jefe de la comisión de evaluación institucional.
- Dr. Alonso Alvares Asesor
- Elvia Cabay Desarrollador

b. Recurso Software

La esPOCH cuenta con el recurso software necesario para la elaboración de este proyecto como lo es:

Servidor de Base de Datos: Microsoft SQL Server 2000

Sistema Operativo: Windows XP.

Servidor Web: Apache Tomcat 5.0

Herramienta de desarrollo: Netbeans IDE 6.1 con JSP de Java

c. Recurso Hardware

La esPOCH cuenta con el hardware requerido para el correcto funcionamiento del sistema planteado.

4.3.5.3 Factibilidad Económica

La escuela Superior Politécnica de Chimborazo por medio de la comisión de evaluación institucional ha dado conocer que cuenta con la infraestructura necesaria para la implementación del sistema y que poseen todos los equipos tanto hardware como software por lo que no se tendrá que realizar ninguna inversión en la adquisición de recursos.

4.3.5.4 Factibilidad Legal

Se cuenta con la autorización y respectiva dirección de la comisión de evaluación institucional y de la ESPOCH, así como también se cuenta con las respectivas licencias de las herramientas que así lo requieren; por lo que no existe ningún impedimento legal para el desarrollo del sistema.

4.4 DEFINIR CASOS DE USO DE ALTO NIVEL

En este punto del desarrollo del proyecto se dará a conocer los principales requerimientos, además de los principales actores que intervendrán en el módulo de estadísticas del sistema SIEDD.

Para la descripción de lo antes mencionado utilizaremos los Casos de Uso con una descripción realizada en lenguaje natural para una mejor comprensión.

**TABLA IV.27 CASO DE USO GENERAL RESULTADOS DE LA
EVALUACIÓN A UN DOCENTE**

Id. de C.U.:	C.U. EVALUACIÓN _ DOCENTE_general	
Nombre C.U:	Resultados de la evaluación docente	
Actores:	Docente(iniciador)	
Propósito:	Mostrar los resultados obtenidos en el proceso de evaluación docente por parte de los estudiantes así como en la autoevaluación.	
Visión General:	<p>El docente ingresa al sistema con su respectiva autenticación y se mostrará el resultado obtenido por la evaluación de sus estudiantes, también se muestra los resultados obtenidos en la autoevaluación.</p> <p>El docente requiere observar los detalles del cuestionario de la evaluación por parte de los estudiantes donde se mostrará la carrera en las que imparte clase el docente con su respectivo promedio.</p> <p>Para observar una materia determinada el docente seleccionará primero la carrera a la que pertenece. De la misma manera para observar la autoevaluación.</p>	
Tipo:	Primario y Esencial	
Referencia:	R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12,	
Curso Critico de Eventos		
ACTOR		SISTEMA
1. Docente ingresa al sistema SIEDD	2. Presenta opciones.	
2. Docente selecciona al módulo de estadísticas de la ESPOCH	3. Solicita ingresar cédula y contraseña del Docente	
4. Ingresar cédula y contraseña	5. Valida datos y despliega promedio de evaluación por estudiantes, Autoevaluación y promedio Total.	
6. Selecciona Evaluación por Estudiantes.	7. Despliega Asignatura indicando la carrera a la que pertenece con su respectivo promedio.	
8. Selecciona Materia	9. Despliega resultados por pregunta y por grupos con su respectivo promedio.	
10. Regresa a menú principal		
11. Selecciona Autoevaluación Docente.	12. Despliega Carreras en las que docente imparte clases con su respectivo promedio.	
13. Selecciona una Carrera.	14. Despliega resultados por pregunta y por grupos con su respectivo promedio.	
15. Cerrar Sesión.		
Cursos Alternativos		
4. Cédula o contraseña inválida se indica error y se cancela la operación.		

**TABLA IV.28 CASO DE USO GENERAL RESULTADOS DE LA
EVALUACIÓN A UN NIVEL INSTITUCIONAL.**

Id. de C.U.:	C.U. EVALUACIÓN _ DOCENTE_institucional
Nombre C.U:	Resultados de la evaluación docente a nivel institucional
Actores:	Usuario General(iniciador)
Propósito:	Mostrar los resultados obtenidos en el proceso de evaluación docente a nivel institucional.
Visión General:	El usuario general seleccionara resultados institucionales donde se presentará resultados por facultad calculados en base al promedio de evaluación por los estudiantes y a la autoevaluación docente de igual forma por escuela y carrera hasta mostrar el cuestionario de dicha carrera con el promedio de la respectiva pregunta.
Tipo:	Primario y Esencial
Referencia:	R13, R14, R15, R16, R17.
Curso Critico de Eventos	
ACTOR	SISTEMA
1. Usuario General ingresa al sistema SIEDD	2. Presenta opciones
3. Selecciona Evaluación Institucional	4. Presenta promedios totales en autoevaluación, evaluación por estudiantes y promedio total por facultad.
5. Selecciona Facultad.	6. Despliega Escuelas con promedio de autoevaluación, evaluación por estudiante y promedio total.
7. Selecciona una Escuela.	8. Despliega Carreras con promedio de autoevaluación, evaluación por estudiante y promedio total.
9. Selecciona Carrera	10. Despliega Carreras con promedio de autoevaluación, evaluación por estudiante y promedio total.
11. Selecciona Autoevaluación de Carrera.	12. Despliega Cuestionario de autoevaluación con promedio por preguntas de la carrera.
13. Selecciona Evaluación por Estudiantes.	14. Despliega Cuestionario de evaluación de estudiantes con promedio por preguntas de la carrera.

4.5 MODELO CONCEPTUAL

Para definir el modelo conceptual tomaremos en cuenta el sistema de evaluación del desempeño docente, enfatizando el módulo de estadísticas.

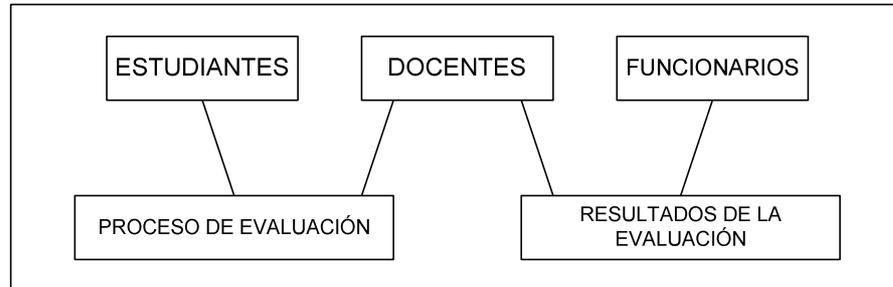


Figura IV.6 Modelo conceptual del módulo de estadísticas SIEDD

4.6 DEFINIR BORRADOR DE LA ARQUITECTURA DEL SISTEMA

Puesto que las estadísticas del sistema SIEDD son un módulo que compone todo el sistema damos a conocer la arquitectura del sistema tomando en cuenta la inclusión del módulo estadístico.

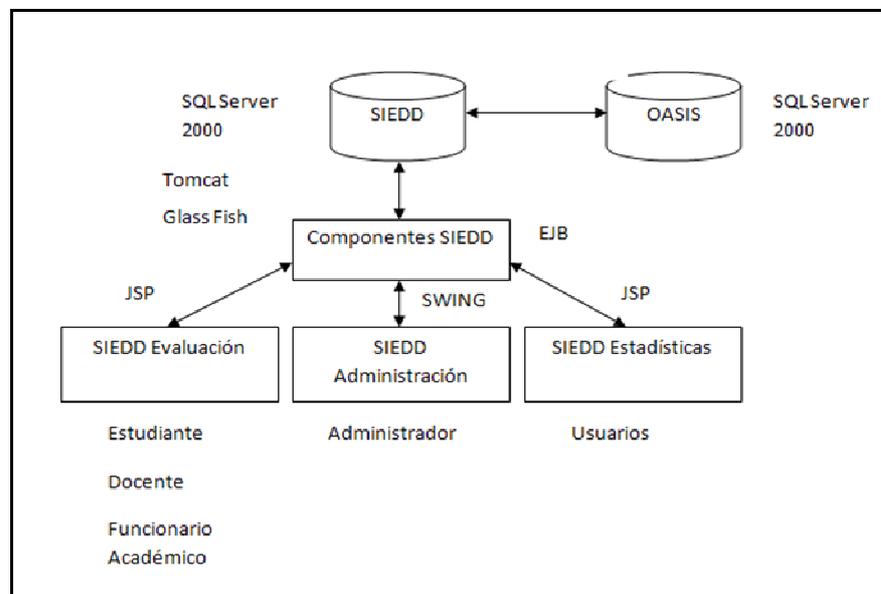


Figura IV.7 Arquitectura del Sistema SIEDD

ELABORACIÓN

4.7 DEFINIR CASOS DE USO ESENCIALES EN FORMATO EXPANDIDO

A continuación se definen de forma general los casos de uso existentes en el módulo de Estadísticas de la ESPOCH.

Cabe recalcar que los siguientes casos de uso se han realizado tomando en cuenta que el módulo de administración ha realizado el cierre del periodo de evaluación y que el módulo de estadísticas posee datos con los cuales puede trabajar.

Caso de Uso General Resultados de la evaluación a un Docente

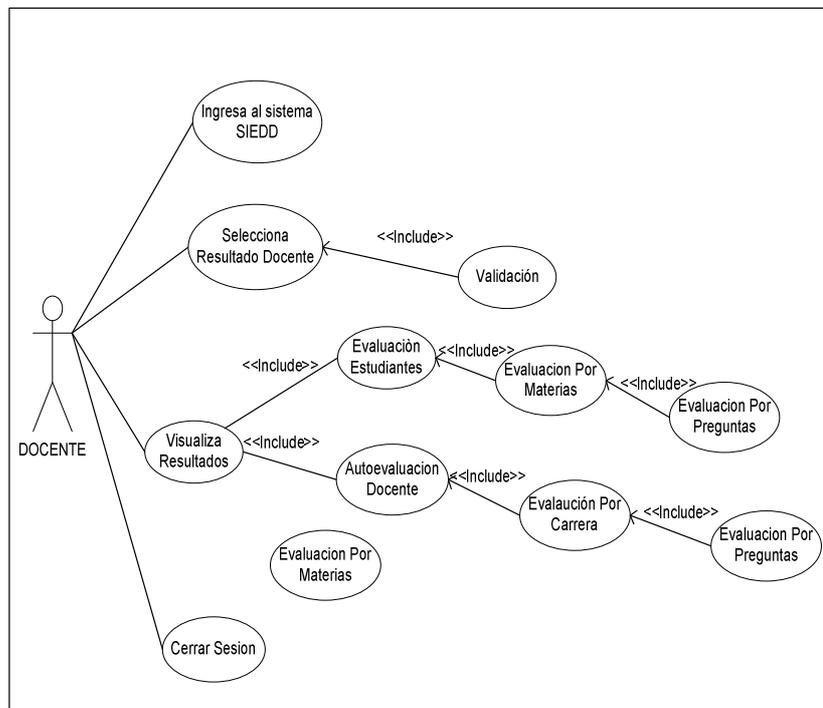


Figura IV.8 Caso de Uso General Resultados de la evaluación a un Docente

Caso de Uso General Resultados de la evaluación a un nivel Institucional.

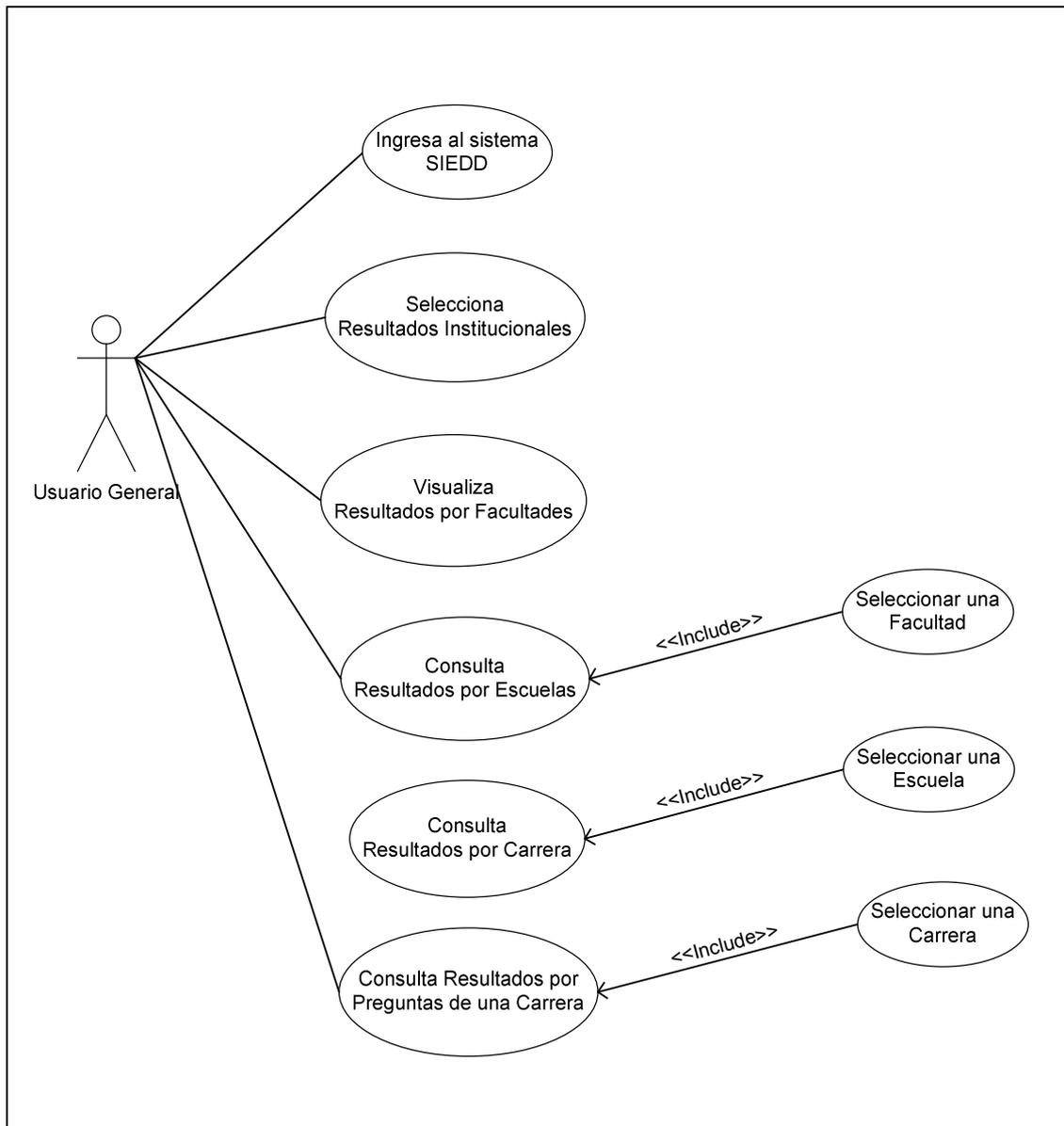


Figura IV.9 Caso de Uso General Resultados de la evaluación a un nivel Institucional.

4.8 REFINAR DIAGRAMAS DE CASOS DE USO

Para definir de forma precisa las acciones del módulo de estadísticas del sistema SIEDD se dará a conocer como se presentará los resultados según las opciones que el usuario seleccione.

RESULTADOS PARA LOS DOCENTES

RESULTADOS DE LA EVALUACIÓN ESTUDIANTE DOCENTE

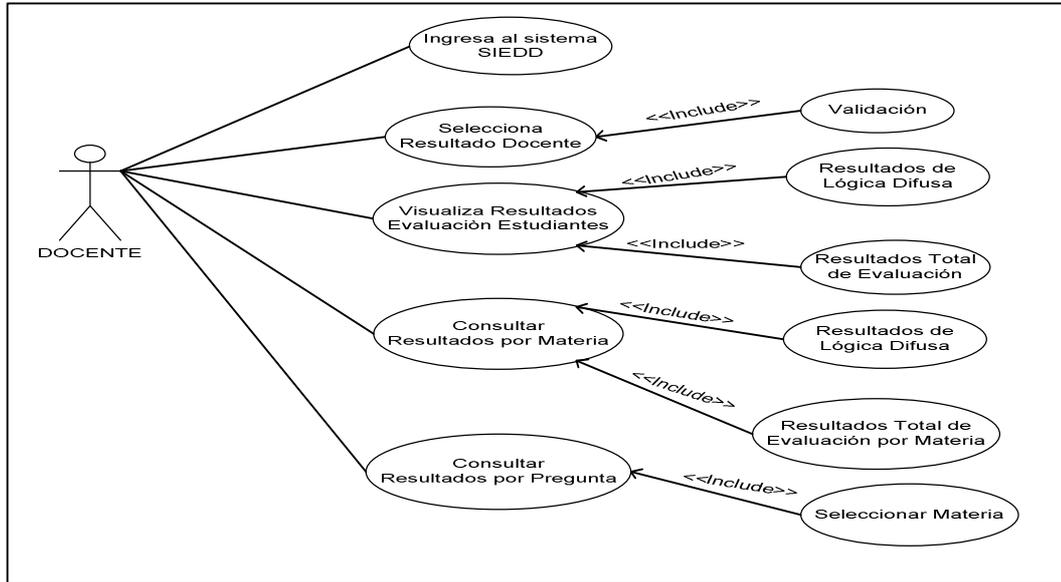


Figura IV.10 Resultados De La Evaluación Estudiante Docente

RESULTADOS DE LA AUTOEVALUACIÓN DOCENTE

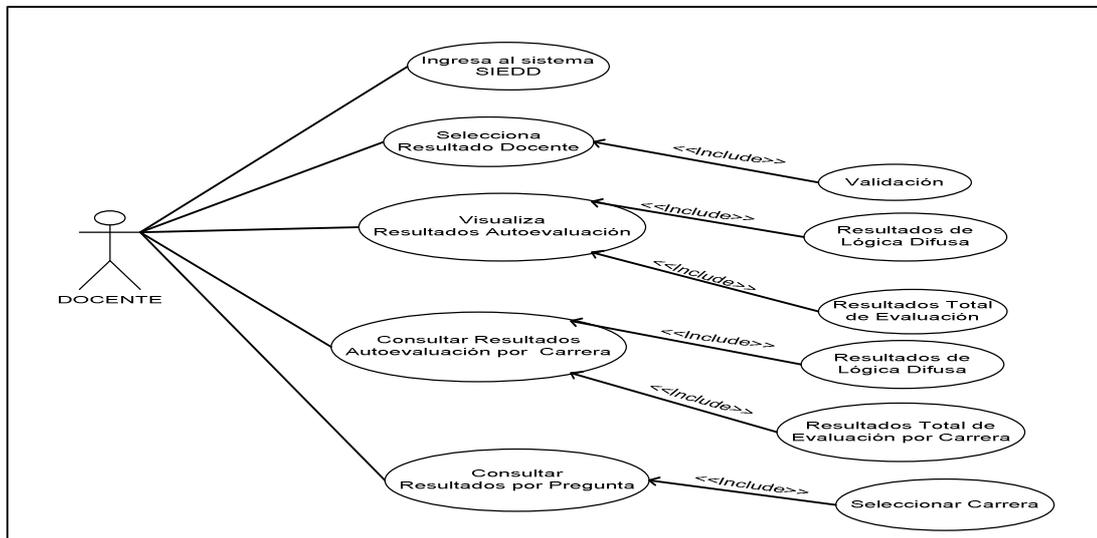


Figura IV.11 Resultados De La Autoevaluación Docente

RESULTADOS INSTITUCIONALES

POR FACULTAD

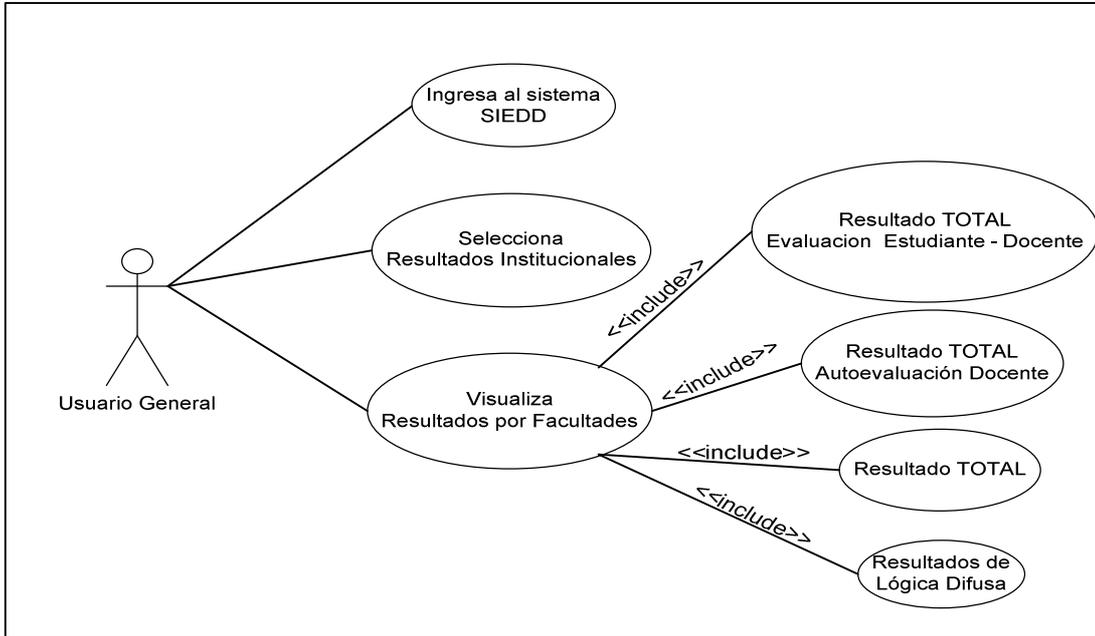


Figura IV.12 Caso de uso Resultado por Facultad

POR ESCUELA

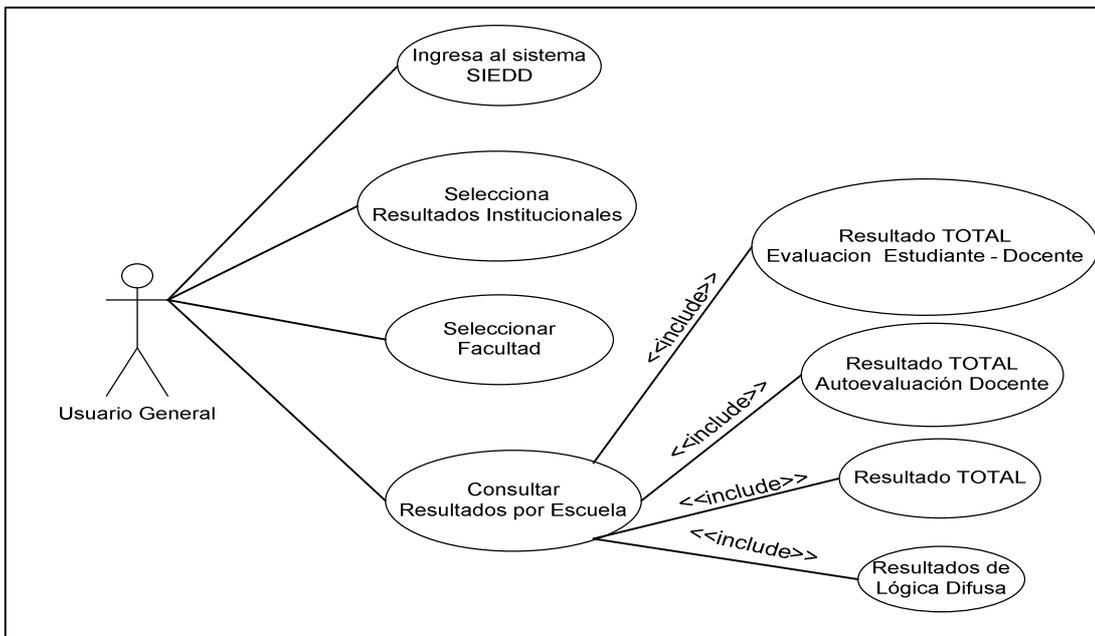


Figura IV.13 Caso de uso resultados por Escuela

POR CARRERA

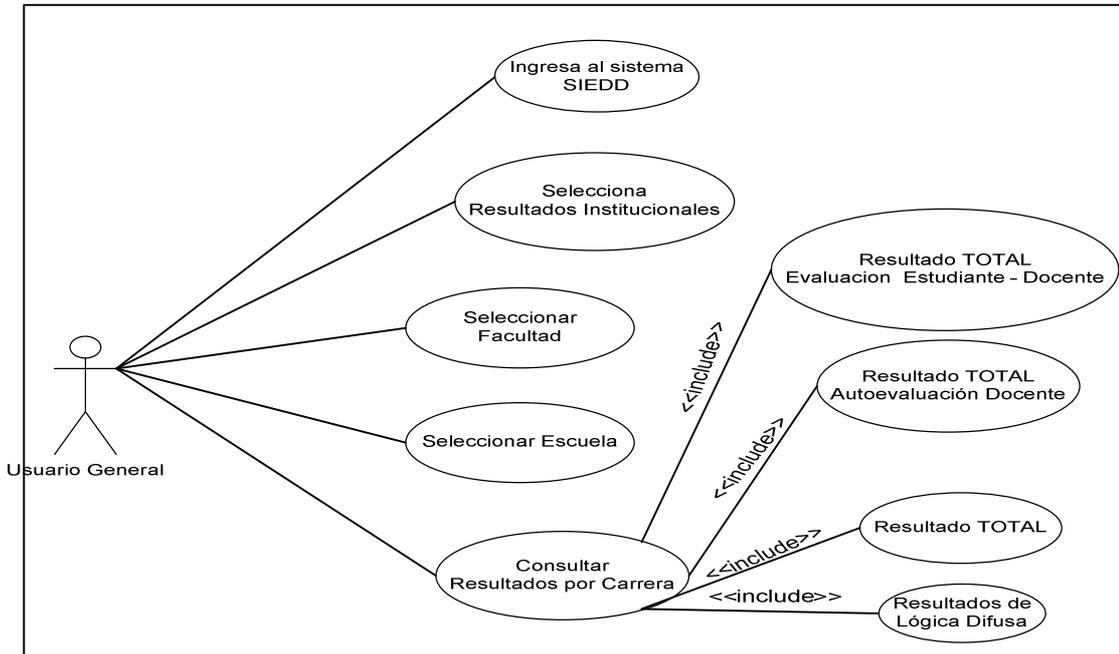


Figura IV.14 Caso de uso resultados por Carrera.

POR PREGUNTA DE LA EVALUACIÓN DE UN ESTUDIANTE A UN DOCENTE EN UNA CARRERA

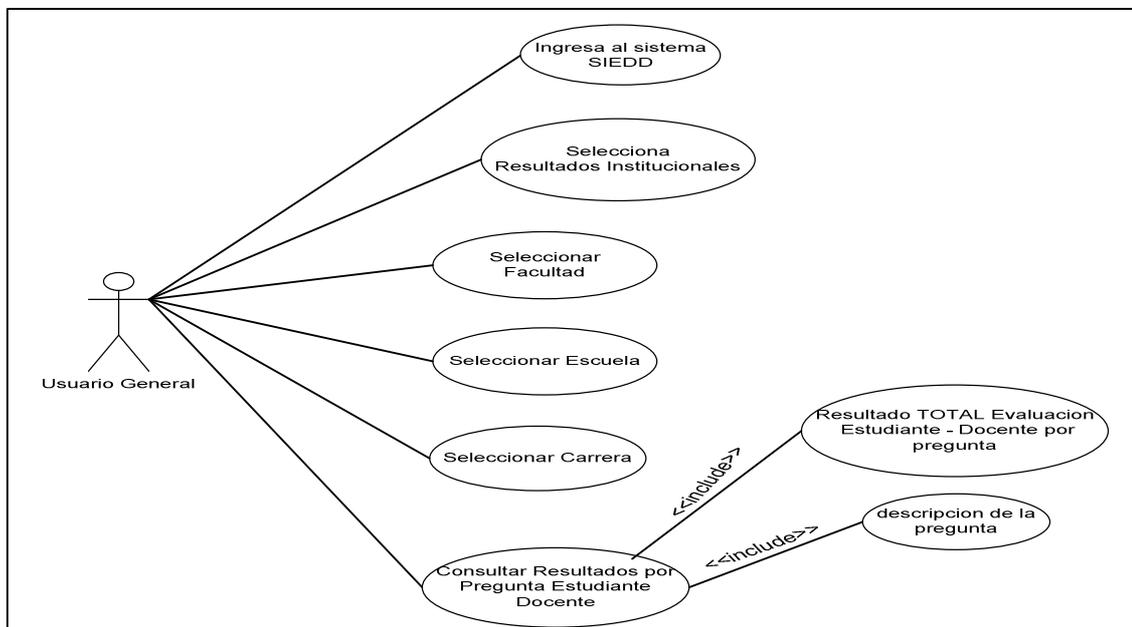


Figura IV.15 Caso de uso resultado por preguntas de una Carrera

POR PREGUNTA DE LA AUTOEVALUACIÓN DOCENTE EN UNA CARRERA

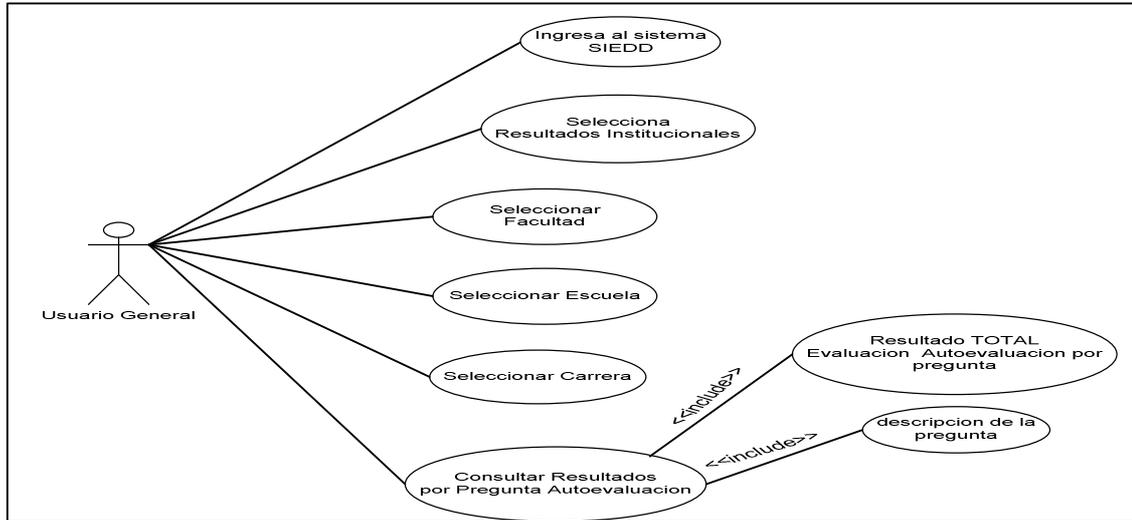


Figura IV.16 Caso de uso resultados por pregunta de un carrera

4.9 REFINAR EL MODELO CONCEPTUAL

Basándonos en el modelo conceptual construido en la fase anterior podemos determinar algunos nuevos aspectos que el sistema propuesto contemplará.

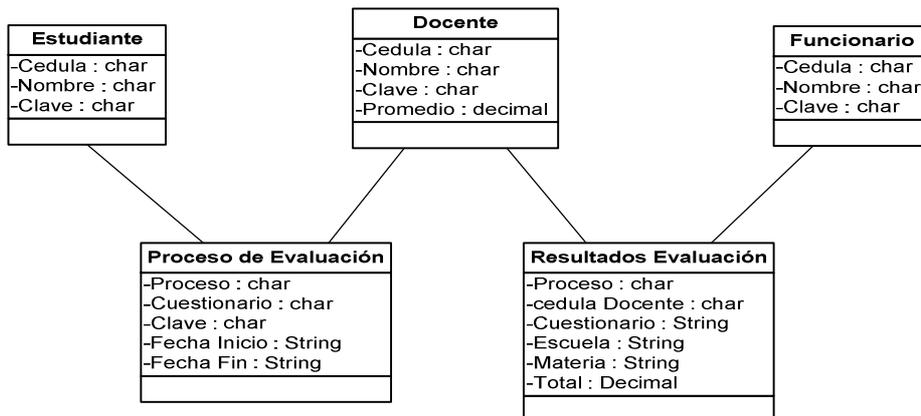


Figura IV.17 Modelo conceptual

4.10 DEFINIR CONTRATOS DE OPERACIÓN

TABLA IV.29 CONTRATO DE OPERACIÓN VALIDARSE

Nombre	Validarse(Cédula, contraseña)
Responsabilidades	Verificar que los datos por el usuario se correctos para acceder al uso del módulo de estadísticas del sistema SIEDD.
Tipo	Operación del Sistema
Referencias cruzadas	Validación
Notas	Algoritmo: 1.-ingresar datos 2.-buscar datos en la base de datos de SIEDD 3.-habilitar servicio. 4.-fin
Excepciones	Si los datos ingresados fueron incorrectos, volver a ingresar datos
Salidas	Acceso al módulo de estadísticas activado.
Pre-condiciones	Solicitar servicio de Internet
Post-condiciones	Navegar en Internet

TABLA IV.30 CONTRATO DE OPERACIÓN CONSULTAR RESULTADOS

Nombre	Consultar Resultados Docente (Código o cedula)
Responsabilidades	Proporcionar a los usuarios los datos requeridos por una consulta.
Tipo	Operación del Sistema
Referencias cruzadas	R1, R2, R3, R4, R5 ,R6, R7, R8 ,R9, R10, R11 ,R12. Caso de Uso General Resultados de la evaluación a un Docente
Notas	Algoritmo: 1.- Ingresar Resultados Docentes del sistema SIEDD 2.- Validarse. 4.- Visualizar detalles de consulta 5.- Fin
Excepciones	3.- Si los parámetros ingresados no corresponde a ningún usuario cancelar operación.
Salidas	Detalles de la consulta.
Pre-condiciones	Validar usuario
Post-condiciones	Si desea detalles de los resultados de la evaluación por estudiantes deberá seleccionar evaluación por estudiantes, y de la misma forma para detalles de la autoevaluación.

TABLA IV.31 CONTRATO DE OPERACIÓN CONSULTAR RESULTADOS INSTITUCIONALES

Nombre	Consultar resultados Institucionales
Responsabilidades	Proporciona al usuario general datos de la autoevaluación y la evaluación docente.
Tipo	Operación del Sistema
Referencias cruzadas	R13, R14, R15, R16, R17. Caso de Uso General Resultados de la evaluación a un nivel Institucional.
Notas	Algoritmo: 1.- Ingresar Resultados Docentes del sistema SIEDD 4.- Visualizar detalles de consulta por facultades 5.- Fin
Salidas	Resultados de la evaluación por facultades.
Post-condiciones	Si desea consultar valores por escuela deberá seleccionar una facultad. Si desea consultar valores por carrera deberá seleccionar una carrera.

4.11 DEFINIR LOS DIAGRAMAS DE ESTADO

Diagramas de estado usuario DOCENTE

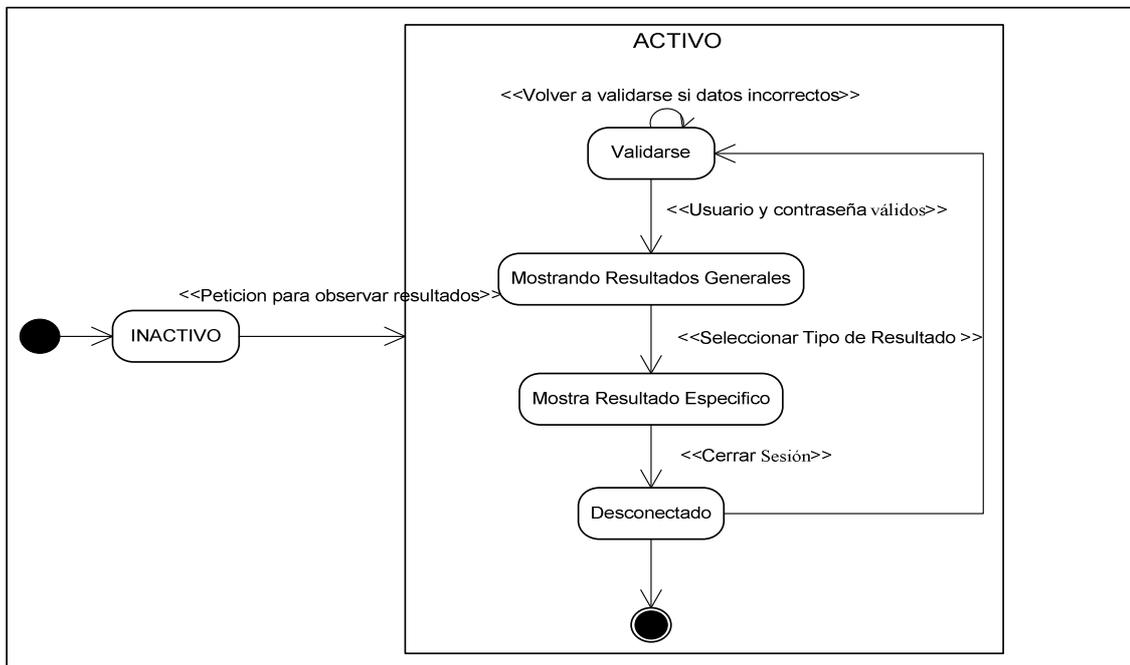


Figura IV.18 Diagrama de Estado usuario Docente

Diagrama de estado usuario GENERAL

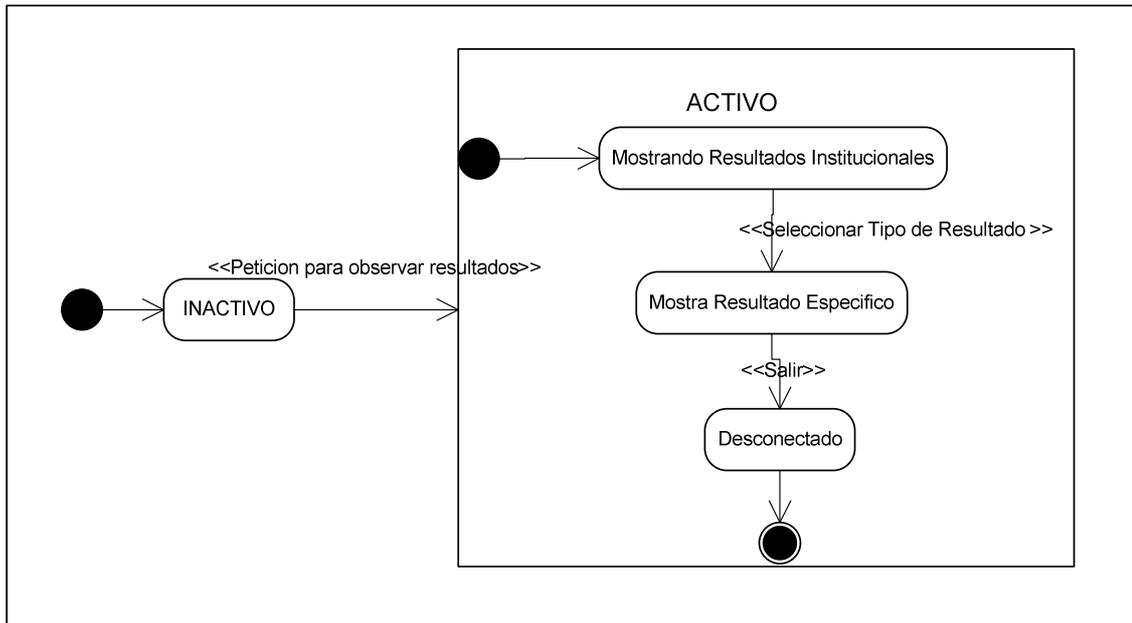


Figura IV.19 Diagrama de Estado Usuario General

4.12 DEFINIR LOS DIAGRAMAS DE ACTIVIDAD

Diagrama de Actividad DOCENTE

Es siguiente diagrama de actividad muestra las operaciones realizadas por el docente en cuanto a las evaluaciones realizadas por los estudiantes.

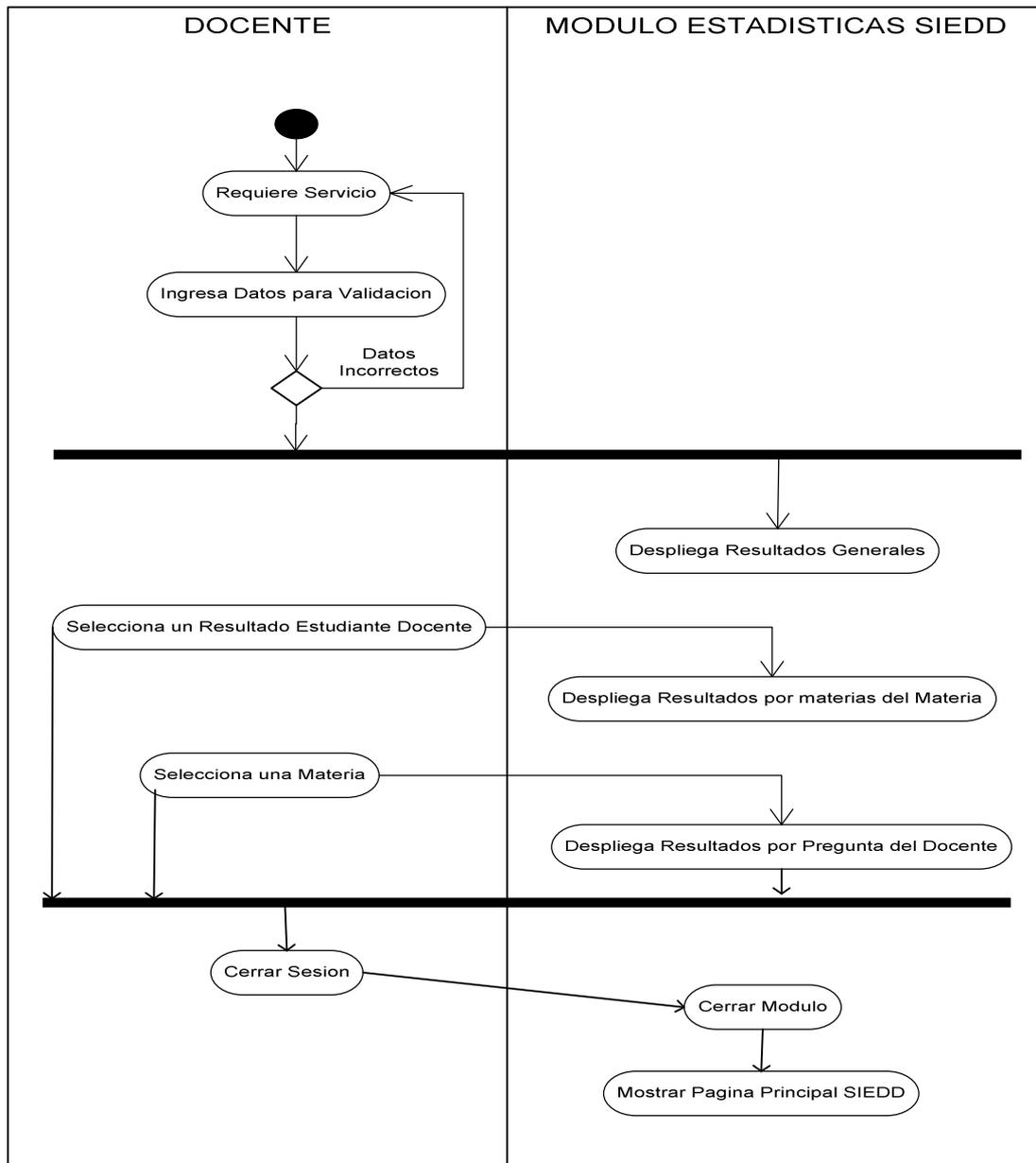


Figura IV. 20 Diagrama De Actividades Evaluación Estudiantes

Es siguiente diagrama de actividad muestra las operaciones realizadas por el docente en cuanto a las autoevaluaciones.

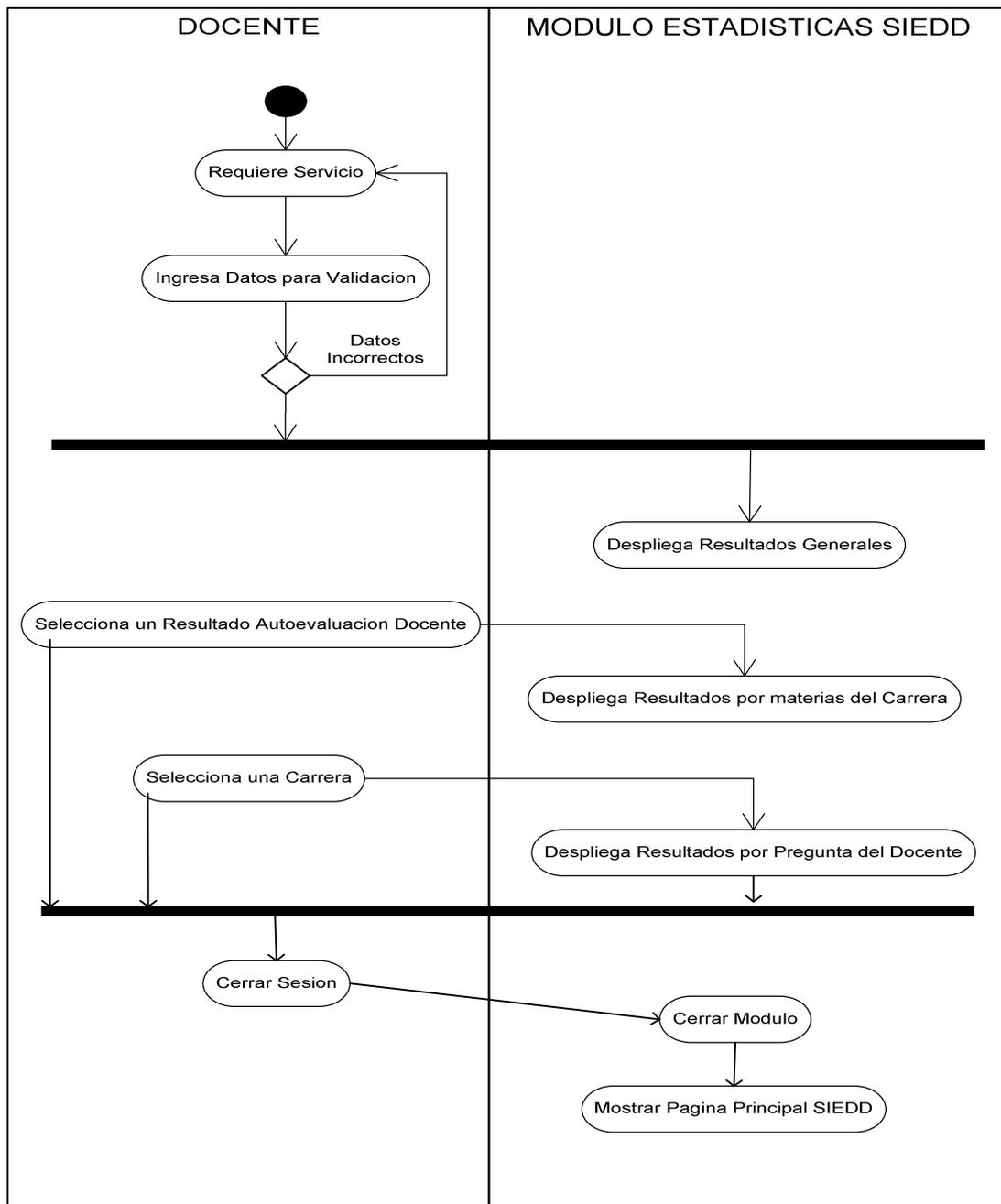


Figura IV.21 Diagrama de Actividades Autoevaluaciones

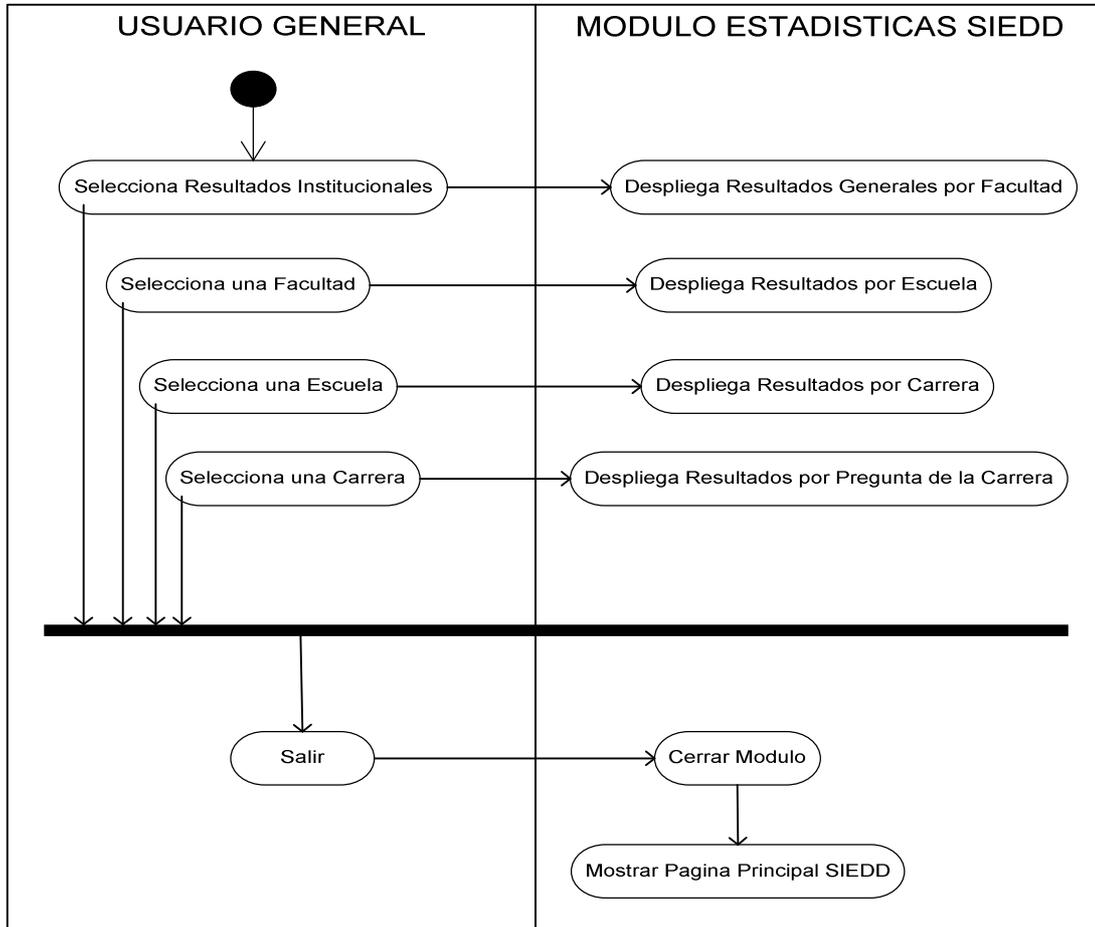
Diagrama de Actividad USUARIO GENERAL

Figura IV.22 Diagrama de actividades Usuario General.

CONSTRUCCION**4.13 DEFINIR CASOS DE USO REALES****TABLA IV.32 RESULTADOS DE LA EVALUACIÓN ESTUDIANTE DOCENTE**

Id. de C.U.:	C.U. EVALUACIÓN _ DOCENTE _ ESTUDIANTE	
Nombre C.U:	Resultados de la evaluación docente	
Actores:	Docente(iniciador)	
Propósito:	Mostrar los resultados obtenidos en el proceso de evaluación docente por parte de los estudiantes.	
Visión General:	El docente ingresa al sistema con su respectiva autenticación y se mostrará el resultado obtenido por la evaluación de sus estudiantes, Para observar una materia determinada el docente seleccionará primero la carrera a la que pertenece. De la misma manera para observar la autoevaluación.	
Tipo:	Primario y Esencial	
Referencia:	R1, R2, R3, R4, R5, R6.	
Curso Critico de Eventos		
ACTOR	SISTEMA	
1. Docente ingresa al sistema SIEDD	2. Presenta opciones.	
3. Docente selecciona al módulo de estadísticas de la ESPOCH	4.Solicita ingresar cédula y contraseña del Docente	
5. Ingresa cédula y contraseña	6. Valida datos y despliega promedio de evaluación por estudiantes, Autoevaluación y promedio Total y resultado difuso.	
7. Selecciona Evaluación por Estudiantes.	8. Despliega Asignatura indicando la carrera a la que pertenece con su respectivo promedio y resultado difuso.	
9. Selecciona Materia	10. Despliega resultados por pregunta y por grupos con su respectivo promedio.	
Cursos Alternativos		
4. Cédula o contraseña inválida se indica error y se cancela la operación.		

TABLA IV.33 RESULTADOS DE LA AUTOEVALUACIÓN DOCENTE

Id. de C.U.:	C.U. EVALUACIÓN _ DOCENTE_AUTOEVALUACION	
Nombre C.U.:	Resultados de la evaluación docente	
Actores:	Docente(iniciador)	
Propósito:	Mostrar los resultados obtenidos en el proceso de evaluación docente en la autoevaluación.	
Visión General:	El docente ingresa al sistema con su respectiva autenticación y se mostrará el resultado obtenido en la autoevaluación. El docente requiere observar los detalles del cuestionario de la autoevaluación donde el docente seleccionará primero la carrera a la que pertenece. De la misma manera para observar la autoevaluación.	
Tipo:	Primario y Esencial	
Referencia:	R7, R8, R9, R10, R11, R12,	
Curso Critico de Eventos		
ACTOR		SISTEMA
1.Docente ingresa al sistema SIEDD		2. Presenta opciones.
3.Docente selecciona al módulo de estadísticas de la ESPOCH		4. Solicita ingresar cédula y contraseña del Docente
5.Ingresa cédula y contraseña		6. Valida datos y despliega promedio de evaluación por estudiantes, Autoevaluación y promedio Total.
7. Selecciona Autoevaluación Docente.		8. Despliega Carreras en las que docente imparte clases con su respectivo promedio.
9. Selecciona una Carrera.		10. Despliega resultados por pregunta y por grupos con su respectivo promedio y resultado difuso.
11. Cerrar Cesión.		
Cursos Alternativos		
4. Cédula o contraseña inválida se indica error y se cancela la operación.		

TABLA IV.34 RESULTADOS INSTITUCIONALES POR FACULTAD

Id. de C.U.:	C.U. EVALUACIÓN _ DOCENTE_institucional	
Nombre C.U.:	Resultados de la evaluación docente a nivel institucional	
Actores:	Usuario General(iniciador)	
Propósito:	Mostrar los resultados obtenidos en el proceso de evaluación docente a nivel institucional.	
Visión General:	El usuario general seleccionara resultados institucionales donde se presentará resultados por facultad calculados en base al promedio de evaluación por los estudiantes y a la autoevaluación docente de igual.	
Tipo:	Primario y Esencial	
Referencia:	R13, R14, R15, R16, R17.	
Curso Critico de Eventos		
ACTOR		SISTEMA
1. Usuario General ingresa al sistema SIEDD		2. Presenta opciones
3. Selecciona Evaluación Institucional		4. Presenta promedios totales en autoevaluación, evaluación por estudiantes y promedio total por facultad, resultado difuso.

TABLA IV.35 RESULTADOS INSTITUCIONALES POR ESCUELA

Id. de C.U.:	C.U. EVALUACIÓN _ DOCENTE_institucional	
Nombre C.U.:	Resultados de la evaluación docente a nivel institucional	
Actores:	Usuario General(iniciador)	
Propósito:	Mostrar los resultados obtenidos en el proceso de evaluación docente a nivel institucional.	
Visión General:	El usuario general seleccionara resultados institucionales donde se presentará resultados por facultad calculados en base al promedio de evaluación por los estudiantes y a la autoevaluación docente de igual forma por escuela y carrera hasta mostrar el cuestionario de dicha carrera con el promedio de la respectiva pregunta.	
Tipo:	Primario y Esencial	
Referencia:	R13, R14, R15, R16, R17.	
Curso Critico de Eventos		
ACTOR		SISTEMA
1. Usuario General ingresa al sistema SIEDD		2. Presenta opciones
3. Selecciona Evaluación		4. Presenta promedios totales en

Institucional	autoevaluación, evaluación por estudiantes y promedio total por facultad, resultado difuso.
5. Selecciona Facultad.	6. Despliega Escuelas con promedio de autoevaluación, evaluación por estudiante y promedio total, resultado difuso.
7. Selecciona una Escuela.	8. Despliega Carreras con promedio de autoevaluación, evaluación por estudiante y promedio total, resultado difuso.

TABLA IV.36 RESULTADOS INSTITUCIONALES POR CARRERA

Id. de C.U.:	C.U. EVALUACIÓN _DOCENTE_institucional	
Nombre C.U:	Resultados de la evaluación docente a nivel institucional	
Actores:	Usuario General(iniciador)	
Propósito:	Mostrar los resultados obtenidos en el proceso de evaluación docente a nivel institucional.	
Visión General:	El usuario general seleccionara resultados institucionales donde se presentará resultados por facultad calculados en base al promedio de evaluación por los estudiantes y a la autoevaluación docente de igual forma por escuela y carrera hasta mostrar el cuestionario de dicha carrera con el promedio de la respectiva pregunta.	
Tipo:	Primario y Esencial	
Referencia:	R13, R14, R15, R16, R17.	
Curso Critico de Eventos		
ACTOR	SISTEMA	
1. Usuario General ingresa al sistema SIEDD	2. Presenta opciones	
3. Selecciona Evaluación Institucional	4. Presenta promedios totales en autoevaluación, evaluación por estudiantes y promedio total por facultad, resultado difuso.	
5. Selecciona Facultad.	6. Despliega Escuelas con promedio de autoevaluación, evaluación por estudiante y promedio total, resultado difuso.	
7. Selecciona una Escuela.	8. Despliega Carreras con promedio de autoevaluación, evaluación por estudiante y promedio total, resultado difuso.	
9. Selecciona Carrera	10. Despliega Carreras con promedio de autoevaluación, evaluación por estudiante y promedio total, resultado difuso.	

TABLA IV.37 RESULTADOS INSTITUCIONALES POR PREGUNTA DE UNA CARRERA

Id. de C.U.:	C.U. EVALUACIÓN _ DOCENTE_institucional	
Nombre C.U.:	Resultados de la evaluación docente a nivel institucional	
Actores:	Usuario General(iniciador)	
Propósito:	Mostrar los resultados obtenidos en el proceso de evaluación docente a nivel institucional.	
Visión General:	El usuario general seleccionara resultados institucionales donde se presentará resultados por facultad calculados en base al promedio de evaluación por los estudiantes y a la autoevaluación docente de igual forma por escuela y carrera hasta mostrar el cuestionario de dicha carrera con el promedio de la respectiva pregunta.	
Tipo:	Primario y Esencial	
Referencia:	R13, R14, R15, R16, R17.	
Curso Critico de Eventos		
ACTOR		SISTEMA
1. Usuario General ingresa al sistema SIEDD		2. Presenta opciones
3. Selecciona Evaluación Institucional		4. Presenta promedios totales en autoevaluación, evaluación por estudiantes y promedio total por facultad.
5. Selecciona Facultad.		6. Despliega Escuelas con promedio de autoevaluación, evaluación por estudiante y promedio total.
7. Selecciona una Escuela.		8. Despliega Carreras con promedio de autoevaluación, evaluación por estudiante y promedio total.
9. Selecciona Carrera		10. Despliega Carreras con promedio de autoevaluación, evaluación por estudiante y promedio total.
11. Selecciona Autoevaluación de Carrera.		12. Despliega Cuestionario de autoevaluación con promedio por preguntas de la carrera.
13. Selecciona Evaluación por Estudiantes.		14. Despliega Cuestionario de evaluación de estudiantes con promedio por preguntas de la carrera.

4.14 DEFINIR INFORMES DE INTERFÁZ DE USUARIO

La interfaz que se presente en el sistema SIEDD a nivel general para la interacción con el usuario es en forma de página web, por lo que el módulo de estadísticas presentará la misma interfaz, de forma que será necesario poseer un navegador o browser para el correcto funcionamiento de la misma.

4.15 DEFINIR DIAGRAMAS DE INTERACCIÓN

4.15.1 Definir Diagramas De Secuencia

En nuestro caso se identifican dos diagramas de secuencia debido a que se identifican 2 tipos de usuarios bien definidos.

Usuario Docente

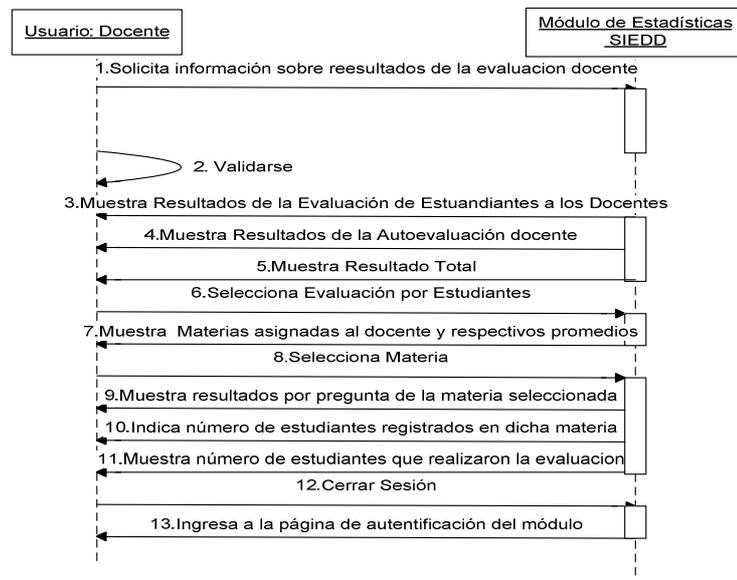


Figura IV.23 Diagrama de secuencia Usuario Docente Evaluacion estudiantes

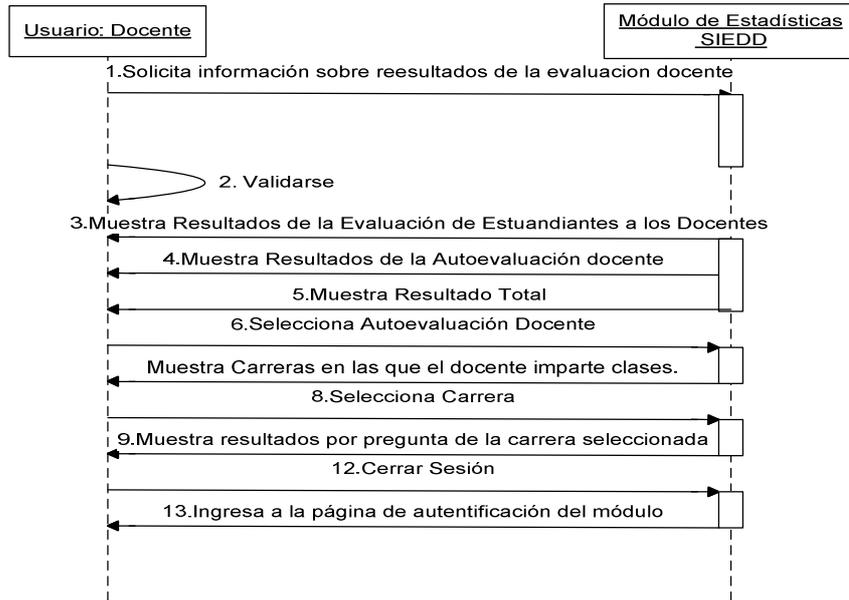


Figura IV.24 Diagrama de Secuencia Usuario Docente Autoevaluación **Usuario General**

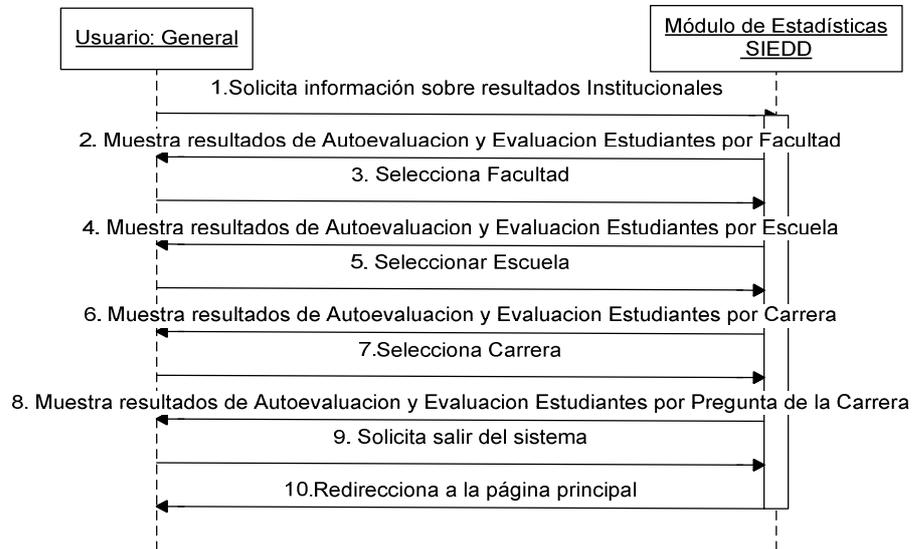


Figura IV.25 Diagrama de Secuencia Resultados institucionales

4.15.2 Definir Diagramas de Colaboración

Usuario Docente

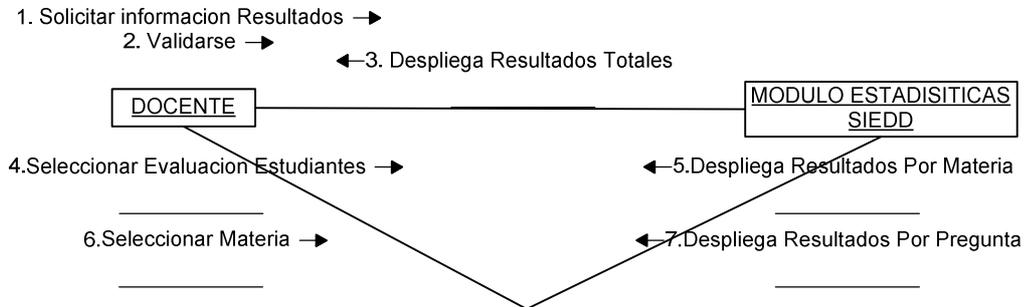


Figura IV.26 Diagrama de Colaboración Resultados evaluación estudiante

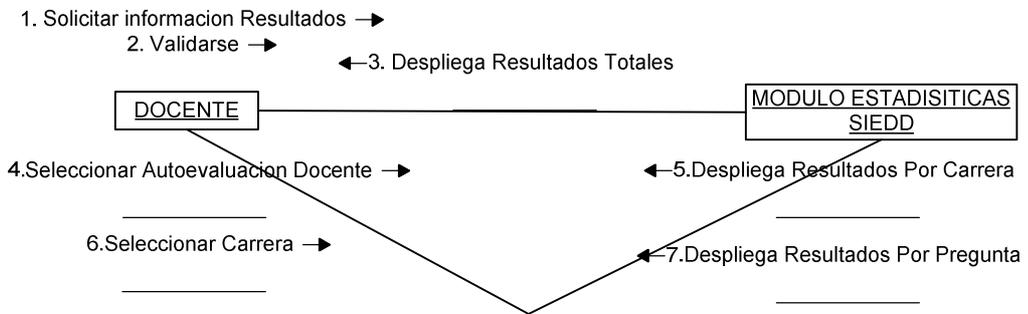


Figura IV.27 Diagrama de Colaboración Resultados Autoevaluación

Usuario General

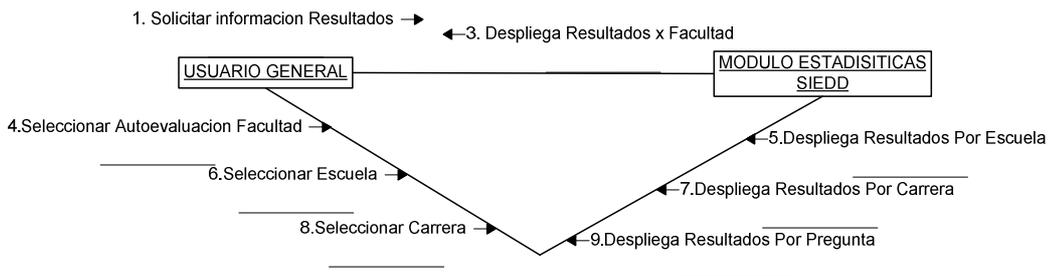


Figura IV.28 Diagrama de Colaboración Resultados institucionales

4.17.1 DICCIONARIO DE DATOS

A continuación mencionamos las tablas con los diferentes campos y una breve descripción de ellos para una mejor comprensión de la base de datos.

TABLA IV.38 DICCIONARIO DE DATOS

Tabla	Columna	Tipo de Dato	Descripción
CARRERA	CarCodigo	VarChar (10)	Código de la carrera
	FacCodigo	VarChar (10)	Facultad a la que pertenece la carrera
	EscCodigo	VarChar (10)	Escuela a la que pertenece la carrera
	CarDetalle	VarChar (100)	Nombre de la carrera
COMENTARIO_DOCENTE_DOCENTE	IdRespuestaDocente	BigInt	Código del comentario
	ComDetalle	VarChar (255)	Detalle del comentario
COMENTARIO_ESTUDIANTE_DOCENTE	IdRespuestaEstudianteDocente	BigInt	Código del Comentario
	ComDetalle	VarChar (255)	Detalle del Comentario
COMENTARIOGEN_DOCENTE_DOCENTE	DocDocId	BigInt	Código del docente
	ComDetalle	VarChar (250)	Detalle del comentario
COMENTARIOGEN_ESTUDIANTE_DOCENTE	EstDocId	BigInt	Código del comentario
	ComDetalle	VarChar (250)	Detalle del comentario
CUESTIONARIO	CueCodigo	VarChar (10)	Código del Cuestionario
	CueNombre	VarChar (50)	Nombre del Cuestionario
	CueDescripcion	VarChar (500)	Descripción del cuestionario
	CueTipo	TinyInt	Tipo de cuestionario
DOCENTE	DocCedula	Char (11)	Cedula del docente
	DocNombreCompleto	VarChar (200)	Nombre Completo del Docente
	DocClave	VarChar (10)	Contraseña del docente para el sistema
DOCENTE_DOCENTE	DocDocId	BigInt	Código de la tabla
	DocCedula	Char (11)	Cédula del docente

	DocRealizada	Bit	Determina si el docente ya a sido evaluado
	ProCodigo	VarChar (10)	Proceso en el que se encuentra evaluado
	FacCodigo	VarChar (10)	Facultad en la que se ha evaluado
	EscCodigo	VarChar (10)	Escuela en la que se ha evaluado
	CarCodigo	VarChar (10)	Carrera en la que se a ha evaluado
ESCUELA	FacCodigo	VarChar (10)	Facultad a la que pertenece la escuela
	EscCodigo	VarChar (10)	Código de la escuela
	EscDetalle	VarChar (100)	Nombre de la escuela
ESTADISTICAS_DOCENTE_DOCENTE	CodigoEstadistica	BigInt	Código para identificar la estadística
	PreCodigo	VarChar (5)	Código de la pregunta
	GruCodigo	VarChar (5)	Código del grupo al que pertenece la pregunta
	CueCodigo	VarChar (5)	Código del cuestionario al que pertenece la pregunta
	ProCodigo	VarChar (10)	Código del proceso de evaluación
	FacCodigo	VarChar (10)	Codigo de la facultad
	EscCodigo	VarChar (10)	codigo de la escuela
	CarCodigo	VarChar (10)	Código de la carrera
	CedDocente	Char (11)	Cedula del docente
	Total	Float	Total obtenido en la autoevaluación
ESTADISTICAS_ESTUDIANTE_DOCENTE	CodigoEstadistica	BigInt	Muestra el código principal de la estadística
	PreCodigo	VarChar (5)	Codigo de la pregunta
	GruCodigo	VarChar (5)	Codigo del grupo al que pertenece la pregunta
	CueCodigo	VarChar (5)	Codigo del cuestionario al que pertenece la pregunta
	ProCodigo	VarChar (10)	Codigo del proceso al que pertenece la pregunta
	FacCodigo	VarChar (10)	Codigo de la facultad
	EscCodigo	VarChar (10)	Codigo de Escuela
	CarCodigo	VarChar (10)	Código de la carrera

	NivCodigo	VarChar (10)	Código del Nivel
	MatCodigo	VarChar (10)	Código de la materia
	CedDocente	Char (11)	Cedula del docente evaluado
	Total	Float	Total obtenido en determinad pregunta
ESTUDIANTE	EstCedula	Char (11)	Cédula del estudiante
	EstNombreCompleto	VarChar (200)	Nombre completo del estudiante
	EstClave	VarChar (10)	Clave de acceso al sistema
ESTUDIANTE_DOCENTE	EstDocId	BigInt	Codigo de la relacion entre estudiante docente
	DocCedula	Char (11)	Cedula del docente
	EstCedula	Char (11)	Cedula del estudiante
	EstRealizada	Bit	Indicador si el estudiante realizó la evaluación al docente
	ProCodigo	VarChar (10)	Código del proceso de evaluación
	MatCodigo	VarChar (10)	Código de la Materia
	NivCodigo	VarChar (10)	Codigo del Nivel
	CarCodigo	VarChar (10)	Codigo de la carrera
	FacCodigo	VarChar (10)	Codigo de la Facultad
	EscCodigo	VarChar (10)	Codigo de la escuela
	MatParalelo	VarChar (1)	Paralelo en la que se dicta la materia.
FACULTAD FACULTAD	FacCodigo	VarChar (10)	Código de la facultad
	FacDetalle	VarChar (100)	Nombre de la facultad
GENERAL GENERAL	GenTipoEvaluacion	TinyInt	
	GenMuestra	Decimal (5,2)	
GRUPOPREGUNTA	CueCodigo	VarChar (10)	Código del cuestionario al que pertenece el grupo
	GruCodigo	VarChar (5)	Codigo del grupo
	GruDetalle	VarChar (50)	Nombre del grupo
	GruOrden	TinyInt	Orden del grupo de preguntas
MATERIA	MatCodigo	VarChar (10)	Código de la materia
	MatNombre	VarChar (50)	Nombre de la materia
	NivCodigo	VarChar (10)	Niveles en los que se dicta la materia
	CarCodigo	VarChar (10)	Carrera en la que se dicta dicha materia
	FacCodigo	VarChar (10)	Facultad en la que se dicta la materia
	EscCodigo	VarChar (10)	Escuela en la que se

			dicta la materia
MATERIA	MatParalelo	VarChar (1)	Paralelo en que se dicta la materia
	DocCedula	Char (11)	Cédula del docente que dicta la materia
OPCIONES	CueCodigo	VarChar (10)	Codigo del cuestionario
	GruCodigo	VarChar (5)	Codigo del grupo de preguntas
	PreCodigo	VarChar (5)	Codigo de la pregunta
	OpcCodigo	VarChar (5)	Codigo de la opción
	OpcDetalle	VarChar (100)	Detalle de la opción
	OpcOrden	TinyInt	Orden en el que se muestra
	OpcPonderacion	Decimal (8,4)	Valor que tomara la opcion para el proceso de cálculo
PREGUNTA	CueCodigo	VarChar (10)	Código del cuestionario
	GruCodigo	VarChar (5)	Código del Grupo
	PreCodigo	VarChar (5)	Código de la pregunta
	PreDetalle	VarChar (500)	Descripción de la pregunta
	PreTipo	TinyInt	Tipo de pregunta
	PreOrden	TinyInt	Orden de la pregunta
	PreObligatoria	Bit	Determina si la pregunta deberá o no ser contestada
	PreComentarios	Bit	Comentarios a la pregunta
PROCESOEVALUACION	ProCodigo	VarChar (10)	Código del proceso de evaluación
	CueCodigo	VarChar (10)	Código del Cuestionario
	CarCodigo	VarChar (10)	Codigo de la carrera que evaluará
	FacCodigo	VarChar (10)	Código de la facultad que evaluará
	EscCodigo	VarChar (10)	Codigo de la escuela que evaluará
PROCESOS	ProCodigo	VarChar (10)	Código del proceso de evaluación
	ProDetalle	VarChar (50)	Detalle del proceso de evaluación
	ProFechaInicio	SmallDateTime	Fecha de inicio del proceso de evaluación
	ProFechaFin	SmallDateTime	Fecha de finalización del proceso de

			evaluación
PROCESOS	ProVigente	Bit	Campo determinante del proceso de evaluación vigente
	ProFechaCierre	SmallDateTime	Fecha que indica el proceso de cierre de la evaluación docente
RESPUESTA_DOCENTE_DOCENTE	IdRespuestaDocente	BigInt	Código de la respuesta
	DocDocId	BigInt	Código de la Autoevaluación
	CueCodigo	VarChar (10)	Código del cuestionario
	GruCodigo	VarChar (5)	Código del grupo
	PreCodigo	VarChar (5)	Código de la pregunta
	OpcCodigo	VarChar (5)	Código de la opción a la que respondió el docente
RESPUESTA_ESTUDIANTE_DOCENTE	IdRespuestaEstudianteDocente	BigInt	Código de la respuesta
	EstDocId	BigInt	Código de la evaluación estudiante docente
	CueCodigo	VarChar (10)	Código del cuestionario
	GruCodigo	VarChar (5)	Código del grupo
	PreCodigo	VarChar (5)	Código de la pregunta
	OpcCodigo	VarChar (5)	Código de la opción
USUARIO	UsuCodigo	SmallInt	Código del usuario
	UsuUsuario	VarChar (20)	Nombre del usuario
	UsuClave	VarChar (20)	Clave de acceso al sistema
	UsuNombreCompleto	VarChar (100)	Nombre completo del usuario
	UsuTipo	TinyInt	Tipo de usuario
	UsuDependencia	VarChar (10)	Dependencia a la que pertenece al usuario

Fuente: Base de datos SIEDD

Elaborado Por: Elvia Mariela Cabay Huebla y Comisión De Evaluación Institucional.

4.18 MODELO FÍSICO Y ARQUITECTURA DEL SISTEMA

4.18.1 Diagrama de Componentes

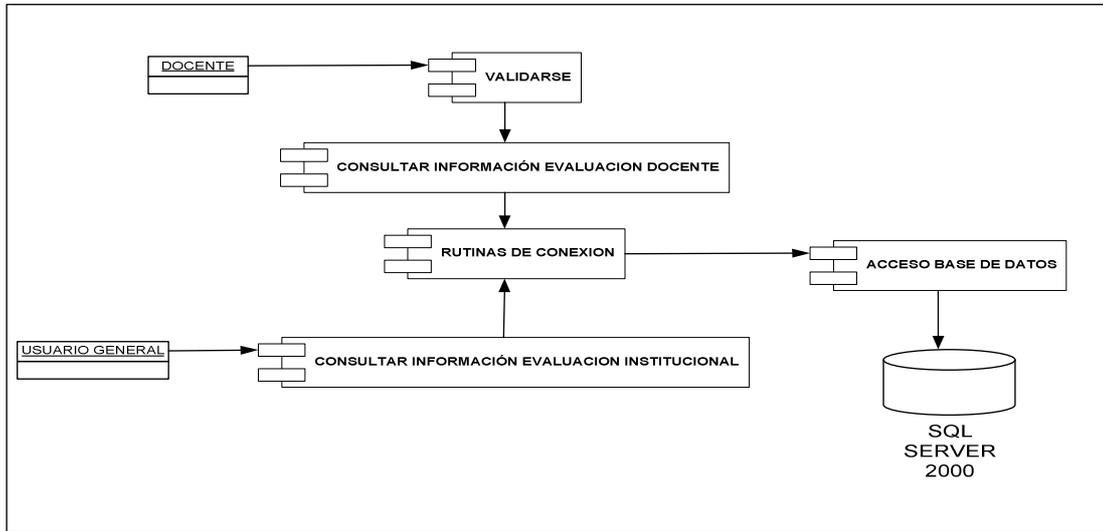


Figura IV.30 Diagrama de Componentes del Sistema

4.18.2 Diagrama de Despliegue

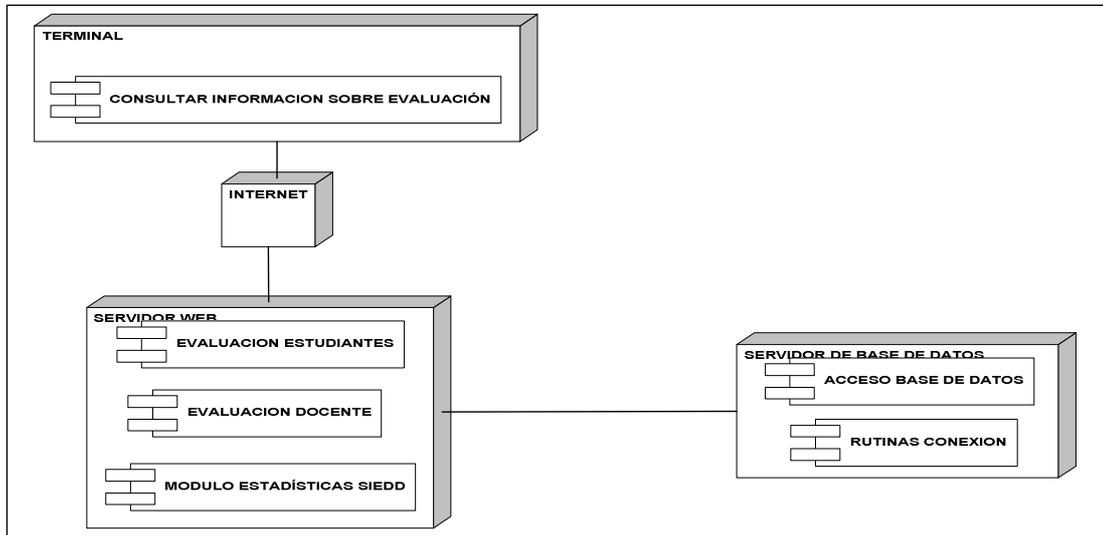


Figura IV.31 Diagrama de Despliegue del sistema

4.19 IMPLEMENTACIÓN Y PRUEBAS

4.19.1 Definición De Estándares De Programación

Para obtener una homogeneidad en cuanto a la programación del sistema se han definido los siguientes estándares de programación:

El nombre de las tablas de la base de datos se describirá en minúscula, en caso de que se componga de dos palabras la siguiente comenzará con mayúscula.

Los campos de las tablas tomará el mismo formato de denominación de las tablas mencionado anteriormente.

Para la interacción con la base de datos se especificará los parámetros necesarios para realizar la conexión como son

- **Driver:** es el controlador que permitirá la interacción con la base de datos creada específicamente en SQL server 2000.
- **Cadena de Conexión:** indica el driver que se usará, habilita el puerto 1433 y especifica el nombre de la base de datos.
- **Usuario:** Con el que se accederá a la base de datos.
- **Password:** La clave de acceso a la base de datos.

4.19.2 Pruebas Unitarias

Para verificar el correcto funcionamiento del sistema se han probado las clases y métodos de forma independiente enviando y consumiendo datos de los mismos y verificando que estos sean correctos.

Se identificó un docente para el cual se realizó el proceso de cálculo de la evaluación de forma manual y se comparó los datos obtenidos con los datos arrojados por el sistema.

4.19.3 Pruebas De Caja Negra

A continuación se presentará los casos de pruebas basándonos en la gestión de casos de prueba del módulo de estadísticas, mismos que se realizarán para los principales casos de usos del módulo, con el siguiente orden:

1. Descripción del Caso de Prueba

2. Condición de ejecución

- **Precondiciones**
- **Valores de entrada**
- **Observaciones**
- **Resultados esperados**
- **Postcondiciones**

1. Descripción del Caso de Prueba: “Validación de ingreso al módulo de estadísticas”.

Las pruebas realizadas a este caso de uso son:

- Ingreso de un docente al módulo de estadísticas.
- Datos incorrectos: Ingreso de número de cédula con dígitos mayores que 11.
- Datos incorrectos: Ingreso de número de cédula con dígitos menores que 11.
- Datos incorrectos: Ingreso de caracteres no permitidos.
- Datos incorrectos: Ingreso de contraseña errónea.

Para realizar el conjunto de pruebas partiremos de la interfaz de usuario del módulo de estadísticas de la epoch.

1.1 Condiciones de Ejecución

Ingresamos al sistema SIEDD y accedemos a la opción de resultados, donde no solicitará la identificación del docente con su respectiva cédula y contraseña.

Ingreso de un docente al módulo de estadísticas.

Para ingresar al módulo de estadísticas se deberá ingresar primero al sistema SIEDD donde se encontrará el enlace para dicho módulo.

Precondiciones

Para poder que un docente pueda ingresar deberá constar en el sistema académico debido a que el ingreso de los parámetros necesarios dependerá de este.

Se debe haber cerrado el proceso de evaluación para poder obtener los resultados de la evaluación.

Valores de Entrada

- Se deberá ingresar en número de cédula con guión en el campo usuario.
- Se deberá ingresar la clave que si no ha cambiado será la misma que se usa para el ingreso al sistema académico en el campo clave.
- Pulsamos el botón Aceptar.

Resultados esperados

Se despliega la pantalla identificando al docente con su nombre y los respectivos promedios obtenidos en las evaluaciones realizadas.

Postcondiciones

Se calcula promedios correspondientes al docente

Datos incorrectos: Ingreso de número de cédula con dígitos mayores que 11.

Para ingresar al módulo de estadísticas se deberá ingresar primero al sistema SIEDD donde se encontrará el enlace para dicho módulo.

Precondiciones

Para poder que un docente pueda ingresar deberá constar en el sistema académico debido a que el ingreso de los parámetros necesarios dependerá de este.

Valores de Entrada

- Se deberá ingresar en número de cédula con guión en el campo usuario.
- Se deberá ingresar la clave que si no ha cambiado será la misma que se usa para el ingreso al sistema académico en el campo clave.
- Pulsamos el botón Aceptar.

Resultados esperados

Se despliega la pantalla de login solicitando nuevamente el usuario y su clave.

Postcondiciones

No se realiza el cálculo de promedios para el docente.

Datos incorrectos: Ingreso de número de cédula con dígitos menores que 11.

Para ingresar al módulo de estadísticas se deberá ingresar primero al sistema SIEDD donde se encontrará el enlace para dicho módulo.

Precondiciones

Para poder que un docente pueda ingresar deberá constar en el sistema académico debido a que el ingreso de los parámetros necesarios dependerá de este.

Valores de Entrada

- Se deberá ingresar en número de cédula con guión en el campo usuario.
- Se deberá ingresar la clave que si no ha cambiado será la misma que se usa para el ingreso al sistema académico en el campo clave.
- Pulsamos el botón Aceptar.

Resultados esperados

Se despliega la pantalla de login solicitando nuevamente el usuario y su clave.

Postcondiciones

No se realiza el cálculo de promedios para el docente.

Datos incorrectos: Ingreso de caracteres no permitidos.

Para ingresar al módulo de estadísticas se deberá ingresar primero al sistema SIEDD donde se encontrará el enlace para dicho módulo.

Precondiciones

Para poder que un docente pueda ingresar deberá constar en el sistema académico debido a que el ingreso de los parámetros necesarios dependerá de este.

Valores de Entrada

- Se deberá ingresar en número de cédula con guión en el campo usuario.
- Se deberá ingresar la clave que si no ha cambiado será la misma que se usa para el ingreso al sistema académico en el campo clave.
- Pulsamos el botón Aceptar.

Resultados esperados

Se despliega la pantalla de login solicitando nuevamente el usuario y su clave.

Postcondiciones

No se realiza el cálculo de promedios para el docente.

2. Descripción del Caso de Prueba: “El módulo de estadísticas deberá mostrar el promedio total obtenido por un docente en la evaluación realizada por los estudiantes y autoevaluación docente.”

Las pruebas realizadas a este caso de uso son:

- Visualizar información con el periodo vigente.
- Visualizar información con un periodo diferente al vigente.

2.1 Condiciones de Ejecución

Visualizar información con el periodo vigente.

Precondiciones

Para poder visualizar esta información el docente deberá estar correctamente validado en el módulo de estadísticas.

Valores de Entrada

- Número de cédula del docente.
- Código del proceso de evaluación.

Resultados esperados

Promedio obtenido en la evaluación realizada por los estudiantes y la autoevaluación en el periodo vigente.

Postcondiciones

Se realiza el cálculo según el docente y el código del proceso.

Visualizar información con un periodo diferente al vigente.

Poder realizar esta prueba se cambiará códigos del proceso de evaluación en la base de datos de forma aleatoria del docente al que se someterá a la prueba.

Precondiciones

Para poder visualizar esta información el docente deberá estar correctamente validado en el módulo de estadísticas.

Valores de Entrada

- Número de cédula del docente.
- Código del proceso de evaluación.

Resultados esperados

Promedio obtenido en la evaluación realizada por los estudiantes y la autoevaluación solo en el proceso de evaluación vigente.

Postcondiciones

Se realiza el cálculo según el docente y el código del proceso vigente. Se descarta los demás códigos de proceso de evaluación.

TRANSICIÓN

4.20 IMPLANTACIÓN DEL SISTEMA

Para la implantación del sistema se nos proporciono un servidor en el que los distintos módulos fueron publicados para su funcionamiento.

Fue necesario tomar la dirección web del sistema de evaluación anterior debido a que los usuarios ya se encontraban identificados con la misma.

Para ingresar directamente al sistema se deberá colocar la siguiente dirección en el browser:

<http://evaluacion.esPOCH.edu.ec>

CAPITULO V

ANÁLISIS DE RESULTADOS

Para poder determinar la validez de la hipótesis planteada: *“El uso de la lógica difusa en un módulo del Sistema de Evaluación del Desempeño del Ejercicio Docente aumentará el número de parámetros analizados para la toma de decisiones”* se hará uso de la estadística inferencial y debido a que la hipótesis muestra la reacción de grupos ante métodos diferentes se aplicará el método **de Distribución Z** que nos permitirá realizar una correcta demostración de la hipótesis, tomando en cuenta que se puede analizar el sistema de evaluación anterior y el sistema de evaluación propuesto. Considerando este aspecto se ha realizado una encuesta que nos permitirá recolectar la información pertinente para poder realizar el proceso de validación de la hipótesis antes mencionada.

5.1 OPERACIONALIZACIÓN DE VARIABLES

5.1.2. OPERACIONALIZACIÓN CONCEPTUAL

VARIABLE	TIPO	CONCEPTO
La lógica difusa	Variable Independiente	La lógica difusa es una extensión de la lógica tradicional (Booleana) que utiliza conceptos de pertenencia de conjuntos, parecidos a la manera de pensar humana. Las proposiciones pueden ser representadas con grados de veracidad usando un lenguaje hablado.
Parámetros analizados para la toma de decisiones en el sistema de evaluación docente de la Epoch.	Variable Dependiente	Para evaluar los resultados obtenidos en el proceso de evaluación se analizan aspectos para determinar la calidad del docente haciendo referencia a que tan explícitos son los resultados, si son suficientes para concluir un resultado y como se muestran o que tan claros y entendibles son para la persona evaluada.

5.1.3. OPERACIONALIZACIÓN METODOLÓGICA

Variable	Categoría	Indicadores	Técnicas	Fuentes de Verificación
El uso de la lógica difusa en un módulo del Sistema de Evaluación del Desempeño del Ejercicio Docente	Utilidad	El uso de la lógica difusa	<ul style="list-style-type: none">• Observación• Encuestas.• Entrevistas	<ul style="list-style-type: none">• Manuales del Sistema de las aplicaciones implantadas en la ESPOCH.• Sistemas expertos desarrollados en la ESPOCH
Parámetros analizados para la toma de decisiones en el sistema de evaluación docente de la Epoch.	Calidad Docente	<ul style="list-style-type: none">• Cantidad de parámetros analizados para el Docente	<ul style="list-style-type: none">• Observación• Entrevistas• Encuestas	<ul style="list-style-type: none">• Docentes evaluados por el sistema de evaluación• Sistema de evaluación anterior.

5.2 ESTADÍSTICA INFERENCIAL

Para la demostración de la hipótesis se requieren de distribuciones que según el tamaño de la muestra o de la desviación que se conozca pueden variar.

En nuestro caso se utilizará la Distribución Z, que es aplicada en su mayoría de veces cuando el tamaño de la muestra es mayor que 30.

Para aplicar la distribución Z cuando se conoce 1 sola muestra se aplicará la siguiente fórmula:

$$Z = \frac{\bar{X} - \mu}{\sigma_{\bar{X}}}$$

$$\sigma_{\bar{X}} = \frac{\sigma}{\sqrt{n}}$$

Si se conoce 2 muestras

$$Z = \frac{\bar{X}_1 - \bar{X}_2}{S_{\bar{X}_1 - \bar{X}_2}}$$

$$S_{\bar{X}_1 - \bar{X}_2} = \sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}$$

\bar{X} = media muestral

μ = media Poblacional

$\sigma_{\bar{X}}$ = Error típico de la media

\bar{X}_1 = media aritmética de la primera muestra.

\bar{X}_2 = media aritmética de la segunda muestra.

S_1 = Varianza de la muestra 1

S_2 = Varianza de la muestra 1

5.2.1 SELECCIÓN Y DETERMINACIÓN DE LA MUESTRA

Para determinar la muestra necesaria se ha tomado en cuenta el cálculo del tamaño de la muestra cuando se conoce la población, tomando como población a los docentes de la escuela de ingeniería que han sido evaluados en semestres anteriores que son 34 por lo que planteando la fórmula se conseguirá lo siguiente:

$$n = \frac{N\sigma^2 Z^2}{(N-1)E^2 + \sigma^2 Z^2}$$

Donde:

n= Tamaño de la muestra

N=Universo o Población

σ^2 =Varianza

Z= Nivel de Confianza

E= Limite aceptable de error muestral.

Entonces aplicando la fórmula para este caso se obtendrá lo siguiente:

$$n = \frac{34(0.5^2)(1.96^2)}{(34-1)0.05^2 + (0.5^2)(1.96^2)} = 31.31$$

Por lo que podemos tomar una muestra aproximadamente de 31 docentes de forma aleatoria.

5.2.2 OBTENCIÓN DE LOS DATOS

Las encuestas realizadas se dirigieron a docentes quienes pudieron visualizar el módulo de estadísticas del sistema SIEDD, mismos que contestaron las preguntas basándose también en lo que recuerdan del sistema anterior.

1. ¿Recuerda usted el sistema de evaluación docente aplicado anteriormente?
SI___ NO___
2. En caso de que la pregunta 1 sea SI, ¿Cómo se evidencian los resultados en el sistema de evaluación anterior?
Muy Bien___ Bien ___ Regular___ Mal___
3. ¿Cuántos parámetros (Tipos de Resultados) puede usted indicar en el sistema de evaluación anterior?
1___ 2___ 3___ Si son más parámetros indique cuantos___ No recuerdo___
4. ¿Cómo se evidencian los resultados en el sistema de evaluación SIEDD?

Muy Bien__ Bien__ Regular__ Mal__

5. ¿Cuántos parámetros (Tipos de Resultados) puede usted visualizar en el sistema de evaluación propuesto SIEDD?

1__ 2__ 3__ Si son más parámetro indique cuantos__

6. ¿Cómo se evidencian los resultados en el sistema de evaluación SIEDD?

Muy Bien__ Bien__ Regular__ Mal__

Dándole opción al encuestado a colocar la respuesta que creyera conveniente.

Tomando en cuenta que las preguntas que serán determinantes para la comprobación de la hipótesis son la segunda y cuarta, las demás ayudarán a recolectar requerimientos adicionales del sistema así como recomendaciones para el mismo.

Además en las opciones se permite que el docente responda las preguntas según su criterio.

Las valoraciones de todas las preguntas se presentan en el Anexo 1.

5.2.3 CLASIFICACIÓN Y ORGANIZACIÓN DE LOS DATOS

Se tomará las respuestas de las preguntas 2 y 4 que servirán para la demostración de la hipótesis, dando a notar que el docente en el momento de ser encuestado puede colocar cualquier respuesta según su criterio.

TABLA IV.39 ORGANIZACIÓN DE RESULTADOS

DOCENTE	ENUNCIADO #3	ENUNCIADO # 5
1	NR	3
2	NR	4
3	3	5
4	3	4
5	3	3
6	NR	3
7	NR	3
8	2	4
9	2	4
10	NR	3
11	NR	3
12	3	5
13	NR	4
14	3	4
15	3	3
16	3	4
17	3	4
18	3	3
19	3	4
20	3	5
21	NR	3
22	NR	3
23	2	4
24	2	3
25	3	4
25	3	4
26	3	3
27	3	4
28	3	5
29	NR	3
30	3	4
31	3	5
PROMEDIO	2,81	3,75
DESVIACION	0,3947	0,7184

5.2.4 ANÁLISIS DE LOS DATOS

Aquí determinaremos los datos necesarios para aplicar la distribución Z, donde nuestras hipótesis son:

H₀: El uso de la lógica difusa en un módulo del Sistema de Evaluación del Desempeño del Ejercicio Docente aumentará el número de parámetros analizados para la toma de decisiones.

H₁: El uso de la lógica difusa en un módulo del Sistema de Evaluación del Desempeño del Ejercicio Docente no aumentará el número de parámetros analizados para la toma de decisiones.

Para evitar aumentar el nivel del error se ha omitido a las personas que respondieron que no recuerdan el sistema de evaluación anterior, por lo que los datos se presentaron de la siguiente manera:

$n_1=22$	$n_2=31$
$\bar{X}_1 = 2.81$	$\bar{X}_2 = 3.75$
$S_1 = 0.39$	$S_2 = 0.71$

Nivel de significancia 5% = 0.05

Tomando la fórmula

$$S_{\bar{X}_1 - \bar{X}_2} = \sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}} = \sqrt{\frac{0.39^2}{22} + \frac{0.71^2}{31}} = -0.1522$$

$$Z = \frac{\bar{X}_1 - \bar{X}_2}{S_{\bar{X}_1 - \bar{X}_2}}$$

$$Z = \frac{2.81 - 3.75}{-0.1522} = -6.17$$

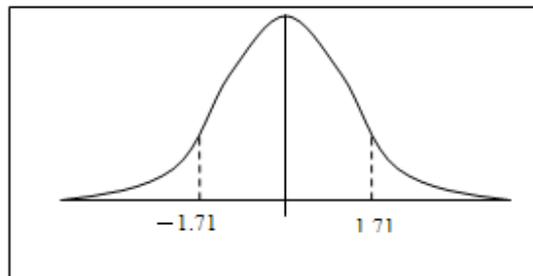


Figura V.1 Distribución Z

5.2.6 VALIDACIÓN DE LA HIPÓTESIS

La lógica difusa permite tener un conocimiento más acercado a la realidad por lo que los docentes que observan estos nuevos valores en el módulo de estadísticas del sistema SIEDD *aumentan el número de parámetros analizados* por el mismo pudiendo determinar así diferentes aspectos en los que se puede cambiar, mantener o mejorar para un mejor desempeño docente.

5.2.7 CONCLUSIONES

- Presentar una encuesta con preguntas alternativas atrae la atención de un mayor número de personas interesadas, así como realizar una encuesta rápida y fácil de entender.

El resultado obtenido al calcular la distribución Z permite determinar que se acepto la hipótesis nula y se rechaza la hipótesis Alternativa, siendo la hipótesis nula: “El uso de la lógica difusa en un módulo del Sistema de Evaluación del Desempeño del Ejercicio Docente aumentará el número de parámetros analizados para la toma de decisiones”.

CONCLUSIONES

1. La presente tesis permitió entender que la lógica difusa está basada en grados de pertenencia asignados a distintos conjuntos difusos por medio de funciones de pertenencia, mismos que se encuentran asociados a variables lingüística, y que arrojan resultados según reglas definidas por un experto.
2. La lógica difusa nos permite expresar los resultados en lenguaje natural, de forma que los usuarios pueden visualizarlos no solo de forma numérica sino también con adjetivos que la describan.
3. Se pudo conocer que la lógica difusa indica el grado en el que un valor está incluido en un conjunto difuso mientras que la probabilidad indica la frecuencia con la que ocurre un determinado evento.
4. Un Sistema de lógica difusa está constituido por cuatro módulos denominados Fuzzificador, Base de Reglas, Motor de Inferencia y Desfuzzificador. Cada uno de estos realizan acciones que permitirán recibir entradas o datos discretos, procesarlos y generar salidas concretas.
5. El proceso de Evaluación realizado en la ESPOCH comienza con la evaluación de los estudiantes y docentes en cada facultad y escuela según el horario definido por

la comisión de evaluación institucional. Luego se realiza el proceso de cierre del periodo, que consiste en ejecutar en el sistema el cálculo de los resultados, mismo que demora entre 8 y 10 días para luego ser puesto en consideración de los usuarios del sistema.

6. Los docentes al visualizar los resultados obtenidos en el proceso de evaluación no pueden observar la escala en la que se encuentra ubicado el mismo, por lo que el docente interpreta a su manera los resultados visualizados y toma la decisión que mejor convenga según su interpretación.
7. La comisión de evaluación realiza el informe de resultados basándose en la documentación provista por el CONEA, que si bien permite realizar el mismo de forma correcta toma más tiempo determinar la escala en la que se hallan los resultados de las distintas facultades.

RECOMENDACIONES

1. Realizar capacitaciones a los usuarios acerca del Sistema Integrado de Evaluación al Desempeño Docente (SIEDD) así como también de lo que es la lógica difusa debido a que este tema no es muy conocido para muchos de ellos.
2. Desarrollar un módulo que permita conocer datos de evaluaciones anteriores para medir la evolución del desempeño docente, puesto que los datos existen pero no una aplicación que los muestre.
3. Incentivar la autoevaluación docente con el fin de conseguir resultados más apegados a la realidad y que no afecten a los resultados institucionales.
4. Estimular a los docentes a que revisen los resultados obtenidos en las evaluaciones debido a que es de gran importancia conocer si su desempeño en el proceso educativo es el correcto según los estudiantes.

RESUMEN

La presente tesis tiene como objetivo estudiar la lógica difusa, que es un modelo matemático, y aplicarla en un módulo en el Sistema de Evaluación Integrado del Desempeño Docente de la Epoch (SIEDD) que aumentará el número de parámetros analizados para la toma de decisiones en cuanto al mejoramiento en el proceso de enseñanza aprendizaje de los estudiantes.

El método utilizado fue el científico general además de encuestas y entrevistas. Se utilizó la herramienta de desarrollo NetBeans IDE 6.5, Apache Tomcat 6, Servidor de Base de Datos SQL server 2000 sobre la plataforma Windows Server 2003.

Con los resultados obtenidos en la evaluación realizada por estudiantes se consiguió datos que permitieron desarrollar el módulo de estadísticas de SIEDD con el parámetro de lógica difusa. Se procedió a establecer los valores de pertenencia a los conjuntos difusos de la forma $A = \{x, A(x) / x \in U\}$, siendo estos: excelente, bueno, regular e insuficiente mediante la función de pertenencia Trapezoidal y según reglas establecer el valor difuso de asignación a una variable lingüística. Así por ejemplo: Un Docente X Tiene un promedio de 78 puntos; Que lo ubica como una fortaleza con un grado de certeza de 23%.

El módulo desarrollado contiene resultados de autoevaluación docente y de evaluación de estudiantes de forma general, también resultados por carrera, por materia y por pregunta en cada tipo de evaluación. Además contiene resultados a nivel institucional.

SUMMARY

The present thesis has as objective to study the diffuse logic that is a mathematical model, and to apply it in a module in the Integrated System of Evaluation of the Educational Acting of the Epoch (SIEDD) that will increase the number of parameters analyzed for the taking of decisions as for the improvement in the process of the students' teaching learning. The general scientific method was applied besides surveys and interviews. The development tool used NetBeans IDE 6.5, Apache Tomcat 6, Database server SQL server 2000 on the platform Windows Server 2003.

With the results obtained in the evaluation carried out by students it was gotten data that allowed to develop the module of statistical of SIEDD with the parameter of diffuse logic. Did you proceed to establish the securities of ownership to the diffuse groups in the way $A = \{x, A(x) / x \in U\}$, in this case: excellent, good, to regulate and insufficient by the function of Trapezoidal ownership and according to rules to establish the diffuse value of assignment to a linguistic variable. This way for example: An Educational X Has an average of 78 points; That locates it like a strength with a grade of certainty of 23%.

The developed module contains results of educational autoevaluación and of students' in a general way evaluation, also been by career, for matter and for question in each evaluation type. It also contains results at institutional level.

GLOSARIO

Java Beans	Son componentes de servidor desarrollados utilizando tecnología Java que encapsula la lógica de negocios de una aplicación.
Jsp	Son documentos de texto que se ejecutan como servlets pero que permiten crear contenidos estáticos
Servlets	Sirve para generar páginas web de forma dinámica a partir de los parámetros de la petición que envíe el navegador web
Defuzzificación	Es el encargado de establecer la relación entre los datos de entrada al sistema y sus correspondientes conjuntos difusos.
Funcion de Perteneacia	Es una curva donde se relaciona términos lingüísticos, no fácilmente mensurables, con un valor numérico.

ANEXOS

**MANUAL DE USUARIO
MÓDULO DE
RESULTADOS DEL
SISTEMA SIEDD**

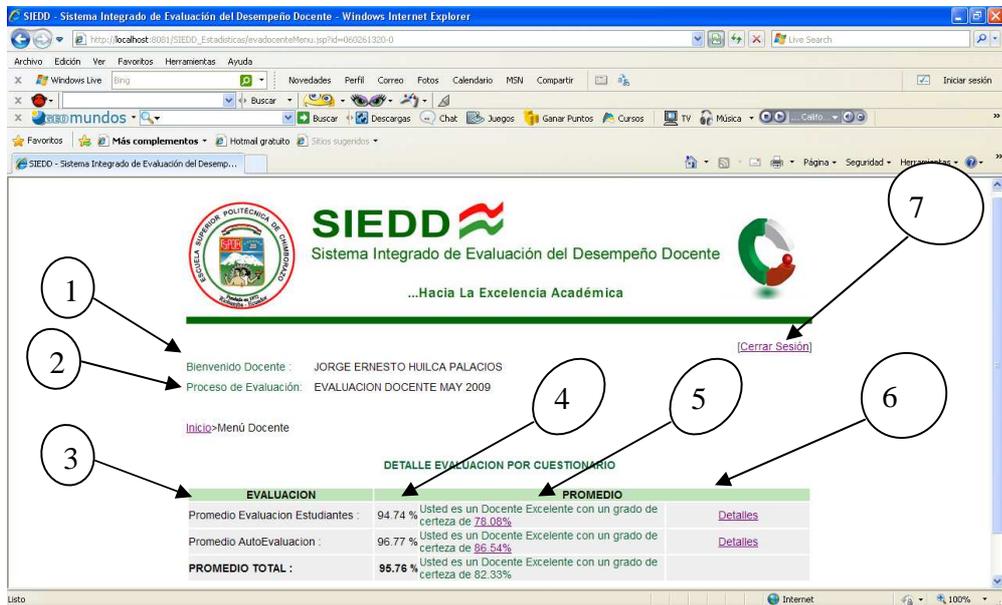
Para ingresar al módulo de resultados del sistema SIEDD es necesario ingresar el link <http://evaluación.esPOCH.edu.ec> se mostrará la página que a continuación se muestra.



Seleccionaremos entonces el link **Resultados de Docentes** donde se desplegará la pantalla de autenticación.



Donde se deberá ingresar el nombre de usuario y password para poder ingresar y poder visualizar los resultados.



La primera pantalla muestra:

1. El nombre del docente

2. El proceso de evaluación vigente
3. El detalle de los cuestionarios que han sido evaluados.
4. El promedio obtenido por el docente en su respectivo cuestionario.
5. Resultado obtenido mediante el cálculo de la lógica difusa.
6. Detalles de los resultados según cada cuestionario.
7. Link que permite cerrar la sesión del usuario.

Podemos hacer clic en el valor calculado por la lógica difusa lo que nos llevará a obtener información de este valor calculado en la siguiente página.

The screenshot shows the SIEDD (Sistema Integrado de Evaluación del Desempeño Docente) web application. The page header includes the SIEDD logo and the text "Sistema Integrado de Evaluación del Desempeño Docente ...Hacia La Excelencia Académica". Below the header, the user is greeted as "Bienvenido Docente : JORGE ERNESTO HUILCA PALACIOS" and the evaluation process is identified as "EVALUACION DOCENTE MAY 2009". A link for "[Cerrar Sesión]" is visible. A table displays the evaluation results for three groups:

GRUPO	PROMEDIO	DETALLES
PEDAGOGÍA Y CONOCIMIENTOS	95.09%	Usted es un Docente Excelente en este grupo, con un grado de certeza de 79.54%
EVALUACIÓN	94.01%	Usted es un Docente Excelente en este grupo, con un grado de certeza de 75.04%
RELACIONES HUMANAS Y ATENCIÓN AL ESTUDIANTE	94.25%	Usted es un Docente Excelente en este grupo, con un grado de certeza de 76.04%

Mediante el navegador que se encuentra debajo del proceso de evaluación vigente podemos regresar a la página anterior y al hacer clic en detalles de cuestionario de evaluación de estudiantes se presentará la pantalla con la siguiente información.

Sistema Integrado de Evaluación del Desempeño Docente

...Hacia La Excelencia Académica

Bienvenido Docente: JORGE ERNESTO HUILCA PALACIOS
Proceso de Evaluación: EVALUACIÓN DE DOCENTES MAY 2009
Promedio Evaluación: 94.74 %

[Inicio](#) > [Menú Docente](#) > [Asignaturas Docente](#)

Carrera	Asignatura	Promedio	
INGENIERIA COMERCIAL	INFORMATICA APLICADA I	90.78%	Usted es un Docente Excelente con un grado de certeza de 61.58%
INGENIERIA COMERCIAL	INFORMATICA APLICADA II	94.13%	Usted es un Docente Excelente con un grado de certeza de 75.54%
INGENIERIA EN SISTEMAS	PROGRAMACION DE REDES	97.08%	Usted es un Docente Excelente con un grado de certeza de 87.83%
INGENIERIA EN SISTEMAS	PROYECTO INTEGRADOR III	96.96%	Usted es un Docente Excelente con un grado de certeza de 87.33%

Desarrollado por: Comisión de Evaluación Institucional - Escuela Superior Politécnica de Chimborazo

1. La carrera donde dicta la cátedra.
2. La asignatura a cargo del docente.
3. El promedio obtenido en dicha materia.
4. La interpretación difusa de cada una de las asignaturas

Sobre cada asignatura se encuentra asignada un link para mostrar detalles por preguntas de dicha asignatura y se presentará de la siguiente forma:

SIEDD - Sistema Integrado de Evaluación del Desempeño Docente
Sistema Integrado de Evaluación del Desempeño Docente
...Hacia La Excelencia Académica

Bienvenido Docente: JORGE ERNESTO HUILCA PALACIOS
Materia: INFORMÁTICA APLICADA I

[Inicio](#) > [Menú Docente](#) > [Asignaturas Docente](#) > [Detalle Materia](#)

PROMEDIO POR GRUPO Y PREGUNTA
Nº Estudiantes Registrados: 70
Nº Estudiantes Que Evaluaron: 56

[Ver Comentarios](#)

Grupo PEDAGOGIA Y CONOCIMIENTOS			
Promedio 91.45			
Orden	Pregunta	Promedio	Desviación Estandar
1	¿Presentó el programa analítico de la asignatura al inicio del curso?	90.67	25.15
2	¿Explicó y acordó con los estudiantes al inicio del curso el sistema de evaluación a utilizar?	95.76	19.06
3	¿Exploró los conocimientos de los estudiantes al inicio del curso?	94.91	19.94
4	¿Establece acuerdos con los estudiantes para la ejecución de las actividades a desarrollar dentro del programa?	83.1	14.52
5	¿Asiste con regularidad a clases?	96.55	10.63
6	¿Comienza y termina las clases a la hora señalada?	90.78	15.26
7	¿Promueve el uso de métodos y/o estrategias para la solución de problemas?	89.22	10.63
8	¿Utiliza diferentes técnicas de trabajo en las sesiones de clases como: trabajos en equipo, discusiones en grupo, talleres u otras?	88.96	19.12
9	¿Enfatiza los contenidos esenciales para su formación profesional y humanística?	88.36	21.38
10	¿Explica con claridad y precisión?	89.83	18.47
11	¿Tiene dominio sobre los conocimientos que imparte en su materia?	94.82	15.22
12	¿Promueve la participación, creatividad y el análisis crítico de los estudiantes?	90.51	15.96
13	¿Realiza actividades prácticas?	95.25	12.67
14	¿Promueve en las clases actividades de investigación?	87.26	21.27
15	¿Promueve el uso de las tecnologías de información?	91.1	18.84
16	¿Mantiene la disciplina, confianza, interés, entusiasmo y atención en clases?	91.94	15.56
17	¿Apoya el desarrollo de la cátedra con la elaboración de textos u otros materiales didácticos?	85.16	25.26

Grupo EVALUACIÓN			
Promedio 90.05			
Orden	Pregunta	Promedio	Desviación Estandar

Donde se mostrará el grupo calificado, su promedio y las preguntas pertenecientes a este grupo con su respectiva desviación estándar. También se mostrarán los diferentes comentarios que los estudiantes de la asignatura indicada colocaron al docente



Regresando con el navegador al menú principal podemos indicar los detalles del cálculo del valor difuso. Seleccionando la interpretación con el valor difuso.



Mostrándonos entonces los valores correspondientes al cuestionario de la autoevaluación.



Igual que en el cuestionario de los estudiantes al dar clic en detalles en la autoevaluación docente nos mostrará las carreras en las que este trabaja con su respectivo promedio e interpretación difusa.



Al dar clic sobre la carrera se desplegará la pantalla con los resultados obtenidos por preguntas. Así como también promedio por grupo

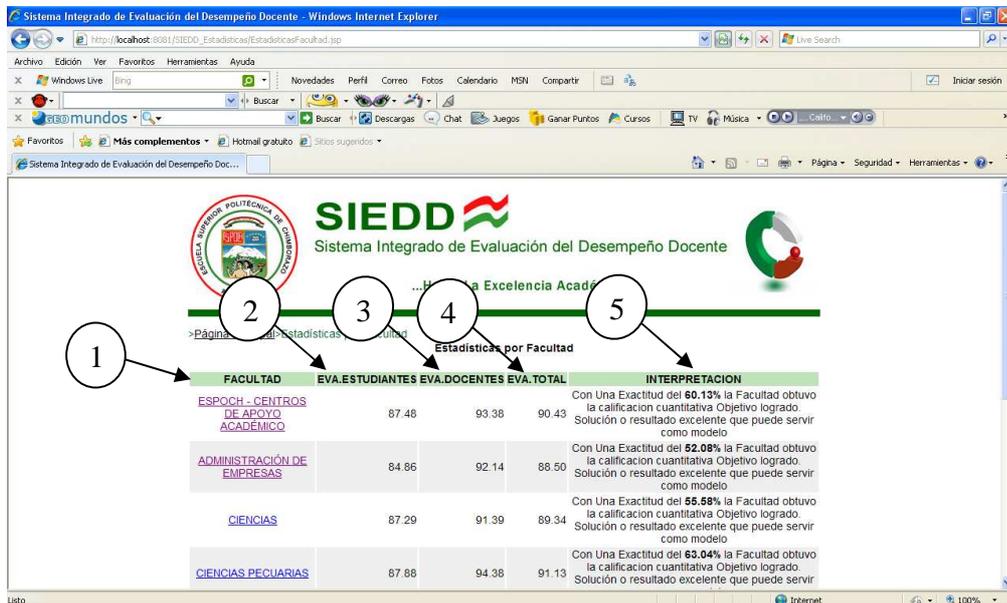
The screenshot shows the SIEDD (Sistema Integrado de Evaluación del Desempeño Docente) interface. At the top, there is a header with the SIEDD logo and the text "Sistema Integrado de Evaluación del Desempeño Docente ...Hacia La Excelencia Académica". Below the header, the user is identified as "JORGE ERNESTO HUILCA PALACIOS" and the evaluation process is "EVALUACION DOCENTE MAY 2009" with an auto-evaluation average of "93.54%". A navigation menu includes "Inicio", "Menú Docente", "Carreras Autoevaluadas", and "Detalle Autoevaluación". The main content area displays a table for the "Grupo PEDAGOGÍA Y CONOCIMIENTO" with a "Promedio 93.05". The table lists five questions and their respective scores:

Orden	Pregunta	Promedio
1	¿Presenté el programa analítico de la asignatura al inicio del curso?	100.0
2	¿Explicé y acordé con los estudiantes al inicio del curso el sistema de evaluación a utilizarse?	100.0
3	¿Exploré los conocimientos de los estudiantes al inicio del curso?	100.0
4	¿Establecí acuerdos con los estudiantes para la ejecución de las actividades a desarrollar dentro del programa?	100.0
5	¿Asisté con regularidad a clases?	100.0

Regresando de nuevo a la página principal encontraremos el link que nos permitirá ingresar a revisar los resultados institucionales.

The screenshot shows the main page of the SIEDD system. It features the same header as the previous screenshot. Below the header, there are several navigation links: "Página Principal", "Sistema de Evaluación", "Resultados de Docentes", "Resultados Institucionales", "Administración", and "Créditos". A central text block describes the teaching function of the Universidad Ecuatoriana. Below this text is a photograph of a building. At the bottom, there is a statement of interest from the Comisión de Evaluación Institucional.

Al hacer clic en este link se presentará los resultados por facultades.



Y este consta de lo siguiente:

1. Nombre de la facultad.
2. Promedio obtenido en la evaluación realizada por los estudiantes.
3. Promedio obtenido en la evaluación realizada por los docentes.
4. Evaluación total que se calcula como un promedio entre los 2 mencionadas anteriormente.
5. Finalmente se muestra la interpretación con la lógica difusa.

Al hacer clic sobre la facultad se desplegará las distintas escuelas de la misma que presentarán información similar a la presentada por las facultades.

The screenshot shows a web browser displaying the 'Sistema Integrado de Evaluación del Desempeño Docente' interface. The page title is 'Sistema Integrado de Evaluación del Desempeño Docente ...Hacia La Excelencia Académica'. The navigation path is '>Estadísticas por Facultad > Estadísticas por Escuela'. The main content is a table with the following data:

ESCUELA	EVA.ESTUDIANTES	EVA.DOCENTES	EVA.TOTAL	INTERPRETACION
CICLO FORMATIVO	81.47	90.38	85.93	Con Una Exactitud del 41.38% la Escuela obtuvo la calificación cuantitativa Objetivo logrado. Solución o resultado excelente que puede servir como modelo
INGENIERIA EN CONTABILIDAD Y AUDITORIA	87.81	90.57	89.19	Con Una Exactitud del 84.96% la Escuela obtuvo la calificación cuantitativa Objetivo logrado. Solución o resultado excelente que puede servir como modelo
INGENIERIA DE EMPRESAS	85.90	92.66	89.28	Con Una Exactitud del 85.33% la Escuela obtuvo la calificación cuantitativa Objetivo logrado. Solución o resultado excelente que puede servir como modelo
INGENIERIA FINANCIERA Y COMERCIO EXTERIOR	84.09	91.94	88.02	Con Una Exactitud del 50.08% la Escuela obtuvo la calificación cuantitativa Objetivo logrado. Solución o resultado excelente que puede servir como modelo
INGENIERIA EN MARKETING	83.72	93.25	88.49	Con Una Exactitud del 82.04% la Escuela obtuvo la calificación cuantitativa Objetivo logrado. Solución o resultado excelente que puede servir como modelo

De la misma forma se muestra las carreras de una escuela seleccionada con su respectiva información.

The screenshot shows the same web application interface, but with the navigation path '>Estadísticas por Facultad > Estadísticas por Escuela > Estadísticas por Carrera'. The main content is a table with the following data:

CARRERA	EVA.ESTUDIANTES	EVA.DOCENTES	EVA.TOTAL	INTERPRETACION
CICLO FORMATIVO	81.47	90.38	85.93	Con Una Exactitud del 41.38% la Carrera obtuvo la calificación cuantitativa Objetivo logrado. Solución o resultado excelente que puede servir como modelo

Desarrollado por :Comisión de Evaluación Institucional - Escuela Superior Politécnica de Chimborazo

En este punto se puede seleccionar el cuestionario de los estudiantes o de la autoevaluación mismo que presentarán los respectivos resultados por preguntas.

Sistema Integrado de Evaluación del Desempeño Docente - Windows Internet Explorer

http://localhost:8081/SIEDD_Estadisticas/EstadisticasPreguntasCarrera.jsp?carrera=CFOR&ipom=2&facultad=FADE&escuela=CFOR

Archivo Edición Ver Favoritos Herramientas Ayuda

Windows Live Bing Novedades Perfil Correo Fotos Calendario MSN Compartir Iniciar sesión

Buscar Descargas Chat Juegos Ganar Puntos Cursos TV Música Celfo...

Sistema Integrado de Evaluación del Desempeño Doc...

SIEDD Sistema Integrado de Evaluación del Desempeño Docente

...Hacia La Excelencia Académica

> Estadísticas por Facultad > Estadísticas por Escuela > Estadísticas por Carrera > Estadísticas por Pregunta

Estadísticas por Carrera

ORDEN	PREGUNTA	PROMEDIO
1	¿Presentó el programa analítico de la asignatura al inicio del curso?	93.33
2	¿Explicó y acordó con los estudiantes al inicio del curso el sistema de evaluación a utilizarse?	100.0
3	¿Exploró los conocimientos de los estudiantes al inicio del curso?	100.0
4	¿Establece acuerdos con los estudiantes para la ejecución de las actividades a desarrollar dentro del programa?	98.33
5	¿Asiste con regularidad a clases?	96.66
6	¿Comienza y termina las clases a la hora señalada?	91.66
7	¿Promueve el uso de métodos y/o estrategias para la solución de problemas?	96.66
8	¿Utiliza diferentes técnicas de trabajo en sus clases como: trabajos en equipo, discusiones en grupo, talleres u otras?	95.0
9	¿Enfatiza los contenidos esenciales para su formación profesional y humanística?	98.33
10	¿Promueve la participación, creatividad y el análisis crítico de los estudiantes?	96.66
11	¿Realiza actividades prácticas?	90.0

Este presentará las preguntas realizadas en cada carrera con el promedio obtenido en cada pregunta.

MANUAL DE INSTALACION DE NETBEANS IDE 6.1

INTRODUCCIÓN

En el presente documento presentamos un manual indicando como hacer funcionar netbeans, con sus funciones para poder programar en J2EE, jsp, html, php, entre otros.

Además veremos paso a paso la instalación de software adicional para la realización de proyectos en J2EE con conexión a Base de Datos, como son Java SDK (indispensable para instalar netbeans).

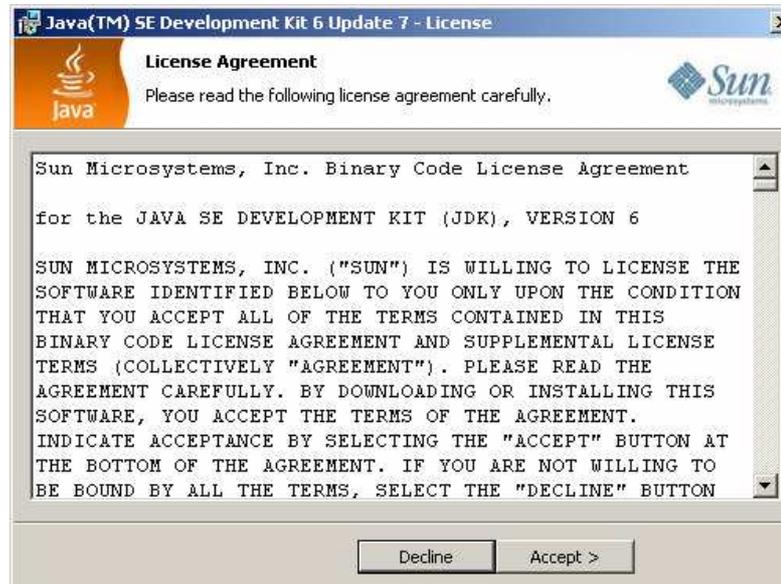
ARCHIVOS NECESARIOS

- Netbeans 6.1 para Windows.
- Java SDK 6 Update 7.

INSTALACION JDK

La instalacion del JDK es un proceso trivial que no amerita una mayor explicación, solo basta decir que dejamos todo tal cual se presenta y damos click en next hasta el final.

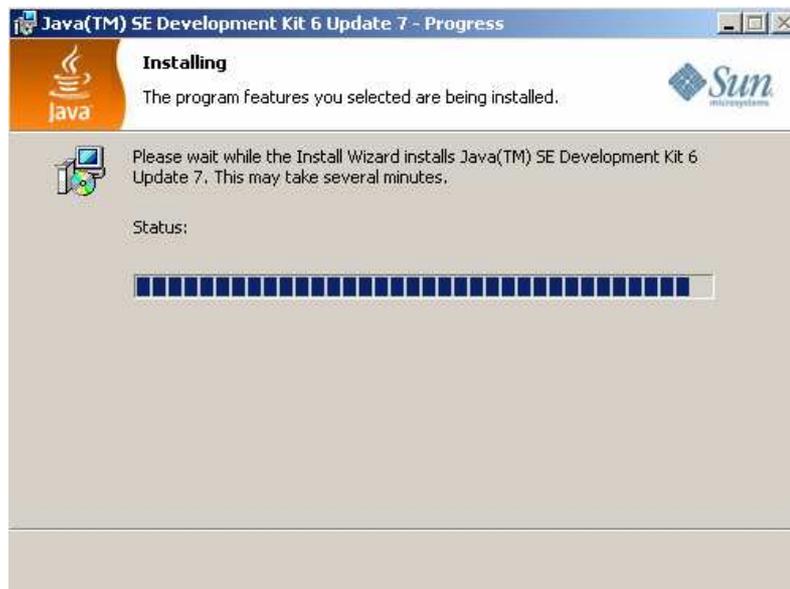
La primera pantalla nos presenta la licencia del producto para lo cual damos clic en aceptar.



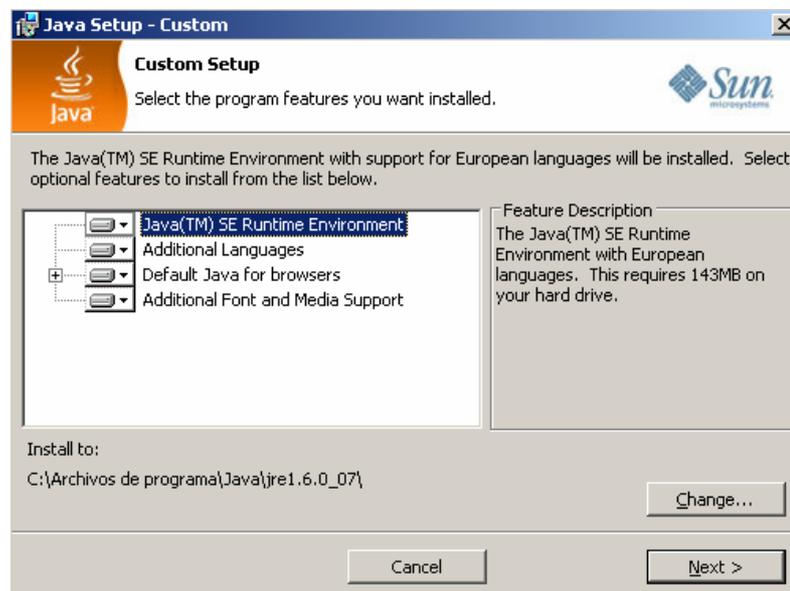
Luego seleccionamos los programas a instalar, por defecto debemos seleccionar todos.



A continuación comienza la instalación de los programas antes seleccionado y por lo que debemos esperar



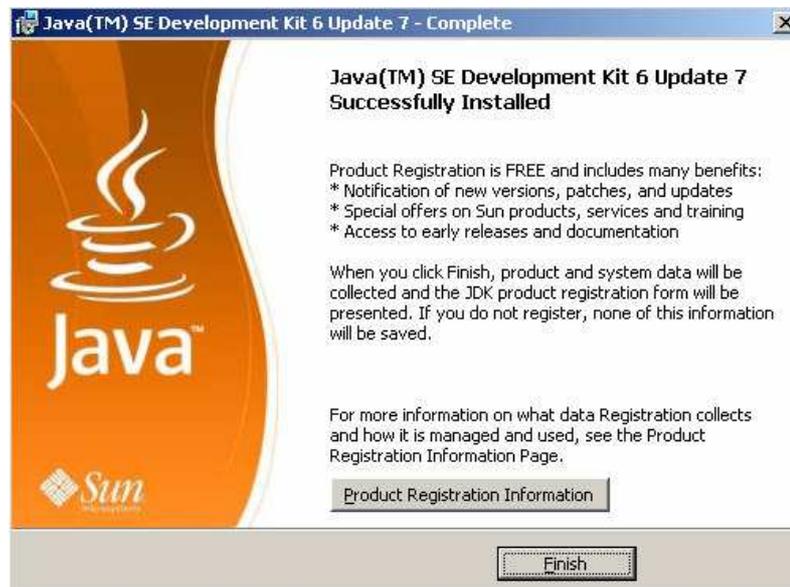
A continuación aparece una pantalla con nuevas configuraciones para lo cual seleccionamos todas las que se presentan por defecto.



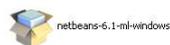
Se inicia nuevamente la instalación de las nuevas configuraciones.



Finalmente se completa la instalación con la presentación de la siguiente pantalla.



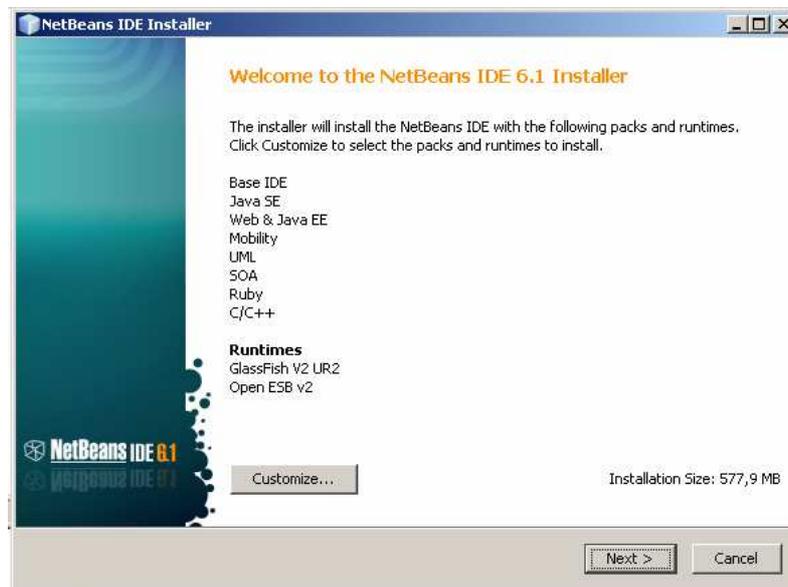
INSTALACION NETBEANS 6.1



Después de abrir el .exe toma un tiempo en configurar el instalador y luego nos da la opción de personalizar la instalación pudiendo agregar o eliminar paquetes,

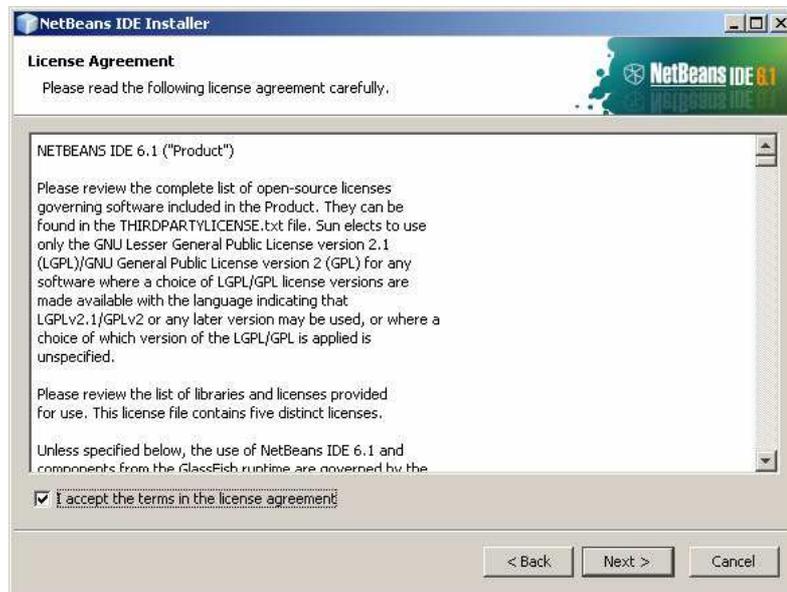


A continuación se presenta la pantalla donde se configurara la instalación del

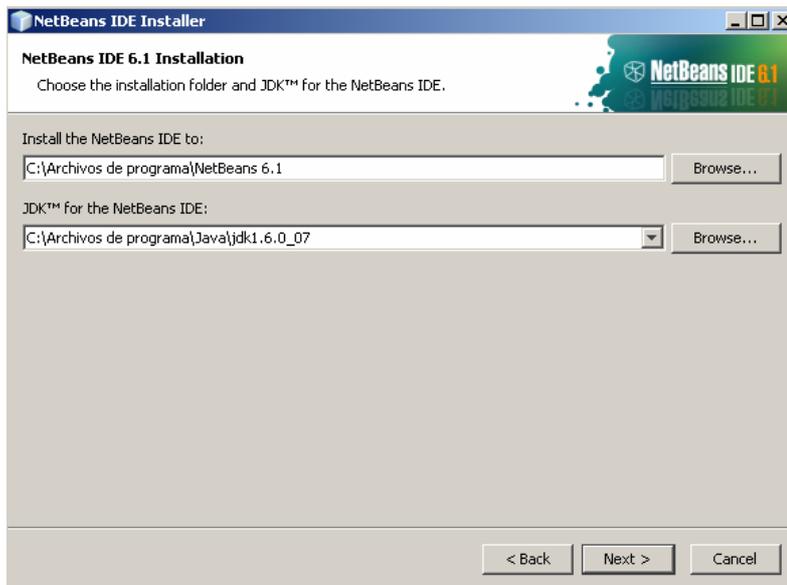


netbeans.

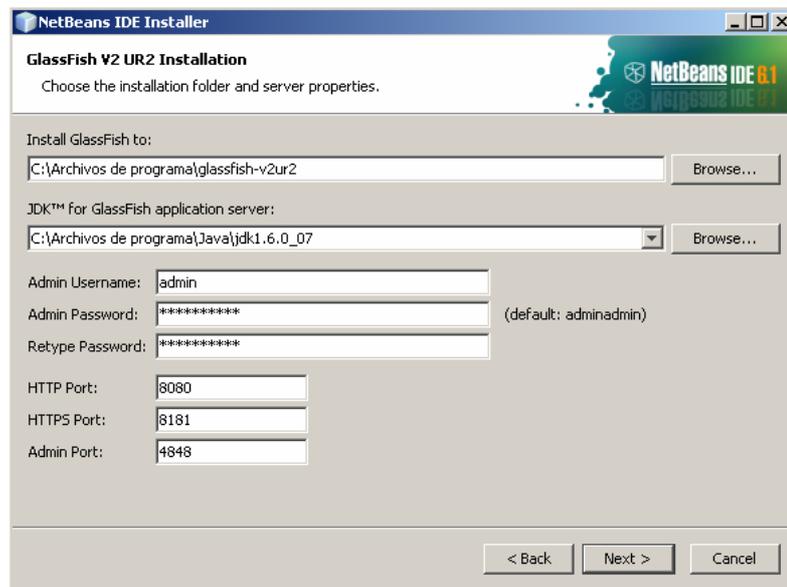
Luego se mostrará la pantalla con la licencia del producto en la cual se deberá dar clic en la opción de I accept.



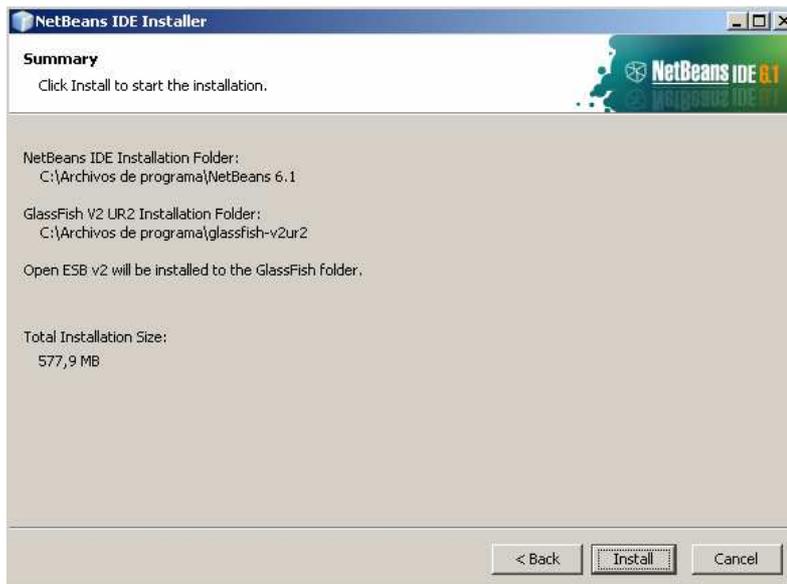
Seleccionamos la ubicación del Netbeans IDE y del JDK que es automático.



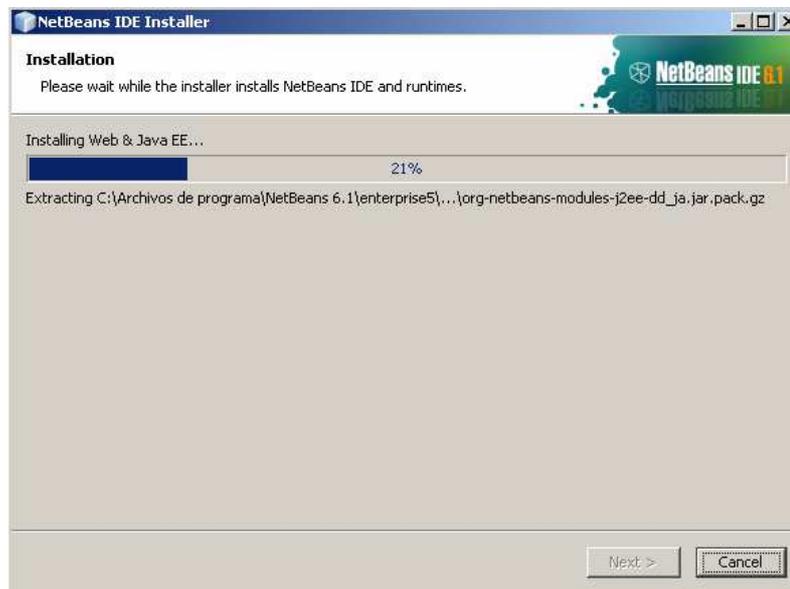
A continuación se muestran las carpetas de instalación y las propiedades del servidor



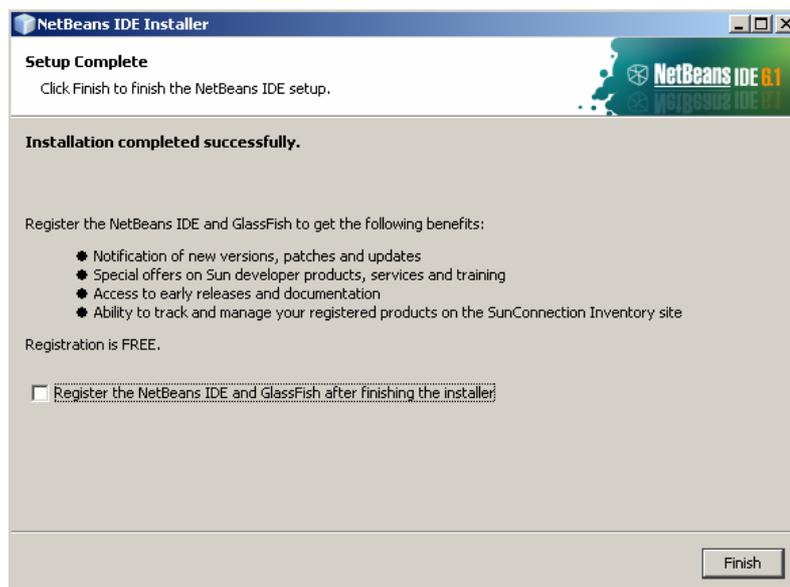
Luego se muestra un resumen de todo lo realizado antes de proceder a la instalación.



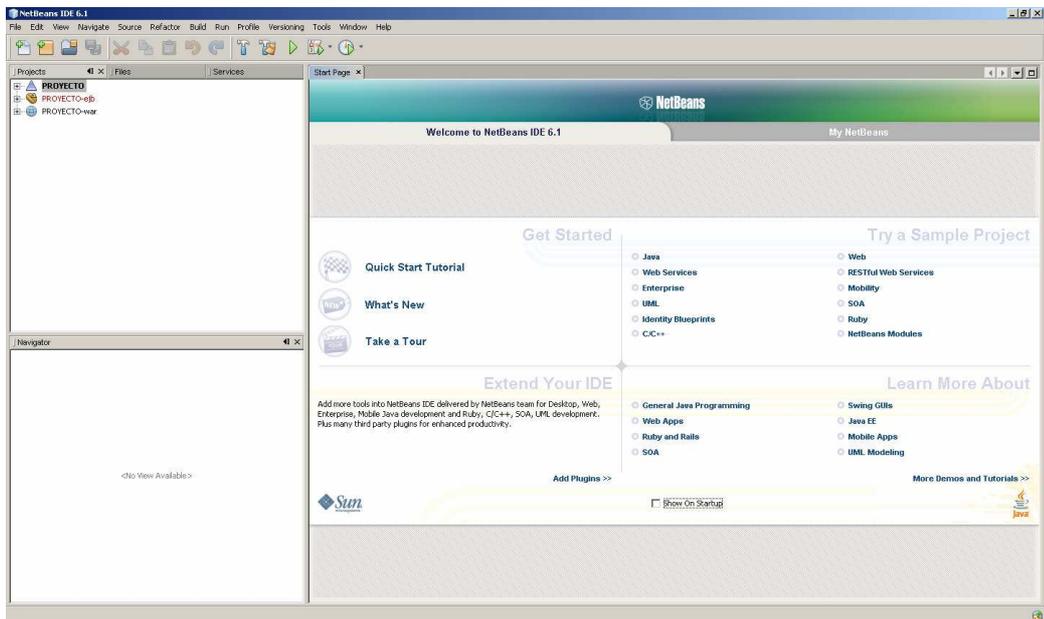
Al dar clic en instalar comienza el proceso.



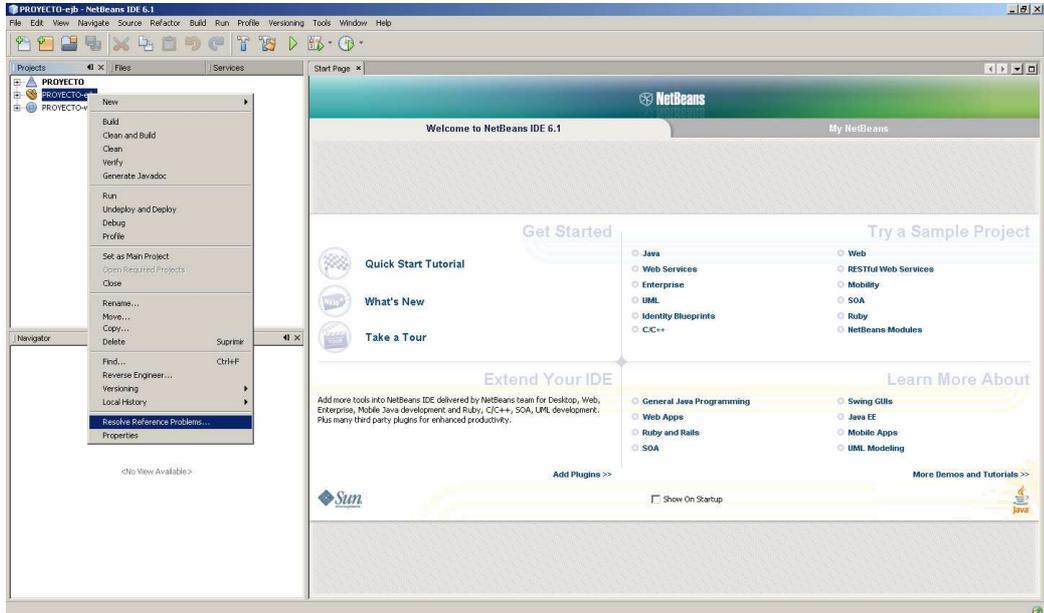
Finalmente se presenta la pantalla de instalación satisfactoria, con lo cual termina el proceso de instalación.



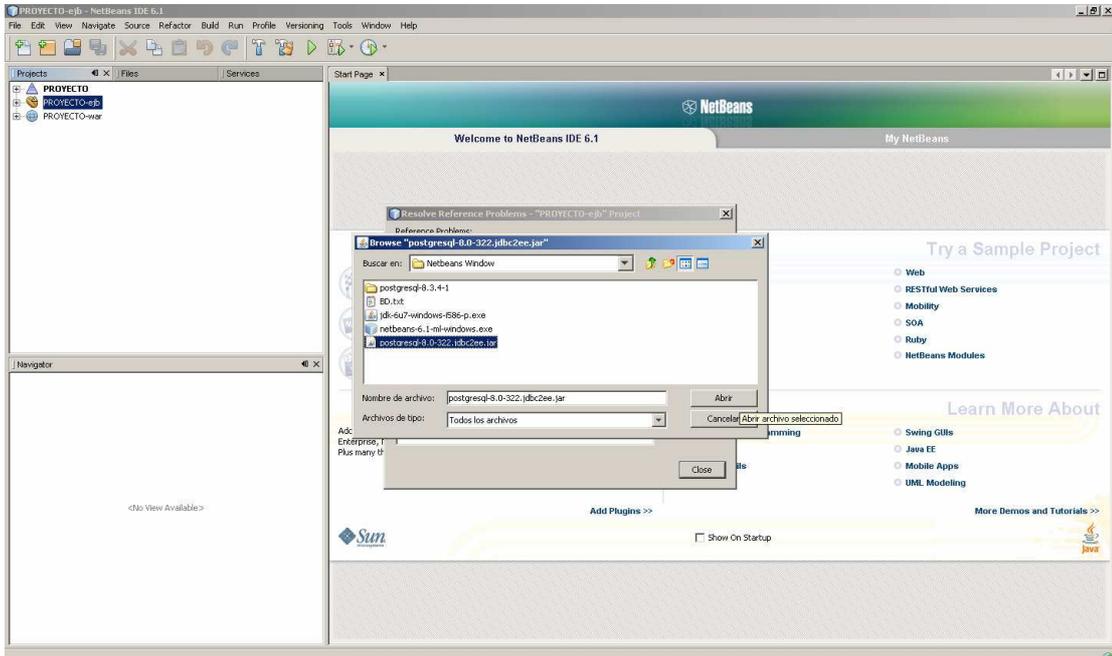
Para que la herramienta se encuentre en correcto funcionamiento abriremos un proyecto en el netbeans. Al tener el proyecto abierto se presentará la siguiente pantalla.



Hacemos clic derecho en el proyecto y seleccionamos la opción **Solucionar problemas de Referencia**.



Entonces seleccionamos el .jar del JDK. Y la herramienta se encuentra lista para su correcto funcionamiento.



MANUAL DE INSTALACION DE APACHE TOMCAT 6.0

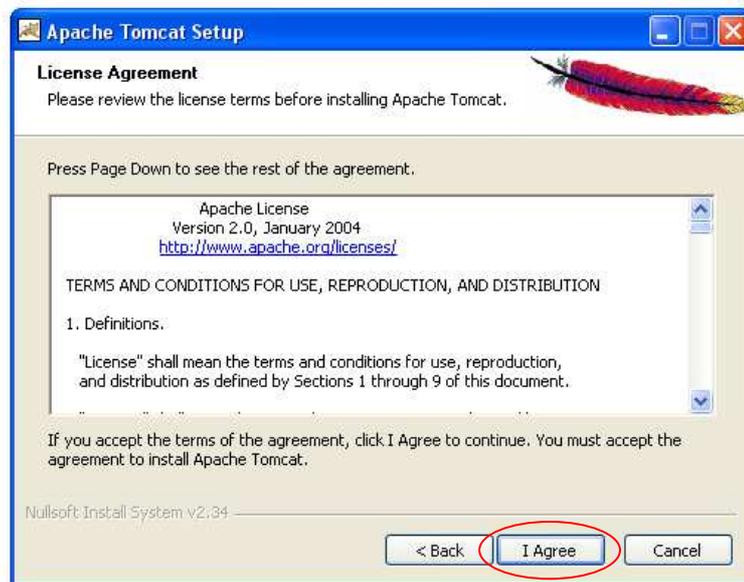
Como primer paso debemos descargar la versión más reciente de Tomcat. En nuestro caso este es el Apache Tomcat 6.0.

Elija el icono  que identifica apache tomcat para comenzar la instalación.

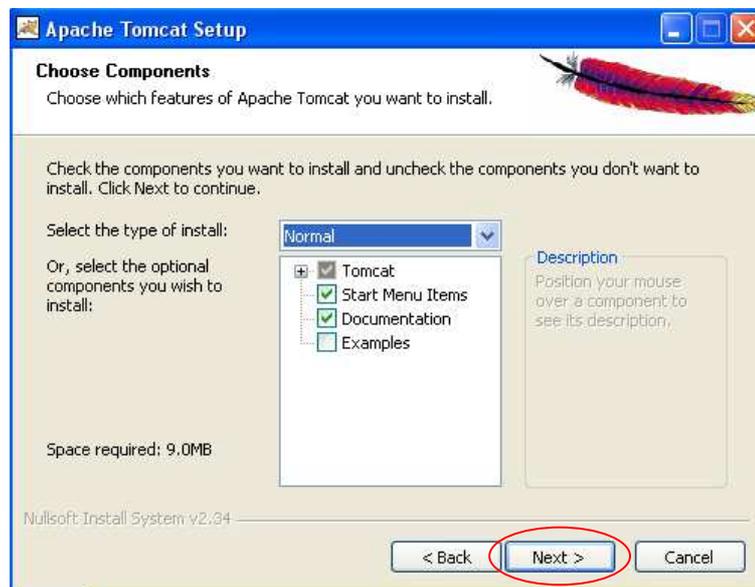
A continuación la ayuda le dará la bienvenida a la instalación de Apache tomcat 6.



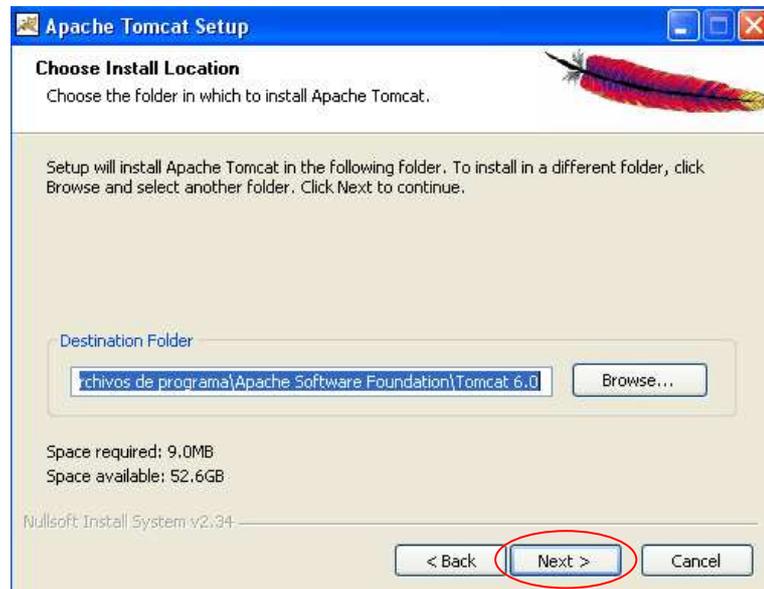
Luego se muestra la licencia de Apache en la cual deberemos hacer clic en el botón I Agree.



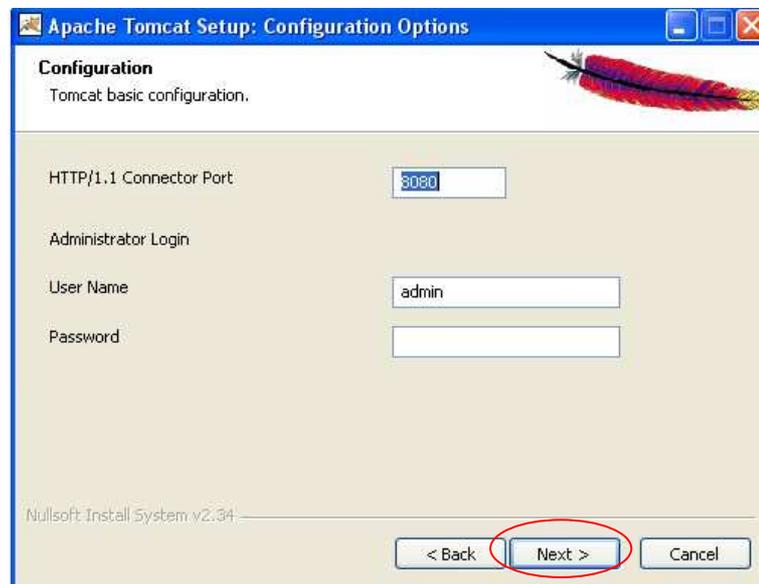
A continuación se presentan los componentes que el usuario desea instalar para el uso de Apache tomcat y hacemos clic en siguiente.



Luego definimos un directorio en donde se realizará la instalación y pulsamos siguiente.



Ahora se ingresará la configuración básica para el correcto funcionamiento de Apache como: El puerto de conexión que por defecto es el 8080, el login del administrador donde indicamos un usuario con su respectivo password.



Para el funcionamiento de Apache es necesario instalar la máquina virtual de Java que viene incluida en el instalador. A continuación seleccionamos un directorio para la instalación de la misma y procedemos a la instalación.

