



UNIVERSIDAD NACIONAL DE LOJA

**ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS
RECURSOS NATURALES NO RENOVABLES**

CARRERA DE INGENIERÍA EN SISTEMAS

**DESARROLLO DE UN SISTEMA INTELIGENTE PARA
PREDICCIÓN DEL DESEMPEÑO ACADÉMICO DE
LOS ESTUDIANTES UNIVERSITARIOS**

**TESIS PREVIA A LA
OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS**

AUTOR:

Jhymer Antonio Martínez Granda

DIRECTOR:

Ing. Henry Patricio Paz Arias, Mg. Sc.

LOJA- ECUADOR

2014

Certificación del Director

Loja, 13 de octubre de 2014

Ing. Henry Patricio Paz Arias, Mg. Sc.

DOCENTE DE LA CARRERA DE INGENIERÍA EN SISTEMAS, DEL ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES DE LA UNIVERSIDAD NACIONAL DE LOJA.

CERTIFICA:

Haber asesorado y revisado detenida y minuciosamente durante todo su desarrollo, el Proyecto de Fin de Carrera titulado: **“DESARROLLO DE UN SISTEMA INTELIGENTE PARA PREDICCIÓN DEL DESEMPEÑO ACADÉMICO DE LOS ESTUDIANTES UNIVERSITARIOS”**. Realizado por el postulante Jhymer Antonio Martínez Granda.

Por lo tanto, autorizo proseguir los trámites legales pertinentes para su presentación y defensa.



Ing. Henry Patricio Paz Arias, Mg. Sc.

DIRECTOR DEL PFC

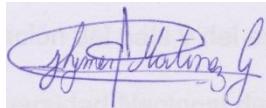
Autoría

Yo **JHYMER ANTONIO MARTÍNEZ GRANDA**, declaro ser el autor del presente trabajo de tesis y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi tesis en el Repositorio Institucional – Biblioteca Virtual.

Autor: Jhymer Antonio Martínez Granda

Firma:



Cédula: 1105113458

Fecha: 10 de marzo de 2015

**CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR,
PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y
PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.**

Yo **JHYMER ANTONIO MARTÍNEZ GRANDA**, declaro ser autor de la tesis titulada: **“DESARROLLO DE UN SISTEMA INTELIGENTE PARA PREDICCIÓN DEL DESEMPEÑO ACADÉMICO DE LOS ESTUDIANTES UNIVERSITARIOS”**, como requisito para optar el grado de: **INGENIERO EN SISTEMAS**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, diez días del mes de marzo del dos mil quince.

Firma:



Autor: Jhymer Antonio Martínez Granda

Cédula: 1105113458

Dirección: Loja, los Geranios (Crisantemos y Anturios N° 23-81).

Correo Electrónico: jamartinezg@unl.edu.ec

Teléfono: 2104374

Celular: 0959757359

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. Henry Patricio Paz Arias, Mg. Sc.

Tribunal de Grado: Ing. Luis Roberto Jácome Galarza, Mg. Sc.

Ing. Franco Hernán Salcedo López, Mg. Sc.

Ing. Jorge Iván Tocto, Mg. Sc.

Agradecimiento

Primeramente agradezco a Dios por la vida y la salud que me brinda día a día y la grata posibilidad de alcanzar uno de mis sueños que es obtener mi título profesional.

A mis padres y a abuelita, que con su apoyo económico y moral supieron alentarme y motivarme para culminar con éxito el desarrollo del presente proyecto y de esta manera poder finalizar una nueva etapa de formación profesional.

A mis tíos, que a pesar de estar lejos me supieron brindar su apoyo de todo tipo y sin el cual no hubiera sido posible el desarrollo del presente proyecto.

A mi esposa y mi hija que son los pilares fundamentales de mi vida que me brindan su apoyo y amor incondicional que constituye la fuerza que me motiva a seguir día a día.

A mis suegros que siempre me brindaron su apoyo de todo tipo para alcanzar este maravilloso objetivo.

De manera muy especial, agradezco a la Universidad Nacional de Loja, a sus autoridades y a todo el personal Docente y Administrativo que labora en tan prestigiosa Institución Educativa, y que gracias a su experiencia supieron guiarme para cumplir con mis metas.

Finalmente un agradecimiento especial a mi Director de Proyecto de Fin de Carrera, Ing. Henry Patricio Paz Arias, Mg. Sc. por su sabia dirección y excelente asesoría durante todo el desarrollo del presente Trabajo de Fin de Carrera.

EL AUTOR

Dedicatoria

El presente trabajo de investigación lo dedico primeramente a Dios, por el regalo de la vida, a mis padres con todo mi cariño ya que supieron inculcarme buenos valores y guiar me por el buen camino.

Con todo el amor del mundo dedico este trabajo a mi esposa y a mi hija que son el tesoro más valioso que poseo.

También quiero dedicar este trabajo a mis tíos y a toda mi familia que gracias a su apoyo hicieron posible el desarrollo del presente proyecto, y así poder alcanzar mi sueño de ser un profesional.

a. Título

“Desarrollo de un Sistema Inteligente para predicción del desempeño académico de los estudiantes universitarios”.

b. Resumen

Una de las principales falencias que se observa dentro de la Universidad es la inexistencia de un medio por el cual el docente comparta las calificaciones continuamente con sus educandos.

El avance de las nuevas tecnologías permite brindar soluciones que facilitan las labores cotidianas y mejoran la productividad, por tal motivo el presente trabajo de titulación está orientado a solventar el problema de incertidumbre que existe en el estudiante ya que permite dar a conocer de manera clara como se está llevando el desempeño académico desde un punto de vista actual y futuro.

Para la obtención de información base del sistema, se realizó una conexión a los Web Services que brinda el Sistema de Gestión Académica (SGA) de la Universidad Nacional de Loja, solventando así el problema de incompatibilidad entre distintas tecnologías.

El presente trabajo de titulación se centra en el uso de agentes inteligentes para plataforma web con el uso de la librería JADE desarrollada en java y la implementación de algoritmos que permitan la generación de calificaciones y gráficas predictivas facilitando al estudiante el mantener un monitoreo constante de su desempeño académico.

El tipo de agente utilizado es un agente deliberativo que mediante una arquitectura BDI (belief, desires, intentions o en español creencias, deseos e intenciones) adquiere un estado mental que permite la resolución de operaciones complejas.

Para el desarrollo de la aplicación web se utilizó la metodología denominada Programación Extrema (XP) y para la creación del agente inteligente se hizo uso de la metodología GAIA.

Como punto final, cabe acotar que el presente trabajo fue realizado en la ciudad de Loja y es aplicable específicamente para la Universidad Nacional de Loja a fin de contribuir al mejoramiento de la calidad académica tanto de los estudiantes como de los docentes.

Summary

A major shortcoming observed within the University is the lack of a mean by which teachers share their academic records with their students.

The advance of new technologies allows to provide solutions to everyday tasks and improve productivity, for this reason the present research work is aimed to solve the problem of uncertainty on the student's questions about their academic performance because it allows the students to know about how the process is being carried out from the beginning to the future.

To obtain basic information system, a connection to the Web Services provided by Sistema de Gestión Académica (SGA) of Universidad Nacional de Loja was carried out to solve the problem of incompatibility between different technologies.

This project focuses on the use of intelligent agents for web platform using the JADE library developed in java and implementation of algorithms for generating predictive ratings and graphs, which help the student to maintain a constant monitoring of their academic performance.

The type of agent used is a deliberative agent which through an architecture BDI (Believes, Desires, Intentions or in Spanish language creencias, deseos e intenciones) acquires a mental state that allows the resolution of complex operations.

For developing the web application it was used the methodology named Extreme Programming (XP) and for the creation of the intelligent agent it was used the GAIA methodology.

Finally it is necessary to say that the present research work was carried out in Loja city, and it is specifically applicable to Universidad Nacional de Loja, to contribute to the improvement of the academic quality of students and teachers.

Índice de Contenidos

Índice General

Certificación del Director	I
Autoría	II
CARTA DE AUTORIZACIÓN DE TESIS	III
Agradecimiento	IV
Dedicatoria	V
a. Título	VI
b. Resumen	VII
Summary	VIII
Índice de Contenidos	IX
Índice General	IX
Índice de Figuras	XIII
Índice de tablas	XV
c. Introducción	1
d. Revisión de Literatura	3
CAPÍTULO I	3
1 Sistemas Web y Desempeño Académico	3
1.1 Introducción	3
1.2 ¿Qué es el desempeño académico?	3
1.3 Factores que influyen en el desempeño académico	4
1.3.1 Determinantes personales	4
1.3.2 Determinantes Sociales	6
1.3.3 Determinantes institucionales	7
1.4 Sistemas Web	8
1.4.1 Antecedentes	8
1.4.2 La Web	9
1.4.3 Definición	9
1.4.4 Arquitecturas	10
1.5 Web Services	14
1.5.1 ¿Qué es un Web Service?	14
1.5.2 Características	15

1.5.3	Ventajas de los Web Services.....	16
1.5.4	Tecnologías utilizadas en los Web Services	17
1.6	Sistemas para el Desempeño Académico	21
1.6.1	CADI	21
1.6.2	Agentes inteligentes y desempeño de los estudiantes.....	22
1.7	Situación Actual	22
CAPÍTULO II.....		24
2	Agentes inteligentes.....	24
2.1	Introducción.....	24
2.2	Definición	25
2.3	Características.....	26
2.4	Comunicación.....	28
2.4.1	KQML.....	28
2.4.2	FIPA-ACL.....	32
2.5	Tipos de Agentes Inteligentes.....	33
2.5.1	Agentes Reactivos	34
2.5.2	Agentes Deliberativos	37
2.5.3	Agentes Colaborativos o Cooperativos.....	38
2.5.4	Agentes de Interfaz.....	38
2.5.5	Agentes Móviles.....	39
2.5.6	Agentes Estáticos	39
2.5.7	Agentes de Información/Internet	39
2.5.8	Agentes Híbridos.....	40
2.6	Arquitecturas de Agentes	40
2.6.1	Arquitecturas reactivas	40
2.6.2	Arquitecturas Deliberativas	41
2.6.3	Arquitecturas Híbridas	43
CAPÍTULO III.....		44
3	Metodologías	44
3.1	Metodología para el Desarrollo de Sistemas	44
3.1.1	Programación Extrema (XP)	44
3.1.2	Comparativa entre Metodologías de Desarrollo de Software	51
3.2	Metodología para el Desarrollo de Agentes	52
3.2.1	Metodología GAIA.....	52

3.2.2	Comparativa entre metodologías de agentes.....	54
e.	Materiales y Métodos.....	55
f.	Resultados.....	58
1	Hipótesis	58
2	Fase I: Análisis Inicial.....	58
2.1	Metodología XP: planificación del sistema	58
2.1.1	Especificación de requerimientos globales del sistema	58
2.1.2	Identificación de las Historias de usuario	59
2.2	Metodología GAIA: Análisis del agente.....	73
2.2.1	Modelo de Agente	73
2.2.2	Modelo de rol del agente.....	74
2.2.3	Definición de protocolos.....	75
2.2.4	Modelo de servicios	76
3	Fase II: Diseño	78
3.1	Metodología XP: Diseño del sistema	78
3.1.1	Análisis general del sistema	78
3.1.2	Patrón de diseño del sistema.....	78
3.1.3	Modelo de la base de datos relacional	79
3.1.4	Diccionario de datos	81
3.1.5	Tarjetas CRC	95
3.1.6	Diseño de interfaces del Sistema	104
3.2	Metodología GAIA: Diseño del Agente	113
3.2.1	Diseño de interacción	113
3.2.2	Arquitectura del agente	114
4	Fase III: Codificación.....	115
4.1	Metodología XP: Codificación del sistema.....	115
4.1.1	Estándares de codificación.....	115
4.1.2	Conexión a los Web Services del SGA	118
4.1.3	Generación de calificaciones y gráficas predictivas.....	119
4.1.1.	Automatización del proceso de registro de nóminas de estudiantes	131
4.1.4	Administración de calificaciones.....	132
4.1.5	Generación de promedios para el docente	134
4.1.6	Generación de promedios para el estudiante	135
4.2	Metodología GAIA: Codificación del agente	136

4.2.1	Puesta en marcha del contenedor de agentes	136
4.2.2	Funcionamiento del agente	137
4.2.3	Integración del Agente Inteligente al sistema web	137
5	Fase IV: Pruebas	140
5.1	Metodología XP: Pruebas unitarias del sistema	140
5.1.1	Pruebas de conexión a los Web Services	140
5.1.2	Pruebas de conexión a la base de datos	141
5.2	Metodología GAIA: Pruebas unitarias del agente inteligente	142
5.2.1	Pruebas del agente inteligente	142
5.3	Pruebas de Funcionalidad.....	143
g.	Discusión	145
1	Desarrollo de la propuesta alternativa	145
2	Valoración técnica económica ambiental.....	146
2.1	Recursos.....	146
h.	Conclusiones	150
i.	Recomendaciones	151
j.	Bibliografía	152
k.	Anexos.....	157

Índice de Figuras

Figura 1. Arquitectura Cliente Servidor	11
Figura 2. Arquitectura de tres niveles.....	13
Figura 3. El modelo de los Web Services	16
Figura 4. Arquitectura SOAP	18
Figura 5. Estructura de un mensaje SOAP	18
Figura 6. Web Services Description Language	20
Figura 7. Interacción de protocolos	21
Figura 8. Actual proceso de calificación en la Universidad Nacional de Loja	23
Figura 9. Abstracción de un agente a partir de su interacción con el medio	28
Figura 10. Funcionamiento Lenguaje de Comunicación de Agentes	32
Figura 11. Funcionamiento de un Agente Reactivo	34
Figura 12. Agente Reactivo Simple.....	35
Figura 13. Agente Reactivo basado en Modelos.....	35
Figura 14. Agente Reactivo basado en objetivos	36
Figura 15. Agentes Reactivos basados en Utilidad	37
Figura 16. Funcionamiento de un Agente Deliberativo.....	38
Figura 17. Esquema básico de un agente de interfaz	39
Figura 18. Arquitectura de un agente reactivo	41
Figura 19. Estructura de un agente deliberativo	42
Figura 20. Estructura de un agente Híbrido	43
Figura 21. Fases que sigue XP	48
Figura 22. Módulos de la metodología GAIA.....	53
Figura 23. Modelo del agente	73
Figura 24. Patrón de diseño para el sistema.....	79
Figura 25. Modelo Físico de Base de Datos	80
Figura 26. Interfaz Ingreso al Sistema	104
Figura 27. Interfaz agregar nuevas materias.....	105
Figura 28. Interfaz administrar materias y calificaciones	105
Figura 29. Interfaz configurar cantidad y porcentaje de parámetros.....	106
Figura 30. Interfaz añadir calificaciones	107
Figura 31. Interfaz editar calificaciones	107
Figura 32. Interfaz detallar calificaciones	108
Figura 33. Interfaz promediar calificaciones parciales.....	108
Figura 34. Interfaz promediar calificaciones finales.....	109
Figura 35. Interfaz seleccionar parámetros iniciales	110
Figura 36. Presentar gráficas de desempeño académico	110
Figura 37. Presentar registró detallado de calificaciones y promedios	111
Figura 38. Presentar factores que influyen en el desempeño	112
Figura 39. Diagrama de interacción del agente.....	113
Figura 40. Arquitectura del agente inteligente dentro del sistema web.....	114
Figura 41. Página inicial para acceso a los Web Services del SGA	118

Figura 42. Área de configuraciones	127
Figura 43. Configuraciones modificadas	128
Figura 44. Gráfica de predicción para evaluaciones	130
Figura 45. Mensaje sobre calificación predictiva	130
Figura 46. Mensaje de notificación por calificaciones bajas	130
Figura 48. Selección de información académica	131
Figura 49. Añadir nuevo registro	131
Figura 50. Administración de calificaciones	132
Figura 51. Botones para administración de calificaciones	132
Figura 52. Agregación de nuevas calificaciones	133
Figura 53. Editando – eliminando calificaciones	133
Figura 54. Promedios parciales en una materia por parámetro a evaluar	134
Figura 55. Promedios generales en una determinada materia	134
Figura 56. Promedios del estudiante	135
Figura 57. Contenedor de agentes en funcionamiento	136
Figura 58. Arquitectura BDI	137
Figura 59. Creación del agente inteligente	139
Figura 60. Resultados del test a los Web Services	141
Figura 61. Resultados del test de conexión a la base de datos	142
Figura 62. Resultados del test para el agente	143

Índice de tablas

TABLA I. ATRIBUTOS DE UN MENSAJE KQML	30
TABLA II. CATEGORIZACIÓN DE PERFORMATIVAS	30
TABLA III. LISTA DE PERFORMATIVOS.....	31
TABLA IV. PARÁMETROS DE UN MENSAJE FIPA ACL.....	33
TABLA V. COMPARATIVA ENTRE METODOLOGÍAS PARTE 1	51
TABLA VI. COMPARATIVA ENTRE METODOLOGÍAS PARTE 2.....	51
TABLA VII. COMPARATIVA ENTRE METODOLOGÍAS DE AGENTES	54
TABLA VIII HISTORIA DE USUARIO: INGRESAR AL SISTEMA	60
TABLA IX. HISTORIA DE USUARIO GENERAR NUEVO REGISTRO DE CALIFICACIONES.....	61
TABLA X. HISTORIA DE USUARIO: CALIFICAR ESTUDIANTES	62
TABLA XI. HISTORIA DE USUARIO: EDITAR CALIFICACIONES	63
TABLA XII. HISTORIA DE USUARIO: ELIMINAR CALIFICACIONES	64
TABLA XIII. HISTORIA DE USUARIO: CONFIGURAR PARÁMETROS DE CALIFICACIÓN	65
TABLA XIV. HISTORIA DE USUARIO: PRESENTAR CALIFICACIONES DETALLADAS	66
TABLA XV. HISTORIA DE USUARIO: PRESENTAR PROMEDIOS DE CALIFICACIONES.....	67
TABLA XVI. HISTORIA DE USUARIO: SELECCIONAR PARÁMETROS INICIALES.	68
TABLA XVII. HISTORIA DE USUARIO: PRESENTAR GRÁFICAS DE DESEMPEÑO ACADÉMICO	69
TABLA XVIII. HISTORIA DE USUARIO: AJUSTAR FACTORES QUE INFLUYEN EN EL DESEMPEÑO ACADÉMICO	70
TABLA XIX. HISTORIA DE USUARIO: GENERAR UN REGISTRO DETALLADO DE CALIFICACIONES.....	71
TABLA XX. HISTORIA DE USUARIO: PRESENTAR PROMEDIOS AL ESTUDIANTE	72
TABLA XXI. HISTORIA DE USUARIO: PRESENTAR SECCIÓN CON INFORMACIÓN PARA EL MEJORAMIENTO DEL RENDIMIENTO ACADÉMICO	72
TABLA XXII. MODELO DE ROLES DEL AGENTE	74
TABLA XXIII. DEFINICIÓN DE PROTOCOLOS	76
TABLA XXIV. MODELO DE SERVICIOS DEL AGENTE	77
TABLA XXV. DICCIONARIO DE DATOS CAREER	81
TABLA XXVI. DICCIONARIO DE DATOS CLASSWORK.....	81
TABLA XXVII. DICCIONARIO DE DATOS CONFIGURATION	82
TABLA XXVIII. DICCIONARIO DE DATOS EVALUATION	83
TABLA XXIX. DICCIONARIO DE DATOS EXTRACLASSWORK	84
TABLA XXX. DICCIONARIO DE DATOS LABORATORY	84
TABLA XXXI. DICCIONARIO DE DATOS LESSON	85
TABLA XXXII. DICCIONARIO DE DATOS MATTER	86
TABLA XXXIII. DICCIONARIO DE DATOS MATTER_MODULE	86
TABLA XXXIV. DICCIONARIO DE DATOS MATTER_OFFER.....	86
TABLA XXXV. DICCIONARIO DE DATOS MODULE	87
TABLA XXXVI. DICCIONARIO DE DATOS OFFER	87

TABLA XXXVII. DICCCIONARIO DE DATOS PARTICIPATION	88
TABLA XXXVIII. DICCCIONARIO DE DATOS REFERENCE	88
TABLA XXXIX. DICCCIONARIO DE DATOS STUDENT.....	89
TABLA XL. DICCCIONARIO DE DATOS STUDENT_CAREER.....	89
TABLA XLI. DICCCIONARIO DE DATOS STUDENT_MATTER	89
TABLA XLII. DICCCIONARIO DE DATOS STUDENT_MODULE.....	90
TABLA XLIII. DICCCIONARIO DE DATOS TEACHER	90
TABLA XLIV. DICCCIONARIO DE DATOS TEACHER_CARRER	90
TABLA XLV. DICCCIONARIO DE DATOS TEACHER_MATTER.....	91
TABLA XLVI. DICCCIONARIO DE DATOS TEACHER_MODULE	91
TABLA XLVII. DICCCIONARIO DE DATOS PERSONALFACTOR	91
TABLA XLVIII.DICCCIONARIO DE DATOS SOCIALFACTOR.....	93
TABLA XLIX. DICCCIONARIO DE DATOS INSTITUTIONALFACTOR	94
TABLA L. TARJETA CRC CAREER	95
TABLA LI. TARJETA CRC CLASSWORK	95
TABLA LII. TARJETA CRC CONFIGURATION	96
TABLA LIII. TARJETA CRC EVALUATION.....	96
TABLA LIV. TARJETA CRC EXTRACLASSWORK	97
TABLA LV. TARJETA CRC LABORATORY	97
TABLA LVI. TARJETA CRC LESSON	98
TABLA LVII. TARJETA CRC MATTER	98
TABLA LVIII. TARJETA CRC MODULE	99
TABLA LIX. TARJETA CRC OFFER.....	99
TABLA LX. TARJETA CRC PARTICIPATION	100
TABLA LXI. TARJETA CRC REFERENCE	100
TABLA LXII. TARJETA CRC STUDENT	101
TABLA LXIII. TARJETA CRC TEACHER.....	102
TABLA LXIV. TARJETA CRC PERSONALFACTOR	102
TABLA LXV. TARJETA CRC SOCIALFACTOR.....	103
TABLA LXVI. TARJETA CRC INSTITUTIONALFACTOR	103
TABLA LXVII. CÓDIGO NECESARIO PARA CONEXIÓN A LOS WEB SERVICES	119
TABLA LXVIII. PESOS DE FACTORES PERSONALES	119
TABLA LXIX. PESOS DE FACTORES SOCIALES.....	120
TABLA LXX. PESOS DE FACTORES INSTITUCIONALES.....	120
TABLA LXXI. MÉTODO PARA GENERACIÓN DE CALIFICACIONES PREDICTIVAS	121
TABLA LXXII. CÓDIGO PARA GENERACION DE GRÁFICAS DE EVALUACIONES	125
TABLA LXXIII. PARÁMETROS Y PESOS POR DEFECTO.....	126
TABLA LXXIV. AJUSTE DE FACTORES PERSONALES.....	128
TABLA LXXV. AJUSTE DE FACTORES SOCIALES.....	129
TABLA LXXVI. AJUSTE DE FACTORES INSTITUCIONALES	129
TABLA LXXVII. CREACIÓN DEL AGENTE INTELIGENTE	138
TABLA LXXVIII. CREACIÓN DE LA CLASE CONTROLADORA PARA EL AGENTE	138
TABLA LXXIX. DECLARACIÓN DEL BEAN EN EL CONTENEDOR DE SPRING....	139
TABLA LXXX. TEST UNITARIO PARA LA CONEXIÓN A LOS WEB SERVICES	140
TABLA LXXXI. TEST DE CONEXIÓN A LA BASE DE DATOS	141

TABLA LXXXII. TEST UNITARIO PARA EL AGENTE INTELIGENTE.....	142
TABLA LXXXIII. CALIFICACIONES REALES DE ESTUDIANTES	143
TABLA LXXXIV. GENERACIÓN DE CALIFICACIONES PARA LA ÚLTIMA EVALUACIÓN	144
TABLA LXXXVI. TALENTO HUMANO.....	146
TABLA LXXXVII. RECURSOS MATERIALES	147
TABLA LXXXVIII. RECURSOS HARDWARE	147
TABLA LXXXIX. RECURSOS SOFTWARE	148
TABLA XC. RECURSOS DE COMUNICACIÓN	148
TABLA XCI. COSTO TOTAL DEL PROYECTO.....	149

c. Introducción

La Universidad Nacional de Loja tiene como objetivo principal la formación de profesionales que brinden soluciones a los problemas que afronta nuestra sociedad mediante el uso de tecnologías encaminadas a la automatización de procesos y mejoramiento de la calidad de vida de las personas.

Hoy en día el proceso de enseñanza aprendizaje ha experimentado muchos cambios gracias a los avances tecnológicos y una mejor formación por parte de los docentes, pero a pesar de todo se presentan ciertas falencias mayoritariamente relacionadas con la socialización de calificaciones con los estudiantes durante el transcurso de la materia.

Es así que se aprecian los siguientes problemas:

- Desconocimiento de calificaciones durante el transcurso de la materia por parte del estudiante. Esto se da debido a que existen docentes que no socializan los resultados en lo concerniente a trabajos, lecciones, evaluaciones, exámenes, etc. lo que origina un gran vacío en el estudiante ya que solo puede obtener su progreso partiendo de un aproximado según su punto de vista.
- Inexistencia de un medio adecuado que permita a los docentes llevar un registro de calificaciones y a su vez de la posibilidad de compartirlas con sus estudiantes, con el fin de permitirles realizar algún reclamo si fuere necesario.
- El estudiante posee poca información de su progreso durante el transcurso del módulo, menos aún, una predicción aproximada de cuál será su desempeño académico a futuro tomando como base sus calificaciones presentes.

Teniendo estos aspectos como base y la aplicabilidad dentro de la Universidad, nació la idea de implementar un sistema inteligente capaz de permitir al estudiante conocer su desempeño académico actual y futuro a más de facilitar al docente un registro detallado de calificaciones de los estudiantes durante el transcurso de la materia. Es así que se plantearon los siguientes objetivos:

- Analizar y diseñar las creencias, deseos e intenciones de Agentes Deliberativos para la predicción del desempeño académico de los estudiantes de la Universidad Nacional de Loja.
- Desarrollar un sistema web con conexión a los Web Services del Sistema de Gestión Académico (SGA) para obtener resultados.
- Integrar Agentes Deliberativos al Sistema Web para el correcto funcionamiento.

Para alcanzar dichos objetivos se procedió al desarrollo de un sistema inteligente orientado a la web que permita al docente administrar de manera eficiente las calificaciones de sus educandos, y para el estudiante existe la generación de gráficas predictivas de su desempeño académico a más de un registro detallado sobre sus calificaciones en todo momento. El documento consta de tres capítulos que se detallan a continuación:

Capítulo 1: Se enfoca en el ámbito del desarrollo de Sistemas Web desde un punto de vista general. Otro aspecto a tratarse en este capítulo es lo relacionado al desempeño académico de los estudiantes en el cual se abordan conceptos básicos y ejemplos.

Capítulo 2: Esta destinado enteramente a los Agentes Inteligentes, su estructura, funcionamiento, utilidad y demás aspectos característicos de los mismos.

Capítulo 3: Orientado a las metodologías más utilizadas tanto para el desarrollo de sistemas como para agentes inteligentes. Es así que se detallan sus fases y el proceso a seguir para aplicar cada una de ellas.

d. Revisión de Literatura

CAPÍTULO I

1 Sistemas Web y Desempeño Académico

En el presente capítulo se abordará todo lo concerniente a Sistemas Web Y Sistemas de Desempeño Académico como base para el desarrollo del presente Proyecto de Fin de Carrera. Dichos temas tienen como finalidad brindar una idea general y conceptos claros de lo que es y cómo funciona un Sistema Web particularmente tomando como referencia los Sistemas de Desempeño Académico utilizados comúnmente en universidades demás centros educativos.

1.1 Introducción

Actualmente gracias al avance de la tecnología y el internet, cada vez son más y de mejor calidad los Sistemas Web, creados principalmente para empresas que tienen la necesidad de anunciar y brindar sus servicios haciendo uso de la red como medio de comunicación. Por otra parte en lo académico, existe un número limitado de Sistemas principalmente orientados a la parte de administración académica como los conocidos sistemas de gestión académica (SGA).

1.2 ¿Qué es el desempeño académico?

El desempeño académico hace referencia a la evaluación del conocimiento adquirido en el ámbito escolar, terciario o universitario. Un estudiante con buen rendimiento académico es aquél que obtiene calificaciones positivas en los exámenes que debe rendir a lo largo de una cursada. [1]

En otras palabras es una medida de las capacidades del alumno, que expresa lo que éste ha aprendido a lo largo del proceso formativo. También supone la capacidad del alumno para responder a los estímulos educativos. En este sentido, el rendimiento académico está vinculado a la aptitud.

Entre los factores que influyen en el rendimiento académico se puede encontrar la dificultad de algunas asignaturas, la cantidad de las mismas, así como de exámenes

que pueden coincidir en una misma fecha, entre otros más que pueden llevar a un alumno a mostrar un pobre rendimiento académico.

Otras cuestiones están directamente relacionadas al factor psicológico, como la poca motivación, el desinterés o las distracciones en clase, que dificultan la comprensión de los conocimientos impartidos por el docente y termina afectando al rendimiento académico a la hora de las evaluaciones [1].

Por otra parte, el desempeño académico puede estar asociado a la subjetividad del docente cuando corrige. Ciertas materias, en especial aquéllas que pertenecen a las ciencias sociales, pueden generar distintas interpretaciones o explicaciones, que el profesor debe saber analizar en la corrección para determinar si el estudiante ha comprendido o no los conceptos. [1].

En todos los casos, los especialistas recomiendan la adopción de hábitos de estudio saludables para mejorar el rendimiento escolar; por ejemplo, no estudiar muchas horas seguidas en la noche previa al examen, sino repartir el tiempo dedicado al estudio [2].

1.3 Factores que influyen en el desempeño académico

Existen numerosos factores que intervienen en lo relacionado al desempeño académico ya sean estos internos o externos con respecto al estudiante. Los mismos se pueden clasificar en tres tipos distintos: factores personales, sociales e institucionales. A continuación se describen los más importantes y su influencia en el rendimiento de los estudiantes.

1.3.1 Determinantes personales [3]

Estas relacionados estrechamente con aspectos de índole personal que influyen directamente en el rendimiento académico. Dentro de estos podemos citar:

La competencia cognitiva: Es la autoevaluación de la propia capacidad del individuo para cumplir una determinada tarea cognitiva es decir su percepción sobre su capacidad y habilidades intelectuales.

La motivación: Es un factor de gran importancia ya que influye directamente en el desempeño del estudiante y su facilidad de aprendizaje.

El autoconcepto académico: Se define como el conjunto de percepciones y creencias que una persona posee sobre sí misma, es así como la mayoría de variables personales

que orientan la motivación parten de las creencias y percepciones que el individuo tiene sobre aspectos cognitivos.

Autoeficacia percibida: Se puede presentar casos en los que el estudiante carece de autoeficacia y se presenta principalmente por falta de motivación que permita al estudiante cumplir con un desempeño académico aceptable. Está asociado a estados de agotamiento, desinterés y falta de proyección con sus estudios.

Bienestar psicológico: está relacionado con estados de satisfacción y felicidad con respecto a sus estudios y es común en aquellos estudiantes que proyectan el no abandonar sus estudios.

Satisfacción con respecto a los estudios: Hace referencia al bienestar del estudiante en relación con sus estudios, e implica una actitud positiva hacia la universidad y la carrera.

Asistencia a clases: Se refiere a la presencia del alumno en las distintas lecciones, participaciones, etc. relacionadas con el día a día en el proceso de enseñanza aprendizaje.

Inteligencia: Es una de las variables más estudiadas dentro de los determinantes de índole personal. Incluye pruebas de comprensión verbal y razonamiento matemático (pruebas psicométricas). La inteligencia es un buen predictor de los resultados académicos, que sobresale en el rendimiento académico, lo cual produce una relación significativa entre inteligencia y rendimiento académico.

Aptitudes: Se asocian a la habilidad del estudiante para realizar determinadas tareas de una manera fácil.

Género: No se puede afirmar del todo una relación directa con el rendimiento académico y el sexo; sin embargo, hay estudios que le dan a la mujer una ligera tendencia al rendimiento superior que a los hombres.

Edad: Influye en menor medida que las demás pero después de algunos estudios se concluye que el estudiantes con menor edad logran mejor rendimiento que los que son de mayor edad.

Estado civil: Aunque su influencia no es grande, se puede considerar que el estado civil influye dependiendo la situación en la que se encuentre el estudiante.

Número de cargas familiares: Esta claro que cuanto mayor sea el número de cargas familiares a cargo de un estudiante padre de familia, incrementará el grado de responsabilidad y disminuirá el rendimiento académico llegando al extremo de dejar los estudios con el fin de trabajar.

Formación previa a la Universidad: Diferentes estudios explican que el rendimiento académico previo a la universidad es un claro indicador del éxito académico en los estudios universitarios. Se han estudiado en esta variable indicadores como las calificaciones obtenidas en la educación secundaria, incluyendo pruebas de bachillerato.

Nota de ingreso a la Universidad: Estudios realizados en la enseñanza superior asociados al rendimiento académico enfatizan el valor de la nota obtenida en las pruebas de admisión a la universidad como un predictor de los más importantes en el rendimiento académico, junto con los rendimientos académicos previos a la universidad.

1.3.2 Determinantes Sociales [3]

Son aquellos factores asociados al rendimiento académico de índole social que interactúa con la vida académica del estudiante. Podemos citar a:

Diferencias sociales: Ciertas diferencias en cuanto a los social y económico pueden llegar a influenciar en la parte académica, aunque no es estrictamente probable que pueda ser un aspecto negativo.

Entorno familiar: Ocupa un lugar importante. Se entiende por entorno familiar un conjunto de interacciones propias de la convivencia familiar, que afectan el desarrollo del individuo, manifestándose también en la vida académica.

Contexto socioeconómico: Puede influir en el aprendizaje y en el desempeño del estudiante en la parte académica y está relacionado con el nivel económico del mismo.

Variable demográfica: condiciones como la zona geográfica de procedencia, zona geográfica en la que vive el estudiante en época lectiva entre otros, son factores que eventualmente se relacionan con el rendimiento académico en forma positiva o negativa.

1.3.3 Determinantes institucionales [3]

Son componentes no personales que intervienen en el proceso educativo, donde al interactuar con los componentes personales influye en el rendimiento académico alcanzado, dentro de estos se encuentran: metodologías docentes, horarios de las distintas materias, cantidad de alumnos por profesor, dificultad de las distintas materias entre otros que seguidamente se abordarán en forma individual.

Elección de los estudios según interés del estudiante: Se refiere a la forma o vía por la cual el estudiante ingresó a la carrera, si fue su primera elección, si fue por traslado de carrera o por no haber encontrado cupo en otra carrera por ejemplo.

Complejidad de los estudios: Se refiere a la dificultad de algunas materias de las distintas carreras o áreas académicas que usualmente las universidades las clasifican basándose en estadísticas de aquellas materias con mayores índices de reprobación.

Condiciones institucionales: Elementos como las condiciones de las aulas, servicios, plan de estudios y formación del profesorado, se presentan como obstaculizadores del rendimiento académico; que a su vez también pueden ser facilitadores.

Servicios institucionales de apoyo: Se refiere a todos aquellos servicios que la institución ofrece al estudiantado, principalmente según su condición económica, como lo son: sistemas de becas, servicio de préstamo de libros, asistencia médica, apoyo psicológico, entre otros.

Ambiente estudiantil: Un ambiente marcado por una excesiva competitividad con los compañeros puede ser un factor tanto obstaculizador como facilitador del rendimiento académico. Se destaca la solidaridad, el compañerismo, y el apoyo social como importantes elementos que inciden positivamente.

Relación estudiante – profesor: Las expectativas que el estudiante tiene sobre las relaciones con sus profesores y con sus compañeros de clase son factores importantes que intervienen en los resultados académicos.

Todos estos factores influyen de alguna manera ya sea positiva o negativamente en el desempeño académico de los estudiantes universitarios.

1.4 Sistemas Web

Gracias al avance de la tecnología en los últimos años, las páginas web que dominaban la Internet han quedado obsoletas, siendo remplazadas por verdaderos Sistemas Web o también llamadas Aplicaciones Web orientadas a cubrir las necesidades de las personas y empresas.

1.4.1 Antecedentes

A principios de los años noventa, el concepto de CMS era desconocido. El principal factor que influyó en su aparición fue, en gran medida, que los sistemas de información dependieran de las personas que procesan los datos; es decir, que durante el proceso de generación de contenido se necesitaban de algunas personas de diversos perfiles (autores, editores, diseñadores y publicadores). Las tareas asignadas a todos los involucrados, eran realizadas cada vez con tiempos de retraso muy altos. Todo ello, comenzó a conspirar contra la efectividad y evolución de los sistemas, debido a que el tiempo de disposición de la información para los usuarios finales aumentaba al tener que circular por cada uno de los puntos de la generación de contenido. Otro problema que aconteció como consecuencia de esta dependencia era que cualquier artículo publicado luego de su fecha de caducidad seguía siendo presentado en los sitios web, lo que no era bien visto por los usuarios [4].

Desde el punto de vista de las organizaciones, los procesos de integración de información generalmente se realizaban con la idea de que la información es un recurso común para todos sus departamentos, por tanto de acuerdo a ese precedente, al momento de generar contenido las organizaciones pretendían vincular a todas sus áreas, provocando que las organizaciones creen estrategias ligadas a la integración de recursos y al flujo constante de información [4].

Debido a la necesidad de solucionar los problemas que presentaron los sistemas de información [5], las organizaciones comenzaron a apoyar sus sistemas de información en sistemas de administración de contenidos; se dieron cuenta de que las aplicaciones que les brindaban los sistemas de información apoyadas en páginas web estáticas eran

1.4.2 La Web

La Web se puede considerar como una plataforma o “sistema operativo” en el cual los recursos están distribuidos en la Red y están siendo extendidos en todo momento con posibilidades ilimitadas. Se ha hecho popular con aplicaciones tales como clientes de correo, buscadores, portales, foros, chats, IRC, RSS, blogs, etc. Además de estas aplicaciones de propósito general, existe adicionalmente una gran diversidad de soluciones que se acomodan al ambiente web, como son: Administradores de contenido (CMS), Administrador de proyectos, Suites para trabajo colaborativo, Administración de relaciones con el cliente (CRM), ERP, etc. [6]

La Web se reinventa día a día para mejorar dando ejemplos claros como google docs, juegos en línea, google maps, etc. que permiten desde el navegador realizar tareas complejas que antes únicamente eran posibles con aplicaciones locales o de escritorio.

Es cierto que la arquitectura cliente-servidor de la web ha ofrecido muchas ventajas, pero también es cierto que carece de la riqueza gráfica de las aplicaciones de escritorio que cuentan con controles inteligentes que dan mayor fluidez al trabajo del usuario. Esto ha sido resuelto con varias estrategias o tecnologías tales como AJAX, Flash y Web 2, entre otras. Así que en vez de ir perdiendo fuerza debido a la pobreza en sus interfaces gráficas, la Web busca alternativas que le permitan ofrecer todas sus ventajas, pero con la posibilidad de ofrecer controles visuales más amigables al trabajo del usuario [6].

1.4.3 Definición

Una Sistema Web es cualquier aplicación que es accedida vía web por una red como internet o una intranet. En general, el término también se utiliza para designar aquellos programas informáticos que son ejecutados en el entorno del navegador (por ejemplo, un applet de Java) o codificado con algún lenguaje soportado por el navegador (como JavaScript, combinado con HTML); confiándose en el navegador web para que reproduzca (renderice) la aplicación [7].

Una de las ventajas de las aplicaciones web cargadas desde internet (u otra red) es la facilidad de mantener y actualizar dichas aplicaciones sin la necesidad de distribuir e instalar un software en, potencialmente, miles de clientes. También la posibilidad de ser ejecutadas en múltiples plataformas.

1.4.4 Arquitecturas

Se estudiarán dos tipos de arquitecturas que en general son las más utilizadas para el desarrollo de Aplicaciones Web

1.4.4.1 Arquitectura Cliente-Servidor

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema. [8]

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

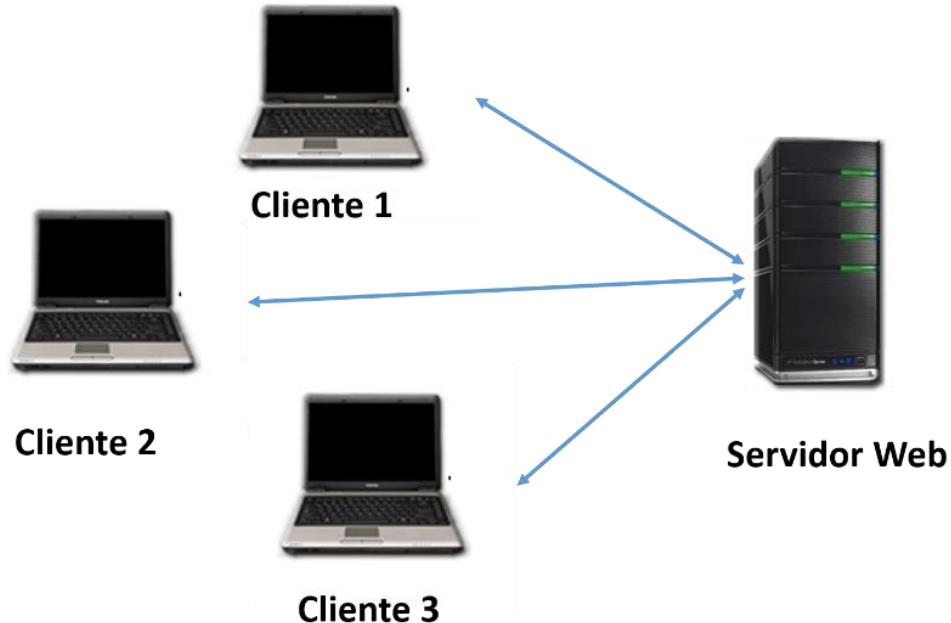


Figura 1. Arquitectura Cliente Servidor

Los clientes están conectados a un servidor, en el que se centralizan los diversos recursos y aplicaciones con que se cuenta; y que los pone a disposición de los clientes cada vez que estos son solicitados. Esto significa que todas las gestiones que se realizan se concentran en el servidor, de manera que en él se disponen los requerimientos provenientes de los clientes que tienen prioridad, los archivos que son de uso público y los que son de uso restringido, los archivos que son de sólo lectura y los que, por el contrario, pueden ser modificados, etc. [8]

a. Ventajas de la arquitectura cliente-servidor:

- El desarrollo de aplicaciones en un ambiente cliente-servidor funciona adecuadamente y cumple los requerimientos.
- Las herramientas para el desarrollo de sistemas cliente-servidor son robustas y ampliamente evaluadas.
- Las técnicas de ingeniería de software de prototipado se aplican fácilmente. Las soluciones de dos capas trabajan en ambientes no dinámicos estables.

b. Desventajas de la arquitectura cliente-servidor:

- Las aplicaciones cliente-servidor requieren control excesivo de las versiones y demandan esfuerzo de distribución de la aplicación cuando se realizan cambios. Esto se da debido al hecho de que la mayor parte de la aplicación lógica existe en la estación de trabajo del cliente.
- No son completamente eficientes en términos de rendimiento.
- No se ejecutan bien en organizaciones rápidamente cambiantes
- La seguridad del sistema en un diseño cliente-servidor es compleja y a menudo requiere administración de las bases de datos; debido al número de dispositivos con acceso directo al ambiente de esas bases de datos.
- Las herramientas del cliente y de la base de datos, utilizadas en diseños de dos capas (cliente-servidor), constantemente están cambiando. La dependencia a largo plazo de cualquier herramienta, puede complicar el escalamiento futuro o las implementaciones

1.4.4.2 Arquitectura de Tres Niveles

La arquitectura de las aplicaciones Web suelen presentar un esquema de tres niveles:

- El primer nivel consiste en la capa de presentación que incluye no sólo el navegador, sino también el servidor web que es el responsable de presentar los datos un formato adecuado.
- El segundo nivel está referido habitualmente a algún tipo de programa o script.
- El tercer nivel proporciona al segundo los datos necesarios para su ejecución.

Su funcionamiento es el siguiente: una aplicación Web típica recogerá datos del usuario (primer nivel), los enviará al servidor, que ejecutará un programa (segundo y tercer nivel) y cuyo resultado será formateado y presentado al usuario en el navegador (primer nivel otra vez)



Figura 2. Arquitectura de tres niveles

Las diferentes componentes suelen ser:

- Un **cliente (Capa 1)**, es decir, el equipo que solicita los recursos, equipado con una interfaz de usuario (generalmente un navegador web) para la presentación.
- El **servidor de aplicaciones (Capa 2)** (también denominado software intermedio), cuya tarea es proporcionar los recursos solicitados, pero que requiere de otro servidor para hacerlo.
- El **servidor de datos (Capa 3)**, que proporciona al servidor de aplicaciones los datos que éste le solicitó [6].

Ventajas de la arquitectura de tres capas:

- Las llamadas de la interfaz del usuario en la estación de trabajo, al servidor de capa intermedia, son más flexibles que en el diseño de cliente-servidor, ya que la estación solo necesita transferir parámetros a la capa intermedia.

- Con la arquitectura de tres capas, la interfaz del cliente no es requerida para comprender o comunicarse con el receptor de los datos. Por lo tanto, esa estructura de los datos puede ser modificada sin cambiar la interfaz del usuario en la PC.
- El código de la capa intermedia puede ser reutilizado por múltiples aplicaciones si está diseñado en formato modular.
- La separación de roles en tres capas, hace más fácil reemplazar o modificar una capa sin afectar a los módulos restantes.

Desventajas de las Arquitecturas de Tres Capas

- Los ambientes de tres capas pueden incrementar el tráfico en la red y requiere más balance de carga u tolerancia a las fallas.
- La estandarización entre diferentes proveedores ha sido lenta en desarrollarse. Muchas organizaciones son forzadas a escoger uno en lugar de otro, mientras que cada uno ofrece sus propias y distintas ventajas.

1.5 Web Services

Debido al crecimiento del Internet a niveles tan altos y el impacto causado por las tecnologías de la información en las últimas dos décadas, la manera de hacer negocios y la comunicación entre las personas y las empresas cambió de rotundamente. Bajo este contexto se hacía cada vez mayor la necesidad de integrar y compartir información entre distintas plataformas tanto de software y hardware.

En los últimos años ha surgido con mucha fuerza el concepto de “web services”, principalmente debido a la necesidad de intercambiar información independientemente de la plataforma o el lenguaje con que se haya desarrollado un sistema con la finalidad de construir una arquitectura orientada a servicios [9].

1.5.1 ¿Qué es un Web Service?

Web Services es una aplicación computacional determinada por un identificador cuyas interfaces públicas y conexiones están definidas y descritas usando XML de tal manera que otros sistemas y software puedan reconocerla. Estos sistemas pueden interactuar con los Web Services usando XML para obtener contenidos en mensajes transportados por los protocolos Internet [10].

1.5.2 Características [11]

- Un Web Service puede ser registrado para poder dejarlo a disposición de otros usuarios y para que los mismos puedan localizarlo. Un mecanismo para registrar estos servicios es por medio de UDDI, sigla que corresponde a **Universal Description, Discovery and Integration**, un “repositorio de Web Services”. Para registrar un servicio tendrás que tener en cuenta que debes suministrar la información de tu empresa, en qué categorías ubicarías tu servicio y la interfaz a utilizar para consumir este servicio.
- El mecanismo utilizado por un Web Service para especificar de qué forma hay que proporcionarle los datos, de manera tal que cualquiera pueda interaccionar con el mismo, es por medio de lenguaje XML. Esta información se almacena en un archivo llamado WSDL (Web Services Description Language), el cual contiene un documento XML junto con la descripción de ciertos mensajes SOAP y cómo deben intercambiarse, así como también dónde está el recurso del servicio y con qué protocolo debe dialogar quien lo consume.
- El protocolo de comunicación utilizado es el SOAP generalmente, el cual es relativamente sencillo de utilizar.
- Los Web Services utilizan protocolos comúnmente conocidos y difundidos tales como el formato XML, TCP/IP como protocolo de transporte y HTTP como protocolo de transferencia de hipertexto.

Para el correcto funcionamiento de los Web Services se hace uso de SOAP que es un protocolo que define el formato XML para los mensajes de intercambio en el uso de un Web Service. Adicionalmente soporta RPC, además es posible definir un mensaje HTTP y este punto es de especial interés puesto que el protocolo imprescindible para Internet es HTTP.

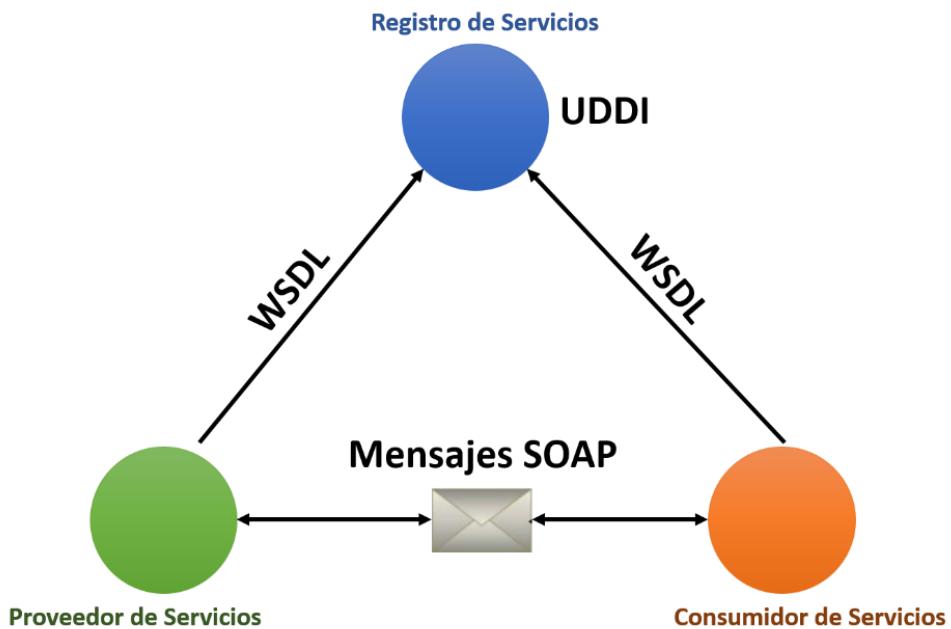


Figura 3. El modelo de los Web Services

1.5.3 Ventajas de los Web Services

Los web services operan utilizando estándares abiertos, lo que permite la comunicación entre aplicaciones escritas en diferentes lenguajes y entre diferentes plataformas.

- Un web service puede ser utilizado para la comunicación con múltiples compañías.
- Comparativamente con otras soluciones, los web services son fáciles y no muy costosos de implementar, esto porque utiliza la infraestructura existente.
- Los web services pueden reducir significativamente los costos de comunicaciones negocio a negocio (Business-to-Bussines o B2B) e integración de aplicaciones ofreciendo así un tangible retorno de inversión.

La mayor ventaja es la de usar estándares abiertos, Word Wide Web Consortium (W3C) [12] es una organización que define tecnologías para el Web, está encargada de asegurar que tales especificaciones se mantengan abiertas a cualquier vendedor. Además Microsoft e IBM están promoviendo la interoperabilidad entre implementaciones de web services y son dos de los miembros fundadores de Web Services Interoperability Organization (WS-I) [13].

1.5.4 Tecnologías utilizadas en los Web Services

A continuación se aborda las tecnologías que comúnmente son utilizadas en los web Services y cuál es el papel que desempeñan.

1.5.4.1 SOAP

Simple Object Access Protocol (SOAP) es un protocolo ligero basado en XML para intercambiar información estructurada y con tipo. SOAP es el que permite usar la funcionalidad de máquinas remotas sin necesidad de saber algo específico sobre la máquina. XML es usado en los Web Services para representar datos, y los datos necesitan tener un esquema en común, para esto SOAP es el que define el esquema.

En un intercambio de mensajes SOAP se procede de la siguiente forma [13]:

- La aplicación del consumidor crea un mensaje SOAP que es una petición para invocar al web service provisto por el proveedor de servicio.
- El documento XML en el cuerpo del mensaje puede ser una petición SOAP RPC como es indicado en la descripción del servicio.
- El cliente SOAP interactúa con el protocolo de red apropiado como por ejemplo HTTP para enviar el mensaje de red.
- Uno de los propósitos del mensaje SOAP es ejecutar un procedimiento a través de la red, es por eso que un mensaje SOAP enviado por la red hacia un web service es un RPC.
- Cuando se entrega el mensaje al proveedor de servicio, el servidor SOAP enruta el mensaje hacia el web service y es el responsable de convertir el mensaje XML en objetos comprensibles por el lenguaje de programación utilizado donde se utilizan las reglas de codificación incluidas en el mensaje SOAP.
- El web service es el responsable de procesar el mensaje y conceder una respuesta que es también un mensaje SOAP e igualmente se envía el mensaje por la red con el protocolo establecido.
- Cuando la respuesta llega, el mensaje XML es convertido en objetos específicos del lenguaje de programación del consumidor y es presentado a la aplicación correspondiente.

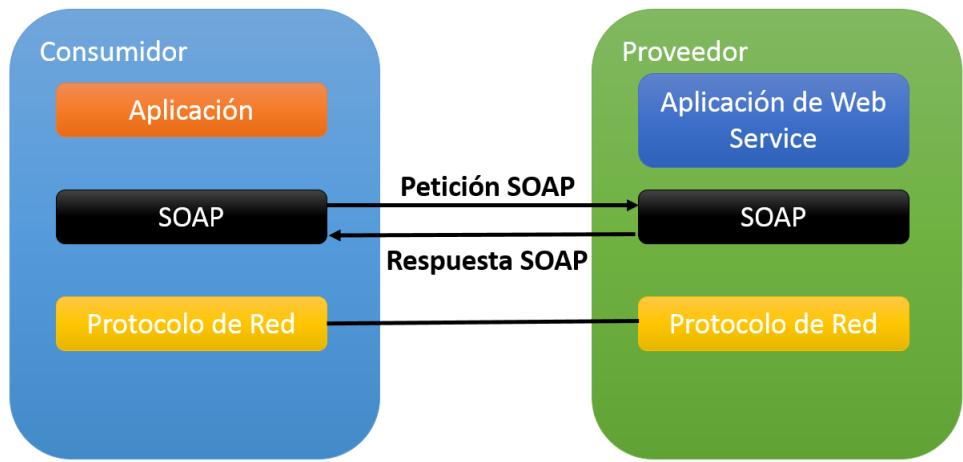


Figura 4. Arquitectura SOAP [13]

a. Estructura de un mensaje SOAP

Un mensaje SOAP se codifica como un documento XML, que consta de un elemento **<Envelope>**, que contiene un elemento **<Header>** opcional y un elemento **<Body>** obligatorio. El elemento **<Fault>**, que se encuentra en **<Body>**, se utiliza para notificar errores [14]

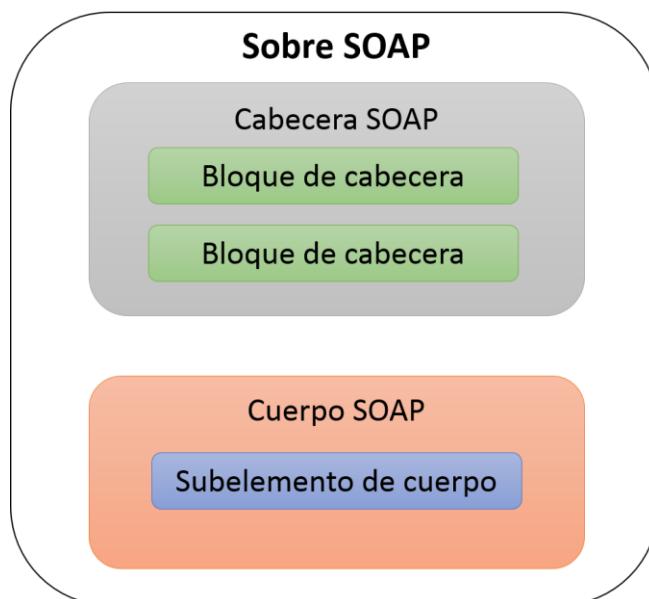


Figura 5. Estructura de un mensaje SOAP [14]

El sobre SOAP: <Envelope> es el elemento raíz en cada mensaje SOAP y contiene dos elementos hijo, un elemento <Header> opcional y un elemento <Body> obligatorio.

La cabecera SOAP: <Header> es un subelemento opcional del sobre SOAP y se utiliza para pasar información relacionada con la aplicación que los nodos SOAP van a procesar a lo largo de la vía de acceso del mensaje; consulte La cabecera SOAP.

El cuerpo SOAP: <Body> es un subelemento obligatorio del sobre SOAP, que contiene información dirigida al destinatario final del mensaje; consulte El cuerpo SOAP.

El error SOAP: <Fault> es un subelemento del cuerpo SOAP, que se utiliza para notificar errores; consulte El error SOAP.

Los elementos XML de <Header> y <Body> están definidos por las aplicaciones que hacen uso de ellos, aunque la especificación SOAP establece algunas restricciones en su estructura.

1.5.4.2 WSDL

Es por la descripción del servicio que un proveedor comunica todas las especificaciones para que el consumidor del servicio invoque al web service. Para este fin nace el Web Services Description Language (WSDL) el cual es un lenguaje basado en XML que es utilizado por la aplicación cliente para obtener información técnica del web service con la cual se puedan comunicar [13].

Muchos web services publican el documento WSDL por el Internet que contiene definiciones que describen al web service en formato XML. El documento WSDL especifica las capacidades que tiene el servicio, su ubicación en la web e instrucciones sobre cómo acceder al web service. Además define cuál va ser la estructura de los mensajes que el web service puede enviar o recibir.

Cuando un web service es publicado, el administrador publica un enlace hacia el documento WSDL del web service en alguno de los registros XML u otro repositorio, entonces, el archivo WSDL está disponible cuando una aplicación busca el registro y logra ubicar el Web Service. El cliente accede a documento WSDL para obtener la información sobre el web service y para crear los mensajes SOAP con la correcta

estructura. Utilizando esta información, la aplicación cliente invoca al Web service. El documento WSDL puede estar publicado ya sea en un repositorio o en cualquier otro lugar ya que es un URL que direcciona hacia este documento [13].

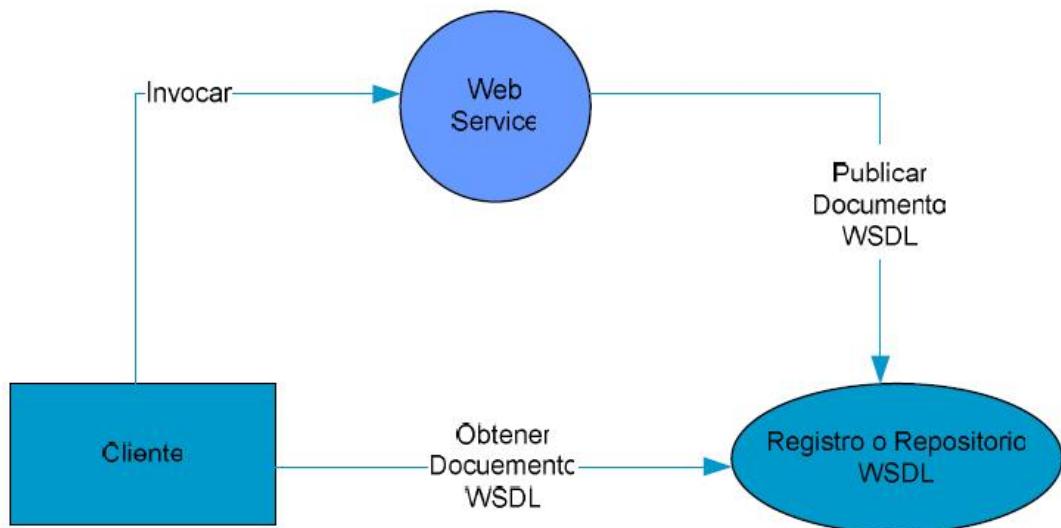


Figura 6. Web Services Description Language [13]

1.5.4.3 UDDI

Universal Description, Discovery and Integration (UDDI) permite la creación de registros web service para ser buscados por consumidores [15].

Las especificaciones UDDI definen la manera para publicar y descubrir información sobre los web services. Descubrimiento (Discovery) es el proceso por el cual se ubica un web service mediante registros. Usando un registro, una empresa puede localizar todos los servicios disponibles requeridos y puede comparar entre los servicios ofrecidos para ubicar el que cumpla sus requerimientos.

La información de UDDI se aloja en nodos de operador, empresas que se han comprometido a ejecutar un nodo público conforme a la especificación que rige el consorcio UDDI.org [15]. En la actualidad existen dos nodos públicos que se ajustan a la versión 1 de la especificación UDDI: Microsoft aloja uno e IBM el otro. Hewlett Packard se ha comprometido a alojar un nodo bajo la versión 2 de la especificación. Los

operadores del host deben replicar datos entre ellos a través de un canal seguro, para conseguir la redundancia de la información en el registro UDDI. Se pueden publicar los datos en un nodo y descubrirlos en otro tras la réplica. Actualmente, la réplica se produce cada 24 horas. En el futuro, este intervalo entre réplicas se reducirá, ya que habrá más aplicaciones que dependan de los datos de UDDI [15].

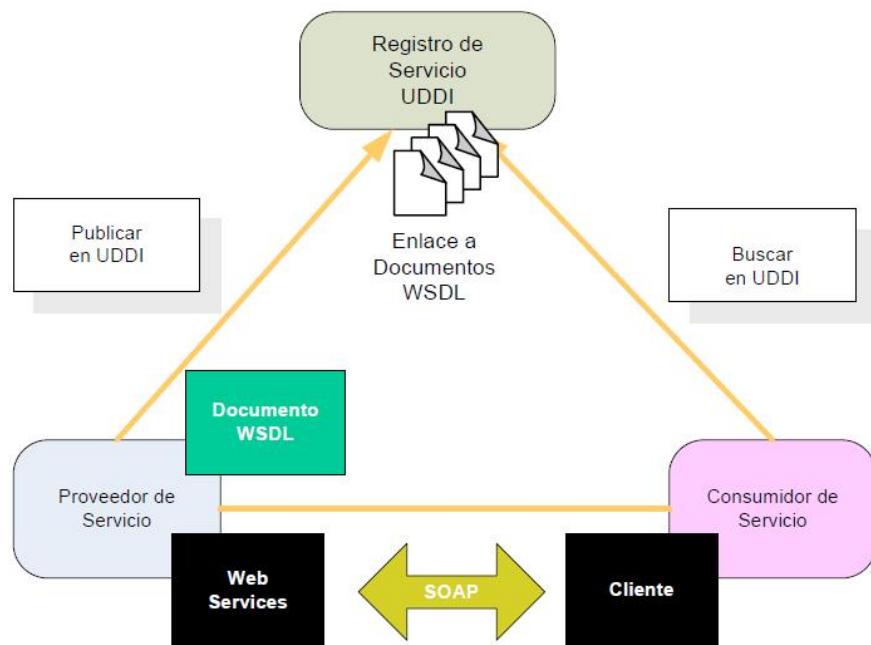


Figura 7. Interacción de protocolos [13]

1.6 Sistemas para el Desempeño Académico

Los Sistemas para el Desempeño Académico existentes en el mercado están encaminados a llevar un control tanto del progreso como del desempeño actual del estudiante. En la web se presentan algunas opciones que analizaremos a continuación.

1.6.1 CADI [16]

Desarrollada por Investigadores de la Universidad Politécnica de Cartagena (UPCT), CADI es un complemento para los alumnos y les permite preguntar y aportar sus opiniones sin tener que hablar en público, ya que esto supone una importante barrera y prefieren quedarse con dudas y no aprender antes que tener que intervenir en clase.

Además, el hecho de tener que plantear la pregunta a través de un móvil, tableta o portátil obliga al alumnos a estructurar la cuestión y pensarla, lo que es también un "ejercicio muy constructivo" para el aprendizaje.

A diferencia de otras herramientas docentes, este software funciona con todo tipo de sistemas operativos sin necesidad de instalar ninguna aplicación, y funciona a través de la red wifi, pero no conecta el dispositivo a internet, lo que evita que el alumno se distraiga consultando otras páginas web.

Además de permitir hacer preguntas, sirve para descargar apuntes, realizar test, exámenes, controles de asistencia y distribuir documentos electrónicos, entre otras funciones.

1.6.2 Agentes inteligentes y desempeño de los estudiantes.

En este trabajo los autores se valen de la utilización de agentes inteligentes móviles para medir los indicadores que influyen en el desempeño académico de los estudiantes [17]

1.7 Situación Actual

Actualmente nuestra Universidad no cuenta con un sistema específicamente enfocado al registro de calificaciones durante el módulo y menos aún un sistema para la predicción del desempeño académico de los estudiantes, el único registro de calificaciones con el que se cuenta es el SGA (Sistema de Gestión Académica) pero su principal falencia es que únicamente da la posibilidad de ingresar calificaciones de las materias una vez finalizado el ciclo es decir solo el promedio final en una materia específica.

A continuación se presenta una imagen que describe gráficamente como es la administración de calificación que lleva el SGA y que únicamente está disponible una vez finalizado el módulo.

Acreditaciones		Nota final de cada materia		
Unidad / Taller / Curso	Obligatoria	Nota Ingresada	Nota Ponderada	
Modelamiento Matemático	SI	0.00		
Lenguajes Formales y Teoría de Autómatas	SI	0.00		
Compiladores	SI	0.00		
Sistemas de Información I	SI	0.00		
Sistemas de Información II	SI	0.00		
Ingeniería de Software	SI	0.00		
Proceso Investigativo	NO	0.00		
Total				

Figura 8. Actual proceso de calificación en la Universidad Nacional de Loja

CAPÍTULO II

2 Agentes inteligentes

El presente capítulo pretende adentrarnos al mundo de la Inteligencia Artificial, específicamente al ámbito de los Agentes Inteligentes y todo lo referente a los mismos de una manera clara y concisa.

2.1 Introducción

A medida que evoluciona la Inteligencia Artificial (IA) encontramos una gran variedad de definiciones. Es así que puede ser entendida como la disciplina que estudia cómo lograr que las computadoras actúen racionalmente, es decir, buscan lograr que éstas se asemejen cada vez más al comportamiento humano.

De ahí que uno de los temas interesante a tratar dentro de esta ciencia es la de agentes inteligentes que operan en cualquier entorno o ambiente y que pueden ser llamados inteligentes [18].

Para entender de una mejor manera propongamos un ejemplo [19]: Supongamos que al llegar por la mañana al trabajo y luego de sentarse al teclado de su ordenador y ejecutar la aplicación de correo electrónico preferida, nota una inundación de decenas de correos electrónicos. Algunos de esos mensajes contienen información que puede conducir a una venta, o bien son simplemente un mensaje del jefe ordenando taxativamente hacer algo. Otros, sin embargo, contienen correo basura, publicidad no deseada, o son mensajes de una lista de correo electrónico a la que últimamente no le prestamos atención. El programa, además, solo nos proporciona información mínima sobre el contenido del mensaje: a veces el nombre del autor, a veces simplemente su dirección de correo, a veces una línea de asunto o tema que tiene que ver con el contenido, pero a veces no.

Por el contrario imaginemos el siguiente escenario: al sentarnos al teclado del ordenador, una ventana se abre en pantalla y nos muestra dos o tres mensajes de correo, clasificados como importantes o urgentes. También nos da la opción de leer los menos urgentes, y muestra claramente un cubo de basura al cual han sido enviados

otros mensajes. Al día siguiente, los mensajes similares a los "menos urgentes" leídos anteriormente son considerados más importantes, y colocados más arriba en la lista de lectura. Además, uno de los mensajes de la lista de correo se considera importante, y se coloca en la lista de "importantes", y se presenta una notificación informándonos que el periódico al cual estamos suscritos en línea contiene una noticia "interesante". Este último escenario supone la existencia de un agente autónomo inteligente [19].

2.2 Definición

Rusell y Norvig en su libro "Inteligencia Artificial: Un enfoque moderno" [20] nos dan una definición de Agente: "Es la de aquel ente que percibe su ambiente mediante sensores y que responde o actúa en dicho ambiente mediante efectores." Su definición de agente inteligente o racional es obviamente, más restrictiva. Un agente inteligente es aquel que toma sus decisiones de actuación para favorecer lo máximo posible su medida de rendimiento. Es decir que el agente toma decisiones que le permitan desempeñar lo mejor posible sus objetivos [21].

Dicho concepto es considerado por otros autores como un tanto simple ya que, según sus interpretaciones, un Agente Inteligente es más complejo. En muchos otros casos se considera que una definición no es suficiente para describir claramente lo que es una Agente Inteligente, y en cambio se cree que se debe reunir un conjunto de características que necesarias para ser considerado como tal [22].

Hípolo y Vargas-Quesada (1999) definen agente inteligente "como una entidad software que, basándose en su propio conocimiento, realiza un conjunto de operaciones destinadas a satisfacer las necesidades de un usuario o de otro programa, bien por iniciativa propia o porque alguno de éstos se lo requiere" [23].

Otra definición que se ajusta al concepto de agentes inteligentes es la dada por Jiménez y Ramos (2000) para los que un agente inteligente es:

"Pieza de software que ejecuta una tarea dada utilizando información recolectada del ambiente, para actuar de manera apropiada hasta completar la tarea de manera exitosa. El software debe ser capaz de auto-ajustarse basándose en los cambios que ocurren en su ambiente de forma tal que un cambio en las circunstancias producirá un resultado esperado" [23]

Franklin and Graesser por su parte afirman que afirma que “Un agente autónomo es un sistema situado y que forma parte de un entorno, lo siente y actúa sobre él, a través del tiempo, persiguiendo sus propios objetivos de forma que afecte lo que sentirá en el futuro” [24].

Shoham “Un agente es una entidad cuyo estado es visto como compuesto de componentes mentales tales como creencias, elecciones, aptitudes y compromisos”.

Maes “Un agente autónomo es un sistema computacional que habita en un entorno dinámico-complejo, percibiendo y actuando autónomamente en este entorno, y haciendo esto para realizar un conjunto de objetivos o tareas para los cuales fueron diseñados”.

FIPA (Foundation for Intelligent Physical Agents) concluye “Un agente es una entidad de software encapsulado con su propio estado, conducta, hilo de control y la habilidad para interactuar y comunicarse con otras entidades (gente, otros agentes o sistemas)” [25].

2.3 Características

Por otra parte, un agente puede tener los siguientes atributos [26]:

- **Autonomía:** Un agente puede operar por sí solo, sin la intervención directa de un humano o alguna especie de control sobre sus acciones y estado interno.
- **Movilidad:** Es la capacidad de una entidad de software de poder migrar de ambiente en caso de ser necesario. Esto se ve aplicado cuando el agente tiene acceso a Internet y puede moverse en distintos ambientes.
- **Inteligencia:** Los agentes deben tener la capacidad de reconocer eventos, determinar su significado y de ahí llevar a cabo acciones o respuestas a dichos eventos.

- **BDI: (belief, desire, intentions)** Un agente debe poseer creencias, deseos e intenciones. Las creencias son todo aquello que se tiene definido como conocimiento inicial de su entorno y responsabilidades, los deseos son los objetivos que deba de cumplir el agente y por último las intenciones es el plan que se llevará a cabo para cumplir esos objetivos.
- **Reactividad:** Los agentes perciben su entorno (el cual puede ser el mundo físico o por medio de la interfaz de un usuario, Internet o una combinación) y responden según los cambios percibidos.
- **Proactividad:** Los agentes no solo deben responder hacia un estímulo del ambiente sino también ser capaces de demostrar iniciativa.
- **Sociabilidad:** Una característica muy significativa del agente es la capacidad de comunicación ya sea con otros agentes o con el ser humano.

Una caracterización más detallada de autonomía es presentada por Covrigaru y Lindsay. La misma expresa que un agente se percibe como autónomo en la medida en que [27]:

- Su comportamiento está orientado por sus metas y es capaz de seleccionar que meta va a procesar a cada instante.
- Su existencia se da en un período relativamente mayor al necesario para satisfacer sus metas.
- Es lo suficientemente robusto como para seguir siendo viable a pesar de los cambios en el ambiente.
- Puede interaccionar con su ambiente en la modalidad de procesamiento de información.
- Es capaz de exhibir una variedad de respuestas, incluyendo movimientos de adaptación fluidos; y su atención a los estímulos es selectiva.

- Ninguna de sus funciones, acciones o decisiones, está totalmente gobernada por un agente externo.
- Una vez en operación, el agente no necesita ser programado nuevamente por un agente externo.

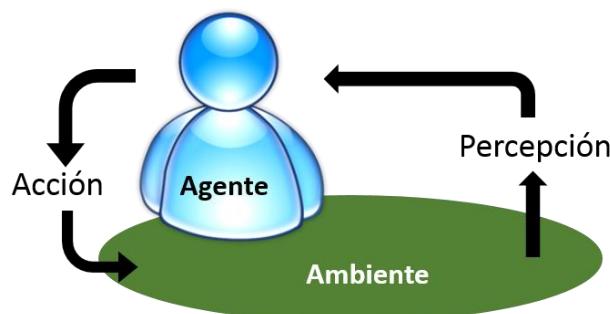


Figura 9. Abstracción de un agente a partir de su interacción con el medio [27]

Un agente percibe su entorno a través de sensores y actúa sobre el mismo mediante “efectores”. Por ejemplo los humanos, robots y software pueden ser considerados agentes.

2.4 Comunicación

Existen algunos estándares para comunicación entre agentes inteligentes de los cuales se analizarán KQML y FIPA-ACL por ser considerados de mayor utilización en el desarrollo de sistemas multiagentes.

2.4.1 KQML

La consulta de conocimientos y de lenguaje de manipulación, o CCLM (Knowledge Query and Manipulation Language KQML), es un lenguaje y protocolo para la comunicación entre los agentes inteligentes y sistemas basados en conocimiento. Fue desarrollado en la década de 1990 es parte DARPA knowledge Sharing Effort. Originalmente fue concebido como una interfaz para sistemas basados en el conocimiento, pero pronto fue reutilizado como idioma de comunicación de agentes inteligentes.

El trabajo en KQML fue dirigido por Tim Finin [28] de la Universidad de Maryland, Baltimore County y Jay Weber de EITech y las contribuciones de muchos investigadores involucrados.

Capas: El lenguaje KQML está dividido en 3 capas [29]:

1. La capa de contenido
 - Relacionada con el contenido del mensaje
 - Puede expresarse en cualquier lenguaje
 - KQML sólo tiene interés en identificar el inicio y el final del contenido
2. La capa de mensaje
 - Es el núcleo del lenguaje KQML
 - Determina los tipos de interacción que puede realizar un agente
 - Identifica el protocolo, la ontología y el acto del habla (performative)
3. La capa de comunicación
 - Identifica las características del mensaje que describen los parámetros de bajo nivel de la comunicación (emisor, receptor e identificador de mensaje)

Mensajes: Un mensaje en KQML representa la intención de realizar alguna acción (performativa o acto del habla). Se representa mediante una lista tipo LISP en la que el primer elemento de la lista es el identificador del acto del habla y el resto de la lista son pares atributo-valor sin un orden predefinido por ejemplo [29]:

```
(ask-one
  :sender agente1
  :receiver servidor-telefonica
  :reply-with accion-telefonica
  :content (Precio Telefonica ?p)
  :language prolog
  :ontology IBEX-35
)
```

KQML se basa en un conjunto extensible de performativas, que definen las operaciones permitidas que los agentes pueden ejecutar en la base de conocimiento de los otros agentes. Además, el lenguaje KQML proporciona una arquitectura básica para la compartición de conocimiento mediante un tipo de agentes mediadores llamados communication facilitators (facilitadores), que coordinan las relaciones e interacciones entre los otros agentes

Atributos Reservados: Un mensaje KQML posee los siguientes atributos [29]

TABLA I. ATRIBUTOS DE UN MENSAJE KQML

Atributo	Significado
:content	La información o contenido en sí
:sender	Emisor del mensaje
:receiver	Receptor del mensaje
:language	El nombre del lenguaje de epresentación empleado en el atributo :content
:ontology	El nombre de la ontología utilizada en el atributo :content
:reply-with	Etiqueta para la respuesta (si es que el emisor la espera)
:in-reply-to	o La etiqueta esperada en la respuesta

Performativas: Las performativas se clasifican en distintas categorías [29]:

TABLA II. CATEGORIZACIÓN DE PERFORMATIVAS

Categoría	Nombre
Información	tell, untell
Consulta	evaluate, reply, ask-if, ask-about, ask-one, ask-all, sorry
Consulta con múltiple respuesta	stream-about, stream-all
Orden	achieve, unachieve
Control de respuestas	standby, ready, next, rest, discard
Capacidad	Advertise
Notificación	subscribe, monitor
Gestión de red	register, unregister, forward, broadcast, pipe, break
Facilitación	broker-one, broker-all, recommend-one, recommend-all, recruit-one, recruit-all

La lista completa de performativas se aprecia a continuación [29]:

TABLA III. LISTA DE PERFORMATIVOS

Atributo	Significado
Tell	E comunica una información que se encuentra en su BC (Bloque de Contenido)
evaluate	E quiere que R evalúe la expresión
reply	E comunica a R una respuesta esperada
ask-if	E quiere saber si la expresión se encuentra en la BC de R
ask-about	E quiere conocer todo el conocimiento de R relacionado con la expresión
ask-one	E quiere que R conteste a una pregunta
stream-about	Versión en respuesta múltiple de ask-about
achieve	S quiere que R convierta en verdadera la expresión en su BC
standby	S quiere que R esté preparado para responder a una performative
Ready	S está preparado para responder a R
Next	S quiere que R le devuelva la siguiente respuesta (de una consulta múltiple)
advertise	S está preparado para responder a una determinada performative
register	S puede responder a performatives de un determinado agente
broadcast	S quiere que R envíe la performative a todos los agentes conectados
recommend-one	S quiere el nombre de un agente que pueda contestar a una performative
recruit-one	S quiere que R le proporcione el nombre de otro agente que conteste a la preformative

2.4.2 FIPA-ACL

La **Foundation for Intelligent Physical Agents (FIPA)** es un organismo para el desarrollo y establecimiento de estándares de software para agentes heterogéneos que interactúan y sistemas basados en agentes [25]

Su objetivo es definir un conjunto completo de normas y estándares para la implementación de sistemas en los que se puedan ejecutar agentes (plataformas de agentes) y especificación de cómo los propios agentes se deben comunicar e interactuar.



Figura 10. Funcionamiento Lenguaje de Comunicación de Agentes [25]

FIPA Agent Communication especifica el Lenguaje de Comunicación de Agentes (Agent Communication Language) o ACL, el cual se basa en pasos de mensajes entre agentes con protocolos de interacción para intercambio de mensajes, actos de habla basados en la teoría de los actos comunicativos y representaciones lingüísticas contenido [25].

Un mensaje ACL contiene un conjunto de uno o más parámetros. Precisamente qué parámetros son necesarios para una eficiente comunicación entre agentes variará de acuerdo a la situación; el único parámetro que es obligatorio en todos los mensajes de ACL es lo performativo (performative), aunque se espera que la mayoría de los mensajes ACL también contendrán parámetros emisor (sender), receptor (receiver) y contenido (content).

2.4.2.1 Parámetros

El conjunto completo de parámetros de mensajes FIPA ACL se muestra en la TABLA IV sin tener en cuenta sus codificaciones específicas en una implementación. La codificación está aprobada por la FIPA y los ordenamientos de los parámetros para mensajes de ACL se dan en otras especificaciones. Cada especificación de representación mensaje ACL contiene descripciones de sintaxis precisas para codificaciones de mensajes ACL basado en XML, cadenas de texto y varios otros esquemas [30].

Un mensaje FIPA ACL se corresponde con un conjunto de parámetros abstractos de la carga útil identificado en la FIPA00001 [30].

TABLA IV. PARÁMETROS DE UN MENSAJE FIPA ACL

Parámetro	Categoría de Parámetros
Performative	Tipo de actos comunicativos
Sender	Participante en la comunicación
Receiver	Participante en la comunicación
reply-to	Participante en la comunicación
Content	Contenido del mensaje
Language	Descripción del contenido
Encoding	Descripción del contenido
Ontology	Descripción del contenido
Protocol	Control de conversación
conversation-id	Control de conversación
reply-with	Control de conversación
in-reply-to	Control de conversación
reply-by	Control de conversación

2.5 Tipos de Agentes Inteligentes

Una característica fundamental de los agentes inteligentes es que pueden dividirse en distintos tipos según su utilidad y las tareas que van a realizar. A continuación se detalla su clasificación.

2.5.1 Agentes Reactivos

Se puede considerar como el tipo de agente más simple que existe. Los agentes reactivos no poseen modelos simbólicos internos de sus entornos [31]. En su lugar reaccionan en modo estímulo-respuesta para representar el estado de su entorno, en otras palabras se trata de un agente que simplemente recibe entradas mediante algún tipo de sensor, el agente las procesa y posteriormente va a producir una respuesta o salida.

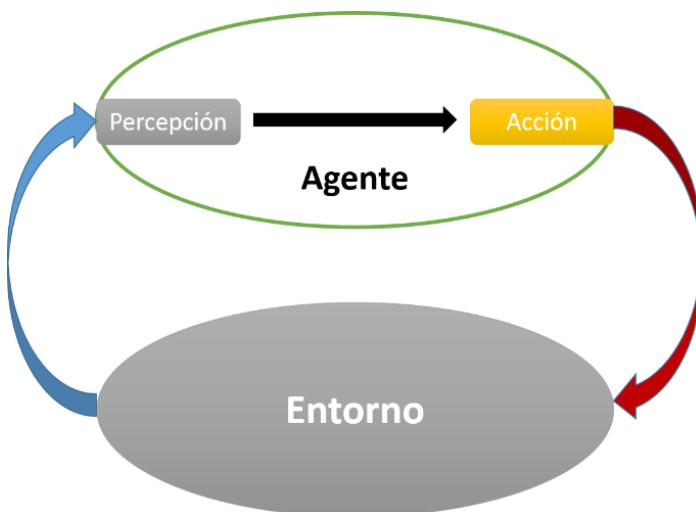


Figura 11. Funcionamiento de un Agente Reactivo [31]

Los Agentes Reactivos se clasifican en Agentes Reactivos simples, basados en Modelos, basados en Objetivos y basados en utilidad.

2.5.1.1 Agentes Reactivos Simples

Las reacciones son conexiones mentales que siguen una regla sencilla [32]:

Si <percepción> entonces <acción>

Por ejemplo: si el auto que va adelante enciende la luz de freno, entonces disminuyo mi velocidad. Este tipo de reglas se llaman reglas de condición-acción (CA).

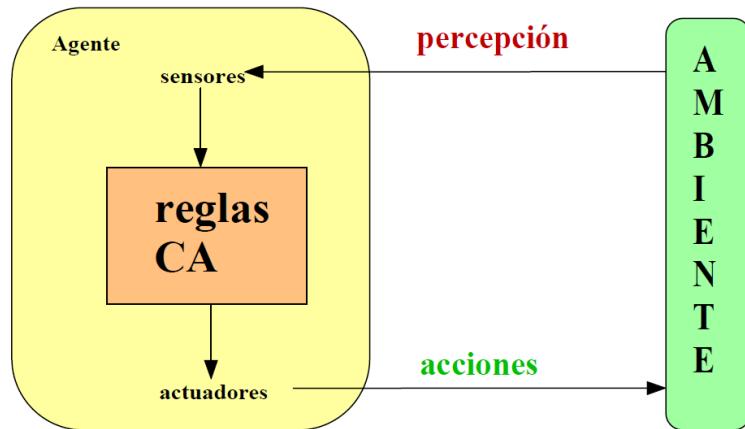


Figura 12. Agente Reactivo Simple

2.5.1.2 Agentes Reactivos basados en Modelos

El problema de los Agentes Reactivos Simples es el no poseer visibilidad del mundo. Para solucionar este inconveniente se puede generar una visibilidad parcial almacenando aquellas partes del mundo que no se ven. Para lo cual se requiere de un estado interno que dependa de la historia de las percepciones. Por ejemplo: al conducir, saber dónde están los otros vehículos [32].

Se necesita información sobre el cómo evoluciona el mundo, independiente del agente: saber por ejemplo que los autos se acercan cuando se encienden las luces de freno traseras. El cómo es lo que se conoce como modelo.

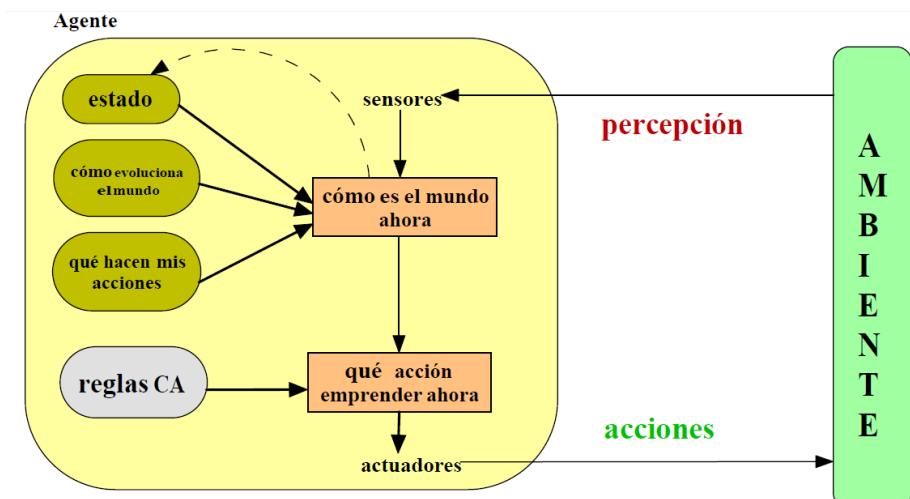


Figura 13. Agente Reactivo basado en Modelos

2.5.1.3 Agentes Reactivos basados en Objetivos

En ocasiones no basta con el conocimiento sobre el estado actual del mundo. Por ejemplo: el taxista automático al verse enfrentado a un cruce de calles, puede decidir girar a la izquierda, derecha o bien continuar su marcha hacia adelante dependiendo hacia dónde quiere ir, y NO de las percepciones del mundo. Se requiere, por lo tanto, información sobre un objetivo o meta para el agente. Para esto, se puede emplear el resultado que generen las acciones, de un modo similar a cómo opera el agente basado en modelos [32].

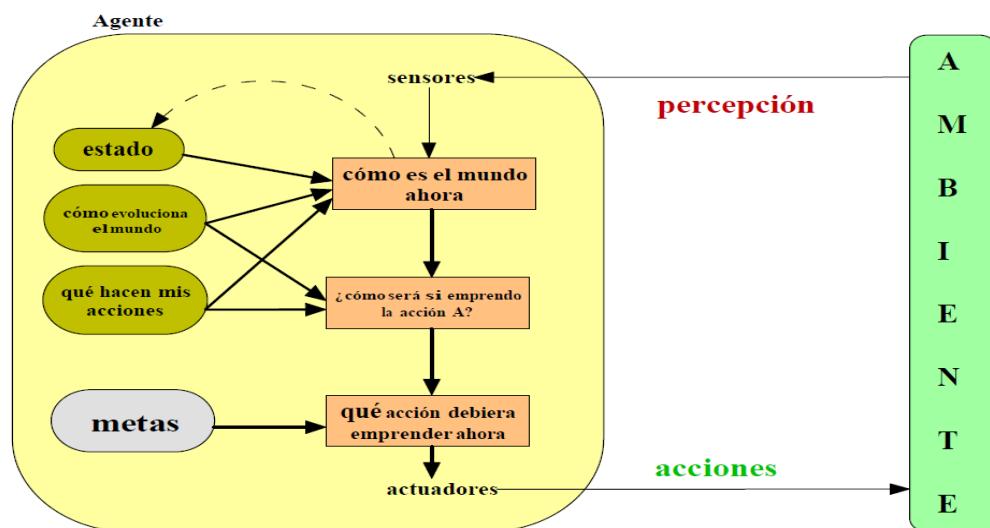


Figura 14. Agente Reactivo basado en objetivos

2.5.1.4 Agentes Reactivos basados en Utilidad

Una vez más se puede dar el caso que las metas por sí solas no son suficientes para generar comportamiento de gran calidad. Por ejemplo, hay muchas maneras de que el taxi automático llegue a su destino, en este caso no está en discusión que el taxista cumple con su objetivo pero no todas las rutas resultan cómodas o baratas para el usuario. Por lo tanto, las metas definen vagamente lo que es felicidad y tristeza, con respecto al cumplimiento de objetivos [32].

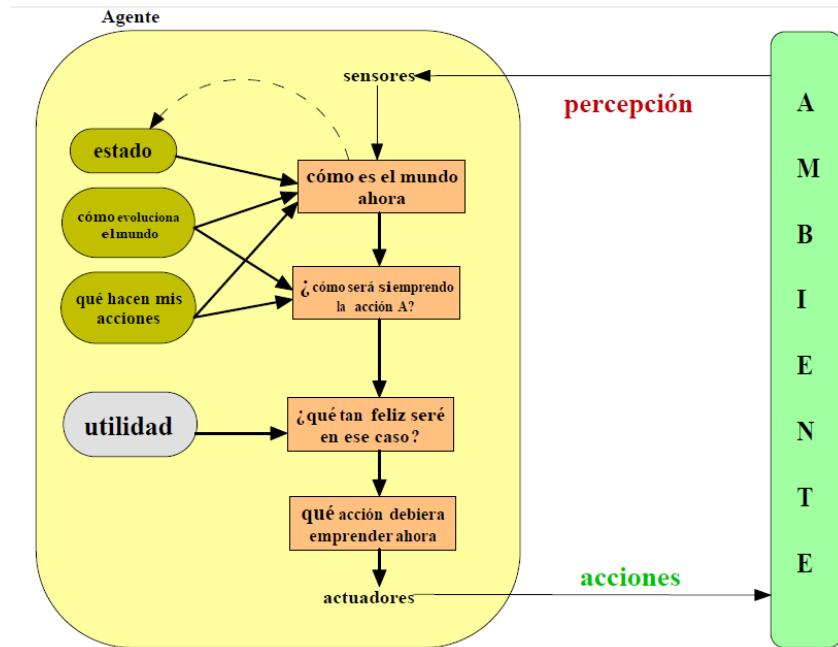


Figura 15. Agentes Reactivos basados en Utilidad

2.5.2 Agentes Deliberativos

Poseen un modelo de razonamiento simbólico interno, comprometido en la planeación y negociación para realizar coordinación con otros agentes. Son capaces de razonar acerca de sus intenciones y conocimientos, crear planes de acción, y ejecutar dichos planes además posee la capacidad de efectuar operaciones complejas.

Una característica fundamental es que son individualmente inteligentes (este tipo de agentes puede ser considerado como un sistema experto, con capacidad de razonamiento sobre su base de conocimiento), pueden comunicarse con los demás agentes y llegar a un acuerdo con todos o algunos de estos, sobre alguna decisión a tomar [33].

Un agente deliberativo funciona de la siguiente manera [33]:

1. una entrada sensorial es recibida por la función de revisión de creencias y las creencias del agente se alteran.
2. La función de generación de opciones evalúa las creencias e intenciones alteradas y crea las opciones disponibles para el agente. Constituyen los deseos del agente.

3. La función de filtro considera las actuales creencias, deseos e intenciones y produce nuevas intenciones.
4. La función de selección de la acción recibe entonces la función de filtro intenciones y decide qué acción realizar.



Figura 16. Funcionamiento de un Agente Deliberativo

2.5.3 Agentes Colaborativos o Cooperativos

Los agentes colaborativos enfatizan en la autonomía y la cooperación para ejecutar tareas por ellos mismos. Se enfocan principalmente en la autonomía y cooperación entre agentes para llevar a cabo ciertas tareas. Deben abarcar los siguientes aspectos [34]:

- Resolver problemas que para un solo agente resultarían muy complejos.
- Proporcionar soluciones a partir de las fuentes de información distribuidas.
- Permitir la interconexión e interoperación de múltiples sistemas.
- Proporcionar soluciones donde la experiencia es distribuida.

2.5.4 Agentes de Interfaz

Los agentes de interfaz enfatizan en la autonomía y el aprendizaje para desempeñar tareas para sus propietarios. Principalmente los agentes de interfaz soportan y proporcionan asistencia, para que el usuario aprenda a utilizar una aplicación [35]:

- Reducen el trabajo para el usuario final y para el desarrollador de la aplicación.
- Pueden adaptarse sobre la marcha a las preferencias y hábitos de sus usuarios.

- Pueden ser usados en el desarrollo de aplicaciones reales porque son simples, operan en dominios limitados y no requieren en general, cooperación con otros agentes



Figura 17. Esquema básico de un agente de interfaz [35]

2.5.5 Agentes Móviles

Este tipo de agentes tiene la capacidad de recorrer por redes WAN como la WWW, interactuando con hosts extraños, recogiendo información para su propietario y realizando tareas propuestas por sus usuarios. Para conseguir este objetivo deben funcionar remotamente, ser autónomos y cooperativos capaces de ser creados en una máquina y distribuirse a otras para ejecutarse. Pueden ser de dos tipos:

- **Mono-salto.**- Se mueven a un único y específico lugar.
- **Multi-salto.**- Transitan por la red de uno a otro lugar

2.5.6 Agentes Estáticos

Estos programas tienen funciones específicas que cumplir. Están instalados en un sistema o una red y no pueden realizar tareas que no se le han sido asignadas. Pueden interactuar con otros agentes estáticos o bien con agentes móviles, de manera que puedan realizar en conjunto un objetivo específico.

2.5.7 Agentes de Información/Internet

Se encargan de manejar, manipular y colecciónar información de muchas fuentes distribuidas para sacar de ésta mayores beneficios. Pueden ser móviles aplicándoles los cambios de los agentes móviles, o estáticos similares a los agentes de interfaz aplicándoles los cambios de los agentes de interfaz.

2.5.8 Agentes Híbridos

Se refiere a aquellos que son una combinación de dos o más clasificaciones de agentes que forman un agente. La arquitectura consiste en los subsistemas de percepción y acción, constando con diferentes capas. Estas capas pueden ser reactivas y deliberativas, las cuales son independientes de su acción [36].

2.6 Arquitecturas de Agentes

Una arquitectura permite descomponer un sistema computacional en componentes más pequeños y determinar cómo es la relación entre estos. Una arquitectura para agentes en particular permite descomponer un sistema de agentes, y determinar cómo deben interactuar entre ellos y con el ambiente.

2.6.1 Arquitecturas reactivas

Los agentes reactivos se caracterizan porque pueden operar rápida y efectivamente sin la necesidad de procesar una representación simbólica del entorno, ya que representan una categoría de agentes que no posee modelos simbólicos externos del entorno en el que se encuentran. Los agentes reactivos toman decisiones basadas totalmente en el presente, sin hacer uso de lo que ha ocurrido en el pasado, porque no conocen su historia [36].

Este tipo de agentes actúan siguiendo un esquema estímulo-respuesta según el estado actual del entorno en el que se encuentran embebido. La característica más importante de los agentes reactivos es el hecho de que los agentes son relativamente simples e interaccionan con otros agentes de manera sencilla. Sin embargo, si se observa de una forma global el conjunto de agentes, las interacciones pueden dar lugar a patrones de comportamiento muy complejos [36].

Por el hecho de ser agentes sencillos, la capacidad de realizar razonamientos complejos desaparece en su mayor parte, y al contrario que los agentes deliberativos la inteligencia que puedan mostrar no proviene de modelos internos, sino de la interacción con su entorno.

Un agente reactivo está formado por los siguientes tipos de módulos:

- Módulos de interacción con el entorno (sensores y actuadores).
- Módulos de competencia (cada una de la capas).

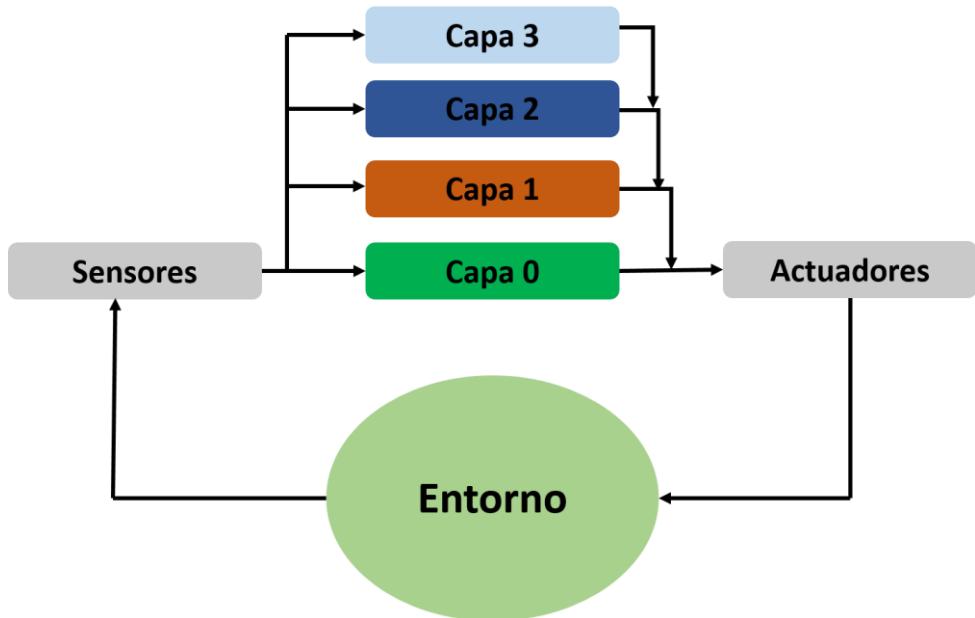


Figura 18. Arquitectura de un agente reactivo [36]

2.6.2 Arquitecturas Deliberativas

Los agentes con una arquitectura deliberativa se caracterizan porque contienen explícitamente un modelo simbólico del entorno, y las decisiones. Además de esta representación estos agentes deliberativos poseen la capacidad de llevar a cabo decisiones a partir de los conocimientos que posee y modificando su estado interno comúnmente llamado 'estado mental', es decir tienen la capacidad de tomar decisiones basándose en el pasado [36].

En el estado mental lo más común en los agentes deliberativos o agentes BDI (Belief, Desire, Intention), estos puntos creencias, deseos e intenciones tienen su significado:

Creencias: Es Aquel conocimiento que el agente tiene de sí mismo y de su entorno, que puede ser no totalmente correcto.

- Deseos: Son objetivos que el agente desea llevar a cabo a largo plazo.
- Metas: Subconjunto de los deseos que el agente puede conseguir.
- Intenciones: Son el subconjunto de los deseos que el agente intenta conseguir.
- Planes: Combinan las intenciones en unidades consistentes.

Para comprender mejor esta estructura se representa en el siguiente gráfico:

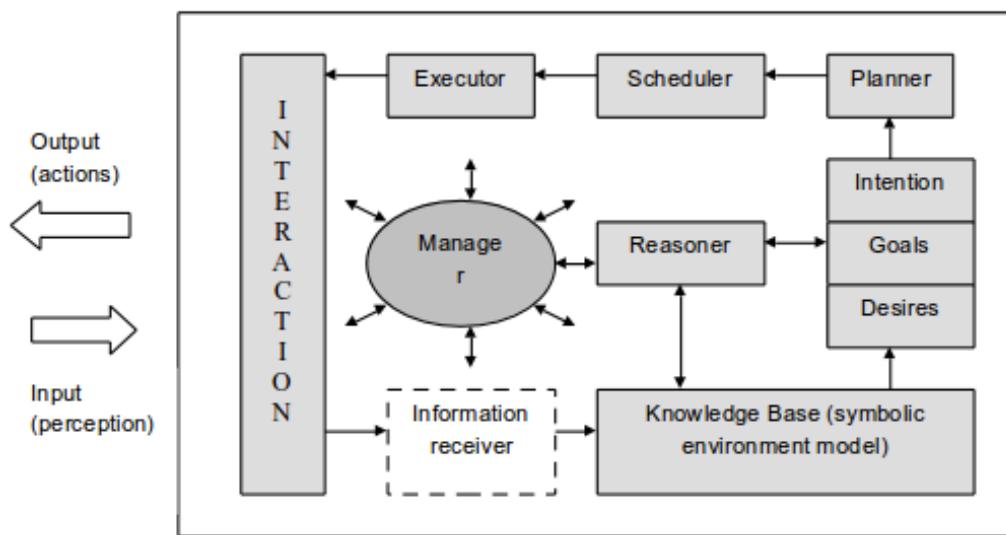


Figura 19. Estructura de un agente deliberativo [36]

Se puede decir que los agentes deliberativos a partir de la base de conocimientos se derivan, por medio de un resolutor, los deseos, metas e intenciones. El planificador toma las intenciones, las combina y forma un plan global consistente en un proceso incremental dinámico, es decir, el planificador testea nuevas intenciones por si hay dependencias con los planes existentes y de esta forma los planes están continuamente adaptándose a las situaciones que se producen por la llegada de nuevas intenciones.

El organizador recibe del planificador los planes actuales, donde cada plan consiste en un número determinado de acciones simples que pueden ser procesadas de forma secuencial o de forma paralela, y decide cuando se ejecutan las acciones que contienen, asociado un tiempo de ejecución a cada acción. Para esto es necesario que el organizador conozca continuamente los recursos disponibles para el agente [36].

El organizador pasa cada acción con su información complementaria al ejecutor que ejecuta la acción y monitorea su correcto funcionamiento.

2.6.3 Arquitecturas Híbridas

Las arquitecturas híbridas combinan componentes de tipo reactivo con componentes de tipo deliberativo. La parte reactiva interacciona con el entorno y reacciona rápidamente a los eventos que en él se producen sin invertir tiempo en realizar razonamiento, mientras que la parte deliberativa planifica y se encarga de la parte de toma de decisiones, es decir, realiza tareas a un nivel de abstracción superior [37].

Las arquitecturas híbridas pretenden aprovechar los beneficios que proporcionan las arquitecturas reactivas y los beneficios que proporcionan las deliberativas, ya que para la mayoría de los problemas no es adecuada ni una arquitectura puramente deliberativa ni una arquitectura puramente reactiva.

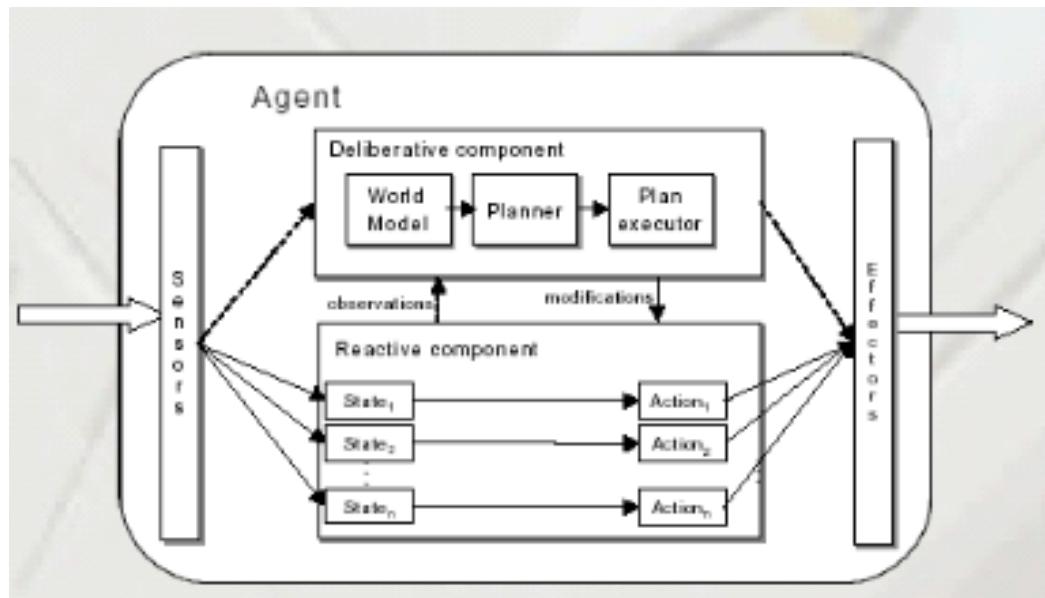


Figura 20. Estructura de un agente Híbrido [36]

Típicamente los sistemas híbridos se diseñan siguiendo una arquitectura jerárquica en capas, en la que las capas más bajas son principalmente reactivas y las capas más altas son principalmente deliberativas.

CAPÍTULO III

3 Metodologías

En este capítulo se abordará el tema de metodologías utilizadas tanto para desarrollo de Sistemas como para el desarrollo de Agentes Inteligentes analizando sus características principales y funcionamiento en general.

3.1 Metodología para el Desarrollo de Sistemas

Como metodología de desarrollo de software se aborda XP (Programación extrema) por las distintas facilidades y la posibilidad de obtener resultados rápidos y de calidad.

3.1.1 Programación Extrema (XP)

La programación extrema o Extreme Programming fue formulado por Kent Beck en el año de 1999 enfocándose como punto principal de su metodología en la simplicidad, la comunicación y el reciclado continuo de código [38].

3.1.1.1 Introducción

XP ha sido notablemente destacado como el más ágil de los procesos de desarrollo de software debido a su manera de adaptarse a los cambios de requerimientos en lugar de tener la previsibilidad que tanto cuesta en el proceso de planificación [39].

Los seguidores de XP aseveran que los cambios sobre la marcha son un aspecto natural e inevitable y que todo gerente de proyecto debe ser capaz de resolver.

Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto y aplicarlo de manera dinámica durante el ciclo de vida del software.

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo [39]. Un aspecto importante es la realimentación continua entre el cliente y el equipo de desarrollo, es decir mantener una comunicación constante con el cliente

durante el desarrollo del sistema además de fomentar una comunicación fluida entre todos los participantes. Otros aspectos importantes son simplicidad en las soluciones implementadas y coraje para enfrentar los cambios [39].

3.1.1.2 Elementos de la Metodología

Las historias de usuario: son la técnica utilizada para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible.

Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarlas en unas semanas. Las historias de usuario se descomponen en tareas de programación y se asignan a los programadores para ser implementadas durante una iteración.

En XP se presentan ciertos roles descritos a continuación [40]:

- **Programador:** Escribe las pruebas unitarias y produce el código del sistema
- **Cliente:** Escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en apoyar mayor valor al negocio.
- **Encargado de pruebas (tester):** Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para las pruebas.
- **Encargado de seguimiento (tracker):** Proporciona realimentación al equipo. Además de verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones.
- **Entrenador (coach):** Es el responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.
- **Consultor:** Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas.
- **Gestor (big boss):** es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

3.1.1.3 Los Cinco Valores de XP

XP se rige en base a 5 valores descritos a continuación [39]:

Comunicación: XP fomenta la comunicación entre los miembros del grupo del desarrollo como con el cliente, una buena comunicación permite un desarrollo sincronizado entre el pequeño grupo que suele conformar el equipo de trabajo en XP, una mala comunicación al contrario permite una desorganización, discusión, mala comprensión de requerimientos e incluso enfados entre las partes.

Sencillez: El sistema debe ser funcional sin ningún añadido como colores, sonidos u otros que no estén dentro de los requerimientos, si al terminar toda la funcionalidad principal hay tiempo suficiente se puede aplicar cualquier otro requerimiento secundario.

Retroalimentación: El sistema al ser funcional y teniendo la suficiente comunicación con el cliente se puede lograr que la información que se necesite, sea el mismo sistema quien la responda, es una manera de que el cliente pruebe un sistema que está en desarrollo dando un índice de fiabilidad.

Valentía: XP nos advierte que existirán cambios durante el desarrollo, y el impacto que esto significa es proporcional a que tan cambiante sea el código del sistema, esto nos obliga a tomar decisiones que pueden ser cruciales para el proyecto.

Respeto: El respeto a cada integrante del grupo pues todos son importantes para la culminación del proyecto. El respeto en XP se refiere a comprender que el trabajo de los integrantes está ligado entre si y que cualquier retraso puede acarrear al atraso de otras personas.

3.1.1.4 Fases

En XP se presentan cuatro fases que son análisis, diseño, codificación y pruebas. Se describen a continuación [39]:

Fase 1. Planificación del proyecto: El primer paso para cualquier proyecto que siga XP, son las Historias de Usuario, éstas tienen la misma finalidad que los casos de uso pero con algunas diferencias: [39]

- Son escritas por el cliente en 3 o 4 líneas utilizando un lenguaje no técnico, sin detalles y sin hacer referencias a diseños o algoritmos para la codificación.
- Las Historias de Usuario son usadas para estimar tiempos y también en la fase de pruebas para verificar que el sistema cumple con lo deseado.

Luego de definir las historias, es preciso elaborar un plan de publicaciones o Release Plan, que indicará las historias de usuario que se crearán para cada versión del programa y las fechas en las que se publicarán estas versiones. Todo proyecto que utilice XP se ha de dividir en Iteraciones de 3 semanas de duración, al inicio de cada iteración se definirá el Release Planing y al final se determinarán las Historias de Usuario que no pasaron las pruebas de aceptación.

La Velocidad del Proyecto es la medida de la rapidez con que se desarrolla el proyecto; la estimación es sencilla, basta con contar el número de historias de usuario que se pueden implementar por una iteración. [39]

XP como metodología recomienda la Programación en Pareja, pues incrementa la productividad y la calidad del software desarrollado, este trabajo involucra a dos programadores en el mismo equipo. Es necesario que los programadores mantengan Reuniones Diarias para que se expongan sus problemas, soluciones e ideas de forma conjunta, las reuniones tienen que ser fluidas y todo el mundo tiene que tener voz y voto

Fase 2. Diseño: La metodología XP sugiere que se manejen Diseños Simples que procuren hacer todo lo menos complicado posible para conseguir un diseño entendible y fácil de implementar. [39]

Usar Glosarios de Términos y una correcta especificación de nombres de clases y métodos ayudará a comprender el diseño y facilitará una futura reutilización de código. No se debe añadir Funcionalidad Extra al programa aunque se piense en un futuro que ésta será utilizada, esto implica un desperdicio de tiempo y recursos.

Refactorizar es mejorar y modificar la estructura y codificación de códigos ya creados sin alterar su funcionalidad. Es muy común usar códigos ya creados que contienen funcionalidades que no serán usadas y diseños obsoletos.

Fase 3. Codificación: El cliente es una parte más del equipo de desarrollo; su presencia es indispensable en las distintas fases de XP. A la hora de codificar una historia de usuario su presencia es aún más necesaria. No olvidemos que los clientes son los que

crean las historias de usuario y negocian los tiempos en los que serán implementadas. [39]

Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que ésta hará y también tendrá que estar presente cuando se realicen las pruebas que verifiquen que la historia implementada cumple la funcionalidad especificada.

La optimización del código siempre se debe dejar para el final. Hay que hacer que funcione y que sea correcto, más tarde se puede optimizar. XP afirma que la mayoría de los proyectos que necesiten más tiempo extra que el planificado para ser finalizados no podrán ser terminados a tiempo se plantea la solución de realizar un nuevo "Release plan" para concretar los nuevos tiempos de publicación y de velocidad del proyecto.

Fase 4: Pruebas: Uno de los pilares de la metodología XP es el uso de pruebas para comprobar el funcionamiento de los códigos que vayamos implementando. Las pruebas permiten verificar que un cambio en la estructura de un código no tiene por qué cambiar su funcionamiento.

Además las pruebas sirven para evaluar las distintas tareas en las que ha sido dividida una historia de usuario. Para asegurar el funcionamiento final de una determinada historia de usuario se deben crear "Test de aceptación"; estos test son creados y usados por los clientes para comprobar que las distintas historias de usuario cumplen su cometido. [39]

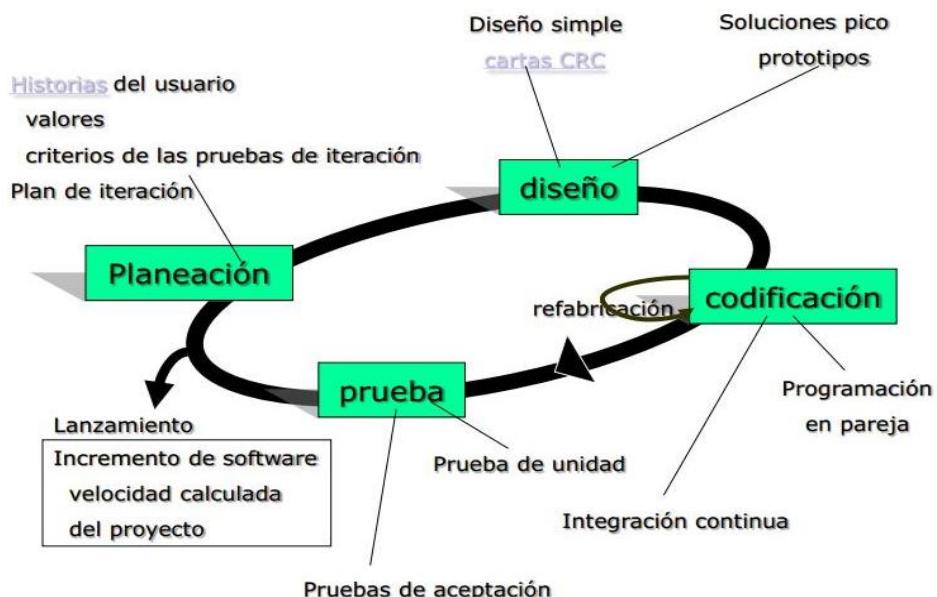


Figura 21. Fases que sigue XP

3.1.1.5 Buenas prácticas de XP

XP se basa en la idea de que es mejor hacer cambios durante el tránsito del proyecto y no correcciones al final que suponen un costo de dinero y tiempo mayor. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las prácticas que describiremos a continuación.

El juego de la planificación: Se consigue gracias a una comunicación frecuente con el cliente. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración. Esta práctica se puede ilustrar como un juego, donde existen dos jugadores: Cliente y Programador. El cliente establece la prioridad de cada historia de usuario, de acuerdo con el valor que aporta para el negocio. Los programadores estiman el esfuerzo asociado a cada historia de usuario. Se ordenan las historias de usuario según prioridad y esfuerzo, y se define el contenido de la entrega y/o iteración, apostando por enfrentar lo de más valor y riesgo cuanto antes. Este juego debe ser realizado durante la planificación de la entrega y de cada iteración o cuando sea necesario reconducir el proyecto. [41]

Entregas pequeñas: Se consigue produciendo versiones estables del sistema de una forma rápida, aunque obviamente no cuenten con toda la funcionalidad pretendida para el sistema pero sí que constituyan un resultado de valor para el negocio. Una entrega no debería tardar más 3 meses. [41]

Utilización de Metáforas: La filosofía de XP es no definir tempranamente una arquitectura estable para el sistema. En cambio va evolucionando y los posibles inconvenientes que se generarían por no contar con ella explícitamente en el comienzo del proyecto se solventan con la existencia de una metáfora. El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora consiste en formar un conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema. Este conjunto de nombres ayuda a la nomenclatura de clases y métodos del sistema. [42]

Diseño simple: Consiste en procurar diseñar la solución más simple que pueda funcionar y que sea implementada en un momento determinado del proyecto. La complejidad innecesaria y el código extra debe ser removido inmediatamente.

Pruebas: La producción de código está dirigida por las pruebas unitarias. XP establece que se debe escribir la prueba antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. Los clientes escriben las pruebas funcionales para cada historia de usuario que deba validarse.

Refactorización: Es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. La refactorización mejora la estructura interna del código sin alterar su comportamiento. [41]

Programación en parejas: Toda la producción de código debe realizarse con trabajo en parejas de programadores. Las ventajas que brinda este estilo de programación son por ejemplo corrección continua de errores en el código ya que el mismo es revisado continuamente, continua discusión de ideas entre los dos programadores, los problemas de programación se resuelven más rápido, la conversación mejora el flujo de información y la dinámica del equipo, y finalmente, los programadores disfrutan más su trabajo.

Propiedad colectiva del código: Cualquier programador puede cambiar cualquier parte del código en cualquier momento. Esta práctica motiva a todos a contribuir con nuevas ideas en todos los segmentos del sistema, evitando a la vez que algún programador sea imprescindible para realizar cambios en alguna porción de código. [41]

Integración continua: Cada porción de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día. Todas las pruebas son ejecutadas y tienen que ser aprobadas para que el nuevo código sea incorporado definitivamente. [41]

40 horas por semana: El máximo de horas trabajadas por semana debe ser 40. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre, probablemente está ocurriendo un problema que debe corregirse. La carga de trabajo puede desmotivar al equipo y minorar el rendimiento para el desarrollo del proyecto produciendo retrasos.

Cliente in-situ: se refiere a la presencia constante del cliente y su disposición para estar en continua comunicación con el equipo de trabajo. La comunicación oral es más efectiva que la escrita, ya que esta última toma mucho tiempo en generarse y puede tener más riesgo de ser mal interpretada.

Estándares de programación: Los estándares deben estar presentes permitiendo la comunicación entre programadores mediante el código y facilitando posibles cambios en el mismo.

3.1.2 Comparativa entre Metodologías de Desarrollo de Software

Se ha realizado una comparativa entre las distintas metodologías de desarrollo de Software comúnmente utilizadas. En la TABLA V se ha tomado aspectos como la curva de aprendizaje de la metodología, las herramientas de integración y el soporte que existe para la misma.

TABLA V. COMPARATIVA ENTRE METODOLOGÍAS PARTE 1

Modelo de Proceso	Curva de aprendizaje	Herramienta de integración	Soporte Externo
RUP	Lenta	Alto Soporte	Alto Soporte
ICONIX	Rápida	Algún Soporte Disponible	Algún Soporte Disponible
XP	Rápida	No mencionado	Algún Soporte Disponible
SCRUM	Rápida	No mencionado	Algún Soporte Disponible

Ahora se comparan otros aspectos como el tamaño del proceso, tamaño del equipo de trabajo y complejidad del problema a resolver.

TABLA VI. COMPARATIVA ENTRE METODOLOGÍAS PARTE 2

Modelo de Proceso	Tamaño del Proceso	Tamaño del Equipo	Complejidad del Problema
RUP	Medio / Extenso	Medio / Extenso	Medio / Alto
ICONIX	Pequeño / Medio	Pequeño / Medio	Medio / Alto
XP	Pequeño / Medio	Pequeño	Pequeño / Medio
SCRUM	Pequeño / Medio	Pequeño / Medio	Medio / Alto

Se puede apreciar que metodologías como RUP se enfocan a proyectos grandes gracias a su extensa y completa documentación mientras que otras como XP es más simplista y esta enfocados a proyectos medianos y pequeños.

3.2 Metodología para el Desarrollo de Agentes

De igual manera que existen metodologías para desarrollo de software, también se presentan un conjunto de metodologías optimizadas y adaptadas para el desarrollo de sistemas multiagentes a fin de seguir procesos que facilitan el desarrollo e incrementan la posibilidad de éxito. Para el presente caso se analizará la metodología GAIA.

3.2.1 Metodología GAIA

GAIA es una metodología que se enfoca en los procesos de análisis y diseño orientados a agentes. Para conseguir este objetivo se hace uso de una serie de modelos de análisis y diseño centrándose en la idea de que la construcción de sistemas basados en agentes es un proceso de diseño organizacional [43].

GAIA es conocida como una metodología para análisis de requerimientos y diseño de sistemas multiagentes, desarrollada para dominios con las siguientes características [43]:

- 1.** Los agentes son parte central del sistema.
- 2.** Los agentes son heterogéneos. Diferentes agentes pueden ser implementados utilizando diferentes lenguajes de programación, arquitecturas y técnicas. La metodología no hace referencia alguna acerca de la plataforma de implementación.
- 3.** La estructura de organización del sistema es estática, las relaciones entre los agentes no cambian en tiempo de ejecución.
- 4.** Las habilidades de los agentes y los servicios que realizan son estáticos, no cambian en tiempo de ejecución.
- 5.** Hay un número manejable de agentes (menos de 100).

GAIA es un proceso sistemático que, partiendo desde sentencias de requerimientos, arriba a un diseño que está lo suficientemente detallado como para ser implementado directamente. El esquema de módulos de la metodología se observa en la siguiente figura:

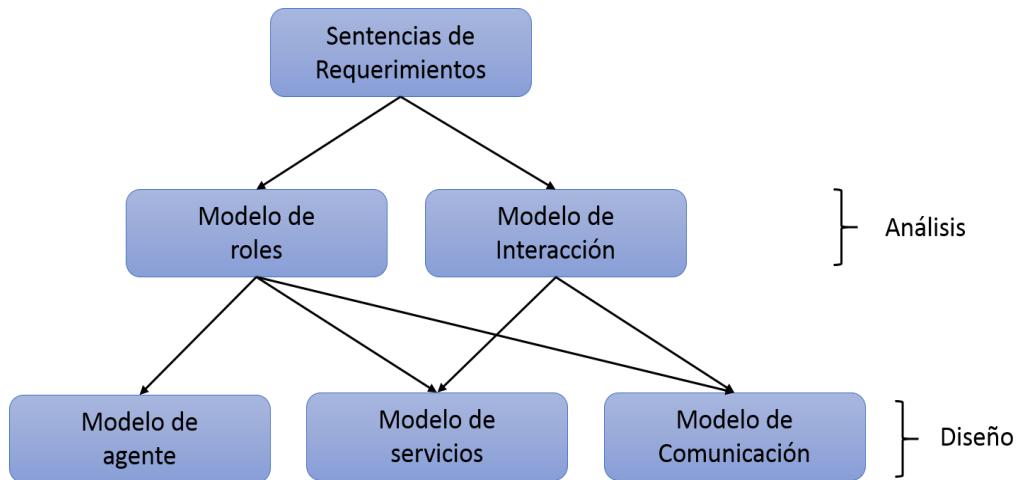


Figura 22. Módulos de la metodología GAIA

En la etapa de análisis tenemos dos tipos de modelos:

El Modelo de Roles: Permiten identificar los distintos tipos de roles que se encuentran en el sistema. Con cada rol hay asociados unos permisos y unas responsabilidades. Las responsabilidades son de dos tipos, de vida y de seguridad. Con estos conceptos, se puede especificar un esquema de rol para cada tipo de rol en la organización y conformar así el modelo de roles. Un esquema de rol presenta una descripción del rol, sus protocolos y actividades, sus permisos y las responsabilidades de vida y seguridad. [43].

El Modelo de Interacción: Es una serie de definiciones de protocolo, una para cada tipo distinto de interacción entre los roles. Cada definición de protocolo consiste de un propósito de la interacción, los roles iniciador y receptor de la misma, las entradas y salidas y el procesamiento ejecutado. [43].

El proceso de diseño en GAIA involucra la transformación de los modelos de análisis a un nivel de abstracción suficientemente bajo para que sea posible implementar los agentes haciendo uso de las técnicas de diseño tradicionales, tales como las que hacen uso de la orientación

En la etapa de diseño, GAIA hace uso de tres tipos de modelos:

El Modelo de Agentes: Permite definir los diferentes tipos de agente que se podrán encontrar en el sistema y el número de instancias que se tendrán de cada uno en tiempo de ejecución. Un tipo de agente es un conjunto de roles, es decir, un tipo de agente puede asumir uno o más roles aunque lo contrario no es cierto. El modelo de agentes

se construye mediante un árbol de tipos de agentes, en el cual los nodos hoja corresponden a los roles y los restantes a los tipos de agentes. Para cada tipo de agente se debe definir un cuantificador de instancia, el cual precisa el número de instancias que se tendrán en el sistema en tiempo de ejecución. [43].

El Modelo de Servicios: Expone los servicios que cada tipo de agente va a implementar, entendiendo por servicio, cierta funcionalidad. Cada servicio es derivado de las actividades y protocolos, así como de sus propiedades de vida y seguridad encontradas en la etapa de análisis. El modelo de servicios se compone de las propiedades de cada uno de los servicios: las entradas, las salidas, las precondiciones y pos condiciones. Las entradas y salidas proceden en forma directa del modelo de protocolos. Las precondiciones y las pos condiciones constituyen limitantes en los servicios y son derivadas de las propiedades de un rol. [43].

El Modelo de Conocidos: Permite precisar los enlaces de comunicación que existen entre tipos de agentes e identificar posibles problemas de embottellamiento surgidos por el uso de estos canales de comunicación. El modelo es un grafo dirigido, en donde cada nodo corresponde a un tipo de agente y las aristas se relacionan con los caminos de comunicación. Así pues, un grafo A→B, muestra que hay un camino de A a B, pero no necesariamente de B a A. [43].

3.2.2 Comparativa entre metodologías de agentes

A continuación se presenta una tabla con características comunes entre distintas metodología utilizadas para el desarrollo de sistemas multiagente y que facilita una visión global en qué casos pueden ser aplicadas.

TABLA VII. COMPARATIVA ENTRE METODOLOGÍAS DE AGENTES

Metodología	Tamaño del equipo	Complejidad del problema
GAIA	Pequeño / Medio	Pequeño / Medio
TROPOS	Pequeño / Medio	Medio / Grande
MESSAGE	Medio / Grande	Medio / Grande
INGENIAS	Medio / Grande	Medio / Grande

Cabe resaltar que siempre es importante conocer primero el problema a solucionar para así escoger la metodología que más se adapte a estas necesidades.

e. Materiales y Métodos

La implementación del sistema inteligente para predicción del desempeño académico se lo realizó haciendo uso de métodos como el deductivo, inductivo y científico a fin de realizar un análisis

Método inductivo: Partiendo de lo particular a lo general. Este método permitió que mediante la obtención de requerimientos y su posterior análisis, se obtuviera una visión global del alcance y funcionalidad del sistema a desarrollar enfocándose en el cumplimiento de los objetivos planteados.

Método deductivo: Partiendo de lo general a lo particular, La aplicación de este método ha permitido conocer de manera específica cada uno de los problemas más comunes que se presentan al momento de llevar un continuo control del desempeño académico por parte de los estudiantes y docentes guías.

Para la obtención de información, se buscó bibliografía relacionada tema de aplicación de agentes inteligentes en el desarrollo de sistemas web siendo estos artículos científicos, tesis doctorales, sitios especializados y oficiales de la temática a buscar, etc.

Otro aspecto fundamental es la utilización de metodologías tanto para el desarrollo del sistema web como para el análisis e implementación del agente inteligente a fin de cumplir con los objetivos planteados al principio.

Metodología XP (Programación extrema): Programación Extrema es una metodología ágil para el desarrollo de aplicaciones software. Su principal filosofía es satisfacer totalmente las necesidades planteadas por el cliente. Su utilización se reflejó durante todo el transcurso del proyecto:

En la fase de planificación se realizó el levantamiento de requerimientos del sistema mediante el uso de historias de usuario.

En la fase de diseño se lo realizó haciendo uso de las tarjetas CRC, interfaces de usuario y procurando funcionalidad mínima con soluciones simples.

En la fase de desarrollo lo principal fue la especificación de los estándares bajo los cuales se programó el sistema a más del desarrollo de pruebas unitarias de código para garantizar la calidad del mismo.

Finalmente en la fase de pruebas se realizó las pruebas de aceptación garantizando que el sistema cumple con los objetivos para los cuales fue creado.

Metodología GAIA: Es una metodología para el desarrollo de sistemas multiagente. Su principal objetivo es maximizar la calidad global del sistema.

Dentro del desarrollo del proyecto, esta metodología fue fundamental para el análisis, diseño e implementación del agente inteligente que posteriormente fue integrado al sistema web. Para conseguir este fin se realizó ciertos modelos y diagramas que permitían abstraerse de mejor manera para crear la lógica del agente.

En la fase de análisis se modelo ciertos aspectos fundamentales como: Modelo de agentes que a su vez permite definir el modelo de roles, protocolos y servicios.

En la fase de diseño lo más importante es el diseño de interacción del agente para definir claramente su funcionalidad.

Posteriormente para la codificación se procuró seguir con los estándares establecidos anteriormente en XP a sí mismo para las pruebas unitarias y demás.

Materiales

Lenguaje de programación Java: Se utilizó para realizar la codificación del sistema gracias a la gran cantidad de librerías y frameworks que permiten el desarrollo de aplicaciones robustas. A más de estos aspectos se suma la amplia documentación que posee con ejemplos claros y concretos.

Entorno de desarrollo Eclipse: Es una herramienta muy completa para el desarrollo de aplicaciones basadas en Java. Se caracteriza por la fácil integración de plugins para ampliación de funcionalidad y demás.

ORM Hibernate: Facilita en gran medida la interacción entre el sistema y la base de datos permitiendo al desarrollador enfocarse de lleno en la lógica de negocio. Su aplicación se vio plasmada en todo momento ya sea para consulta o grabado de información en Base de Datos.

Base de datos MySql: Una de las mejores bases de datos relacionales, se caracteriza por su velocidad a la hora de consulta de información. Gracias a esta ventaja se optó por su aplicación como medio de persistencia de datos para el sistema.

Framework JSF: Permite el desarrollo de aplicaciones Web basadas en Java mediante la utilización del patrón de diseño Modelo – Vista – Controlador. Claramente constituyó la base donde se desarrolló el sistema.

Framework Spring: Se caracteriza por su facilidad en el desacoplamiento de aplicaciones Java mediante la utilización de inyección de dependencias (DI). Su aplicación se vio reflejada en todo momento como medio donde se declaran las clases (Beans) y su posterior inyección.

Librería Primefaces: Una de las mejores librerías para JSF que se enfoca en agregar funcionalidad a sus distintos componentes. Su utilización fue base para el diseño de las vistas tanto para el docente como para el estudiante, facilitando la utilización de AJAX y la interacción con los distintos componentes visuales.

Librería JAX-WS: Se enfoca en facilitar la conexión con un Web Service y la generación de código para interacción con el mismo. Su aplicación dentro del sistema se vio plasmada al momento de conseguir una correcta conexión con los Web Services del SGA a más de la generación de métodos que permiten la recuperación de información.

Framework JADE: Constituye un entorno completo para desarrollo de sistemas multiagente. Su utilización dentro del sistema se vio plasmada al momento de realizar la codificación e integración del agente inteligente con el sistema web.

f. Resultados

1 Hipótesis

Un sistema web da la posibilidad al docente de llevar un registro detallado de calificaciones e integrar un agente inteligente que presente al estudiante gráficas predictivas sobre el desempeño académico.

2 Fase I: Análisis Inicial

Constituye la etapa inicial del proyecto donde el principal objetivo es obtener los requerimientos para el sistema.

2.1 Metodología XP: planificación del sistema

Es sistema tiene por objetivo fundamental presentar al estudiante gráficamente su desempeño académico durante el trascurso de módulo, y una predicción sobre cual serán los posibles resultados al final del mismo vía Web.

De acuerdo a la metodología que se está utilizando (Programación Extrema XP), en la fase de diseño vamos a realizar el levantamiento de requerimientos mediante las historias de usuario, plan de entregas, tarjetas CRC, cronograma de entregas, etc.

2.1.1 Especificación de requerimientos globales del sistema

El Sistema Inteligente para predicción del desempeño académico de los estudiantes universitarios se encuentra desarrollado en un entorno Web, utilizando una arquitectura de tres Niveles. El Sistema contiene un módulo que va a permitir al docente administrar las calificaciones de los estudiantes tomando en cuenta la materia y los distintos parámetros de calificación como por ejemplo lecciones, trabajos, evaluaciones, etc. y otro módulo para estudiantes el cual permitirá conocer cómo va el progreso académico actual y una predicción a futuro del mismo, partiendo de las calificaciones ingresadas por el docente, dicha información será presentada gráficamente.

El Sistema posee los siguientes módulos:

- Docente
- Estudiante

De acuerdo a XP los requerimientos se los debe tomar mediante las historias de usuario que a continuación se las va a identificar y describir.

2.1.2 Identificación de las Historias de usuario

Es necesario conocer como parte inicial los usuarios que van para acceder al sistema, para esto se los va a describir y son los siguientes:

Docente: Es la persona que tiene acceso a las listas de los estudiantes en los distintos módulos y materias sobre las cuales imparte clases, a su vez puede administrar las calificaciones de lecciones, deberes, trabajos, evaluaciones, etc.

Estudiante: Es la persona que accede al sistema para conocer el estado de sus calificaciones y una predicción grafica de su desempeño académico del presente módulo que está cursando.

2.1.2.1 Ingresar al Sistema

En esta historia se describe lo necesario para que el internauta pueda ingresar al sistema mediante una cédula y contraseña que son los mismos utilizados para ingreso al Sistema de Gestión Académica (SGA) por lo que no es necesaria ninguna información adicional. En la TABLA VIII se muestra la historia de usuario “*Ingresar al Sistema*”.

Campos a desplegar:

- Cédula
- Contraseña

TABLA VIII HISTORIA DE USUARIO: INGRESAR AL SISTEMA

Historia de Usuario	
Número: 1	Nombre: Ingresar al Sistema
Usuario: Estudiante, Docente	
Prioridad en Negocio: Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Alto	Programador Responsable: Jhymer Martínez
Descripción: El internauta deberá ingresar la información en los campos cedula y contraseña que son las mismas utilizadas en el Sistema de Gestión Académica tendrá acceso al Sistema Web.	
Observaciones:	

2.1.2.2 Generar nuevo registro de calificaciones

En esta historia se describe todo lo necesario para que el usuario *Docente* genere un nuevo registro que contiene la información relacionada con la oferta, carrera, materia, módulo y paralelo al cual el docente imparte clases, a más de la lista de estudiantes perteneciente ha dicho paralelo. En la TABLA IX se muestra la historia de usuario “Generar nuevo registro de calificaciones para una materia específica”.

Campos a desplegar:

- Periodo académico
- Oferta académica
- Carrera
- Módulo
- Paralelo
- Materia

TABLA IX. HISTORIA DE USUARIO GENERAR NUEVO REGISTRO DE CALIFICACIONES

Historia de Usuario	
Número: 2	Nombre: Generar nuevo registro de calificaciones
Usuario: Docente	
Prioridad en Negocio: Alta	Puntos Estimados: 8
Riesgo en Desarrollo: Alto	Programador Responsable: Jhymer Martínez
Descripción: El sistema deberá permitir docente crear un nuevo registro de calificaciones para una de sus materias mediante la especificación del periodo, oferta, módulo y paralelo en el que imparte clases.	
Observaciones: En caso de ser el primer ingreso del docente, esta operación se realizará automáticamente.	

2.1.2.3 Calificar estudiantes

En esta historia se describe lo necesario para que el docente realice la calificación individual de cada uno de sus estudiantes en las distintas materias que son impartidas por el docente y tomando en cuenta los distintos parámetros de calificación como son evaluaciones, lecciones, trabajos autónomos, deberes o tareas, participaciones en clase y laboratorios o talleres. En la TABLA X se muestra la historia de usuario “*Calificar estudiantes*”.

Campos a desplegar:

- Nombre y número de parámetro
- Fecha del aporte
- Referencia general sobre el aporte
- Nombre del estudiante
- Calificación
- Anotación adicional

TABLA X. HISTORIA DE USUARIO: CALIFICAR ESTUDIANTES

Historia de Usuario	
Número: 3	Nombre: Calificar estudiantes
Usuario: Docente	
Prioridad en Negocio: Alta	Puntos Estimados: 13
Riesgo en Desarrollo: Alto	Programador Responsable: Jhymer Martínez
Descripción: El sistema debe permitir al docente calificar a sus estudiantes dependiendo de la materia y tomando en cuenta los distintos parámetros de calificación (evaluaciones, lecciones, trabajos autónomos, etc.)	
Observaciones:	

2.1.2.4 Editar calificaciones

En esta historia se describe todo lo necesario para que el docente edite los registros de calificaciones que han sido ingresadas previamente. En la TABLA XI se muestra la historia de usuario “*Editar calificaciones*”.

Campos a desplegar:

- Nombre y número de parámetro
- Fecha del aporte
- Referencia general sobre el aporte
- Nombre del estudiante
- Calificación
- Anotación adicional

TABLA XI. HISTORIA DE USUARIO: EDITAR CALIFICACIONES

Historia de Usuario	
Número: 4	Nombre: Editar calificaciones
Usuario: Docente	
Prioridad en Negocio: Alta	Puntos Estimados: 8
Riesgo en Desarrollo: Medio	Programador Responsable: Jhymer Martínez
Descripción: El sistema dará la posibilidad al docente de editar las calificaciones previamente ingresadas.	
Observaciones:	

2.1.2.5 Eliminar calificaciones

En esta historia se describe todo lo necesario para que el docente elimine los registros de calificaciones que han sido ingresadas previamente. En la TABLA XII se muestra la historia de usuario “*Eliminar calificaciones*”.

Campos a desplegar:

- Nombre y número de parámetro
- Fecha del aporte
- Referencia general sobre el aporte
- Nombre del estudiante
- Calificación
- Anotación adicional

TABLA XII. HISTORIA DE USUARIO: ELIMINAR CALIFICACIONES

Historia de Usuario	
Número: 5	Nombre: Eliminar calificaciones
Usuario: Docente	
Prioridad en Negocio: Alta	Puntos Estimados: 8
Riesgo en Desarrollo: Medio	Programador Responsable: Jhymer Martínez
Descripción: El sistema dará la posibilidad al docente de eliminar las calificaciones previamente ingresadas.	
Observaciones: Por seguridad solo se permite eliminar el último registro de calificación.	

2.1.2.6 Configurar parámetros de calificación

En esta historia se describe todo lo necesario para que el docente realice configuraciones relacionadas con la cantidad de parámetros que se estiman evaluar en el módulo a más de los pesos que poseerán los mismos, dichos pesos son utilizados para generar el promedio final. En la TABLA XIII se muestra la historia de usuario “Configurar parámetros de calificación”.

Campos a desplegar:

- Cantidad de evaluaciones
- Cantidad de lecciones
- Cantidad de trabajos autónomos
- Cantidad de tareas - deberes
- Cantidad de participaciones
- Cantidad de laboratorios – talleres
- Porcentaje de evaluaciones
- Porcentaje de lecciones
- Porcentaje de trabajos autónomos
- Porcentaje tareas - deberes
- Porcentaje de participaciones
- Porcentaje de laboratorios – talleres

TABLA XIII. HISTORIA DE USUARIO: CONFIGURAR PARÁMETROS DE CALIFICACIÓN

Historia de Usuario	
Número: 6	Nombre: Configurar parámetros de calificación
Usuario: Docente	
Prioridad en Negocio: Alta	Puntos Estimados: 8
Riesgo en Desarrollo: Medio	Programador Responsable: Jhymer Martínez
Descripción: El sistema debe dar la opción al docente de configurar la cantidad y peso (porcentaje de calificación a la nota final) de los distintos parámetros de calificación que son evaluaciones, lecciones, tareas, trabajos autónomos, participaciones y laboratorios o talleres.	
Observaciones: El parámetro perteneciente a evaluaciones debe ser obligatorio y uno como mínimo	

2.1.2.7 Presentar calificaciones detalladas

En esta historia se describe todo lo necesario para que el sistema presente al docente la información detallada sobre las calificaciones almacenadas, de una manera más específica y detallada. En la TABLA XIV se muestra la historia de usuario “*Presentar calificaciones detalladas*”.

Campos a desplegar:

- Nombre y número de parámetro
- Fecha del aporte
- Referencia general sobre el aporte
- Nombre del estudiante
- Calificación
- Anotación adicional

TABLA XIV. HISTORIA DE USUARIO: PRESENTAR CALIFICACIONES DETALLADAS

Historia de Usuario	
Número: 7	Nombre: Presentar calificaciones detalladas
Usuario: Docente	
Prioridad en Negocio: Media	Puntos Estimados: 8
Riesgo en Desarrollo: Medio	Programador Responsable: Jhymer Martínez
Descripción: El sistema debe generar un registro detallado de las calificaciones llevadas por el docente de una manera más específica e individualizada para cada estudiante calificado.	
Observaciones:	

2.1.2.8 Presentar promedios al docente

En esta historia se describe todo lo necesario para que el sistema presente al docente los promedios generales de cada estudiante en las distintas materias impartidas. En la TABLA XV se muestra la historia de usuario “*Presentar promedios al docente*”.

Campos a desplegar:

- Número de estudiante
- Nombres y apellidos del estudiante
- Promedios parciales
- Promedio final

TABLA XV. HISTORIA DE USUARIO: PRESENTAR PROMEDIOS DE CALIFICACIONES

Historia de Usuario	
Número: 8	Nombre: Presentar promedios de calificaciones
Usuario: Docente	
Prioridad en Negocio: Media	Puntos Estimados: 5
Riesgo en Desarrollo: Medio	Programador Responsable: Jhymer Martínez
Descripción: El sistema debe generar un registro detallado de las calificaciones llevadas por el docente de una manera más específica e individualizada para cada estudiante calificado.	
Observaciones:	

2.1.2.9 Seleccionar parámetros iniciales

En esta historia se describe todo lo necesario para que el usuario *Estudiante* seleccione parámetros iniciales para la presentación de calificaciones y gráficas. Estos parámetros están relacionados con la oferta académica, carrera, módulo y paralelo. En la TABLA XVI se muestra la historia de usuario “*Seleccionar parámetros iniciales*”.

Campos a desplegar:

- Oferta académica
- Carrera
- Módulo
- Paralelo

TABLA XVI. HISTORIA DE USUARIO: SELECCIONAR PARÁMETROS INICIALES

Historia de Usuario	
Número: 9	Nombre: Seleccionar parámetros iniciales
Usuario: Estudiante	
Prioridad en Negocio: Alta	Puntos Estimados: 8
Riesgo en Desarrollo: Alto	Programador Responsable: Jhymer Martínez
Descripción: El estudiante podrá seleccionar la oferta académica, carrera y módulo sobre el cual se realizará la consulta de calificaciones en las distintas materias.	
Observaciones:	

2.1.2.10 Presentar gráficas de desempeño académico

En esta historia se describe todo lo necesario para el sistema genere y presente al estudiante, gráficas relacionadas con su desempeño académico desde un punto de vista actual y con una predicción a futuro. En la TABLA XVII se muestra la historia de usuario “Presentar gráficas de desempeño académico”.

Campos a desplegar:

- Oferta académica
- Carrera
- Módulo
- Paralelo
- Materia

TABLA XVII. HISTORIA DE USUARIO: PRESENTAR GRÁFICAS DE DESEMPEÑO ACADÉMICO

Historia de Usuario	
Número: 10	Nombre: Presentar gráficas de desempeño académico
Usuario: Estudiante	
Prioridad en Negocio: Alta	Puntos Estimados: 21
Riesgo en Desarrollo: Alto	Programador Responsable: Jhymer Martínez
<p>Descripción: El sistema presentará graficas con información sobre el desempeño académico actual y predicción a futuro de las distintas materias que cursa el estudiante. Cada gráfica le pertenece a un parámetro de calificación en específico como es evaluaciones, lecciones, tareas, trabajos autónomos, participaciones y laboratorios o talleres.</p>	
<p>Observaciones:</p>	

2.1.2.11 Ajustar factores que influyen en el desempeño académico

En esta historia se describe todo lo necesario para el sistema genere y presente al estudiante un formulario con una lista de parámetros que influyen en el desempeño académico con el fin de que el estudiante los ajuste según su realidad. En la TABLA XVIII se muestra la historia de usuario “Ajustar factores que influyen en el desempeño académico”.

Campos a desplegar (por cada aporte):

- Factores personales
- Factores sociales
- Factores institucionales

**TABLA XVIII. HISTORIA DE USUARIO: AJUSTAR FACTORES QUE INFLUYEN EN
EL DESEMPEÑO ACADÉMICO**

Historia de Usuario	
Número: 11	Nombre: Ajustar factores que influyen en el desempeño académico
Usuario: Estudiante	
Prioridad en Negocio: Alta	Puntos Estimados: 13
Riesgo en Desarrollo: Medio	Programador Responsable: Jhymer Martínez
Descripción: El sistema presentará un formulario el cual contiene distintos parámetros que influyen en la parte académica del estudiante. Este formulario podrá ser ajustado según la realidad del estudiante.	
Observaciones:	

2.1.2.12 Generar un registro detallado de calificaciones

En esta historia se describe todo lo necesario para el sistema genere y presente al estudiante una vista con información detallada de sus calificaciones en las distintas materias y tomando en cuenta los distintos parámetros de calificación (evaluaciones, lecciones, etc.). En la TABLA XIX se muestra la historia de usuario “*Generar un registro detallado de calificaciones*”.

Campos a desplegar (por cada aporte):

- Nombre y número de parámetro
- Fecha del aporte
- Referencia general sobre el aporte
- Calificación
- Anotación adicional

TABLA XIX. HISTORIA DE USUARIO: GENERAR UN REGISTRO DETALLADO DE CALIFICACIONES

Historia de Usuario	
Número: 12	Nombre: Generar un registro detallado de calificaciones
Usuario: Estudiante	
Prioridad en Negocio: Alta	Puntos Estimados: 13
Riesgo en Desarrollo: Alto	Programador Responsable: Jhymer Martínez
Descripción: El sistema generará automáticamente un registro con información detallada de las calificaciones impartidas por el docente en los distintos parámetros de calificación (lecciones, evaluaciones, deberes, etc.)	
Observaciones:	

2.1.2.13 Presentar promedios al estudiante

En esta historia se describe todo lo necesario para el sistema genere y presente al estudiante los promedios generales de sus calificaciones en las distintas materias cursadas. En la TABLA XX se muestra la historia de usuario “*Presentar promedios al estudiante*”.

Campos a desplegar:

- Promedio evaluaciones
- Promedio lecciones
- Promedio trabajos autónomos
- Promedio deberes o tareas
- Promedio participaciones
- Promedio laboratorios o talleres
- Promedio final

TABLA XX. HISTORIA DE USUARIO: PRESENTAR PROMEDIOS AL ESTUDIANTE

Historia de Usuario	
Número: 13	Nombre: Presentar promedios al estudiante
Usuario: Estudiante	
Prioridad en Negocio: Alta	Puntos Estimados: 5
Riesgo en Desarrollo: Alto	Programador Responsable: Jhymer Martínez
Descripción: El sistema generará automáticamente los promedios parciales y finales con las calificaciones del estudiante.	
Observaciones:	

2.1.2.14 Presentar sección con información para el mejoramiento del rendimiento académico

En esta historia se describe todo lo necesario para el sistema genere y presente al estudiante una sección con información para el mejoramiento del rendimiento académico. En la TABLA XXI se muestra la historia de usuario “*Presentar sección con información para el mejoramiento del rendimiento académico*”.

TABLA XXI. HISTORIA DE USUARIO: PRESENTAR SECCIÓN CON INFORMACIÓN PARA EL MEJORAMIENTO DEL RENDIMIENTO ACADÉMICO

Historia de Usuario	
Número: 14	Nombre: Presentar sección con información para el mejoramiento del rendimiento académico
Usuario: Estudiante	
Prioridad en Negocio: Baja	Puntos Estimados: 5
Riesgo en Desarrollo: Bajo	Programador Responsable: Jhymer Martínez
Descripción: El sistema presentará al estudiante una biblioteca con información selecta enfocada al mejoramiento del rendimiento académico.	
Observaciones:	

Una vez comprendidos en su totalidad los requerimientos del sistema, ya se tiene las bases para empezar con la parte del diseño que es el siguiente punto a analizar.

2.2 Metodología GAIA: Análisis del agente

La metodología GAIA facilita el análisis y diseño del agente inteligente que se utilizará para la generación de calificaciones predictivas que serán presentadas mediante gráficas de desempeño al estudiante. Para alcanzar este fin se procede a modelar aspectos relacionados con la funcionalidad que poseerá el agente.

2.2.1 Modelo de Agente

Este modelo tiene como finalidad determinar las actividades que realizará el agente dentro del sistema.

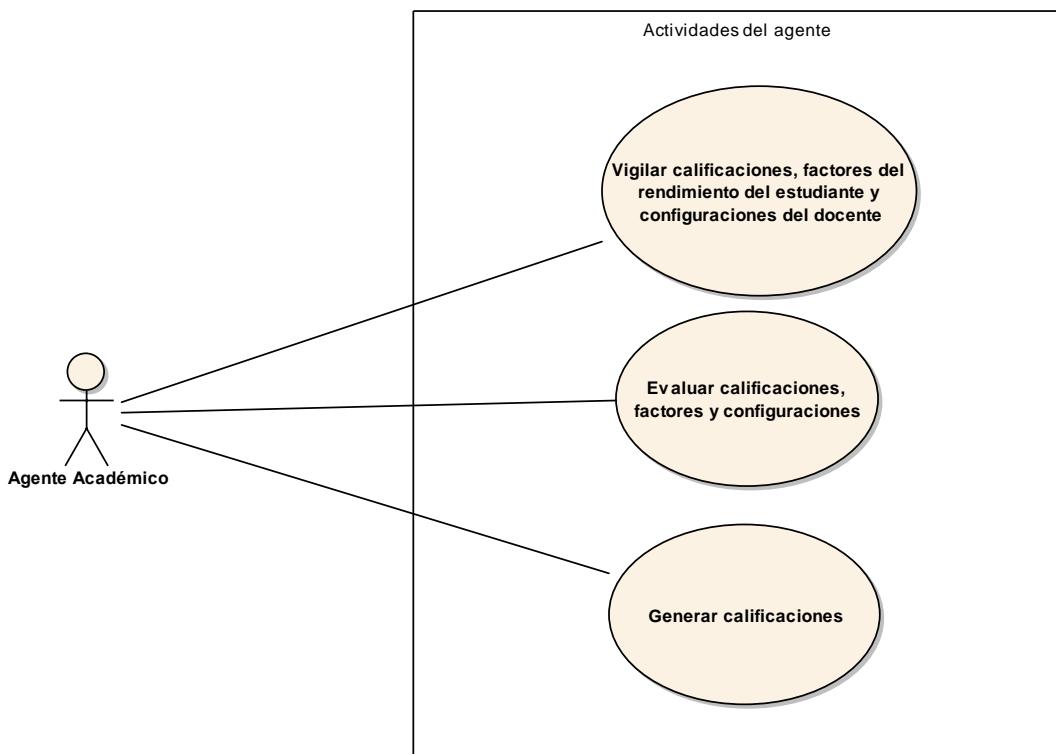


Figura 23. Modelo del agente

En esta fase se define además los roles del agente mediante la utilización de una plantilla donde se especifica una descripción de lo que va a realizar el agente, luego se detallan los protocolos, actividades, permisos y responsabilidades.

2.2.2 Modelo de rol del agente

Un rol necesita un conjunto de permisos que identifique aquellos recursos que puede acceder para poder cumplir con sus responsabilidades. Para el presente caso estos recursos son fuentes de información que han sido extraídas de la base de datos.

TABLA XXII. MODELO DE ROLES DEL AGENTE

Esquema de rol	Agente Académico
Descripción. Este rol involucra que el agente académico mediante los datos de entrada como las calificaciones pertenecientes a evaluaciones, lecciones, trabajos autónomos, tareas, participaciones, talleres a más de información relacionada con los factores de rendimiento académico (personales, sociales e institucionales) del estudiante y la configuración de cantidad y porcentajes para cada parámetro que ha sido ingresada por el docente a fin de generar posibles calificaciones futuras.	
Protocolos y actividades Actividades. <ul style="list-style-type: none">Vigilar calificaciones, factores de rendimiento y configuraciones.- Consiste en observar las calificaciones obtenidas hasta una fecha determinada, a más de los factores de rendimiento ajustados por el estudiante y la cantidad y porcentajes de parámetros que el docente estableció para la materia. Dicha información constituye las creencias del agente.Evaluar calificaciones, factores y configuraciones.- Se realiza un proceso de diagnóstico para verificar que sea posible generar calificaciones futuras con los datos actuales y la factibilidad de generación de nuevas calificaciones que sirvan de guía al estudiante. En esta parte se evidencia los deseos que cumplirá el agente.	

Protocolo.

- **Generar calificaciones.-** En base a la evaluación de los diferentes parámetros se realizara.
 - Procesamiento de la información de entrada
 - Generación de calificaciones predictivas
 - Retorno de información a la aplicación para generación de gráficas con los resultados
 - Alerta al estudiante en caso de calificaciones bajas
 - Información adicional sobre promedios y desempeño en general.

Permisos

- Leer calificaciones del estudiante.
- Leer factores del rendimiento del estudiante.
- Leer configuraciones del docente.

Responsabilidades

Agente Inteligente (Vigilar calificaciones, factores y configuraciones, Evaluarlos, Generar calificaciones)

2.2.3 Definición de protocolos

Constituyen las interacciones entre los roles, en otras palabras es una actividad que necesita una interacción entre roles. A continuación se presenta la definición de protocolos para el agente.

TABLA XXIII. DEFINICIÓN DE PROTOCOLOS

Protocolo: Generar calificaciones
Nombre: Agente Académico
<p>Generación calificaciones predictivas</p> <p>Teniendo como base las calificaciones hasta la fecha, factores del rendimiento y configuraciones, se generan nuevas calificaciones adaptadas al progreso del desempeño académico del estudiante presente y futuro.</p> <p>Retorno de información a la aplicación para generación de gráficas con los resultados.</p> <p>Toda esta nueva información generada es enviada a la aplicación con el fin de ser renderizada y presentada a manera de graficas estadísticas.</p> <p>Alerta e información de promedios al estudiante.</p> <p>En caso de que las calificaciones obtenidas por el estudiante sean muy bajas y no exista suficiente tiempo durante el transcurso del módulo para recuperar el promedio en ese parámetro evaluable, el agente enviará al sistema una alerta que será mostrada al estudiante a través de un mensaje que informará la situación. A más de esto también durante el transcurso del módulo se presentan promedios e información adicional sobre el estado de su desempeño tanto para calificaciones reales como para las generadas por el sistema</p>

2.2.4 Modelo de servicios

En este modelo se detalla las entradas, salidas con su respectiva pre-condición y post-condición dentro de su uso en el sistema.

TABLA XXIV. MODELO DE SERVICIOS DEL AGENTE

Servicio	Entrada	Salidas	Pre-condición	Post-condición
Vigilar calificaciones, factores del rendimiento y configuraciones	<ul style="list-style-type: none"> • Evaluaciones • Lecciones • Trabajos autónomos • Deberes o tareas • Participaciones en clase • Laboratorios o talleres • Factores personales • Factores sociales • Factores institucionales • Configuraciones 	Creencias del agente	<ul style="list-style-type: none"> • Evaluaciones \neq nulo • Lecciones \neq nulo • Trabajos autónomos \neq nulo • Deberes o tareas \neq nulo • Participaciones en clase \neq nulo • Laboratorios o talleres \neq nulo • Factores personales \neq nulo • Factores sociales \neq nulo • Factores institucionales \neq nulo • Configuraciones \neq nulo 	Datos Insuficientes para evaluación de calificaciones
Evaluar calificaciones, factores y configuraciones	<ul style="list-style-type: none"> • Evaluaciones • Lecciones • Trabajos autónomos • Deberes o tareas • Participaciones en clase • Laboratorios o talleres • Factores personales • Factores sociales • Factores institucionales • Configuraciones 	Deseos del agente	<ul style="list-style-type: none"> • Vigilar calificaciones == true 	No se puede evaluar calificaciones sin datos, se debe vigilar antes de poder evaluar.
Generar calificaciones	Notificaciones entrantes de Vigilar y evaluar	Intenciones del agente	<ul style="list-style-type: none"> Vigilar calificaciones == true Evaluar calificaciones == true 	Se debe vigilar y evaluar calificaciones para proceder

3 Fase II: Diseño

En esta fase y una vez que se tienen claros los requerimientos, se procede a diseñar el sistema y el agente inteligente.

3.1 Metodología XP: Diseño del sistema

Siguiendo con la metodología XP en esta fase se realiza el diseño del sistema haciendo uso de tarjetas CRC y los prototipos de pantalla

3.1.1 Análisis general del sistema

El sistema consta de dos módulos claramente definidos que son el módulo de docentes y el módulo de estudiantes, cada uno a su vez permite realizar diferentes operaciones las mismas que fueron detalladas en las historias de usuario. El sistema está enfocado completamente a un entorno web y hace uso de los web services que proporciona el Sistema de Gestión Académica (SGA) de la universidad.

3.1.2 Patrón de diseño del sistema

El sistema está desarrollado siguiendo el patrón MVC lo que facilita la reutilización de código y la flexibilidad. Gracias a estas ventajas permite captarse a cualquier tipo de proyecto como en el presente caso. A continuación se muestra el diagrama del modelo MVC

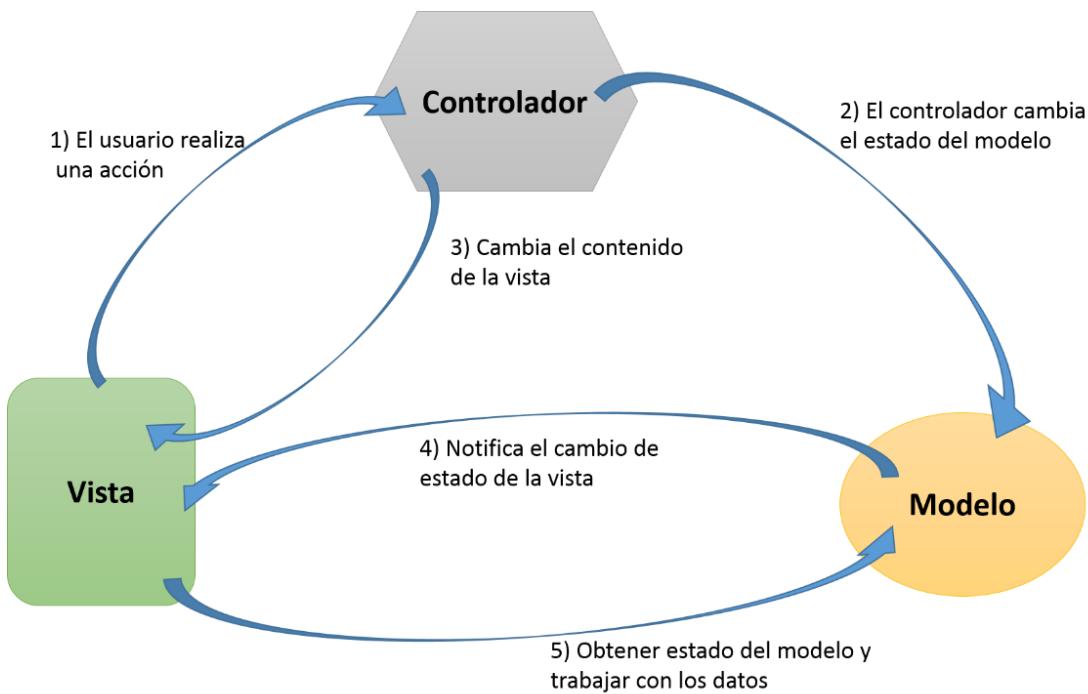


Figura 24. Patrón de diseño para el sistema

3.1.3 Modelo de la base de datos relacional

Mediante el uso de herramientas CASE e Ingeniería inversa se generó el modelo que se presenta a continuación:

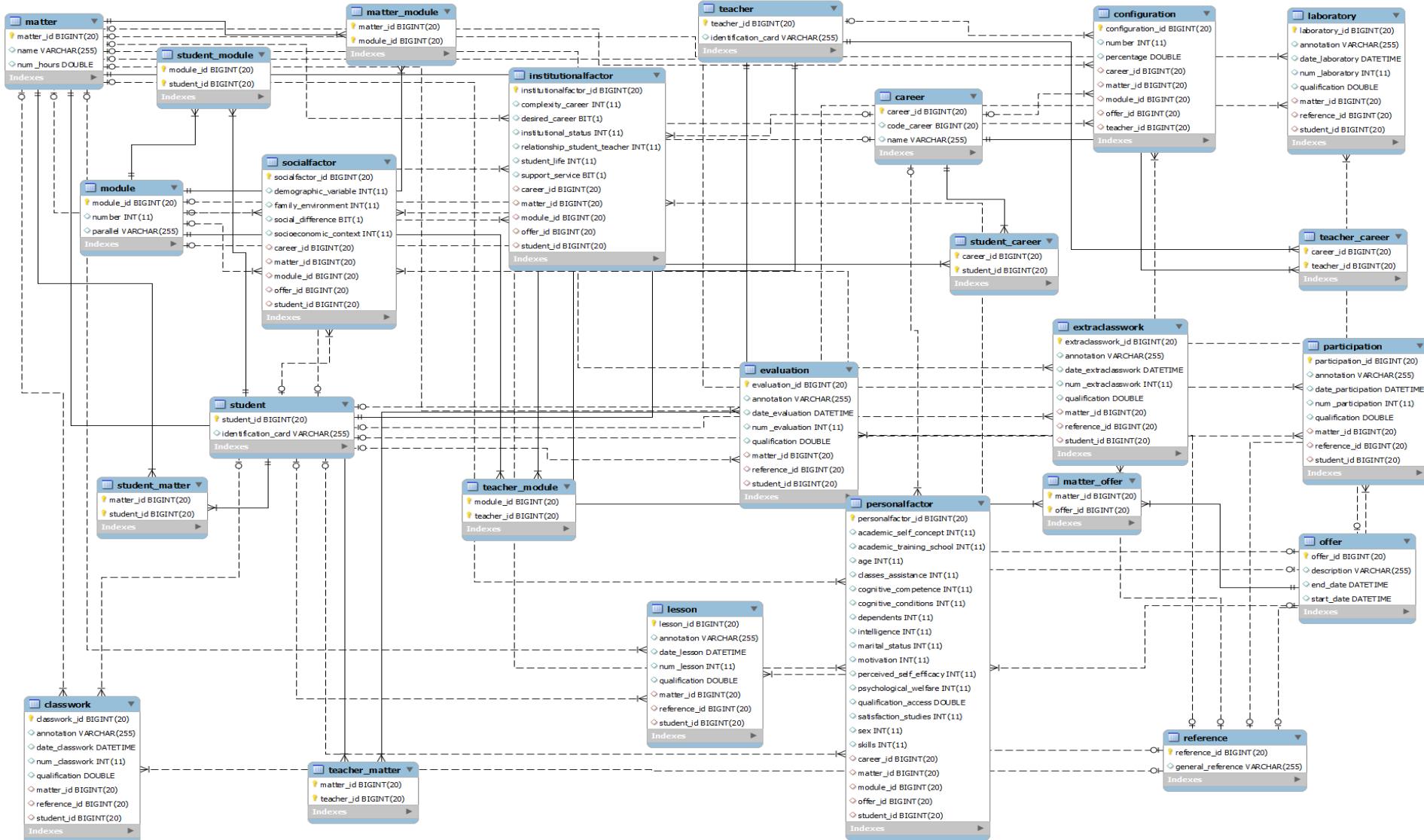


Figura 25. Modelo Físico de Base de Datos

3.1.4 Diccionario de datos

El diccionario de datos permite conocer las características importantes de los datos como el nombre, tipo y funcionalidad por lo que a continuación se describe estos aspectos para cada tabla de la base de datos:

career

En la TABLA XXV se muestra la descripción de la tabla que almacena información relacionada con las distintas carreras universitarias:

TABLA XXV. DICCCIONARIO DE DATOS CAREER

Campo	Tipo de dato	PK	FK	Descripción
career_id	BIGINT	X		Clave única de la tabla career
code_career	BIGINT			Código identificativo de la carrera perteneciente a la base de datos del SGA
Name	VARCHAR			Contiene el nombre de la carrera

classwork

En la TABLA XXVI se muestra la descripción de la tabla que almacena información relacionada con el parámetro de calificación para trabajos autónomos del estudiante:

TABLA XXVI. DICCCIONARIO DE DATOS CLASSWORK

Campo	Tipo de dato	PK	FK	Descripción
classwork_id	BIGINT	X		Clave única de la tabla <i>classwork</i>
annotation	VARCHAR			Contiene anotaciones individuales para cada estudiante relacionadas con este parámetro en específico
date_classwork	DATETIME			Contiene la fecha en la que se realizó el aporte
num_classwork	INT			Contiene un número incremental para identificar cuantos aportes se tiene

qualification	DOUBLE			Contiene la calificación del estudiante
matter_id	BIGINT		X	Clave foránea de la tabla <i>matter</i>
reference_id	BIGINT		X	Clave foránea de la tabla <i>reference</i>
student_id	BIGINT		X	Clave foránea de la tabla <i>student</i>

configuration

En la TABLA XXVII se muestra la descripción de la tabla que almacena las configuraciones para el docente como la posible cantidad de parámetros a evaluar y sus respectivos pesos a la nota final.

TABLA XXVII. DICCIONARIO DE DATOS CONFIGURATION

Campo	Tipo de dato	PK	FK	Descripción
configuration_id	BIGINT	X		Clave única de la tabla <i>configuration</i>
Number	BIGINT			Contiene la cantidad total de parámetros que van a ser evaluados
Percentage	DOUBLE			Contiene el porcentaje o peso de cada parámetro con relación a la nota final.
carrer_id	INT		X	Clave foránea de la tabla <i>carrer</i>
matter_id	BIGINT		X	Clave foránea de la tabla <i>matter</i>
module_id	BIGINT		X	Clave foránea de la tabla <i>module</i>
offer_id	BIGINT		X	Clave foránea de la tabla <i>offer</i>
teacher_id	BIGINT		X	Clave foránea de la tabla <i>student</i>

evaluation

En la TABLA XXVIII se muestra la descripción de la tabla que almacena información relacionada con el parámetro de calificación para evaluaciones:

TABLA XXVIII. DICCIONARIO DE DATOS EVALUATION

Campo	Tipo de dato	PK	FK	Descripción
evaluation_id	BIGINT	X		Clave única de la tabla <i>evaluation</i>
Annotation	VARCHAR			Contiene anotaciones individuales para cada estudiante relacionadas con este parámetro en específico
date_evaluation	DATETIME			Contiene la fecha en la que se realizó el aporte
num_evaluation	INT			Contiene un número incrementable para identificar cuantos aportes se tiene
Qualification	DOUBLE			Contiene la calificación del estudiante
matter_id	BIGINT		X	Clave foránea de la tabla <i>matter</i>
reference_id	BIGINT		X	Clave foránea de la tabla <i>reference</i>
student_id	BIGINT		X	Clave foránea de la tabla <i>student</i>

extraclasswork

En la TABLA XXIX se muestra la descripción de la tabla que almacena información relacionada con el parámetro de calificación para deberes - tareas:

TABLA XXIX. DICCIONARIO DE DATOS EXTRACLASSWORK

Campo	Tipo de dato	PK	FK	Descripción
extraclasswork_id	BIGINT	X		Clave única de la tabla <i>extraclasswork</i>
Annotation	VARCHAR			Contiene anotaciones individuales para cada estudiante relacionadas con este parámetro en específico
date_extraclasswork	DATETIME			Contiene la fecha en la que se realizó el aporte
num_extraclasswork	INT			Contiene un número incrementable para identificar cuantos aportes se tiene
Qualification	DOUBLE			Contiene la calificación del estudiante
matter_id	BIGINT		X	Clave foránea de la tabla <i>matter</i>
reference_id	BIGINT		X	Clave foránea de la tabla <i>reference</i>
student_id	BIGINT		X	Clave foránea de la tabla <i>student</i>

laboratory

En la TABLA XXX se muestra la descripción de la tabla que almacena información relacionada con el parámetro de calificación para laboratorios - talleres:

TABLA XXX. DICCIONARIO DE DATOS LABORATORY

Campo	Tipo de dato	PK	FK	Descripción
laboratory_id	BIGINT	X		Clave única de la tabla <i>laboratory</i>
Annotation	VARCHAR			Contiene anotaciones individuales para cada estudiante relacionadas con este parámetro en específico

date_laboratory	DATETIME			Contiene la fecha en la que se realizó el aporte
num_laboratory	INT			Contiene un número incremental para identificar cuantos aportes se tiene
Qualification	DOUBLE			Contiene la calificación del estudiante
matter_id	BIGINT		X	Clave foránea de la tabla <i>matter</i>
reference_id	BIGINT		X	Clave foránea de la tabla <i>reference</i>
student_id	BIGINT		X	Clave foránea de la tabla <i>student</i>

lesson

En la TABLA XXXI se muestra la descripción de la tabla que almacena información relacionada con el parámetro de calificación para lecciones:

TABLA XXXI. DICCIONARIO DE DATOS LESSON

Campo	Tipo de dato	PK	FK	Descripción
lesson_id	BIGINT	X		Clave única de la tabla <i>lesson</i>
annotation	VARCHAR			Contiene anotaciones individuales para cada estudiante relacionadas con este parámetro en específico
date_lesson	DATETIME			Contiene la fecha en la que se realizó el aporte
num_lesson	INT			Contiene un número incremental para identificar cuantos aportes se tiene
qualification	DOUBLE			Contiene la calificación del estudiante
matter_id	BIGINT		X	Clave foránea de la tabla <i>matter</i>
reference_id	BIGINT		X	Clave foránea de la tabla <i>reference</i>
student_id	BIGINT		X	Clave foránea de la tabla <i>student</i>

matter

En la TABLA XXXII se muestra la descripción de la tabla que almacena información relacionada con las distintas materias:

TABLA XXXII. DICCIONARIO DE DATOS MATTER

Campo	Tipo de dato	PK	FK	Descripción
matter_id	BIGINT	X		Clave única de la tabla matter
Name	VARCHAR			Contiene el nombre asignado a la materia
num_hours	INT			Contiene el número de horas asignada a la materia en la oferta académica seleccionada.

matter_module

En la TABLA XXXIII se muestra la relación entre las tablas *matter* y *module*:

TABLA XXXIII. DICCIONARIO DE DATOS MATTER_MODULE

Campo	Tipo de dato	PK	FK	Descripción
matter_id	BIGINT		X	Clave foránea de la tabla matter
module_id	BIGINT		X	Clave foránea de la tabla module

matter_offer

En la TABLA XXXIV se muestra la relación entre las tablas *matter* y *offer*:

TABLA XXXIV. DICCIONARIO DE DATOS MATTER_OFFER

Campo	Tipo de dato	PK	FK	Descripción
matter_id	BIGINT		X	Clave foránea de la tabla matter
offer_id	BIGINT		X	Clave foránea de la tabla offer

module

En la TABLA XXXV se muestra la descripción de la tabla que almacena información relacionada con los posibles módulos y paralelos:

TABLA XXXV. DICCIONARIO DE DATOS MODULE

Campo	Tipo de dato	PK	FK	Descripción
module_id	BIGINT	X		Clave única de la tabla <i>module</i>
Number	INT			Contiene el número asignado al módulo en cuestión
Parallel	VARCHAR			Contiene el paralelo perteneciente a un módulo en específico.

offer

En la TABLA XXXVI se muestra la descripción de la tabla que almacena información relacionada con las ofertas académicas:

TABLA XXXVI. DICCIONARIO DE DATOS OFFER

Campo	Tipo de dato	PK	FK	Descripción
offer_id	BIGINT	X		Clave única de la tabla <i>offer</i>
description	VARCHAR			Contiene la descripción detallada sobre la oferta académica.
end_date	DATETIME			Contiene la fecha de inicio de actividades para la oferta académica
start_date	DATETIME			Contiene la fecha de culminación de actividades para la oferta académica

participation

En la TABLA XXXVII se muestra la descripción de la tabla que almacena información relacionada con el parámetro de calificación para participaciones en clase:

TABLA XXXVII. DICCIONARIO DE DATOS PARTICIPATION

Campo	Tipo de dato	PK	FK	Descripción
participation_id	BIGINT	X		Clave única de la tabla <i>participation</i>
Annotation	VARCHAR			Contiene anotaciones individuales para cada estudiante relacionadas con este parámetro en específico
date_participation	DATETIME			Contiene la fecha en la que se realizó el aporte
num_participation	INT			Contiene un número incrementable para identificar cuantos aportes se tiene
Qualification	DOUBLE			Contiene la calificación del estudiante
matter_id	BIGINT		X	Clave foránea de la tabla <i>matter</i>
reference_id	BIGINT		X	Clave foránea de la tabla <i>reference</i>
student_id	BIGINT		X	Clave foránea de la tabla <i>student</i>

reference

En la TABLA XXXVIII se muestra la descripción de la tabla que almacena información relacionada con las referencias que se asignan al momento de realizar un aporte en una materia:

TABLA XXXVIII. DICCIONARIO DE DATOS REFERENCE

Campo	Tipo de dato	PK	FK	Descripción
reference_id	BIGINT	X		Clave única de la tabla <i>reference</i>
general_reference	VARCHAR			Contiene la referencia general a la cual se hace referencia el momento de realizar algún aporte

student

En la TABLA XXXIX se muestra la descripción de la tabla que almacena información relacionada con el estudiante:

TABLA XXXIX. DICCIONARIO DE DATOS STUDENT

Campo	Tipo de dato	PK	FK	Descripción
student_id	BIGINT	X		Clave única de la tabla <i>student</i>
Identification_card	VARCHAR			Contiene la cédula de ciudadanía perteneciente al estudiante

student_career

En la TABLA XL se muestra la relación entre las tablas *student* y *career*.

TABLA XL. DICCIONARIO DE DATOS STUDENT_CAREER

Campo	Tipo de dato	PK	FK	Descripción
career_id	BIGINT		X	Clave foránea de la tabla <i>career</i>
student_id	BIGINT		X	Clave foránea de la tabla <i>student</i>

student_matter

En la TABLA XLI se muestra la relación entre las tablas *student* y *matter*.

TABLA XLI. DICCIONARIO DE DATOS STUDENT_MATTER

Campo	Tipo de dato	PK	FK	Descripción
student_id	BIGINT		X	Clave foránea de la tabla <i>student</i>
matter_id	BIGINT		X	Clave foránea de la tabla <i>matter</i>

student_module

En la TABLA XLII se muestra la relación entre las tablas *student* y *module*:

TABLA XLII. DICCIONARIO DE DATOS STUDENT_MODULE

Campo	Tipo de dato	PK	FK	Descripción
module_id	BIGINT		X	Clave foránea de la tabla <i>module</i>
student_id	BIGINT		X	Clave foránea de la tabla <i>student</i>

teacher

En la TABLA XLIII se muestra la descripción de la tabla que almacena información relacionada con el estudiante:

TABLA XLIII. DICCIONARIO DE DATOS TEACHER

Campo	Tipo de dato	PK	FK	Descripción
teacher_id	BIGINT		X	Clave única de la tabla <i>teacher</i>
identification_card	VARCHAR		X	Contiene la cédula de ciudadanía perteneciente al docente

teacher_carrer

En la TABLA XLIV se muestra la relación entre las tablas *teacher* y *carrer*:

TABLA XLIV. DICCIONARIO DE DATOS TEACHER_CARRER

Campo	Tipo de dato	PK	FK	Descripción
carrer_id	BIGINT		X	Clave foránea de la tabla <i>carrer</i>
teacher_id	BIGINT		X	Clave foránea de la tabla <i>teacher</i>

teacher_matter

En la TABLA XLV se muestra la relación entre las tablas *teacher* y *matter*:

TABLA XLV. DICCCIONARIO DE DATOS TEACHER_MATTER

Campo	Tipo de dato	PK	FK	Descripción
matter_id	BIGINT		X	Clave foránea de la tabla <i>matter</i>
teacher_id	BIGINT		X	Clave foránea de la tabla <i>teacher</i>

teacher_module

En la TABLA XLVI se muestra la relación entre las tablas *teacher* y *module*:

TABLA XLVI. DICCCIONARIO DE DATOS TEACHER_MODULE

Campo	Tipo de dato	PK	FK	Descripción
module_id	BIGINT		X	Clave foránea de la tabla <i>module</i>
teacher_id	BIGINT		X	Clave foránea de la tabla <i>teacher</i>

personalfactor

En la TABLA XLVII se muestra la descripción de la tabla que almacena información sobre los factores del desempeño académico de carácter personal:

TABLA XLVII. DICCCIONARIO DE DATOS PERSONALFACTOR

Campo	Tipo de dato	PK	FK	Descripción
personalfactor_id	BIGINT	X		Clave única de la tabla <i>personalfactor</i>
cognitive_competence	INT			Contiene el nivel de competencia cognitiva ingresado por el estudiante
Motivation	INT			Contiene el nivel de motivación

cognitive_conditions	INT			Contiene las condiciones cognitivas
academic_self_concept	INT			Contiene el autoconcepto académico
perceived_self_efficacy	INT			Contiene la autoeficacia percibida por el estudiante
psychological_welfare	INT			Contiene el bienestar psicológico
satisfaction_studies	INT			Contiene el nivel de satisfacción de la carrera seleccionada
classes_assistance	INT			Contiene el porcentaje de asistencia a clases
intelligence	INT			Contiene el nivel de inteligencia del estudiante
skills	INT			Contiene el grado de aptitudes que posee el estudiante
Sex	INT			Contiene el género
Age	INT			Contiene la edad
marital_status	INT			Contiene el estado civil
Dependents	INT			Contiene el número de cargas familiares
academic_training_school	INT			Contiene el nivel de formación previa a la Universidad
qualification_access	DOUBLE			Contiene la calificación de ingreso a la Universidad.
offer_id	BIGINT		X	Clave foránea de la tabla <i>offer</i>
career_id	BIGINT		X	Clave foránea de la tabla <i>career</i>
module_id	BIGINT		X	Clave foránea de la tabla <i>module</i>
matter_id	BIGINT		X	Clave foránea de la tabla <i>matter</i>
student_id	BIGINT		X	Clave foránea de la tabla <i>student</i>

socialfactor

En la TABLA XLVIII se muestra la descripción de la tabla que almacena información sobre los factores del desempeño académico de carácter social:

TABLA XLVIII.DICCIONARIO DE DATOS SOCIALFACTOR

Campo	Tipo de dato	PK	FK	Descripción
socialfactor_id	BIGINT	X		Clave única de la tabla <i>socialfactor</i>
social_difference	BIT			Contiene el nivel de diferencia social existente
family_environment	INT			Contiene el valor que indica qué tan bueno es el entorno familiar
socioeconomic_context	INT			Contiene el valor que indica el contexto socioeconómico del estudiante.
demographic_variable	INT			Contiene el valor que indica su ubicación demográfica con respecto a la universidad
offer_id	BIGINT		X	Clave foránea de la tabla <i>offer</i>
career_id	BIGINT		X	Clave foránea de la tabla <i>career</i>
module_id	BIGINT		X	Clave foránea de la tabla <i>module</i>
matter_id	BIGINT		X	Clave foránea de la tabla <i>matter</i>
student_id	BIGINT		X	Clave foránea de la tabla <i>student</i>

institutionalfactor

En la TABLA XLIX se muestra la descripción de la tabla que almacena información sobre los factores del desempeño académico de carácter institucional:

TABLA XLIX. DICCIONARIO DE DATOS INSTITUTIONALFACTOR

Campo	Tipo de dato	PK	FK	Descripción
institutionalfactor_id	BIGINT	X		Clave única de la tabla <i>socialfactor</i>
desired_career	BIT			Contiene el valor que indica si se eligió la carrera deseada
complexity_career	INT			Contiene el valor de complejidad de la materia
institutional_status	INT			Contiene el valor que indica el estado actual de la institución
support_service	INT			Contiene el valor que indica si existen servicios de apoyo en la institución
student_life				Contiene el valor que indica el ambiente estudiantil
relationship_student_teacher				Contiene el valor que indica la relación entre estudiante y profesor.
offer_id	BIGINT		X	Clave foránea de la tabla <i>offer</i>
career_id	BIGINT		X	Clave foránea de la tabla <i>career</i>
module_id	BIGINT		X	Clave foránea de la tabla <i>module</i>
matter_id	BIGINT		X	Clave foránea de la tabla <i>matter</i>
student_id	BIGINT		X	Clave foránea de la tabla <i>student</i>

3.1.5 Tarjetas CRC

La metodología XP promueve utilizar una correcta especificación de clases, atributos y métodos que se estén utilizando dentro del sistema mediante el uso de tarjetas denominadas CRC (Clase-Responsabilidad-Colaboración) que nos ayudan a la identificación de objetos que intervienen dentro de la misma. Para el presente caso, lo que se va a realizar es el estudio de clases y métodos más importantes para el funcionamiento del sistema.

En la TABLA L se muestra la tarjeta Career, donde se indica todos los atributos como name, code_career, studentCareers, etc. y la relación con otras clases como son Student, Teacher y Configuration.

TABLA L. TARJETA CRC CAREER

Career	
Atributos	
name	Student
code_career	Teacher
studentCareers	Configuration
teacherCareers	
configurations	

En la TABLA LI se muestra la tarjeta ClassWork, donde se indica todos los atributos como num_classWork, qualification, annotation, date_classwork, etc. y la relación con otras clases como son Matter, Student, y Reference.

TABLA LI. TARJETA CRC CLASSWORK

ClassWork	
Atributos	
num_classWork	Matter
qualification	Student
annotation	Reference
date_classwork	
matter	
student	
reference	

En la TABLA LII se muestra la tarjeta Configuration, donde se indica todos los atributos como offer, career, module, matter, etc. y la relación con otras clases como son Career, Module, Matter y Teacher.

TABLA LII. TARJETA CRC CONFIGURATION

Configuration	
Atributos	
offer	Offer
career	Career
module	Module
matter	Matter
teacher	Teacher
percentage	
number	

En la TABLA LIII se muestra la tarjeta Evaluation, donde se indica todos los atributos como num_classWork, qualification, annotation, etc. y la relación con otras clases como son Matter, Student y Reference.

TABLA LIII. TARJETA CRC EVALUATION

Evaluation	
Atributos	
num_classWork	Matter
qualification	Student
annotation	Reference
date_classwork	
matter	
student	
reference	

En la TABLA LIV se muestra la tarjeta ExtraClassWork, donde se indica todos los atributos como num_extraClassWork, qualification, annotation, etc. y la relación con otras clases como son Matter, Student y Reference.

TABLA LIV. TARJETA CRC EXTRACLASSWORK

ExtraClassWork	
Atributos	
num_extraClassWork	Matter
qualification	Student
annotation	Reference
date_extraclasswork	
matter	
student	
reference	

En la TABLA LV se muestra la tarjeta Laboratory, donde se indica todos los atributos como num_laboratory, qualification, annotation, etc. y la relación con otras clases como son Matter, Student y Reference.

TABLA LV. TARJETA CRC LABORATORY

Laboratory	
Atributos	
num_laboratory	Matter
qualification	Student
annotation	Reference
date_laboratory	
matter	
student	
reference	

En la TABLA LVI se muestra la tarjeta Lesson, donde se indica todos los atributos como num_lesson, qualification, annotation, etc. y la relación con otras clases como son Matter, Student y Reference.

TABLA LVI. TARJETA CRC LESSON

Lesson	
Atributos	
num_lesson	Matter
qualification	Student
annotation	Reference
date_lesson	
matter	
student	
reference	

En la TABLA LVII se muestra la tarjeta Matter, donde se indica todos los atributos como name, num_hours, studentMatters, teacherMatters, lessons, etc. y la relación con otras clases como son Student, Teacher, Lesson, Evaluation, ClassWork, etc.

TABLA LVII. TARJETA CRC MATTER

Matter	
Atributos	
name	Student
num_hours	Teacher
studentMatters	Lesson
teacherMatters	Evaluation
lessons	ClassWork
evaluations	ExtraClassWork
classWorks	Participation
extraClassWorks	Laboratory
participations	Module
laboratorios	Offer
matterModules	Configuration
matterOffers	
configurations	

En la TABLA LVIII se muestra la tarjeta Module, donde se indica todos los atributos como number, parallel, studentModules, etc. y la relación con otras clases como son Student, Teacher, Matter y Configuration.

TABLA LVIII. TARJETA CRC MODULE

Module	
Atributos	
number	Student
parallel	Teacher
studentModules	Matter
teacherModules	Configuration
matterModules	
configurations	

En la TABLA LIX se muestra la tarjeta Offer, donde se indica todos los atributos como description, startDate, endDate, etc. y la relación con otras clases como son Matter, y Configuration.

TABLA LIX. TARJETA CRC OFFER

Offer	
Atributos	
description	Matter
startDate	Configuration
endDate	
matterOffers	
configurations	

En la TABLA LX se muestra la tarjeta Participation, donde se indica todos los atributos como num_participation, qualification, annotation, etc. y la relación con otras clases como son Matter, Student y Reference.

TABLA LX. TARJETA CRC PARTICIPATION

Participation	
Atributos	
num_participation	Matter
qualification	Student
annotation	Reference
date_participation	
matter	
student	
reference	

En la TABLA LXI se muestra la tarjeta Reference, donde se indica todos los atributos como general_reference, lessons, evaluations, etc. y la relación con otras clases como son Lesson, Evaluation, ClassWork, ExtraClassWork, Participation y Laboratory.

TABLA LXI. TARJETA CRC REFERENCE

Reference	
Atributos	
general_reference	Lesson
lessons	Evaluation
evaluations	ClassWork
classWorks	ExtraClassWork
extraClassWorks	Participation
participations	Laboratory
laboratorios	

En la TABLA LXII se muestra la tarjeta Student, donde se indica todos los atributos como identification_card, first_name, last_name, lessons, etc. Los métodos principales como generateName() para generar el nombre del estudiante y la relación con otras clases como son Matter, Career y Module.

TABLA LXII. TARJETA CRC STUDENT

Student	
Atributos	
identification_card	Matter
first_name	Career
last_name	Module
lessons	
evaluations	
classWorks	
extraClassWorks	
participations	
laboratorios	
studentMatters	
studentCareers	
studentModules	
Operaciones	
generateName()	

En la TABLA LXIII se muestra la tarjeta Teacher, donde se indica todos los atributos como identification_card, first_name, last_name, teacherMatters, etc. Los métodos principales como generateName() para generar el nombre del docente y la relación con otras clases como son Matter, Career y Module.

TABLA LXIII. TARJETA CRC TEACHER

Teacher	
Atributos	
identification_card	Matter
first_name	Career
last_name	Module
teacherMatters	Configuration
teacherModules	
teacherCareers	
configurations	
Operaciones	
generateName()	

En la TABLA LXIV se muestra la tarjeta PersonalFactor, donde se indica todos los atributos como cognitiveCompetence, motivation, cognitiveConditions, etc. y la relación con otras clases como son Offer, Matter, Career, Module y Student.

TABLA LXIV. TARJETA CRC PERSONALFACTOR

PersonalFactor	
Atributos	
cognitiveCompetence	Offer
motivation	Matter
cognitiveConditions	Career
academicSelfConcept	Module
perceivedSelfEfficacy	Student
psychologicalWelfare	
satisfactionStudies	
classesAssistance	
intelligence	
skills	
sex	
age	

maritalStatus	
dependents	
academicTrainingSchool	
qualificationAccess	

En la TABLA LXV se muestra la tarjeta SocialFactor, donde se indica todos los atributos como socialDifference, familyEnvironment, etc. y la relación con otras clases como son Offer, Matter, Career, Module y Student.

TABLA LXV. TARJETA CRC SOCIALFACTOR

SocialFactor	
Atributos	
socialDifference	Offer
familyEnvironment	Matter
socioeconomicContext	Career
demographicVariable	Module
	Student

En la TABLA LXVI se muestra la tarjeta InstitutionalFactor, donde se indica todos los atributos como desiredCareer, complexityCareer, institutionalStatus, etc. y la relación con otras clases como son Offer, Matter, Career, Module y Student.

TABLA LXVI. TARJETA CRC INSTITUTIONALFACTOR

InstitutionalFactor	
Atributos	
desiredCareer	Offer
complexityCareer	Matter
institutionalStatus	Career
supportService	Module
studentLife	Student
relationshipStudentTeacher	

3.1.6 Diseño de interfaces del Sistema

A continuación se presenta las interfaces candidatas para el sistema, procurando una eficiente navegabilidad y brindando facilidad al usuario. Primero se presentan las interfaces para el usuario Docente.

Ingreso al sistema: Se refiere a la página inicial donde el usuario ingresa los datos iniciales para conseguir acceso al sistema. En la Figura 26 se muestra la interfaz de ingreso al sistema

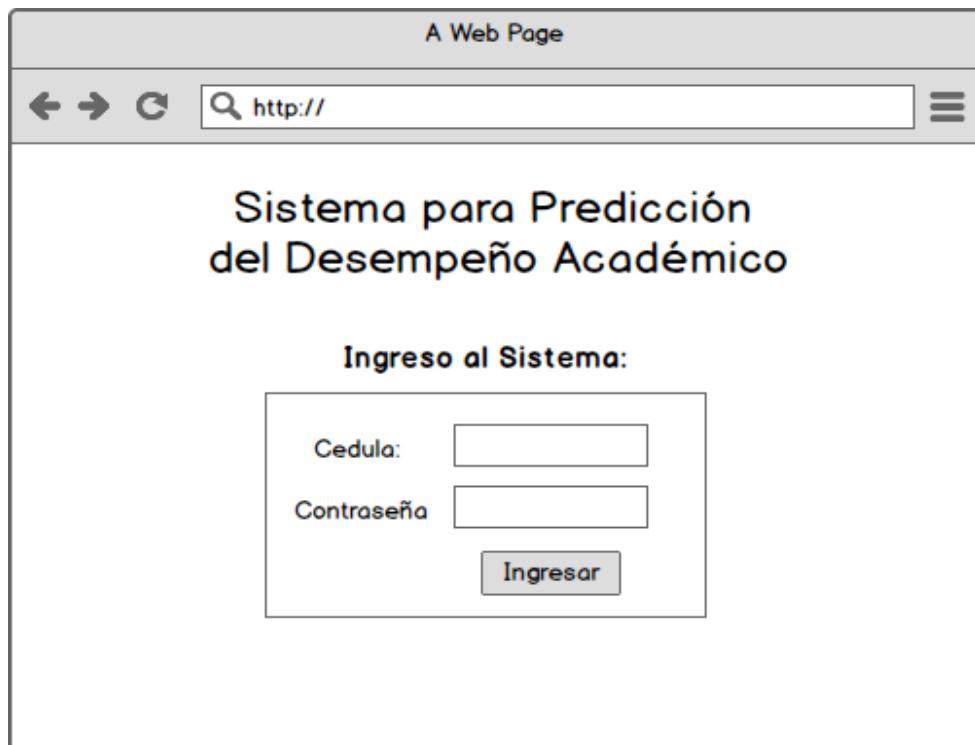


Figura 26. Interfaz Ingreso al Sistema

Agregar nuevas materias: En esta página se realiza la selección las materias asignadas al docente a fin de realizar el proceso de calificación. Se empieza desde la selección de la oferta académica hasta culminar con la materia buscada, todo esto ocurre automáticamente ya que los datos son obtenidos de los Web Services del SGA.

A Web Page

Ing. Jose Rivas [Salir](#)

Seleccione:

Periodo: 2013-2014

Oferta: Marzo-Julio 2014

Modulo: Decimo Módulo

Paralelo: B

Materia: Inteligencia Artificial

[Cancelar](#) [Crear](#)

Figura 27. Interfaz agregar nuevas materias

Administrar materias y calificaciones: En esta pantalla y luego de que se han seleccionado las distintas materias a calificar, se presenta una lista de las mismas con sus distintas opciones que son la de calificar, presentar promedios y realizar configuraciones. Para facilidad del usuario se implementó un filtro en la parte izquierda a fin de presentar únicamente la información necesaria en la tabla de materias.

A Web Page

Ing. Jose Rivas [Salir](#)

[Añadir Nueva Materia](#)

Seleccione:

Oferta: Marzo-Julio 2014

Carrera: Ingeniería en Sistemas

Modulo: Decimo Módulo

Paralelo: B

[Recuperar datos](#)

Mis Materias

Mo	Par	Materia	Calificar	Promedios	Configuraci
10	B	Inteligencia Artificial	Calificar	Promedios	Configurar
10	B	Anteproyectos de Tesi	Calificar	Promedios	Configurar

Figura 28. Interfaz administrar materias y calificaciones

Configurar cantidad y porcentaje de parámetros: En esta página se realiza la configuración relacionada con la cantidad y peso de los distintos parámetros de evaluación (exámenes, lecciones, tareas, etc.). Los pesos están relacionados con la nota final es decir exámenes puede tener un 70% de peso con relación a la nota final. En la Figura 29 se presenta la interfaz *Configurar cantidad y porcentaje de parámetros*.

The screenshot shows a web browser window with the title 'A Web Page'. The address bar contains 'http://'. The top right corner shows 'Ing. Jose Rivas' and a 'Salir' link. Below the header, the page title is 'Configuraciones' with a 'Atras' button. The main content area contains six rows of configuration fields:

Cantidad de exámenes	<input type="text" value="7"/>	peso	<input type="text" value="70"/>	%
Cantidad de lecciones	<input type="text" value="3"/>	peso	<input type="text" value="6"/>	%
Cantidad de trabajos autónomos	<input type="text" value="3"/>	peso	<input type="text" value="6"/>	%
Cantidad de tareas	<input type="text" value="3"/>	peso	<input type="text" value="6"/>	%
Cantidad de participaciones	<input type="text" value="3"/>	peso	<input type="text" value="6"/>	%
Cantidad de talleres	<input type="text" value="3"/>	peso	<input type="text" value="6"/>	%

Total: 100%

Figura 29. Interfaz configurar cantidad y porcentaje de parámetros

Añadir calificaciones: En esta pantalla se presentan los componentes necesarios para que el docente pueda calificar a sus estudiantes de una manera individualizada con los debidos controles y validaciones necesarios para un óptimo funcionamiento del sistema. En la Figura 30 se presenta la interfaz *Añadir calificaciones*.

A Web Page

http:// Ing. Jose Rivas Salir

Lección N° 1

Fecha: 20/05 /2014

Tema: Nuevo tema 1

Calificaciones		
Estudiante	Calificación	Apuntes
Juan Perez	7.6	
Rosa Gonzalez	8.3	
Ana Lima	7.4	Apunte sobre este estudiante
Juan Cuenca	8.2	
Pedro Loja	9.5	

Figura 30. Interfaz añadir calificaciones

Editar calificaciones: En esta página se realiza la edición de las calificaciones que previamente fueron ingresadas. En la Figura 31 se presenta la interfaz *Editar calificaciones*.

A Web Page

http:// Ing. Jose Rivas Salir

Editando Lección N° 1

Nueva fecha: 20/05 /2014

Nuevo tema: Nuevo tema 1

Calificaciones		
Estudiante	Calificación	Apuntes
Juan Perez	7.8	Nuevo apunte
Rosa Gonzalez	8.9	
Ana Lima	7.9	Apunte sobre este estudiante
Juan Cuenca	8.2	
Pedro Loja	9.5	

Figura 31. Interfaz editar calificaciones

Detallar calificaciones: En esta página se presenta la información sobre las calificaciones de una manera más detallada facilitando el entendimiento y usabilidad al usuario. En la Figura 32 se presenta la interfaz *Detallar calificaciones*.

The screenshot shows a web browser window titled "A Web Page". At the top, there are navigation buttons (back, forward, search), a URL bar with "http://", and user information "Ing. Jose Rivas" and "Salir". Below the header, there is a form field labeled "Seleccione Lección N°" with a dropdown menu showing the value "1". Underneath, there are two text fields: "Nueva fecha:" with the value "20/05/2014" and "Nuevo tema:" with the value "Referencia sobre esta lección". A section titled "Calificaciones" contains a table with the following data:

Estudiante	Calificación	Apuntes
Juan Perez	7.8	Nuevo apunte
Rosa Gonzalez	8.9	
Ana Lima	7.9	Apunte sobre este estudiante
Juan Cuenca	8.2	
Pedro Loja	9.5	

At the bottom right of the table area is a "Atras" button.

Figura 32. Interfaz detallar calificaciones

Promediar calificaciones parciales: En esta página se presenta los promedios relacionados con cada parámetro de calificación (exámenes, lecciones, etc.) de forma parcial con respecto a la nota final del módulo. En la Figura 33 se presenta la interfaz *Promediar calificaciones parciales*.

The screenshot shows a web browser window titled "A Web Page". At the top, there are navigation buttons (back, forward, search), a URL bar with "http://", and user information "Ing. Jose Rivas" and "Salir". Below the header, there is a section titled "Promedio Lecciones" and a table titled "Calificaciones". The table displays the following data:

Estudiante	Nota1	Nota2	Nota3	PROMEDIO
Juan Perez	7.8	5.6	6.6	6.67
Rosa Gonzalez	8.9	7.7	8.8	8.47
Ana Lima	7.9	6.6	7.29	7.26
Juan Cuenca	8.2	5.7	6.96	6.95
Pedro Loja	9.5	9.7	9.8	9.67

At the bottom right of the table area is a "Atras" button.

Figura 33. Interfaz promediar calificaciones parciales

Promediar calificaciones finales: En esta página se presenta los promedios finales de la materia tomando como referencia las configuraciones de porcentajes realizadas por el docente (ver Figura 29). En la Figura 34 se presenta la interfaz *Promediar calificaciones finales*.

A Web Page

Ing. Jose Rivas [Salir](#)

Atras

PROMEDIOS GENERALES

Calificaciones

Estudiante	Examenes (70%)	Lecciones(6%)	Tareas(6%)	Trabajos(6%)	Participaciones(6%)	Talleres(6%)	PROMEDIO FINAL
Juan Perez	6.4	0.6	0.55	0.34	0.6	0.4	8.89
Rosa Gonzalez	6.0	0.4	0.4	0.37	0.66	0.3	8.13
Ana Lima	5.4	0.2	0.25	0.33	0.1	0.4	6.68
Juan Cuenca	6.8	0.34	0.5	0.4	0.56	0.2	8.80
Pedro Loja	4.8	0.5	0.5	0.5	0.6	0.4	7.3

Figura 34. Interfaz promediar calificaciones finales

A continuación las interfaces del usuario Estudiante. Se obviara la interfaz de ingreso al sistema por ser igual a la del docente.

Seleccionar parámetros iniciales: En esta página se presenta los parámetros iniciales que debe seleccionar el estudiante para cargar la información necesaria. En la Figura 35 se presenta la interfaz *Seleccionar parámetros iniciales*.

A Web Page

← → ⌂ http:// ⌂ Estudiante Juan Perez [Salir](#)

Oferta:

Carrera:

Módulo:

Paralelo:

Figura 35. Interfaz seleccionar parámetros iniciales

Presentar gráficas de desempeño académico: En esta página se presenta las gráficas de desempeño junto con las calificaciones predictivas generadas el agente inteligente. En la Figura 36 se presenta la interfaz *Presentar gráficas de desempeño académico*.

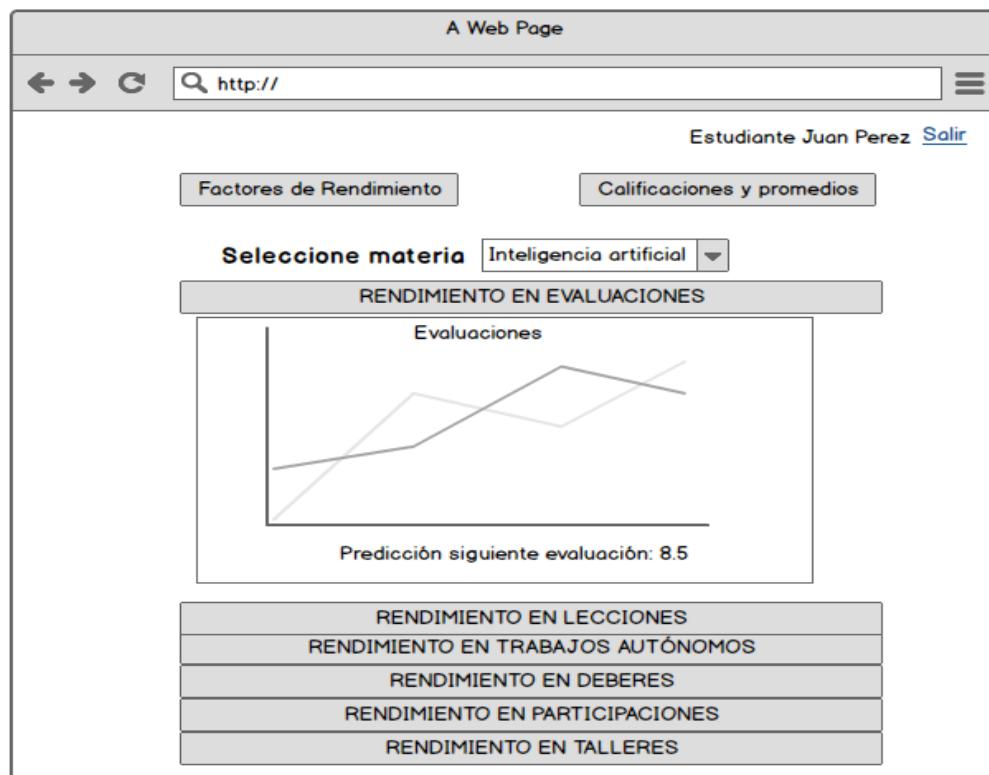


Figura 36. Presentar gráficas de desempeño académico

Presentar registró detallado de calificaciones y promedios: En esta página se presenta las toda la información concerniente a calificaciones obtenidas por el estudiante y sus promedios. En la Figura 37 se presenta la interfaz *Presentar registró detallado de calificaciones y promedios*.

The screenshot shows a web browser window with the title 'A Web Page'. The address bar displays 'http://'. The user is identified as 'Estudiante Juan Perez' with a 'Salir' link. The main content area is titled 'Inteligencia Artificial'.

Calificaciones:

Exámenes	Lecciones	Trabajos autónomos	Deberes	Participaciones	Talleres
Fecha	Referencia Generc	Descripción	Calificación (sobre 10)	Anotación	
10-03-2014	Referencia 1	Lección 1	8.5	anotación para est	
16-03-2014		Lección 2	8.7		
23-03-2014		Lección 3	9.2		
05-04-2014		Lección 4	4.8		

Promedio parcial de lecciones: 7.8 / 10

Promedios finales:

Evaluaciones	5.3	70%
Lecciones	0.5	6%
Trabajos autónomos	0.4	6%
Deberes	0.2	6%
Participaciones	0.6	6%
Talleres	0.3	6%
TOTAL:	7.30/10.00	

Figura 37. Presentar registró detallado de calificaciones y promedios

Presentar factores que influyen en el desempeño: En esta página se presenta un formulario con distintos factores que influyen en el desempeño académico para ser ajustados por el estudiante según su realidad. En la Figura 38 se presenta la interfaz *Presentar factores que influyen en el desempeño*.

A Web Page

← → ⌛ 🔍 http:// ⌂

Estudiante Juan Perez [Salir](#)

Factores personales	Factores sociales	Factores Institucionales
---------------------	-------------------	--------------------------

Factor 1

Factor 2

Factor 3

Factor 4

Factor 5

Bajo Medio Alto

Si No

Guardar cambios

Figura 38. Presentar factores que influyen en el desempeño

Una vez terminado el prototipado de pantallas se tiene una visión más clara de lo que se va a desarrollar y la mejor manera de hacerlo lo siguiente es empezar a codificar el sistema.

3.2 Metodología GAIA: Diseño del Agente

Esta fase comprende el diseño de interacción y la arquitectura del agente inteligente.

3.2.1 Diseño de interacción

Este modelo tiene como finalidad definir la funcionalidad entre el agente y su conexión entre Creencias, Deseos, Intenciones y las posibles fuentes de obtención de información a más de la interacción con el usuario.

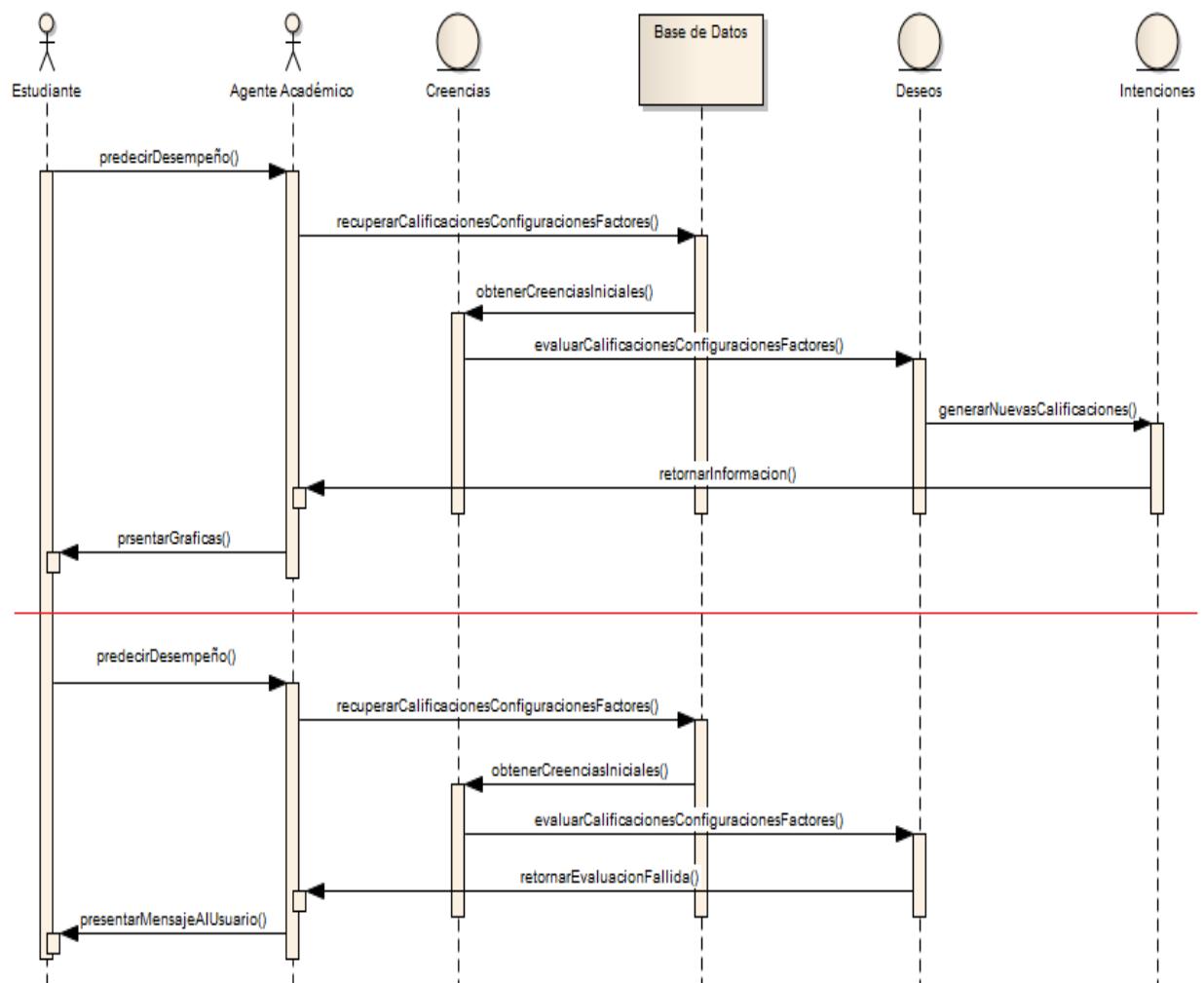


Figura 39. Diagrama de interacción del agente

3.2.2 Arquitectura del agente

La arquitectura muestra la ubicación del agente dentro del sistema, dicha ubicación está dentro de un contenedor de agentes principal que se encuentra ubicado en la nube.

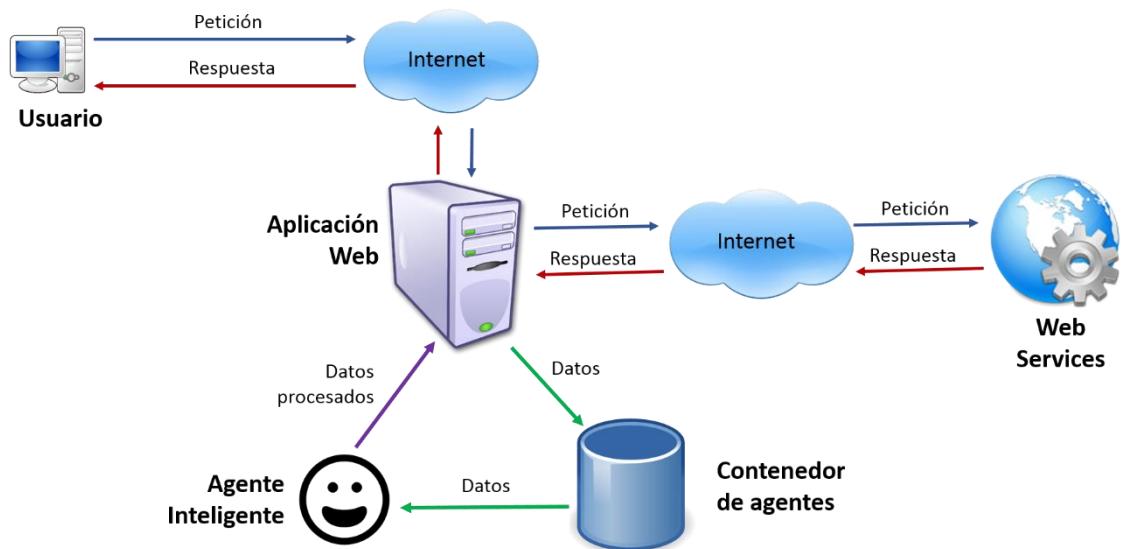


Figura 40. Arquitectura del agente inteligente dentro del sistema web

4 Fase III: Codificación

Una vez terminado el diseño del sistema y del agente, se procede a su codificación declarando como primer paso los estándares a utilizar.

4.1 Metodología XP: Codificación del sistema

Siguiendo con la metodología Programación Extrema (XP), se recomienda seguir un proceso de desarrollo adecuado definiendo estándares de codificación y pruebas de aceptación a fin de verificar el correcto funcionamiento, los resultados esperados, etc.

4.1.1 Estándares de codificación

A continuación se describen algunos estándares y patrones de diseño que se utilizarán para codificar la aplicación

4.1.1.1 Patrón de programación

Modelo-Vista-Controlador (MVC): Se utiliza este patrón de diseño con la finalidad de separar la lógica de negocio de las vistas y el modelo.

4.1.1.2 Estándares de programación web

Los estándares descritos a continuación constituyen buenas prácticas para el desarrollo de aplicaciones basadas en JAVA las mismas que se basan en estándares recomendados que buscan facilitar la comprensión del código para a su vez lograr la propiedad colectiva del mismo.

- Todo paquete debe iniciar con minúscula.
- Toda clase debe iniciar con mayúscula.
- Todo método debe iniciar con minúscula.
- Toda variable debe iniciar con minúscula.

- Sea clase, método o variable. Si consta de dos o más palabras. Todas van sin espacios y de la segunda en adelante con mayúscula inicial.
- Al crear JavaDoc: Antes de todo método, variable y clase debe ir una explicación del mismo empleando el siguiente formato:
 - `/** comentario */`
 - `/**
 * comentario
 */`
- Se hará excepción de aquellos en que su comprensión resulte intuitiva a juicio del desarrollador, acogiéndose a la simplicidad según lo plantea XP.
- Todo el código debe estar correctamente identado:
 - La llave se abre en la misma fila donde se especifica el nombre del método o constructor luego de especificar los parámetros.
 - Cuando es una clase, luego de las posibles implementaciones.
 - En caso de ser una estructura de control (for, if, etc.) se seguirá la misma lógica.
 - No se escribe nada más en esa línea
 - Se cierra en la columna donde se encuentra la primera letra del nombre que identifica al método, constructor o estructura de control
 - Tampoco se escribe nada en la línea donde se cerró la llave
 - EXCEPCIONES
 - Para el control de excepciones en el código (try-catch)
- Solo se declaran variables al inicio de una clase, método o ciclo.
- El nombre de las tablas es en singular y minúscula.
- Los nombres de las clases entidad tienen el mismo nombre que la tabla en base de datos
- Las columnas de una tabla se convierten en atributos privados en la respectiva clase manteniendo el tipo de dato.
- Todos los atributos de una clase están encapsulados. Son privados y su acceso se da por medio de los métodos `setAtributo()` y/o `getAtributo()`.

4.1.1.3 Tecnologías a utilizar

HTML (HyperText Markup Language): Estandar utilizado para la creación de páginas web. Está compuesto por una serie de etiquetas que el navegador interpreta y da forma en la pantalla. HTML dispone de etiquetas para imágenes, hipervínculos que nos permiten dirigirnos a otras páginas, saltos de línea, listas, tablas, etc.

CSS: Es un lenguaje de estilo que define la presentación de los documentos HTML. CSS se encarga de cuestiones relativas a fuentes, colores, márgenes, líneas, altura, anchura, imágenes de fondo, etc. [12].

JavaScript: Es el lenguaje de programación encargado de administrarle funcionalidad a las páginas web. Al ser un lenguaje de scripting no necesita compilación sino que es ejecutado directamente por el navegador al descargar la página y al ser llamado por un evento [12].

Java EE: La plataforma Java Enterprise Edition (Java EE) son un conjunto de especificaciones que facilitan el desarrollo y despliegue de aplicaciones empresariales multi-capa. Java EE ofrece un conjunto de especificaciones y técnicas que proporcionan soluciones completas, seguras, estables y escalables para el desarrollo, despliegue y gestión de aplicaciones de múltiples niveles de funcionalidad basadas en servidores. Se reduce el costo y complejidad de desarrollo, lo cual resulta en servicios que se pueden desplegar y extender fácilmente [44].

JSF (Java Server Faces): Es un framework para el desarrollo de aplicaciones java para la web. Se enfoca en la simplificación de interfaces de usuario en aplicaciones Java EE. Sigue el patrón de diseño MVC [45].

Primefaces: Es una librería de componentes para el framework JSF que cuenta con un conjunto de componentes mejorados y enriquecidos a fin de facilitar la creación de aplicaciones web [46].

Spring: Es un framework para el desarrollo de aplicaciones web y contenedor de inversión de control. Se integra fácilmente con otro tipo de tecnologías como JSF e Hibernate [47].

Hibernate: Es una herramienta de Mapeo objeto-relacional (ORM) disponible para la plataforma Java. Su función principal es facilitar el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación, mediante la utilización de archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones [48].

Tomcat: Es un contenedor de servlets para la plataforma Java. Implementa las especificaciones de los servlets y de JavaServer Pages (JSP) desarrolladas por Oracle [49].

4.1.2 Conexión a los Web Services del SGA

Un punto fundamental constituía la conexión de la aplicación con los Web Services proporcionados por el Sistema de Gestión Académica (SGA).

Para acceder a los WS se lo realiza mediante la siguiente dirección: <http://ws.unl.edu.ec/> y se desplegará la página principal como se aprecia en la Figura 41.



Figura 41. Página inicial para acceso a los Web Services del SGA

La conexión se la realizó mediante la utilización de la librería JAX WS [50] la cual mediante la utilización de la herramienta **wsimport** permite la generación automática de código en java partiendo de las especificaciones del documento WSDL. Una vez realizado el proceso y para tener una conexión constante se utilizó la porción de código descrita a continuación.

TABLA LXVII. CÓDIGO NECESARIO PARA CONEXIÓN A LOS WEB SERVICES

```
/**sgaValidacion es instancia de la clase
 * SGAWebServicesValidacionPortType generada por JAX-WS*/
BindingProvider bp = (BindingProvider) sgaValidacion;
Map<String, Object> map = bp.getRequestContext();
map.put(bp.USERNAME_PROPERTY, "usuario");
map.put(bp.PASSWORD_PROPERTY, "contrasenia");
```

En el código se muestra el acceso a los métodos para validación tanto del docente como del estudiante.

4.1.3 Generación de calificaciones y gráficas predictivas

Para la generación de calificaciones predictivas y una vez teniendo las creencias para el agente inteligente, se procede primero a verificar las configuraciones del docente para evaluar si no existen inconvenientes para llevar a cabo el proceso de predicción. El siguiente paso es promediar las calificaciones anteriores (dato histórico) con el fin de obtener el 50% de la nueva calificación. Para completar el otro 50% se asigna valores (pesos) a cada factor de rendimiento configurado por el estudiante como se aprecia en la siguiente tabla considerando que el valor más bajo para cada factor es cero. A continuación se describen los factores personales, sociales e institucionales.

TABLA LXVIII. PESOS DE FACTORES PERSONALES

FACTORES PERSONALES	
Campo	Peso
Competencia cognitiva	0.3
Motivación	0.2
Condiciones cognoscitivas	0.3

Autoconcepto académico	0.3
Autoeficacia percibida	0.2
Bienestar psicológico	0.2
Satisfacción en los estudios	0.3
Asistencia a clases	0.1
Inteligencia	0.3
Aptitudes	0.2
Género	0.05
Edad	0.05
Estado civil	0.05
Cargas familiares	0.2
Formación previa a la Universidad	0.1
Calificación de ingreso	0.1

TABLA LXIX. PESOS DE FACTORES SOCIALES

FACTORES SOCIALES	
Campo	Peso
Diferencia social	0.05
Entorno familiar	0.2
Contexto socioeconómico	0.1
Variable demográfica	0.2

TABLA LXX. PESOS DE FACTORES INSTITUCIONALES

FACTORES INSTITUCIONALES	
Campo	Peso
Carrera deseada	0.3
Complejidad carrera	0.3
Estado de la institución	0.2
Servicios de apoyo institucionales	0.1
Ambiente estudiantil	0.3
Relación estudiante – profesor	0.3

El Método principal encargado de generar las nuevas calificaciones se aprecia a continuación tomando como ejemplo la generación de calificaciones para evaluaciones.

TABLA LXXI. MÉTODO PARA GENERACIÓN DE CALIFICACIONES PREDICTIVAS

```

public static final double MINIMUM_QUALIFICATION_POSSIBLE = 7.0;
public static final double REAL_QUALIFICATION_VALUE = 5.0;
public static final double MAX_VALUE = 0.3;
public static final double MED_VALUE = 0.2;
public static final double MIN_VALUE = 0.1;
public static final double MIN_MIN_VALUE = 0.05;

public static final double VALUE_10 = 10.0;
public static final double VALUE_2 = 2.0;
public static final double VALUE_3 = 3.0;
public static final double VALUE_100 = 100.0;

public List<Object> generatePrediction(List<?> listParam,
                                         double percentage,
                                         int totalNumberParams,
                                         int daysToCompleteOffer,
                                         int totalDaysOffer,
                                         Date dateLastParam,
                                         Date dateEndOffer,
                                         PersonalFactor personalFactor,
                                         SocialFactor socialFactor,
                                         InstitutionalFactor institutionalFactor) {

    List<Object> listFinal = new ArrayList<Object>();
    double aux=0.0;

    for(int i =0;i<listParam.size();i++){
        EvaluationUtil data = (EvaluationUtil) listParam.get(i);
        aux += data.getQualificationEval();
        listFinal.addAll((List<EvaluationUtil>) listParam);
    }

    /*Promedio*/
    double generalAverage = aux / listParam.size();

    /*Minimo necesario 70% (7.0)*/
    double min_average = percentage *
                         MINIMUM_QUALIFICATION_POSSIBLE;

    /*Cantidad de parametros que faltan por calificar*/
    int missingParams = (totalNumberParams - listParam.size());

    /*Método para verificar si las calificaciones son
     * muy bajas */
    Boolean isNotLow = chekingDataQualification(listParam,
                                                 min_average, missingParams,percentage);

    Operations operation = new OperationsImpl();

    int listSize = listParam.size();
}

```

```

/*Almacenara las calificaciones actuales más una
* que será la predicción*/
double [] newQualification = new double[listSize+1];
boolean isCompleteAllEvaluations=false;

if(missingParams != 0){
    double [] finalValues = new double[26];

    /*factores personales*/
    finalValues[0] = (personalFactor.getcognitiveCompetence()
                      * MAX_VALUE) / VALUE_10;
    finalValues[1] = (personalFactor.getMotivation()
                      * MED_VALUE) / VALUE_2;
    finalValues[2] = (personalFactor.getcognitiveConditions()
                      * MAX_VALUE) / VALUE_10;
    finalValues[3] = (personalFactor.getAcademicSelfConcept()
                      * MAX_VALUE) / VALUE_3;
    finalValues[4]=(personalFactor.getPerceivedSelfEfficacy()
                      * MED_VALUE) / VALUE_2;
    finalValues[5]=(personalFactor.getPsychologicalWelfare()
                      * MED_VALUE) / VALUE_3;
    finalValues[6] = (personalFactor.getSatisfactionStudies()
                      * MAX_VALUE) / VALUE_2;
    finalValues[7] = (personalFactor.getClassesAssistance()
                      * MIN_VALUE) / VALUE_100;
    finalValues[8] = (personalFactor.getIntelligence()
                      * MAX_VALUE) / VALUE_10;
    finalValues[9] = (personalFactor.getSkills()
                      * MED_VALUE) / VALUE_2;

    if(personalFactor.getSex() == 0){
        finalValues[10] = MIN_MIN_VALUE;
    }else{
        finalValues[10] = 0.0;
    }

    if(personalFactor.getAge() >= 0 &&
       personalFactor.getAge() <= 25 ){
        finalValues[11] = MIN_MIN_VALUE;
    }else{
        finalValues[11] = 0.0;
    }

    if(personalFactor.getMaritalStatus() == 0){
        finalValues[12] = MIN_MIN_VALUE;
    }else{
        finalValues[12] = 0.0;
    }

    if(personalFactor.getDependents() == 0){
        finalValues[13] = MED_VALUE;
    }else{
        if(personalFactor.getDependents() >= 1 &&
           personalFactor.getDependents() <= 3){
            finalValues[13] = MIN_VALUE;
        }else{
            finalValues[13] = 0.0;
        }
    }
}

```

```

        }

finalValues[14]=(personalFactor.getAcademicTrainingSchool()
                  * MIN_VALUE) / VALUE_10;
finalValues[15] = (personalFactor.getQualificationAccess()
                  * MIN_VALUE) / VALUE_10;

/*factores sociales*/
if(!socialFactor.isSocialDifference()) {
    finalValues[16] = MIN_MIN_VALUE;
} else{
    finalValues[16] = 0.0;
}

finalValues[17] = (socialFactor.getFamilyEnvironment()
                  * MED_VALUE) / VALUE_3;
finalValues[18] = (socialFactor.getSocioeconomicContext()
                  * MIN_VALUE) / VALUE_2;
finalValues[19] = (socialFactor.getDemographicVariable()
                  * MED_VALUE) / VALUE_2;

/*factores institucionales*/
if(institutionalFactor.isDesiredCareer()) {
    finalValues[20] = MAX_VALUE;
} else{
    finalValues[20] = 0.0;
}
double varAux = (institutionalFactor.getComplexityCareer()
                  * MAX_VALUE) / VALUE_10;
finalValues[21] = MAX_VALUE - varAux;
finalValues[22] = (institutionalFactor.getInstitutionalStatus()
                  * MED_VALUE) / VALUE_10;


if(institutionalFactor.isSupportService()) {
    finalValues[23] = MIN_VALUE;
} else{
    finalValues[23] = 0.0;
}

finalValues[24] = (institutionalFactor.getStudentLife()
                  * MAX_VALUE) / VALUE_3;
finalValues[25]=(institutionalFactor
                .getRelationshipStudentTeacher() * MAX_VALUE) / VALUE_3;

/*almacena calificaciones reales*/
for(int i=0;i<listSize;i++){
    EvaluationUtil data = (EvaluationUtil) listParam.get(i);
    newQualification[i] = operation.roundNumber(
        data.getQualificationEval(),2);
}

double cumSum = 0.0;
for(int i =0;i<finalValues.length;i++){
    cumSum += finalValues[i];
}

```

```

/*Se adiciona el promedio
* de calificaciones reales*/
double valAux =(generalAverage
                * REAL_QUALIFICATION_VALUE) / VALUE_10;
double finalQualification = valAux + cumSum;
newQualification[listSize] = finalQualification;

/*Si no quedan días antes de terminar
* de evaluar todos los parámetros establecidos*/
if(daysToCompleteOffer <missingParams ){
    for(int i=0;i<newQualification.length;i++){
        EvaluationUtil newParam = new EvaluationUtil();
        if(i==(newQualification.length-1)){
            newParam.setDateEvaluation(dateEndOffer);
        }else{
            EvaluationUtil data = (EvaluationUtil)
                listParam.get(i);
            newParam.setDateEvaluation(
                data.getDateEvaluation());
        }
        newParam.setQualificationEval(newQualification[i]);
        listFinal.add(newParam);
    }else{
        /*días a añadir*/
        int daysToAdd = daysToCompleteOffer / missingParams;
        Date dateNewParam = dateLastParam;
        for(int i=0;i<newQualification.length;i++){
            EvaluationUtil newParam = new EvaluationUtil();
            if(i==(newQualification.length-1)){
                dateNewParam = operation.addDate(
                    dateNewParam, daysToAdd);
                newParam.setDateEvaluation(dateNewParam);
            }else{
                EvaluationUtil data = (EvaluationUtil)
                    listParam.get(i);
                newParam.setDateEvaluation(data.getDateEvaluation());
            }
            newParam.setQualificationEval(newQualification[i]);
            listFinal.add(newParam);
        }else{
            isCompleteAllEvaluations = true;
        }
        List<Object> listObject = new ArrayList<Object>();
        listObject.add(listFinal);
        listObject.add(isNotLow);
        listObject.add(isCompleteAllEvaluations);

        return listObject;
    }
}

```

Este método constituye las intenciones del agente inteligente y están encaminadas a informar al estudiante sobre su desempeño actual y futuro a fin de conseguir que apruebe el módulo sin problemas y evitar posibles descuidos o sorpresas al final del mismo.

De igual manera el método utilizado para presentar la gráfica al estudiante se aprecia en el siguiente cuadro. Para muestra se tomó el código perteneciente a evaluaciones

TABLA LXXII. CÓDIGO PARA GENERACION DE GRÁFICAS DE EVALUACIONES

```

String msg1 = "Su promedio en ";
String msg2 = " no alcanza el mínimo necesario (70%) "
+ "a pesar de obtener 10 (100%) en "
+ "los aportes restantes, "
+ "por favor comuníquese con el docente responsable.';

String msgComp=" no alcanzó el mínimo necesario (70%), "
+ "por favor comuníquese con el docente responsable.';

String msg3= "El docente especificó únicamente ";
String msg4 = " a calificar por lo que no se "
+ "pueden generar más notas predictivas.';

String msg7 = "Hasta la fecha no se ha "
+ "ingresado ninguna calificación de ";
String msg8 = " al sistema, una vez que se ingresen "
+ "datos las gráficas serán "
+ "presentadas automáticamente.";
String label1="Calificación real";
String label2="Predicción";
SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");

if(!evaluations.isEmpty()){
    int numberEvaluations = parametersNew.getListConfigurations()
        .get(0).getNumberDefault();

    /*grafica de calificaciones reales*/
    LineChartSeries evaluationChart = new LineChartSeries();
    evaluationChart.setLabel(label1);
    for(EvaluationUtil data:evaluations){
        String dateEval = format.format(
            data.getDateEvaluation());
        evaluationChart.set(
            dateEval, data.getQualificationEval());
    }
    if(numberEvaluations > 1){
        LineChartSeries evaluationPrediction =
            new LineChartSeries();
        evaluationPrediction.setLabel(label2);
        for(EvaluationUtil data:parameters.getEvaluations()){
            String dateEval = format.format(
                data.getDateEvaluation());
            evaluationPrediction.set(
                dateEval, data.getQualificationEval());
        }
    }
}

```

```

        }

modelEvaluation.addSeries(evaluationPrediction);
modelEvaluation.setSeriesColors(lineColorPrediction+",006400");
if(!parameters.getLowEval()) {
    if(evaluations.size()==parameters.getEvaluations().size()) {
        messageEvaluationFATAL = msg1+ "EVALUACIONES"+msgComp;
    }else{
        messageEvaluationFATAL = msg1+ "EVALUACIONES"+msg2;
    }
}
if(evaluations.size() == parameters.getEvaluations().size()){
    messageEvaluationComplete = msgComplete+"EVALUACIONES"
}

}else{
    messageEvaluationOne = msg3+"una evaluación"+msg4;
}
modelEvaluation.addSeries(evaluationChart);
modelEvaluation.setTitle("EVALUACIONES");
this.generateYAxisData(modelEvaluation,matterName);
this.generateXAxisData(modelEvaluation,idOffer,format);
}else{
    messageEvaluationEmpty = msg7 + "evaluación" + msg8;
}

```

Su finalidad es brindar al estudiante una visión actual y futura de su desempeño académico tomando en cuenta ciertos parámetros comúnmente utilizados por los docentes para calificar al estudiante.

TABLA LXXIII. PARÁMETROS Y PESOS POR DEFECTO

Parámetro	Peso a la nota final(Por defecto)
Evaluaciones	70%
Lecciones	6%
Trabajos autónomos	6%
Deberes o tareas	6%
Participaciones en clase	6%
Talleres o laboratorios	6%
TOTAL	100%

En la TABLA LXXIII se presentan los porcentajes por defecto que poseerán cada parámetro con relación al promedio final. Cabe resaltar que el docente tiene la posibilidad de configurar estos valores a su conveniencia o la planificación prevista.

INTELIGENCIA ARTIFICIAL Módulo 10 Paralelo B Pregrado Marzo-Julio 2014					
CANTIDAD DE PARÁMETROS A EVALUAR CON SUS PORCENTAJES					
<small>* Puede especificar 0 en caso de que no se vaya a tomar en cuenta un parámetro * El total de porcentajes debe ser siempre 100%</small>					
<input checked="" type="checkbox"/> Cantidad de evaluaciones	3	con un peso de	0.7	(70.00 %)	a la nota final
<input checked="" type="checkbox"/> Cantidad de lecciones	3	con un peso de	0.06	(6.00 %)	a la nota final
<input checked="" type="checkbox"/> Cantidad de trabajos autónomos	3	con un peso de	0.06	(6.00 %)	a la nota final
<input checked="" type="checkbox"/> Cantidad de deberes - tareas	3	con un peso de	0.06	(6.00 %)	a la nota final
<input checked="" type="checkbox"/> Cantidad de participaciones en clase	3	con un peso de	0.06	(6.00 %)	a la nota final
<input checked="" type="checkbox"/> Cantidad de talleres-laboratorios	3	con un peso de	0.06	(6.00 %)	a la nota final
<input type="button" value="Guardar cambios"/>				TOTAL: 100.00 %	

Figura 42. Área de configuraciones

En la Figura 42 se presentan los parámetros con su cantidad y pesos por defecto los mismos que pueden ser configurados por el docente.

Se aprecia además la posibilidad de especificar un número aproximado de evaluaciones, lecciones etc. en caso de que cierto parámetro no sea considerado por ejemplo laboratorios, se establecerá un valor de cero ajustando los pesos a los nuevos valores.

Continuando con el ejemplo se realizó una modificación en la cantidad de parámetros a calificar quedando de la siguiente manera:

INTELIGENCIA ARTIFICIAL || Módulo 10 Paralelo B || Pregrado Marzo-Julio 2014

CANTIDAD DE PARÁMETROS A EVALUAR CON SUS PORCENTAJES

* Puede especificar 0 en caso de que no se vaya a tomar en cuenta un parámetro
 * El total de porcentajes debe ser siempre 100%

✓ Cantidad de evaluaciones	7 ▲ ▼	con un peso de	0.7	(70.00 %)	a la nota final
✓ Cantidad de lecciones	5 ▲ ▼	con un peso de	0.06	(6.00 %)	a la nota final
✓ Cantidad de trabajos autónomos	6 ▲ ▼	con un peso de	0.06	(6.00 %)	a la nota final
✓ Cantidad de deberes - tareas	10 ▲ ▼	con un peso de	0.06	(6.00 %)	a la nota final
✓ Cantidad de participaciones en clase	4 ▲ ▼	con un peso de	0.06	(6.00 %)	a la nota final
✓ Cantidad de talleres-laboratorios	6 ▲ ▼	con un peso de	0.06	(6.00 %)	a la nota final

TOTAL: 100.00 %

Figura 43. Configuraciones modificadas

Los pesos quedan con los valores por defecto. Una vez realizadas las configuraciones respectivas procede a ingresar ciertas calificaciones al sistema.

Ahora pasando a la parte del estudiante, se realizan modificaciones en los formularios que contienen información de factores que influyen en el desempeño académico.

La configuración establecida por el estudiante se aprecia en las siguientes tablas:

TABLA LXXIV. AJUSTE DE FACTORES PERSONALES

FACTORES PERSONALES		
Campo	Opción elegida	Valor final
Competencia cognitiva	8	0.24
Motivación	Medio	0.1
Condiciones cognoscitivas	8	0.24
Autoconcepto académico	Bueno	0.199
Autoeficacia percibida	Media	0.1
Bienestar psicológico	Bueno	0.133
Satisfacción en los estudios	Satisfecho	0.3
Asistencia a clases	86 %	0.086
Inteligencia	7	0.21
Aptitudes	Medias	0.1

Género	Hombre	0
Edad	23	0.05
Estado civil	Casado	0
Cargas familiares	1	0.1
Formación previa a la Universidad	8	0.08
Calificación de ingreso	8	0.08

TABLA LXXV. AJUSTE DE FACTORES SOCIALES

FACTORES SOCIALES		
Campo	Opción elegida	Valor final
Diferencia social	No	0.05
Entorno familiar	Muy bueno	0.2
Contexto socioeconómico	Clase media	0.05
Variable demográfica	Sector urbano	0.2

TABLA LXXVI. AJUSTE DE FACTORES INSTITUCIONALES

FACTORES INSTITUCIONALES		
Campo	Opción elegida	Valor final
Carrera deseada	Sí	0.3
Complejidad carrera	7	0.089
Estado de la institución	7	0.14
Servicios de apoyo institucionales	Sí	0.1
Ambiente estudiantil	Muy bueno	0.3
Relación estudiante – profesor	Muy buena	0.3

El promedio de calificaciones es: 8.133 y transformado al 50% queda 4.06. La suma de estos valores da un total de 3.749 y sumando ambos valores obtenemos el aproximado de la calificación tomando en cuenta mayor número de decimales es **7.82**

En la Figura 44 se aprecia la gráfica obtenida para el parámetro de evaluaciones.

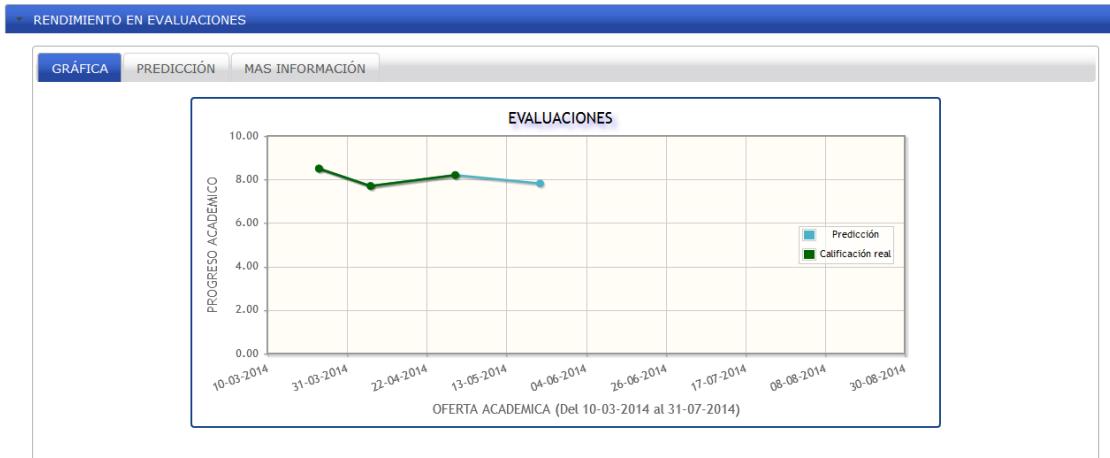


Figura 44. Gráfica de predicción para evaluaciones

Para mayor detalle en la pestaña de predicción se puede apreciar claramente la calificación generada.

El sistema predice que la siguiente calificación para la próxima EVALUACIÓN, podría aproximarse a **7.82** tomando como base sus calificaciones actuales y los factores que intervienen en el rendimiento académico.

Figura 45. Mensaje sobre calificación predictiva

Puede darse el caso de que las calificaciones obtenidas por el estudiante sean demasiado bajas y el tiempo restante o la cantidad de parámetros a evaluar sea insuficientes para alcanzar el mínimo necesario (70%), en este caso se presenta un mensaje al estudiante notificándolo de la situación .

Su promedio en EVALUACIONES no alcanza el mínimo necesario (70%) a pesar de obtener 10 (100%) en los aportes restantes, por favor comuníquese con el docente responsable.

Figura 46. Mensaje de notificación por calificaciones bajas

Cabe destacar que esto no significa específicamente que se perderá el módulo pero sí que se está con altas posibilidades de esto ocurra sabiendo que las evaluaciones ocupan el 70% de la nota final.

1.1.1. Automatización del proceso de registro de nóminas de estudiantes

Gracias a la facilidad que nos brindan los Web Services del SGA, el proceso de registro y administración de nóminas de los estudiantes pertenecientes a cada módulo en las distintas materias y carreras es automático.

Información Académica

* Seleccione los datos académicos necesarios hasta obtener la materia deseada.

Periodo Lectivo:	2013-2014
Oferta Académica:	Pregrado Marzo-Julio 2014
Carrera:	Ingeniería en Sistemas
Módulo:	Modulo 10 Paralelo B
Materia:	INTELIGENCIA ARTIFICIAL

Figura 47. Selección de información académica

Como se aprecia en la Figura 49 el docente puede en cualquier momento, y si no se ha realizado previamente, recuperar información relacionada con los estudiantes a los que imparte clases lo que evita el ingreso manual de la información perteneciente a los mismos. Para acceder basta con seleccionar la opción **Añadir Nuevo Registro**.

Seleccione su Módulo

Oferta Académica:
Pregrado Marzo-Julio 2014

Carrera:
Ingeniería en Sistemas

Modulo:

MIS MATERIAS	
Módulo - Paralelo	Materia
10 - B	INTELIGENCIA ARTIFICIAL

Figura 48. Añadir nuevo registro

En el caso de ser la primera vez que se ingresa al sistema, se redireccionará automáticamente a la página de la Figura 48.

4.1.4 Administración de calificaciones

El docente tiene la posibilidad de administrar las calificaciones pertenecientes a las distintas materias que imparte. Para acceder a dicha opción basta con seleccionar la opción **Calificar** luego de seleccionar la materia deseada. A continuación se presenta el sitio de administración de calificaciones como se aprecia en la figura 50.

La captura de pantalla muestra una interfaz web titulada "INTELIGENCIA ARTIFICIAL || Módulo 10 Paralelo B || Pregrado Marzo-Julio 2014". La barra superior incluye enlaces para "Seleccionar materia", "Configuraciones", "Promedios generales", "Ayuda" y el nombre del docente "DOCENTE HENRY PAZ" con un icono para cerrar sesión. La barra de menú principal contiene los ítems "EVALUACIONES", "LECCIONES", "TRABAJOS AUTÓNOMOS", "DEBERES - TAREAS", "PARTICIPACION EN CLASE" y "TALLERES - LABORATORIOS". Debajo de la barra de menú, hay tres botones: "Nueva evaluación" (con icono de estrella), "Editar o Eliminar" (con icono de lápiz) y "Parciales" (con icono de cuaderno). La tabla principal muestra los datos de 12 estudiantes, cada uno con su número de estudiante, nombre, y calificaciones para cuatro evaluaciones. Los datos son:

Nº	Estudiante	Evaluación N° 1	Evaluación N° 2	Evaluación N° 3	Evaluación N° 4
1	ANDREA ELIZABETH ARMIJOS CARTUCHE	0.00	0.00	0.00	0.00
2	FREDDY PATRICIO LOAYZA ABAD	0.00	0.00	0.00	0.00
3	JENNY LILIANA SARAGURO PACHECO	0.00	0.00	0.00	0.00
4	JOHN PATRICIO SOLANO CABRERA	0.00	0.00	0.00	0.00
5	YESENIA CECIBEL CHAMBA JIMÉNEZ	0.00	0.00	0.00	0.00
6	JHYMER ANTONIO MARTÍNEZ GRANDA	3.00	2.00	2.00	2.00
7	KARLA KATHERINE HERRERA PAREDES	0.00	0.00	0.00	0.00
8	JAIRO ISRAEL BANDA BERMEO	0.00	0.00	0.00	0.00
9	LAURO IBÁÑ JAPA ÁVILA	0.00	0.00	0.00	0.00
10	MARIO RICHAR LEÓN RAMÓN	0.00	0.00	0.00	0.00
11	JORGE ANIBAL MALES CHALÁN	0.00	0.00	0.00	0.00
12	ERIKA KATHERINE MASACHE MONTAÑO	0.00	0.00	0.00	0.00

Figura 49. Administración de calificaciones

Las operaciones administrativas son crear, editar y eliminar calificaciones de los estudiantes. Cabe resaltar que el proceso es para cada calificación perteneciente a cada parámetro (evaluaciones, lecciones, deberes, etc.).



Figura 50. Botones para administración de calificaciones

En la Figura 51 se aprecia los botones encargados de realizar las operaciones de creación, edición y eliminación de calificaciones.

4.1.4.1 Agregar nuevas calificaciones

El sistema permite agregar calificaciones a los estudiantes que reciben la materia impartida por el docente.

INTELIGENCIA ARTIFICIAL || Módulo 10 Paralelo B || Pregrado Marzo-Julio 2014

Evaluación N°	5		
Fecha: *	31-07-2014	Guarda	
- Agrege un tema o referencia -			
Referencia:			
Nº	Estudiante	Calificación	Anotación adicional
1	ANDREA ELIZABETH ARMIJOS CARTUCHE	0.0	¿alguna anotación adicional?
2	FREDDY PATRICIO LOAYZA ABAD	0.0	¿alguna anotación adicional?
3	JENNY LILIANA SARAGURO PACHECO	0.0	¿alguna anotación adicional?
4	JOHN PATRICIO SOLANO CABRERA	0.0	¿alguna anotación adicional?
5	YESENIA CECIBEL CHAMBA JIMÉNEZ	0.0	¿alguna anotación adicional?
6	JHYMER ANTONIO MARTÍNEZ GRANDA	0.0	¿alguna anotación adicional?
7	KARLA KATHERINE HERRERA PAREDES	0.0	¿alguna anotación adicional?

Guardar datos **Cancelar**

Figura 51. Agregación de nuevas calificaciones

Se facilita la calificación ya que se presenta la nómina completa de estudiantes.

4.1.4.2 Editar y eliminar calificaciones

Una vez que se agregó alguna calificación, es posible realizar modificaciones en caso de necesitarlo.

INTELIGENCIA ARTIFICIAL || Módulo 10 Paralelo B || Pregrado Marzo-Julio 2014

Evaluación N°	1 (24-03-2014)		
Se evaluó el: *	24-03-2014	Guarda	
- Agrege un tema o referencia -			
Temas tratados:			
Se está editando la Evaluación N° 1			
Nº	Estudiante	Calificación	Anotación adicional
1	ANDREA ELIZABETH ARMIJOS CARTUCHE	8.00	¿alguna anotación adicional?
2	FREDDY PATRICIO LOAYZA ABAD	7.00	¿alguna anotación adicional?
3	JENNY LILIANA SARAGURO PACHECO	6.00	¿alguna anotación adicional?
4	JOHN PATRICIO SOLANO CABRERA	8.00	¿alguna anotación adicional?
5	YESENIA CECIBEL CHAMBA JIMÉNEZ	7.50	¿alguna anotación adicional?
6	JHYMER ANTONIO MARTÍNEZ GRANDA	7.50	¿alguna anotación adicional?

Guardar cambios **Eliminar** **Cancelar Edición**

Figura 52. Editando – eliminando calificaciones

Se puede eliminar únicamente el último registro ingresado para evitar que existan eliminaciones erróneas de registros anteriores.

4.1.5 Generación de promedios para el docente

El sistema permite la generación automática de promedios parciales y generales. En la Figura 54 se aprecia la generación de promedios parciales, es decir por cada parámetro de calificación, dicha ventana se habilita al seleccionar la opción de promedios en la ventana de administración de calificaciones

PROMEDIOS PARCIALES						
Nº	Estudiante	Nota 1	Nota 2	Nota 3	PROMEDIO	70.0 % DE LA NOTA FINAL
1	ANDREA ELIZABETH ARMIJOS CARTUCHE	8.00	8.00	8.00	8.00	5.60
2	FREDDY PATRICIO LOAYZA ABAD	7.00	8.50	7.50	7.67	5.37
3	JENNY LILIANA SARAGURO PACHECO	6.00	8.00	0.00	4.67	3.27
4	JOHN PATRICIO SOLANO CABRERA	8.00	9.00	8.50	8.50	5.95
5	YESENIA CECIBEL CHAMBA JIMÉNEZ	7.50	8.00	7.50	7.67	5.37
6	JHYMER ANTONIO MARTÍNEZ GRANDA	7.50	9.50	9.30	8.77	6.14
7	KARLA KATHERINE HERRERA PAREDES	9.00	8.40	8.00	8.47	5.93
8	JAIRO ISRAEL BANDA BERMEO	6.00	8.00	9.00	7.67	5.37
9	LAURO IBÁN JAPA ÁVILA	8.00	9.00	7.00	8.00	5.60

Figura 53. Promedios parciales en una materia por parámetro a evaluar

Para la generación de promedios generales se toma los parciales de cada estudiante y según el peso especificado por el docente, se genera el promedio general para un paralelo es específico (Figura 55).

PROMEDIOS GENERALES								
Nº	Estudiante	Evaluaciones 70.0%	Lecciones 6.0%	Trabajos Autónomos 6.0%	Deberes - Tareas 6.0%	Participaciones 6.0%	Laboratorios- Talleres 6.0%	TOTAL
1	ANDREA ELIZABETH ARMIJOS CARTUCHE	5.60	0.55	0.54	0.57	0.48	0.51	8.25
2	FREDDY PATRICIO LOAYZA ABAD	5.37	0.55	0.48	0.57	0.48	0.56	8.02
3	JENNY LILIANA SARAGURO PACHECO	3.27	0.18	0.12	0.18	0.30	0.18	4.23
4	JOHN PATRICIO SOLANO CABRERA	5.95	0.54	0.42	0.51	0.48	0.33	8.23
5	YESENIA CECIBEL CHAMBA JIMÉNEZ	5.37	0.50	0.48	0.57	0.48	0.51	7.90
6	JHYMER ANTONIO MARTÍNEZ GRANDA	6.14	0.44	0.48	0.57	0.42	0.54	8.58
7	KARLA KATHERINE HERRERA PAREDES	5.93	0.51	0.48	0.51	0.48	0.52	8.43
8	JAIRO ISRAEL BANDA BERMEO	5.37	0.54	0.48	0.45	0.48	0.56	7.88
9	LAURO IBÁN JAPA ÁVILA	5.60	0.54	0.48	0.51	0.48	0.33	7.94
10	MARIO RICHAR LEÓN RAMÓN	5.83	0.48	0.48	0.57	0.54	0.57	8.48
11	JORGE ANIBAL MALES CHALÁN	6.30	0.52	0.48	0.48	0.54	0.54	8.86

Figura 54. Promedios generales en una determinada materia

La opción de promedios generales está disponible para cada materia del docente y aparece por defecto en sitio de administración de materias.

4.1.6 Generación de promedios para el estudiante

Al igual que el docente, el estudiante puede mantenerse constantemente informado sobre el promedio académico, para lo cual luego de la presentación de gráficas se debe acceder al enlace de calificaciones actuales y al sub apartado promedios que se presenta de cómo se aprecia en la Figura 56.

RESUMEN CALIFICACIONES			PROMEDIOS FINALES
Promedio evaluaciones:	6.14	Con un peso de 70.0 % a la nota final. Restan por calificar 4 evaluaciones	
Promedio lecciones:	0.44	Con un peso de 6.0 % a la nota final. Restan por calificar 3 lecciones	
Promedio trabajos autónomos:	0.48	Con un peso de 6.0 % a la nota final. Restan por calificar 5 trabajos autónomos	
Promedio deberes - tareas:	0.57	Con un peso de 6.0 % a la nota final. Restan por calificar 9 deberes - tareas	
Promedio participaciones:	0.42	Con un peso de 6.0 % a la nota final. Restan por calificar 3 participaciones	
Promedio laboratorios - talleres:	0.54	Con un peso de 6.0 % a la nota final. Restan por calificar 5 laboratorios - talleres	
TOTAL (hasta la fecha):	8.58	/	10

Figura 55. Promedios del estudiante

Se detalla lo mejor posible la calificación parcial y el peso que posee cada parámetro calificable para un completo entendimiento del estudiante.

Una vez terminado el análisis de resultados queda verificada la hipótesis planteada al principio y la viabilidad del sistema en desarrollo.

4.2 Metodología GAIA: Codificación del agente

Tomando en cuenta los entandares establecidos al principio, se procede a la codificación del agente.

4.2.1 Puesta en marcha del contenedor de agentes

El contenedor de agentes como su nombre lo indica, tiene la función de contener a los agentes inteligentes que son utilizados por las aplicaciones. Para levantar el contenedor basta con usar el siguiente código una vez descargado él .jar de la página oficial:

```
java -cp jade.jar jade.Boot -gui
```

Esto a su vez levanta el contenedor y la interfaz gráfica para llevar a cabo distintas operaciones de una manera sencilla.

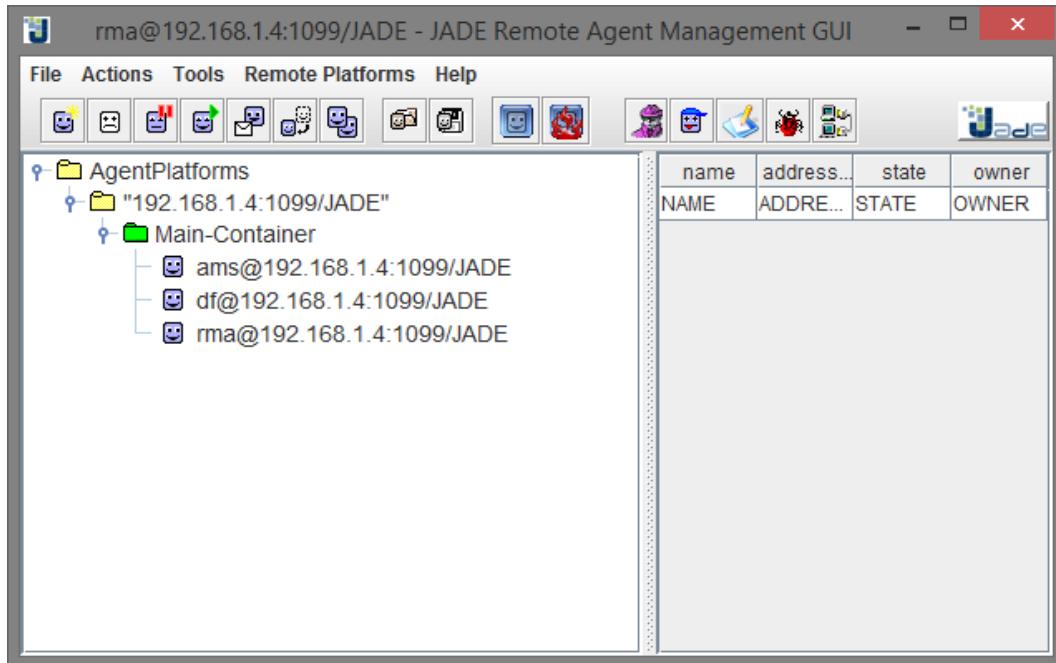


Figura 56. Contenedor de agentes en funcionamiento

En la Figura 57 se precia el contenedor principal levantado, además se visualiza 3 agentes que son utilizados para el correcto funcionamiento del mismo.

4.2.2 Funcionamiento del agente

El funcionamiento del agente inteligente está basado en una arquitectura BDI (Beliefs, Desires, Intentions o en español Creencias, Deseos, Intenciones) como se aprecia en la Figura 58.

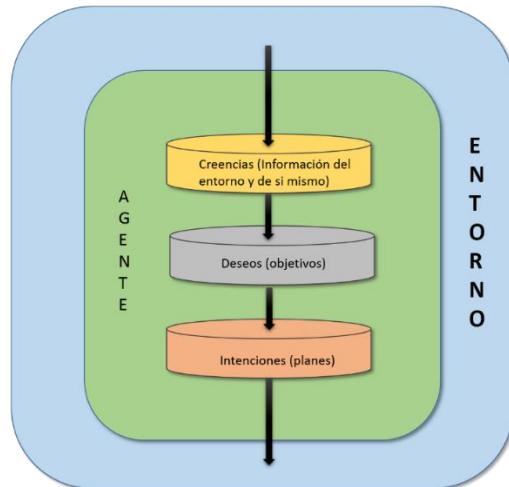


Figura 57. Arquitectura BDI

Esta arquitectura está caracterizada porque los agentes están dotados de los estados mentales que representan las Creencias, Deseos e Intenciones. [53]

4.2.3 Integración del Agente Inteligente al sistema web

JADE es la librería para agentes inteligentes utilizada en el sistema la cual da la opción de integrarse con otras tecnologías, en este caso con JSF y SPRING pero más específicamente con los servlets que son básicamente los controladores de la aplicación. Con el fin de conseguir una correcta integración entre estas tecnologías se heredó de la clase GateWayAgent (ver TABLA LXXVII).

TABLA LXXVII. CREACIÓN DEL AGENTE INTELIGENTE

```
public class AcademicAgent extends GatewayAgent{  
  
    /** realizar operación */  
    @Override  
    protected void processCommand(Object object) {  
        //operaciones correspondientes..  
    }  
  
    /** levantar agente */  
    @Override  
    protected void setup() {  
        super.setup();  
    }  
  
    /** matar agente */  
    @Override  
    protected void takeDown() {  
        super.takeDown();  
    }  
}
```

Posteriormente se utilizó la clase JadeGateway para llevar acabo las operaciones fundamentales mediante un controlador (TABLA LXXVIII)

TABLA LXXVIII. CREACIÓN DE LA CLASE CONTROLADORA PARA EL AGENTE

```
public class AgentController implements Serializable{  
  
    /**Levanta el agente con sus propiedades*/  
    public void startGatewayAgent(){  
        Properties prop = new Properties();  
        prop.put("host", "localhost");  
        prop.put("port", 1099);  
  
        /**Localización del agente dentro del proyecto*/  
        JadeGateway.init("ec.edu.unl.agents.agents.AcademicAgent",prop);  
    }  
    /**Fin de la ejecución del agente*/  
    public void endGateWayAgent(){  
        JadeGateway.shutdown();  
    }  
    /**Método utilizado con el fin de pasar información al agente  
     * para que ejecute las operaciones correspondientes*/  
    public void executeGateWayAgent(Object obj){  
  
        try {  
            JadeGateway.execute(obj);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

Al integrar estos métodos dentro de una clase se facilita la interacción con el agente inteligente. Ahora solo falta definir la clase en el contenedor de beans de Spring de la siguiente manera:

TABLA LXXIX. DECLARACIÓN DEL BEAN EN EL CONTENEDOR DE SPRING

```
<bean id="agentController"
      class="ec.edu.unl.controller.AgentController"
      scope="session"/>
```

Una vez que un usuario estudiante selecciona la opción generar las gráficas, automáticamente el agente entra en funcionamiento y se aprecia como se muestra en la Figura 59.

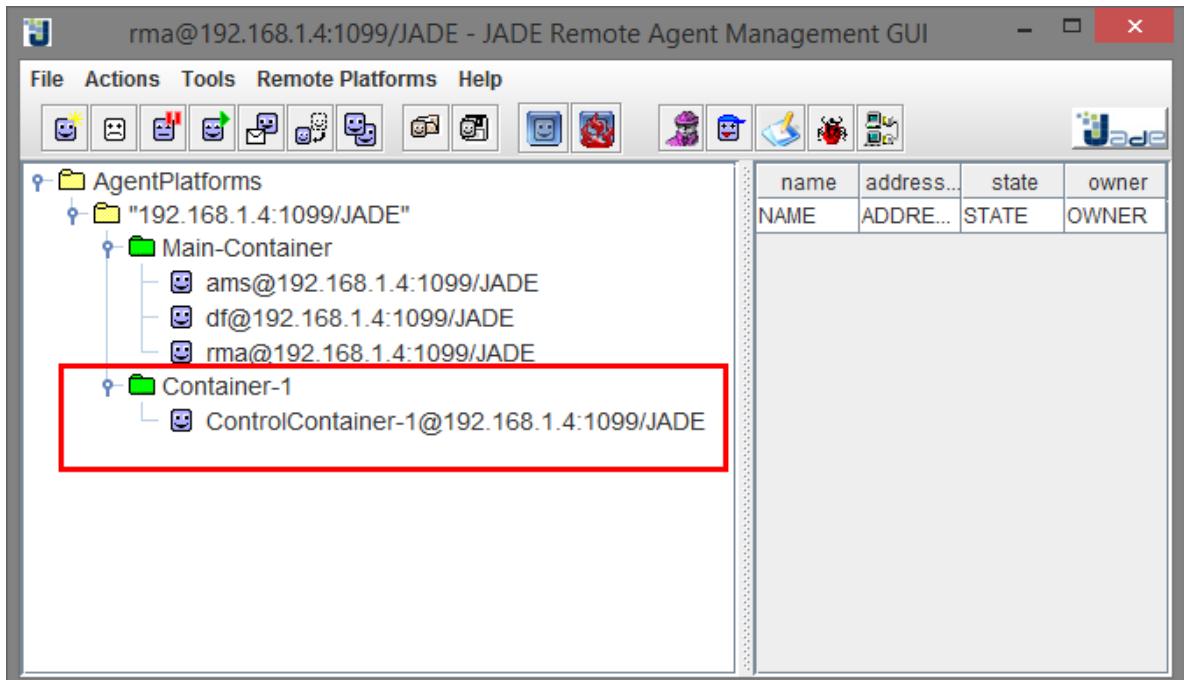


Figura 58. Creación del agente inteligente

5 Fase IV: Pruebas

5.1 Metodología XP: Pruebas unitarias del sistema

Para la realización de las pruebas se utilizó JUnit enfocándose en las partes más esenciales del código y que constituyen la base de la aplicación.

5.1.1 Pruebas de conexión a los Web Services

Estas pruebas están enfocadas al aseguramiento y verificación de obtención de información proveniente del SGA.

TABLA LXXX. TEST UNITARIO PARA LA CONEXIÓN A LOS WEB SERVICES

```
public class TestWS {  
    ConeksiionImpl conection;  
  
    @Before  
    public void init(){  
        conection = new ConeksiionImpl();  
        conection.setUsernameWS("usuarioWS");  
        conection.setPasswordWS("passwordWS");  
    }  
    @Test  
    public void testConectionValidacionStudentWS() {  
        String result = conection.conectionValidacionWS()  
            .sgawsValidarEstudiante("1105113458", "password")  
        assertEquals("true", result);  
    }  
    @Test  
    public void testConectionValidacionTeacherWS() {  
        String result = conection.conectionValidacionWS()  
            .sgawsValidarDocente("1104258460", "abcde");  
        assertEquals("false", result);  
    }  
    @Test  
    public void testConectionAcademicaWS() {  
        String result = conection.conectionAcademicaWS()  
            .sgawsPeriodosLectivos();  
        assertNotEquals(null, result);  
    }  
    @Test  
    public void testConectionInstitucionalWS() {  
        String result = conection.conectionInstitucionalWS()  
            .sgawsListaAreas();  
        assertNotEquals(null, result);  
    }  
    @Test  
    public void testConectionPersonalWS() {  
        String result = conection.conectionPersonalWS()  
            .sgawsDatosEstudiante("1105113458");  
        assertNotEquals(null, result);  
    }  
}
```

Los resultados para todos los test fueron los siguientes

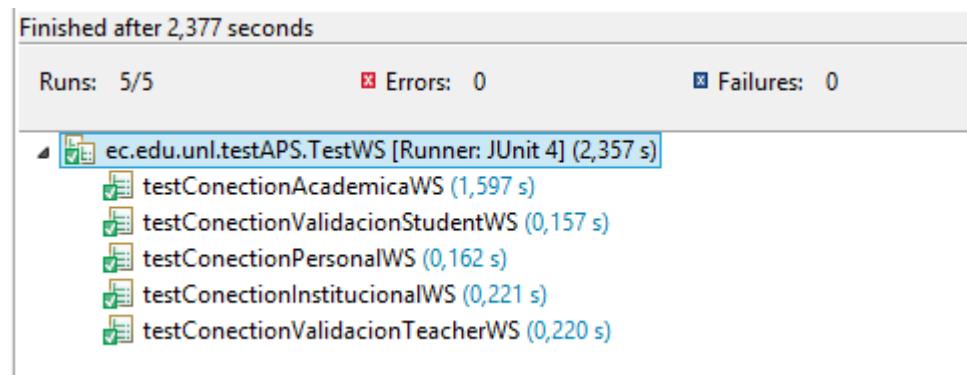


Figura 59. Resultados del test a los Web Services

Tomando en cuenta los resultados se concluye que la información obtenida de los Web Services es la necesaria y la conexión está correctamente establecida.

5.1.2 Pruebas de conexión a la base de datos

Otras pruebas de vital importancia son las realizadas a los métodos que permiten conectarse e interactuar con la base de datos local, ya que es justo aquí donde se almacena la información que permite al sistema llevar a cabo sus operaciones.

TABLA LXXXI. TEST DE CONEXIÓN A LA BASE DE DATOS

```
public class TestDB {  
  
    AcademicDAO academic;  
  
    @Before  
    public void init(){  
        academic = new AcademicDAO();  
    }  
  
    @Test  
    public void testGetListTeachers() {  
        List<Teacher> listNew = academic.getListTeachers();  
        assertTrue(listNew.get(0) instanceof Teacher);  
    }  
  
    @Test  
    public void testReturnStudent() {  
        Student student = academic.returnStudent(6);  
  
        assertTrue(student.getIdentification_card().equals("1105113458"));  
    }  
}
```

Los resultados de estas pruebas se presentan a continuación:

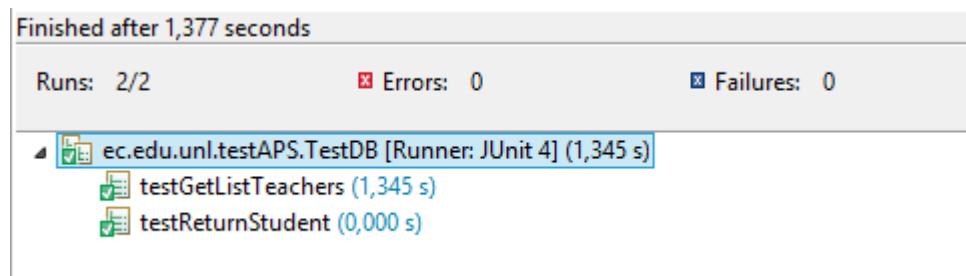


Figura 60. Resultados del test de conexión a la base de datos

Después de revisar los resultados se concluye que la conexión e interacción con la base de datos es exitosa y no presenta ningún fallo.

5.2 Metodología GAIA: Pruebas unitarias del agente inteligente

El agente inteligente constituye básicamente el corazón del sistema por lo que es de vital importancia comprobar su correcto funcionamiento.

5.2.1 Pruebas del agente inteligente

Para la realización de la prueba, como primer método de la clase se tiene la carga de información necesaria para la prueba, como las calificaciones y configuraciones generales.

TABLA LXXXII. TEST UNITARIO PARA EL AGENTE INTELIGENTE

```
public class TestAgent {  
  
    @Test  
    public void testRunAgent() {  
        int numberEvals = parameters.getEvaluations().size();  
        StudentAdminController controller  
            = new StudentAdminController();  
        Parameter newParameters = controller.runAgent(parameters);  
  
        assertTrue(numberEvals < newParameters  
                   .getEvaluations().size());  
    }  
}
```

Los resultados obtenidos para la prueba del agente se presentan a continuación:

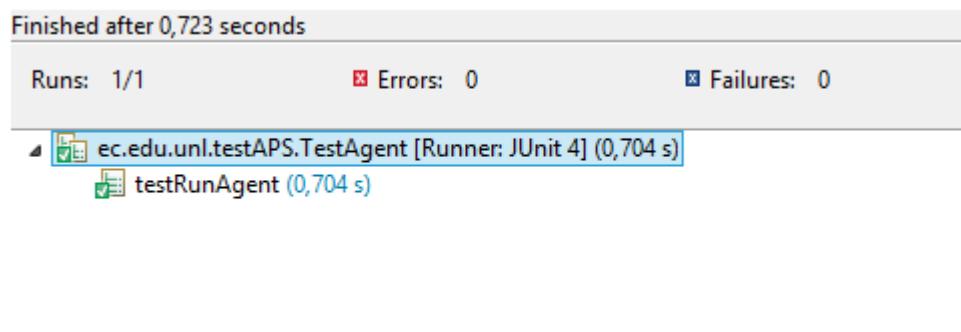


Figura 61. Resultados del test para el agente

Tomando en cuenta estos resultados, se llega a la conclusión de que agente inteligente está realizando las operaciones asignadas de una manera correcta.

5.3 Pruebas de Funcionalidad

Para las pruebas se utilizará calificaciones y promedios reales obtenidos en evaluaciones tomando una muestra de estudiantes del octavo módulo de la carrera de ingeniería en sistemas durante marzo – julio 2014 en la materia de Redes y Telecomunicaciones.

TABLA LXXXIII. CALIFICACIONES REALES DE ESTUDIANTES

Estudiante	C.I.	Nota 1	Nota 2	Nota 3	Nota 4
CHRISTIAN RAUL LARA PACHECO	1104648199	7.86	7.86	5.86	7.00
JÉSSICA MERCEDES CUEVA RIOFRÍO	1104583040	7.14	7.86	4.71	7.00
SORAYA ISAMAR LARA TAPIA	1105537532	7.00	7.71	7.29	5.71
RAUL ALEXANDER GOMEZ ARMIJOS	1724102429	7.14	7.86	5.00	7.14
PRISCILLA LOURDES PACHECO ORDOÑEZ	1104850100	7.86	8.57	4.71	5.14
CLAUDIO ISRAEL RODRIGUEZ SALINAS	1104257579	8.57	9.29	6.43	6.86
SHEIMY LILIANA PATIÑO RIVERA	1105044158	7.86	8.57	6.57	7.79
HENRY PAÚL VIVANCO ENCALADA	1105163875	8.57	9.29	5.71	7.14
JUAN PABLO GUAMÁN RODRÍGUEZ	1104904667	6.57	7.29	7.14	7.14

Haciendo uso del sistema se generará la última calificación es decir se tomarán en cuenta las 3 evaluaciones anteriores y posteriormente se compararán resultados.

TABLA LXXXIV. GENERACIÓN DE CALIFICACIONES PARA LA ÚLTIMA EVALUACIÓN

Estudiante	C.I.	Nota 1	Nota 2	Nota 3	Predic. 4
CHRISTIAN RAUL LARA PACHECO	1104648199	7.86	7.86	5.86	6.61
JÉSSICA MERCEDES CUEVA RIOFRÍO	1104583040	7.14	7.86	4.71	6.34
SORAYA ISAMAR LARA TAPIA	1105537532	7.00	7.71	7.29	5.95
RAUL ALEXANDER GOMEZ ARMIJOS	1724102429	7.14	7.86	5.00	6.74
PRISCILLA LOURDES PACHECO ORDOÑEZ	1104850100	7.86	8.57	4.71	5.90
CLAUDIO ISRAEL RODRIGUEZ SALINAS	1104257579	8.57	9.29	6.43	6.59
SHEIMY LILIANA PATIÑO RIVERA	1105044158	7.86	8.57	6.57	7.00
HENRY PAÚL VIVANCO ENCALADA	1105163875	8.57	9.29	5.71	6.95
JUAN PABLO GUAMÁN RODRÍGUEZ	1104904667	6.57	7.29	7.14	6.80

En conclusión se puede decir que el sistema ha tomado como base las calificaciones existentes y los distintos factores determinantes del rendimiento propios de cada individuo consiguiendo generar nuevas calificaciones en rango bastante aproximado al real.

g. Discusión

1 Desarrollo de la propuesta alternativa

Para llevar a cabo la realización del presente proyecto de titulación se evaluaron cada uno de los objetivos específicos planteados de la manera en que se fueron cumpliendo como se detalla a continuación.

Objetivo Específico 1: Analizar y diseñar las creencias, deseos e intenciones de Agentes Deliberativos para la predicción del desempeño académico de los estudiantes de la Universidad Nacional de Loja.

Para el cumplimiento de este objetivo y mediante el uso de la metodología GAIA se consiguió analizar y diseñar el modelo BDI (creencias, deseos e intenciones) del agente inteligente, todo este proceso se encuentra detallado en el apartado 2 Análisis y diseño para la implementación del agente inteligente mediante la utilización de la metodología GAIA de la sección de resultados.

Objetivo Específico 2: Desarrollar un sistema web con conexión a los Web Services del Sistema de Gestión Académico (SGA) para obtener resultados.

El sistema web se lo desarrolló en base a estándares de programación y siguiendo las fases de la metodología XP como se especifica en la sección 1 Análisis y diseño del sistema mediante la utilización de la metodología XP. La conexión a los Web Services del SGA se la consiguió mediante la utilización de la librería JAX-WS que permite generación automática de métodos partiendo de un archivo WSDL en una arquitectura SOAP (ver Capítulo 1 sección 1.4 Web Services) lo que facilita el consumo de recursos mediante Web Services.

Objetivo Específico 3: Integrar Agentes Deliberativos al Sistema Web para el correcto funcionamiento.

Una vez cumplido con el objetivo 1 y 2 se procede a realizar la integración entre el Sistema Web y el agente inteligente cumpliendo así con los objetivos planteados en cada una de las fases de la metodología GAIA

2 Valoración técnica económica ambiental

2.1 Recursos

Para la realización del presente trabajo de investigación titulado “**Desarrollo de un Sistema Inteligente para predicción del desempeño académico de los estudiantes universitarios**”, se contó con los recursos humanos, técnicos, económicos y tecnológicos como hardware y software, necesarios para su exitosa culminación.

En cuanto al aspecto económico, los recursos materiales fueron adquiridos por el tesista, en su totalidad.

Por lo manifestado anteriormente, la ejecución del proyecto fue totalmente factible y da cumplimiento a todos los objetivos planteados. Los recursos utilizados se detallan a continuación:

- **Recursos Humanos**

Para el desarrollo del proyecto se necesitó el recurso humano conformado por 2 personas, un profesional en el campo quien se encargó de la dirección del proyecto y un egresado de la carrera Ingeniería en Sistemas quien hizo las veces de analista y programador.

En la TABLA LXXXVI refleja el número de horas y el coste que han empleado cada uno de estos perfiles en el desarrollo del proyecto. El número de horas de trabajo multiplicado por el coste por hora de cada uno, da como resultado el coste total del recurso humano.

TABLA LXXXV. TALENTO HUMANO

Talento Humano			
Rol	Número de horas	Costo por hora (\$)	Valor Total (\$)
Director de tesis	-	-	-
Analista – Desarrollador	350	5.00	1750.00
Subtotal			1750.00

- **Recursos Materiales**

Para el desarrollo del proyecto ha sido necesario el uso de material de oficina empleado para la documentación, almacenamiento de la información y demás.

Como resultado se presenta la siguiente tabla que indica los costes de materiales utilizados:

TABLA LXXXVI. RECURSOS MATERIALES

Recursos Materiales			
Rubro	Cantidad	Costo unidad (\$)	Valor Total (\$)
Hojas A4	300	0.03	9.00
Anillado	3	1.50	4.50
Perfil	6	0.50	3.00
Carpeta plástica	3	0.60	1.80
Libros	2	50.00	100.00
Impresiones	200	0.10	20.00
CD's	5	0.35	1.75
DVD's	2	0.50	1.00
Memoria Flash	1	10.00	10.00
Subtotal			151.05

- **Recursos Técnicos/Tecnológicos**

Estos recursos fueron divididos en tres secciones: Recursos de Hardware, software y Comunicaciones, a continuación los detallamos:

Recursos de Hardware

Dentro del hardware consta el equipo necesario para el desarrollo del sistema.

TABLA LXXXVII. RECURSOS HARDWARE

Recursos Hardware			
Rubros	Cantidad	Costo unidad (\$)	Valor Total (\$)
Portátil DELL INSPIRON	1	1500.00	1500.00
Subtotal			1500.00

Recursos de Software

La mayor parte del software utilizado es libre por lo que no implicaría ningún costo.

TABLA LXXXVIII. RECURSOS SOFTWARE

Recursos Software			
Rubros	Cantidad	Costo unidad (\$)	Valor Total (\$)
Java	1	0.00	0.00
Eclipse IDE	1	0.00	0.00
Netbeans IDE	1	0.00	0.00
MySQL	1	0.00	0.00
Frameworks JSF, Spring, etc.	1	0.00	0.00
Git – Github	1	0.00	0.00
Diagramas en Gliffy	1	0.00	0.00
Balsamiq Mockups	1	0.00	0.00
Cronogramas en Gantter	1	0.00	0.00
Enterprise Architect	1	135.00	135.00
Subtotal			135.00

- Recursos de Comunicación**

Un recurso fundamental es Internet, el mismo constituye un elemento imprescindible para la realización de consultas y búsqueda de información para el proyecto.

TABLA LXXXIX. RECURSOS DE COMUNICACIÓN

Recursos de Comunicación			
Rubros	Duración	Valor Unitario (\$)	Valor Total (\$)
Internet	500 h	1.00	500.00
Llamadas telefónicas	10 h	0.06	36.00
Subtotal			536.00

- **Resumen de Presupuesto**

En la TABLA XCI se resume el balance del coste total del proyecto de fin de carrera, para ello se suman los subtotales obtenidos en las tablas anteriores:

TABLA XC. COSTO TOTAL DEL PROYECTO

Costo Total	
Recursos	Valor Total (\$)
Talento Humano	1750.00
Materiales	151.05
Hardware	1500.00
Software	135.00
Comunicación	536.00
TOTAL:	4072.05

El proyecto tiene un coste total de **\$4072.05**

h. Conclusiones

- Los Web Services facilitan la obtención de información sin importar el tipo de tecnología que se esté utilizando lo que a su vez constituye una enorme ventaja para conseguir trabajo colaborativo entre desarrolladores que sean expertos en diferentes lenguajes.
- La utilización de una arquitectura para el agente inteligente basada en creencias, deseos e intenciones da una visión clara de la lógica a implementar como si se tratase de un ente que puede razonar y solucionar problemas de la manera más óptima posible, asimismo facilita la integración de nuevos agentes cada uno con una funcionalidad nueva y comunicación entre ellos.
- La utilización de la metodología XP en el proceso de desarrollo de software permitió un desarrollo rápido del sistema enfocándose en la calidad y reduciendo al mínimo la presencia de errores en el código.
- El uso de la metodología GAIA para la creación del agente inteligente, dio una visión clara y estructurada basada en modelos, facilitando en gran medida el análisis y diseño del mismo.
- Mantener al estudiante en constante información sobre su desempeño académico actual permite eliminar la incertidumbre generada por el desconocimiento de sus calificaciones. La predicción futura a su vez da la posibilidad de enfocarse en ciertas falencias que debería corregir para mejorar su promedio.
- Dar la facilidad al docente que lleve un registro claro y detallado de las calificaciones de sus estudiantes, automatiza en gran medida este proceso a más de que la información se encuentra centralizada y disponible en cualquier momento que lo requiera.

i. Recomendaciones

- Antes de utilizar el sistema es recomendable revisar los manuales tanto para el docente como del estudiante con el fin de conseguir un correcto funcionamiento y rendimiento del mismo.
- Se recomienda la implementación y utilización del sistema para predicción del desempeño académico ya que contribuye a la mejora ciertos aspectos fundamentales para aprobación del módulo académico de los estudiantes universitarios.
- La utilización de agentes inteligentes en los sistemas basados en software que se desarrollan para el beneficio de nuestra universidad, ya permiten la construcción de aplicaciones de calidad valiéndose de técnicas de inteligencia artificial.
- A los docentes evitar la utilización de medios poco adecuados para el almacenamiento de calificaciones como hojas de Excel ya que al ser simples archivos pueden generar perdida de información al no existir respaldos.
- A los estudiantes el mantener un constante seguimiento del progreso académico y así reducir al máximo la probabilidad de pérdida de módulo por exceso de confianza y desconocimiento.
- La utilización de metodologías para el análisis y diseño de software y agentes inteligentes mediante XP y GAIA respectivamente, ya que brindan eficiencia y orden al momento de cumplir con sus tareas.

j. Bibliografía

- [1] M. Navarro y D. Lladó, *La gestión escolar. Una aproximación a su estudio*, Palibrio LLC, 2014.
- [2] E. Y. Morocho Pauta y X. E. Ortiz Jara, «El desarrollo de las tareas escolares influyen en el rendimiento académico de niños y niñas del sexto y séptimo año de educación general básica del centro educativo fiscal “5 de septiembre”,» 2011. [En línea]. Available: <http://www.biblioteca.ueb.edu.ec/bitstream/15001/1121/1/305%20P..pdf>. [Último acceso: 8 Julio 2014].
- [3] G. M. Garbanzo Vargas, «Factores asociados al rendimiento académico en estudiantes universitarios, una reflexión desde la calidad de la educación superior pública,» *redalyc Revista Educación*, nº 31, pp. 43-63, 2007.
- [4] N. Verdezoto Días y G. Caicedo Rossi, «Análisis, diseño e implementación de un sistema de administración de contenidos e interacción de aplicaciones web que soporte un nuevo modelo conceptual orientado a la usabilidad de administradores y usuarios,» [En línea]. Available: <http://www.dspace.espol.edu.ec/bitstream/123456789/4904/1/7700.pdf>.
- [5] A. Hernandez, «Los Sistemas de Información: evolución y desarrollo,» [En línea]. Available: dialnet.unirioja.es/descarga/articulo/793097.pdf. [Último acceso: 4 Junio 2014].
- [6] R.-M. Editoriales, «Conceptos generales de la arquitectura web,» [En línea]. Available: <http://www.rama.es/descargas/descargar.php?fichero=Z3dkZXNjYXJnYXNwcm9mIzIzM5Nzg4NDk5NjQxNjA3X0NhG10dWxvIDEucGRm>. [Último acceso: 3 Junio 2014].
- [7] «Diccionario de Informática y Tecnología,» [En línea]. Available: <http://www.alegsa.com.ar/Dic/aplicacion%20web.php>. [Último acceso: 8 Junio 2014].
- [8] C. Cubillos, «Arquitectura Cliente/Servidor,» 2008. [En línea]. Available: <http://ocw.pucv.cl/cursos-1/arquitectura-de-sistemas-de-software/materiales-de-clases/web-cliente-servidor>. [Último acceso: 7 Octubre 2014].
- [9] GXtechnical, «Web Services,» [En línea]. Available: http://library.gxtechnical.com/gxdlsp/pub/genexus/internet/technicalpapers/web_services.htm. [Último acceso: 10 Junio 2014].
- [10] J. González Cornejo, «Servicios Web XML o Web Services,» 25 Julio 2003. [En línea]. Available: <http://www.docirs.cl/webservices4.html>. [Último acceso: 15 Junio 2014].

- [11] N. Tedeschi, «Web Services, un ejemplo práctico,» [En línea]. Available: <http://msdn.microsoft.com/es-es/library/bb972248.aspx>. [Último acceso: 18 Junio 2014].
- [12] W. W. W. Consortium, «World Wide Web Consortium,» 2014. [En línea]. Available: <http://www.w3c.es/>. [Último acceso: 22 Junio 2014].
- [13] M. Mármol Panama y D. Peñaherrera Pacheco, «Construcción de un benchmark para realizar un estudio comparativo entre web services en plataformas Java y .Net,» Quito, 2006.
- [14] I. K. Center, «La estructura de un mensaje SOAP,» 2013. [En línea]. Available: http://www-01.ibm.com/support/knowledgecenter/SSKM8N_7.0.0/com.ibm.etools.mft.doc/ac55780_.htm?lang=es. [Último acceso: 24 Junio 2014].
- [15] U. XML, «UDDI XML,» 2014. [En línea]. Available: <http://uddi.xml.org/>. [Último acceso: 25 Junio 2014].
- [16] ABC.ES, «Crean un software que mejora el rendimiento académico,» *Actualidad*, 31 10 2012.
- [17] N. Fila y B. Misnevs, «Intelligent Agents' implementation for study process quality performance indicators,» de *Reliability and Statistics in Transportation and Communication*, Riga, Latvia, 2009.
- [18] P. Valdiviezo Díaz y H. Gómez Alvarado, Inteligencia Artificial Avanzada, Loja: EDILOJA Cía. Ltda., 2013.
- [19] J. Merelo, «Agentes Autónomos Inteligentes,» [En línea]. Available: <http://www.redcientifica.com/doc/doc199903310001.html>. [Último acceso: 29 Junio 2014].
- [20] S. Russell y P. Norvig, Inteligencia Artificial: Un enfoque moderno, Prentice Hall, 1996.
- [21] L. Hernández López , «Heurísticas para el control deliberativo en una arquitectura de agentes inteligentes de tiempo real,» Valencia, 2004.
- [22] J. M. Corchado, «Modelos y Arquitecturas de Agente,» [En línea]. Available: [http://bisite.usal.es/archivos/c1%20\(1\).pdf](http://bisite.usal.es/archivos/c1%20(1).pdf). [Último acceso: 29 Junio 2014].
- [23] UPV, «Definición de agentes inteligentes y clasificación,» [En línea]. Available: http://personales.upv.es/ccarrasc/doc/2003-2004/AI_Web/definicion.html. [Último acceso: 2 Julio 2014].
- [24] F. Stan y A. Grasser, Is it an Agent or just a Program? A taxonomy for Autonomous Agents, Springer Verlag, 1996.

- [25] FIPA, «fipa.org,» 2014. [En línea]. Available: <http://www.fipa.org/>. [Último acceso: 5 Julio 2014].
- [26] UDLAP,
 «http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/camargo_m_d/capitulo3.pdf,» [En línea]. Available:
http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/camargo_m_d/capitulo3.pdf. [Último acceso: 8 Julio 2014].
- [27] A. Guerra Hernández, «Agentes Inteligentes,» Xalapa.
- [28] T. Finin, «Tim Finin Blog,» 2014. [En línea]. Available:
<http://www.csee.umbc.edu/~finin/>. [Último acceso: 10 Julio 2014].
- [29] F. Serradilla García y Á. Arroyo Castillo, «Agentes de Información,» 15 Marzo 2012. [En línea]. Available:
http://www.sia.eui.upm.es/isa/doku.php?id=asignaturas:agentes_inteligentes. [Último acceso: 22 Julio 2014].
- [30] FIPA, «FIPA ACL Message Structure Specification,» [En línea]. Available:
<http://www.fipa.org/specs/fipa00061/SC00061G.pdf>. [Último acceso: 25 Julio 2014].
- [31] J. Pavón Mestras, «Modelos y Arquitecturas de Agentes,» [En línea]. Available:
<http://www.fdi.ucm.es/profesor/jpavon/doctorado/arquitecturas.pdf>. [Último acceso: 28 Julio 2014].
- [32] M. Ramírez Klapp, «Agentes Inteligentes,» 2011. [En línea]. Available:
<http://www.slideshare.net/miramire/unidad-no-5-agentes-inteligentes#>. [Último acceso: 28 Julio 2014].
- [33] A. Francisco Javier, M. Jorge Alejandro, J. Moreno y D. Ovalle, «Diseño y Desarrollo de Mecanismos de Razonamiento Multi-Agente para la Negociación de Energía Eléctrica Utilizando JESS Y JADE,» *Redalyc*, vol. I, pp. 51-56, 2006.
- [34] E. Naranjo Díaz y J. Tandazo Rodriguez, «desarrollo de un agente buscador inteligente de metadatos geograficos para la UNISIG,» Quito, 2008.
- [35] J. Balsas Almagro, M. Díaz Galiano, A. Montejo Ráez, F. Martínez Santiago, M. García Vega y L. Ureña López, «Una Arquitectura para Agentes de Interfaz Inteligentes: el ordenador “sugerente”,» Madrid.
- [36] «Diseño y Aplicación de Modelos Multiagente para Ayuda a la Decisión (TIC2000-1370-04-01),» Madrid, 2003.
- [37] H. Paz, «Desarrollar de un Sistema Multiagente mediante una Arquitectura Híbrida, siendo reactivo para el dispositivo móvil y deliberativo para el servidor, en el caso de estudio de citas médicas odontológicas.,» México, D. F., 2012.

- [38] M. Gancino Cordones y E. Martínez Núñez, «Desarrollo de un sistema web de gestión de brigadas médicas del área 2 de salud de Pastaza,» Quito, 2011.
- [39] INTECO, «Curso de desarrollo ágil,» [En línea]. Available: http://www.ulsa.edu.sv/index.php?option=com_phocadownload&view=category&download=111:curso-de-desarrollo-gil&id=8:software-libre. [Último acceso: 20 Julio 2014].
- [40] chromatic, «Roles in Extreme Programming,» de *Extreme Programming Pocket Guide*, O'Reilly Media, 2003, pp. 59-64.
- [41] P. Letelier y C. Penadés, «Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP),» 2004. [En línea]. Available: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>. [Último acceso: 7 Octubre 2014].
- [42] M. Fowler, «Is Design Dead?,» 2001. [En línea]. Available: www.martinfowler.com/articles/designDead.html. [Último acceso: 8 Octubre 2014].
- [43] P. Kogan y G. Parra, «Análisis de requerimientos de un sistema multigente de robots que juegan futbol,» Buenos Aires.
- [44] jatun, «Java Enterprise Edition,» 2014. [En línea]. Available: <http://www.jatun.com/web/company/training/javaee5>. [Último acceso: 1 Septiembre 2014].
- [45] A. Pérez García, «JSF - Java Server Faces,» 2006. [En línea]. Available: <http://www.desarrolloweb.com/articulos/2380.php>. [Último acceso: 2 Septiembre 2014].
- [46] Ç. Çivici, «primefaces documentation,» 2014. [En línea]. Available: <http://www.primefaces.org/documentation>. [Último acceso: 11 Septiembre 2014].
- [47] Spring.io, «Spring Framework,» Spring.io, 2014. [En línea]. Available: <http://projects.spring.io/spring-framework>. [Último acceso: 10 Septiembre 2014].
- [48] hibernate.org, «Hibernate ORM,» hibernate.org, 2014. [En línea]. Available: <http://hibernate.org/orm/>. [Último acceso: 12 Septiembre 2014].
- [49] Apache.org, «Apache Tomcat,» Apache.org, 2014. [En línea]. Available: <http://tomcat.apache.org/>. [Último acceso: 13 Septiembre 2014].
- [50] java.net, «JAX-WS,» java.net, 2014. [En línea]. Available: <https://jax-ws.java.net/>. [Último acceso: Septiembre 2014].
- [51] gestiondeoperaciones.net, «Cómo utilizar una Regresión Lineal para realizar un Pronóstico de Demanda,» gestiondeoperaciones.net, 22 Febrero 2014. [En línea]. Available: <http://www.gestiondeoperaciones.net/proyeccion-de-demanda/como-utilizar-una-regresion-lineal-para-realizar-un-pronostico-de-demanda/>. [Último acceso: 11 Septiembre 2014].

- [52] C. d. I. P. Educar, «EL REGLAMENTO GENERAL A LA LEY ORGÁNICA DE EDUCACIÓN INTERCULTURAL,» Centro de Información Pedagógica Educar, 2014. [En línea]. Available: http://www.educar.ec/servicios/regla_loei-6.html. [Último acceso: 1 Septiembre 2014].
- [53] projectolatin.org, «Arquitecturas de Agentes,» 2014. [En línea]. Available: <http://escritura.projectolatin.org/inteligencia-artificial/arquitectura-de-agentes/>. [Último acceso: 27 Julio 2014].
- [54] C. Llamas Bello, «Protocolos de interacción entre agentes (FIPA),» [En línea]. Available: <http://www.infor.uva.es/~cllumas/MAS/ProInterAg.pdf>. [Último acceso: 29 Agosto 2014].

k. Anexos