ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

DESARROLLO DE UN MÓDULO DE EVALUACIÓN ABIERTO PARA EL LMS (LEARNING MANAGEMENT SYSTEMS) MOODLE

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

MARCO ANTONIO TOSCANO FREIRE

Correo: marcgigolito57@gmail.com

DIRECTOR: Ing. Francisco Hallo

Correo: fhallo@yahoo.com

Quito, Abril 2008

DECLARACIÓN

Yo Marco Antonio Toscano Freire, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

.....

Marco Antonio Toscano Freire

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Marco Antonio Toscano Freire, bajo mi supervisión.

.....

Ing. Francisco Hallo

DIRECTOR DE PROYECTO

AGRADECIMIENTO

Quisiera comenzar agradeciendo al dueño de toda mi vida, el sustento de mis días, a ti señor Jehová por haberme ayudado y dado valor para afrontar y sobresalir en cada reto que la vida me ha entregado.

Agradecer con todo mi alma, corazón a mis viejitos, los cuales siempre me inculcaron los valores, consejos y sobre todo me dieron su amor siempre que lo necesite, a los cuales toda mi vida estaré agradecido.

A mis hermanitos, mi Jimyto, mi Zulemita y mi Elianita los cuales siempre confiaron en mí y me apoyaron en cada decisión tomada, a pesar de no ser siempre la correcta, gracias ñañitos.

A toda mi familia, amigos que de una u otra forma me brindaron su apoyo incondicional y estuvieron conmigo cuando los necesite. Gracias amigos(as).

Y como olvidar a mis profesores de escuela, colegio y universidad; quienes siempre apoyaron mis sueños por más absurdos que fuesen gracias.

DEDICATORIA

Este trabajo se lo dedicó con todo mi corazón a mi papito y a mi mamita, quienes siempre fueron mi ejemplo a seguir, quienes gracias a sus consejos y enseñanzas me hicieron conocer lo bonito que es vivir en unión familiar, quienes desarrollaron en mi todos los valores, aptitudes para ser una buena persona, quienes siempre han sido mi apoyo en cada decisión tomada, a quien más sino a ustedes viejitos, porque se lo merecen hoy y siempre.

Quiero que sepan que los amo mucho y que siempre estaré con ustedes apoyándoles en todo lo que necesiten, viejitos para ustedes siempre todo mi respeto y agradecimiento.

INDICE DE CONTENIDOS

INDICE DE CONTENIDOS ..... **VI**
INDICE DE FIGURAS ..... **XI**
ÍNDICE DE TABLAS ..... **XIV**
RESUMEN ..... **XV**
PRESENTACIÓN ..... **XVII**

CAPÍTULO I **1**
1. ANÁLISIS **1**
1.1 DEFINICIÓN DEL PROBLEMA 1
1.1.1 E-LEARNING 1
1.1.1.1 Fundamentos 1
1.1.1.2 Ventajas 2
1.1.1.3 Desventajas 2
1.1.1.4 Plataformas E- learning 3
1.1.1.5 Estándares E-Learning 4
1.1.2 MOODLE 6
1.1.2.1 Fundamentos 6
1.1.2.2 Características 6
1.1.2.3 Estructura de Moodle 8
1.1.3 PROBLEMA 12
1.2 PLANTEAMIENTO DE SOLUCIONES 12
1.2.1 GENERALIDADES 12
1.2.2 TECNOLOGIA MOODLE 13
1.2.3 AJAX 14
1.2.3.1 Características 14
1.2.3.2 AJAX en Moodle 14
1.2.3.3 AJAX en PHP 15
1.2.4 SELECCIÓN DE ESTRATEGIA 15
1.2.5 ANALISIS DE METODOLOGIAS 16
1.2.5.1 Fundamentos 16
1.2.5.2 Proceso Desarrollo de Moodle 16
1.2.5.3 Estudio de las Metodologías 18
1.2.5.4 Comparativa 21
1.2.5.4.1 Generalidades 22
1.2.5.4.2 Cuadro Comparativo Respecto a Moodle 24
1.2.5.4.3 Adopción de Metodología 26
1.2.5.5 XP (Extreme Programming) [15] 26
1.2.5.5.1 Técnicas de XP 28
1.2.5.5.2 Roles 30
1.3 DEFINICIÓN DEL MÓDULO DE EVALUACIÓN 31
1.3.1 CARACTERÍSTICAS 31
1.3.1.1 Principios de Aprendizaje 32
1.3.1.2 Características 32
1.3.2 LIMITANTES 33

1.3.3 HISTORIAS DE USUARIO 34
1.3.3.1 Planificación Inicial de las Iteraciones 34
1.3.3.2 Desarrollo de las Iteraciones 35
1.3.3.2.1 Iteración 1: Gestión de Elementos de Estudio 36
1.3.3.2.2 Iteración 2: Gestión de la Evaluación 37
1.3.2.2.3 Iteración 3: Gestión de la Retroalimentación 38
1.3.3.3 Desarrollo de Historias de Usuario 39
1.3.3.3.1 Iteración 1: Gestión de Elementos de Estudio 39
1.3.3.3.2 Iteración 2: Gestión de la Evaluación 42
1.3.3.3.3 Iteración 3: Gestión de la Retroalimentación 43
1.3.3.4 Planificación del Versionamiento 45

CAPÍTULO II 47
2. DISEÑO 47
2.1 DISEÑO DE MODELOS 47
2.1.1 FUNDAMENTOS 47
2.1.2 ARQUITECTURA 48
2.1.3 MODELAMIENTO AGIL (AM) 49
2.1.3.1 Fundamentos 49
2.1.3.2 Principios 50
2.1.3.3 Prácticas 51
2.1.4 CRC (Class, Responsabilities and Collaborations) 53
2.1.4.1 Fundamentos 53
2.1.4.2 Modelamiento CRC 54
2.1.4.3 Modelo de EVAAB 55
2.1.4.3.1 Iteración 1: Gestión de Elementos de Estudio 56
2.1.4.3.1.1 Gestión de Cuestionario 56
2.1.4.3.1.2 Gestión de Preguntas 56
2.1.4.3.1.3 Gestión de Respuestas 57
2.1.4.3.1.4 Gestión de Pistas 57
2.1.4.3.1.5 Gestión de Referencias 58
2.1.4.3.2 Iteración 2: Gestión de la Evaluación 59
2.1.4.3.2.1 Presentar Elementos de Ayuda 59
2.1.4.3.2.2 Evaluar Cuestionario 59
2.1.4.3.3 Iteración 3: Gestión de la Retroalimentación 60
2.1.4.3.3.1 Seguimiento de Resolución de Preguntas 60
2.1.4.3.3.2 Mostrar Progreso 61
2.1.4.3.3.3 Generación de Informe de Proceso 62
2.1.5 REFACTORIZACIÓN CRC 62
2.5.1.1 Iteración 1: Gestión de Elementos de Estudio 62
2.5.1.2 Iteración 2: Gestión de Evaluación 63
2.5.1.3 Iteración 3: Gestión de Retroalimentación 64
2.1.6 DIAGRAMAS DE ACTIVIDAD 64
2.1.6.1 Iteración 1: Gestión de Elementos de Estudio 64
2.1.6.1.1 Gestión de Cuestionario 65
2.1.6.1.2 Gestión de Preguntas 65
2.1.6.1.3 Gestión de Respuestas 66
2.1.6.1.4 Gestión de Pistas 67
2.1.6.1.5 Gestión de Referencias 67

2.1.6.2 Iteración 2: Gestión de Evaluación 68
2.1.6.2.1 Presentar Elementos de Ayuda 68
2.1.6.2.2 Evaluar Cuestionario 69
2.1.6.3 Iteración 3: Gestión de Retroalimentación 69
2.1.6.3.1 Seguimiento de Resolución de Preguntas 70
2.1.6.3.2 Mostrar Progreso 70
2.1.6.3.3 Generación de Informe de Proceso 71
2.2 DISEÑO DE BASE DE DATOS 71
2.2.1 GENERALIDADES 71
2.2.2 ESTANDARIZACIÓN 72
2.2.2.1 Diseño del Estándar EVAAB 72
2.2.2.2 Estándar Moodle 73
2.2.3 IDENTIFICACIÓN DE ENTIDADES, ATRIBUTOS Y RELACIONES 74
2.2.3.1 Descripción de Entidades /atributos 74
2.2.3.1.1 Evaab 74
2.2.3.1.2 Evaab_Estado 75
2.2.3.1.3 Evaab_Preguntas 76
2.2.3.1.4 Evaab_Pistas 76
2.2.3.1.5 Evaab_Referencias 76
2.2.3.1.6 Evaab_Respuestas 77
2.2.3.1.7 Evaab_Tipocuestionario 78
2.2.3.1.8 Evaab_Tipopregunta 78
2.2.3.1.9 Evaab_Cue_Estu 79
2.2.3.1.10 Evaab_Opciones 79
2.2.3.1.11 Evaab_Resoluciones 80
2.2.3.2 Definición de Relaciones 80
2.2.3.2.1 Rel_Eva_Eva_Cue_Estu 81
2.2.3.2.2 Rel_Eva_Estu_Cue_Est 81
2.2.3.2.3 Rel_Eva_Eva_Pre 81
2.2.3.2.4 Rel_Eva_Pre_Pis 81
2.2.3.2.5 Rel_Eva_Pre_Ref 81
2.2.3.2.6 Rel_Eva_Pre_Res 82
2.2.3.2.7 Rel_Eva_Tipcue_Eva 82
2.2.3.2.8 Rel_Eva_Tippre_Pre 82
2.2.3.2.9 Rel_Eva_Est_Opc 82
2.2.3.2.10 Rel_Eva_Pre_Opc 82
2.2.3.2.11 Rel_Eva_Eva_Reso 82
2.2.3.2.12 Rel_Eva_Res_Opc 83
2.2.3.3 Modelo Entidad – Relación 83
2.2.3.4 Modelo Físico 85
2.3 DISEÑO DE INTERFACES 86
2.3.1 GENERALIDADES 86
2.3.2 ESTANDARIZACION 86
2.3.2.1 Diseño del Estándar de EVAAB 86
2.3.2.2 Estándar de Moodle 87
2.3.3 DEFINICIÓN DE INTERFACES 88
2.3.3.1 Evaab_Principal 88
2.3.3.2 Evaab_Cuestionario 88
2.3.3.3 Evaab_Pregunta 89

2.3.3.4 Evaab_Respuesta 90
2.3.3.5 Evaab_Pista 91
2.3.3.6 Evaab_Referencia 92
2.3.3.7 Evaab_Resolucion 93
2.3.3.7.1 Evaab_Resolucion_Pregunta_Respuesta 94
2.3.3.7.2 Evaab_Resolucion_Pistas 96
2.3.3.7.3 Evaab_Resolucion_Referencias 97
2.3.3.8 Evaab_Progreso 98
2.3.3.9 Evaab_Calificación 99
2.3.3.10 Evaab_Informe 99
2.3.3.11 Evaab_Ayuda 100

CAPÍTULO III 101
3. IMPLEMENTACIÓN Y PRUEBAS 101
3.1 JUSTIFICACIÓN DEL AMBIENTE DE DESARROLLO 101
3.1.1 AMBIENTE DE DESARROLLO 101
3.1.1.1 Definición de Elementos 101
3.1.1.1.1 LMS Moodle 101
3.1.1.1.2 Servidores Web 101
3.1.1.1.3 Base de Datos y Herramienta de Administración 102
3.1.1.1.4 Herramienta de Diseño 102
3.1.1.1.5 Herramienta para el Desarrollo con PHP 103
3.1.1.1.6 Herramienta para seguimiento de XP 103
3.1.1.1.7 Utilitarios 103
3.1.2 JUSTIFICACIÓN 104
3.2 IMPLEMENTACIÓN DEL MÓDULO DE EVALUACIÓN 104
3.2.1 ESTÁNDAR DE CODIFICACIÓN PARA EVAAB 104
3.2.2 DEFINICIÓN DE EVAAB 105
3.2.2.1 Estructura de EVAAB 106
3.2.2.2 Capas de EVAAB 107
3.2.2.3 Definición de Componentes Complementarios a EVAAB 108
3.2.2.3.1 XAJAX 109
3.2.2.3.2 CPdf y Cezpdf 109
3.2.2.4 Definición del CORE de EVAAB 110
3.2.2.4.1 Clases Modelo 111
3.2.2.4.2 Clases Controladoras 112
3.2.2.4.3 Clases Vista 113
3.2.3 VERSIONAMIENTO EN EVAAB 115
3.2.4 COMUNICACIÓN CON EL USUARIO 116
3.3 PRUEBAS 118
3.3.1 DEFINICIÓN DE AMBIENTE DE PRUEBAS 118
3.3.1.1 Definición de Curso 118
3.3.1.2 Definición de Casos de Prueba 119
3.3.1.2.1 Casos Prueba Profesor 119
3.3.1.2.1.1 Gestionar Cuestionario 119
3.3.1.2.1.2 Gestionar Preguntas 122
3.3.1.2.1.3 Gestionar Respuestas 125
3.3.1.2.1.4 Gestionar Pistas 127
3.3.1.2.1.5 Gestionar Referencias 130

3.3.1.2.2 Casos Prueba Estudiante 132
3.3.1.2.2.1 Resolución Cuestionario 132
3.3.2 RESULTADOS 136
3.3.2.1 Cuestionario General 137
3.3.2.2 Resolución Cuestionario 137
3.3.2.3 Generación Calificación 138
3.3.2.3 Generación del Informe 138

CAPÍTULO IV 140
4. INTEGRACIÓN Y EVALUACIÓN 140
4.1 INTEGRACIÓN CON MOODLE 140
4.1.1 GENERALIDADES 140
4.1.2 EVAAB DENTRO DE MOODLE 141
4.2 EVALUACIÓN DE RESULTADOS 143
4.2.1 FACTORES A EVALUAR 143
4.2.2 RESULTADOS 144
4.2.2.1 Evaab como Tabla 144
4.2.2.2 Evaab como Actividad 144
4.2.2.3 Evaab como Instancia 145
4.2.2.4 Evaab como Módulo 146

CAPÍTULO V 147
5. CONCLUSIONES Y RECOMENDACIONES 147
5.1 CONCLUSIONES 147
5.1.1 METODOLOGÍA DESARROLLO 147
5.1.2 EVAAB 147
5.2 RECOMENDACIONES 148

BIBLIOGRAFIA 150
ANEXOS 1
MANUAL DE INSTALACIÓN DE EVAAB 1
MANUAL DE USUARIO DE EVAAB 3

INDICE DE FIGURAS

CAPÍTULO I

Fig. 1. 1: Estructura de Moodle V.1.6.3 8
Fig. 1. 2: Ingeniería de Software Tradicional [17] 27
Fig. 1. 3: XP (Extreme Programming) [17] 27
Fig. 1. 4: Planificación de las Iteraciones 35
Fig. 1. 5: Desarrollo de la Iteración I 37
Fig. 1. 6: Desarrollo Iteración II 38
Fig. 1. 7: Desarrollo Iteración III 38
Fig. 1. 8: Gestión Cuestionario 40
Fig. 1. 9: Gestión de Preguntas 40
Fig. 1. 10: Gestión de Respuestas 41
Fig. 1. 11: Gestión de Pistas 41
Fig. 1. 12: Gestión de Referencia 42
Fig. 1. 13: Presentación Elementos Ayuda 43
Fig. 1. 14: Evaluar Cuestionario 43
Fig. 1. 15: Seguimiento Resolución Preguntas 44
Fig. 1. 16: Mostrar Progreso 44
Fig. 1. 17: Generación informe proceso 45

CAPÍTULO II

Fig. 2. 1: Arquitectura EVAAB 48
Fig. 2. 2: Tarjeta CRC [21] 53
Fig. 2. 3: Modelo CRC 55
Fig. 2. 4 : Modelo CRC Gestión de Cuestionario 56
Fig. 2. 5: Modelo CRC Gestión de Preguntas 57
Fig. 2. 6: Modelo CRC Gestión de Respuestas 57
Fig. 2. 7: Modelo CRC Gestión de Pistas 58
Fig. 2. 8: Modelo CRC Gestión de Referencia 58
Fig. 2. 9: Modelo CRC Presentar Elementos de Ayuda 59
Fig. 2. 10: Modelo CRC Evaluación del Cuestionario 60
Fig. 2. 11: Modelo CRC Seguimiento de Resolución de 61
Fig. 2. 12: Modelo CRC Mostrar Progreso 61
Fig. 2. 13: Modelo CRC Generación de Informe de Progreso 62
Fig. 2. 14: Modelo CRC para Iteración 1 63
Fig. 2. 15: Modelo CRC para Iteración 2 63
Fig. 2. 16: Modelo CRC para Iteración 3 64
Fig. 2. 17: Diagrama Activad Gestión Cuestionario 65
Fig. 2. 18 : Diagrama de Activad Gestión Preguntas 66
Fig. 2. 19: Diagrama de Actividad Gestión Respuestas 66
Fig. 2. 20: Diagrama de Actividad Gestión de Pistas 67
Fig. 2. 21: Diagrama de Actividad Gestión de Referencias 68
Fig. 2. 22: Diagrama de Actividad Presentar Elementos Ayuda 69
Fig. 2. 23: Diagrama de Actividad Evaluar Cuestionario 69
Fig. 2. 24: Diagrama Actividad Seguimiento Resolución Preguntas 70

Fig. 2. 25: Diagrama de Actividad Mostrar Progreso..... 70
Fig. 2. 26: Diagrama de Actividad Generación Informe Proceso 71
Fig. 2. 27: Modelo Lógico EVAAB 84
Fig. 2. 28: Modelo Físico EVAAB..... 85
Fig. 2. 29: Pantalla Principal 88
Fig. 2. 30: Evaab_Cuestionario 89
Fig. 2. 31: Pantalla Ingreso de Pregunta..... 89
Fig. 2. 32: Pantalla Grabado de Pregunta 90
Fig. 2. 33: Pantalla Ingreso de Respuesta..... 90
Fig. 2. 34: Pantalla Grabado de Respuesta 91
Fig. 2. 35: Pantalla Ingreso de Pista 91
Fig. 2. 36: Pantalla Grabado de Pista..... 91
Fig. 2. 37: Pantalla Ingreso de Referencia..... 92
Fig. 2. 38: Pantalla Grabado de Referencia..... 92
Fig. 2. 39: Pantalla Inicial Resolución..... 93
Fig. 2. 40: Mensaje de Validación Perfil..... 93
Fig. 2. 41: Pantalla Resolución Cuestionario 94
Fig. 2. 42: Pantalla Resolución Pregunta..... 95
Fig. 2. 43: Pantalla Pregunta Marcada..... 95
Fig. 2. 44: Mensaje de Pregunta Correcta 96
Fig. 2. 45: Mensaje de Pregunta Incorrecta..... 96
Fig. 2. 46: Mensaje de Pregunta Incorrecta..... 96
Fig. 2. 47: Pantalla Resolución Pistas 97
Fig. 2. 48: Mensaje de Pistas No existentes 97
Fig. 2. 49: Pantalla Resolución Referencias..... 98
Fig. 2. 50: Mensaje de Referencias No existentes..... 98
Fig. 2. 51: Pantalla de Progreso..... 99
Fig. 2. 52: Pantalla de Calificación 99
Fig. 2. 53: Pantalla de Calificación 100
Fig. 2. 54: Pantalla de Ayuda 100

CAPÍTULO III

Fig. 3. 1: Estructura de EVAAB..... 106
Fig. 3. 2: Evaab dentro de Moodle 107
Fig. 3. 3: Ejemplo de Clase Modelo..... 112
Fig. 3. 4: Ejemplo de Clase Controladora 113
Fig. 3. 5: Clase Vista Moodle view.php 114
Fig. 3. 6: Ejemplo de Clase Vista EVAAB 114
Fig. 3. 7: Ejemplo de Clase Modelo EVAAB- AJAX..... 115
Fig. 3. 8: Pantalla Comunicación Líder Proyecto..... 117
Fig. 3. 9: EVAAB dentro de Moodle Traker..... 117
Fig. 3. 10: Pantalla Inicial Moodle – E.P.N..... 118
Fig. 3. 11: Pantalla Comprobación Cuestionario Ingreso..... 137
Fig. 3. 12: Pantalla Comprobación Resolución..... 138
Fig. 3. 13: Pantalla Comprobación Calificación 138
Fig. 3. 14: Pantalla Comprobación Informe 139

CAPÍTULO IV

Fig. 4. 1: Pantalla Tablas EVAAB 144
Fig. 4. 2: Pantalla Actividad 145
Fig. 4. 3: Pantalla Eliminación Instancia 145
Fig. 4. 4: Pantalla Módulo 146

ANEXOS

Fig. A 1: Agregando Actividad EVAAB 3
Fig. A 2: Opciones Profesor 3
Fig. A 3: Elementos Interfaz Profesor 4
Fig. A 4: Cuestionario Grabado Exitosamente 5
Fig. A 5: Grabada de Cuestionario No Exitosa 6
Fig. A 6: Creación de Pregunta Errada 7
Fig. A 7: Grabada de Pregunta No Exitosa 7
Fig. A 8: Pregunta Grabada Exitosamente 8
Fig. A 9: Grabada de Respuesta No Exitosa 9
Fig. A 10: Respuestas Grabada Exitosamente 9
Fig. A 11: Grabada de Pista No Exitosa 10
Fig. A 12: Pistas Grabadas Exitosamente 10
Fig. A 13: Grabada de Referencia No Exitosa 11
Fig. A 14: Referencias Grabadas Exitosamente 11
Fig. A 15: Pantalla Recursos Referencias 12
Fig. A 16: Pantalla Final Cuestionario 12
Fig. A 17: Elementos Interfaz Estudiante 13
Fig. A 18: Pantalla Retroalimentación Instantánea 14
Fig. A 19: Pantalla Pregunta Postergada 14
Fig. A 20: Pantalla Visualización Pistas 15
Fig. A 21: Pantalla Visualización Referencias 15
Fig. A 22: Mostrando Recurso Referencia 16
Fig. A 23: Pantalla de Progreso Resolución 16
Fig. A 24: Pantalla Preliminar Generación Calificación 17
Fig. A 25: Pantalla Generación Calificación 17
Fig. A 26: Pantalla Generación Informe 18

ÍNDICE DE TABLAS

CAPITULO I

Tabla 1. 1: Tabla de LMS [4] 4
Tabla 1. 2: Tabla de Estándares [6] 5
Tabla 1. 3: RUP (Rational Unified Process)..... 19
Tabla 1. 4: MSF (Microsoft Solution Framework)..... 20
Tabla 1. 5: XP (Extreme Programming)..... 21
Tabla 1. 6: Diferencias entre metodologías Ágiles vs. Fuertes [16]..... 22

CAPITULO III

Tabla 3. 1: Caso de Prueba Guardar Cuestionario..... 120
Tabla 3. 2: Caso de Prueba Actualizar Cuestionario 121
Tabla 3. 3: Caso de Prueba Eliminar Cuestionario..... 122
Tabla 3. 4: Caso de Prueba Guardar Pregunta..... 123
Tabla 3. 5: Caso de Prueba Actualizar Pregunta 124
Tabla 3. 6: Caso de Prueba Eliminar Pregunta..... 125
Tabla 3. 7: Caso de Prueba Guardar Respuesta..... 126
Tabla 3. 8: Caso de Prueba Actualizar Respuesta 127
Tabla 3. 9: Caso de Prueba Eliminar Respuesta..... 127
Tabla 3. 10: Caso de Prueba Guardar Pista 128
Tabla 3. 11: Caso de Prueba Actualizar Pista..... 129
Tabla 3. 12: Caso de Prueba Eliminar Pista 130
Tabla 3. 13: Caso de Prueba Guardar Referencia..... 131
Tabla 3. 14: Caso de Prueba Actualizar Referencia 131
Tabla 3. 15: Caso de Prueba Eliminar Referencia..... 132
Tabla 3. 16: Caso de Prueba Seguimiento Preguntas 133
Tabla 3. 17: Caso de Prueba Visualización de Pistas 134
Tabla 3. 18: Caso de Prueba Visualización de Referencias 135
Tabla 3. 19: Caso de Prueba Mostrar Progreso 136
Tabla 3. 20: Caso de Prueba Generar Informe 136

RESUMEN

El desarrollo del módulo de Evaluación Abierto ***EVAAB*** es un proyecto con licencia ***GPL*** el cual tiene como principal objetivo dar a conocer a las comunidades internacionales de desarrollo de ***Software Libre*** **que el** ***Ecuador*** puede aportar a su crecimiento.

El módulo ***EVAAB*** forma parte del LMS (Learning Management System) ***Moodle*** el cual permite realizar evaluaciones con retroalimentación a través de la utilización de elementos de ayuda como son las pistas y referencias. El desarrollo de ***EVAAB*** se encuentra estructurado de la siguiente manera:

CAPÍTULO I

Trata acerca de los fundamentos del E-learning sus ventajas, desventajas para luego estudiar a ***Moodle*** en sus características fundamentales como el ***LMS*** a utilizar. Posteriormente se realiza una comparativa entre las metodologías más comunes tomando en cuenta los procedimientos aplicables a Moodle con el objetivo de adoptar una de ellas.

Estudiamos a ***XP*** como la metodología a adoptar para luego seguir con la definición del módulo de evaluación ***EVAAB*** como solución a una carencia de Moodle. Terminando este capítulo con el juego de la planificación propuesta por ***XP***.

CAPÍTULO II

Trata el tema del diseño de ***EVAAB***, se establece la arquitectura con la cual va a trabajar, se elaboran los diferentes elementos propuestos por ***XP*** para la etapa del diseño como son las tarjetas ***CRC y la Refactorización***. Posteriormente se elabora el diseño de la base de datos junto con las definiciones de tablas, columnas y relaciones terminando con diseño de interfaces todo en base a las especificaciones estándar de Moodle.

CAPÍTULO III

Trata acerca de implementación y pruebas realizados para la consecución de ***EVAAB***, se estudia en detalle la arquitectura de Moodle y se define la arquitectura de ***EVAAB*** junto con los elementos, procedimientos utilizados en su desarrollo. Se definen los casos de pruebas a realizar y se concluye con la evaluación de los resultados obtenidos.

CAPÍTULO IV

Trata acerca de la integración de ***EVAAB dentro de Moodle***, los pasos y consideraciones a adoptar conjuntamente con los resultados esperados en la integración.

CAPÍTULO V

Se establece las conclusiones y recomendaciones acerca del proyecto ***EVAAB*** tomando en cuenta tanto la metodología como la filosofía utilizada.

PRESENTACIÓN

Actualmente la tendencia a utilizar ***gestores de aprendizaje, aulas virtuales*** para el manejo del aprendizaje está popularizándose cada vez más a tal extremo que hoy en día estos mecanismos se encuentran arraigados y difundidos en muchos sectores sociales como son las universidades, las empresas etc.…, en los cuales el uso de estas herramientas representan un punto fundamental en el desarrollo de sus organizaciones gracias a las características que poseen entre las cuales se puede destacar la independencia de tiempos en la seguimiento de un curso, la utilización de multiplicidad de recursos conjuntamente con el desarrollo y retroalimentación en conjunto etc.

Tomando en cuenta estas referencias se comenzó a investigar cuales son las herramientas de ***Software Libre*** que tienen mayor aceptación a nivel internacional en el manejo del aprendizaje y se concluyó que la mejor elección era ***Moodle***.

Se realizó el estudio de ***Moodle*** con la finalidad de crear un módulo adicional que permitiera gestionar las evaluaciones utilizando la retroalimentación como un concepto clave en el aprendizaje; así nació el módulo de evaluación ***EVAAB***, el mismo que cumple con la filosofía manejada por ***Moodle,*** es decir el Software Libre.

Actualmente la tendencia a utilizar ***Software Libre*** en los ámbitos educativos, financieros y económicos está alcanzando popularidad en Latinoamérica, tal es el caso de que en países como Brasil, Venezuela, Argentina, Chile se ha dado una difusión muy buena en cuanto la utilización de alternativas ***Open Source*** frente a software comercial.

Ecuador no es la excepción, actualmente con la adopción del ***Software Libre*** como política de Estado esta tendencia a ganado espacio; tal es el caso de que hoy por hoy existen proyectos innovadores como ejemplo se puede citar a ***Machangara Soft,*** el cual es un proyecto del Municipio de Quito que tiene como objetivo desarrollar software de calidad utilizando entre sus herramientas al Software Libre como una alternativa en el proceso.

Para concluir se puede decir que el objetivo de este trabajo es contribuir a un proyecto de desarrollo internacional de ***Software Libre***, específicamente ***Moodle***, desde una perspectiva de cambio radical al demostrar que es posible desarrollar para el mundo y contribuir a la educación que tanta falta hace a los pueblos. Así nació ***EVAAB*** como un proyecto orgullosamente ***Ecuatoriano!***

CAPÍTULO I

1. ANÁLISIS

1.1 DEFINICIÓN DEL PROBLEMA

1.1.1 E-LEARNING

1.1.1.1 Fundamentos

El e-learning está considerado como un sistema de aprendizaje a distancia basado en el uso del Internet cuyas características principales radican en la comunicación e interacción entre los participantes, conjuntamente con la difusión de la información a través de la utilización de medios estáticos e interactivos. [1]

Además permite reducir costos de aprendizaje capacitando a un sinnúmero de usuarios en conjunto, de uno a miles, es decir se implementa la actividad grupal característica en la educación a distancia.

La diversificación de contenido, el seguimiento de los estudiantes son características sumamente imprescindibles en un e-learning, por lo que siempre será necesario contar con elementos innovadores que llamen la atención a los estudiantes conjuntamente con elementos estadísticos que permitan cuantificar el desempeño en el desarrollo de los cursos. “Entre los componentes fundamentales de un e-learning se puede mencionar los siguientes” [1]:

- Desarrollo de Contenidos
- Gestión de la Plataforma de Aprendizaje
- Servicios de Apoyo

1.1.1.2 Ventajas

Entre las ventajas que presenta un e-learning al momento del aprendizaje, que hacen de esté un gestor de conocimiento muy interesante se puede mencionar [1]:

- Elimina las distancias físicas a través de la utilización de herramientas como correo electrónico, foros, chat entre los participantes facilitando la comunicación y colaboración entre ellos.
- Las formas de enseñanza pueden ser realizadas alternadamente, tanto individual como grupal.
- Facilita la predisposición del tiempo de los participantes, ya que no existe horarios para seguir un curso, éste se lo puede tomar cuando se disponga de un tiempo favorable.
- La interacción entre los estudiantes, profesores es una característica muy interesante que facilita la retroalimentación continua de ambas partes.
- Presenta una disposición de contenido muy variado mediante la utilización de recursos como videos, páginas Web, imágenes, animaciones, sin exceptuar los recursos adicionales disponibles en el Internet.
- Capacitación a gran cantidad de usuarios con los mismos recursos y metodologías de enseñanza.

1.1.1.3 Desventajas

A pesar de que la tecnología avance tan rápidamente y que el uso creciente del Internet ha permitido dar pasos favorables en todos los ámbitos. En el aprendizaje en línea siempre existirán ciertos limitantes que no permitirán explotar todas las características involucradas en un aprendizaje offline, las cuales representan desventajas en el e-learning. Entre éstas se puede mencionar:

- La tecnología no suplirá el contacto cara a cara ni tampoco el computador se podrá comparar a una aula de aprendizaje.
- La soledad a la que está sumergida el alumno al momento de recibir un curso es un factor preocupante ya que las clases son recibidas de una forma independiente.
- La interacción continua con el computador puede provocar cansancio o incomodidad para aprender, tomando en cuenta la culturización informática de la cual no todos estamos adaptados.
- La necesidad de un equipo computacional y los conocimientos informáticos supera mucha de las veces las expectativas al momento de enrolarse en uno de estos cursos.

1.1.1.4 Plataformas E- learning

Para el desarrollo de un e-learning se necesita de una plataforma sobre la cual trabajar "razón por la cual a partir del año de 1990 se crearon plataformas de aprendizaje denominadas con el nombre de ***LMS (Learning Management System)***" [2] la mismas que serán las encargadas de gestionar todas las actividades relacionadas con el aprendizaje, como la distribución de contenidos formativos, el manejo de herramientas de comunicación, herramientas de seguimiento y monitoreo de estudiantes etc. "Entre las herramientas que debería tener una plataforma e-learning para garantizar el buen desenvolvimiento de un curso están las siguientes" [3]:

- Herramientas para la distribución de contenidos
- Herramientas para comunicación y colaboración
- Herramientas para seguimiento y evaluación
- Herramientas para administración y asignación de servicios.

Actualmente existen varios ***LMS*** (***Learning Management System***) tanto propietarios como de libre de difusión, ***Software Libre***, diferenciándose unos de otros por factores como el tamaño del sistema, la filosofía de

uso que manejan y la posibilidad de integración con otros LMS. A continuación en la tabla 1.1 se presenta una lista de algunos de ellos:

LMS	***Características***
SumTotal	Parecido a un ERP integrado
Claroline	Sistema Open Source
WebCT	Sistema con orientación universitaria
Blackboard	Sistema con orientación universitaria
Dokeos	Sistema Open Source
Moodle	Sistema Open Source

Tabla 1. 1: Tabla de LMS [4]

1.1.1.5 Estándares E-Learning

Uno de los mayores problemas en las nuevas tecnologías de información y comunicación aplicadas a la educación es la falta de estándares que garanticen la transparencia, interoperabilidad, accesibilidad, adaptabilidad, durabilidad, productividad y reusabilidad entre los sistemas de aprendizaje de modo que no sea un impedimento la transición entre un sistema y otro al momento de cumplir con el objetivo primordial de aprender. [5]

Entre los estándares que se persigue alcanzar en un e-learning como requisitos fundamentales para tener éxito en la era del conocimiento y a futuro, se puede mencionar estándares en los siguientes aspectos [5]:

- Metadatos de Contenidos
- Secuencia de Contenidos
- Interoperabilidad de Preguntas y Test
- Perfil de Alumnos/Profesores
- Interacción en Tiempo de Ejecución

Actualmente existen algunas organizaciones involucradas en el proceso de estandarización e-learning las cuales se las puede apreciar en la Tabla 1.2:

Organización	***Estándar***
ADL	SCORM
AICC	CMI
IEEE	LOM
LTSC	LTSA
IMC	IMS

Tabla 1. 2: Tabla de Estándares [6]

Entre los aspectos comunes que concuerdan estos estándares están: ***la separación entre la plataforma y el contenido***, utilizando dos caminos [7]:

- El primer camino tiene que ver con ***las especificaciones de empaquetamiento***, lo cual define la forma de empaquetar el contenido de tal forma que puede ser reconocido por el LMS y cargado en el sistema.

- El segundo tiene que ver con ***las especificaciones de test*** que consisten en la forma en la que el contenido debe registrar las actividades de los alumnos junto con las actividades de evaluación, es decir un API que permita la comunicación entre el contenido y el LMS.

1.1.2 MOODLE

1.1.2.1 Fundamentos

Moodle[1], **Modular Object Oriented Dynamic Learning Environment,** es un sistema de gestión de enseñanza (LMS) que se distribuye bajo la licencia ***GNU GPL***[2], el cual inició su desarrollo en USA en la universidad **Curtin University of Technology.** Su estilo de aprendizaje se basa en la pedagogía de constructivismo social, la cual afirma que el conocimiento se construye en la mente del estudiante cuando interactúa con material de aprendizaje, cuando aporta con materiales para otros estudiantes, cuando interactúa con los mismos acerca del material en curso en lugar de ser transmitido sin cambios a partir de libros y enseñanzas. Siendo la **diferencia principal la discusión en lugar de la lectura que caracteriza a las clases comunes [8]**.

La frase que define a Moodle en su ambiente de aprendizaje son las "**experiencias de aprendizaje en línea**" más que "**cursos en línea**" ya que éste último denota un sinnúmero de páginas Web, imágenes, animaciones todo de una forma estática y secuencial mientras que **las experiencias de aprendizaje en línea** connota una mayor actividad es decir interacciones entre los estudiantes, adoptar el papel de profesores y alumnos, utilizar todos los recursos interactivos como chat, foro, entre otras, y no seguir necesariamente una secuencia de enseñanza [8].

1.1.2.2 Características

Entre las características que presenta ***Moodle*** se puede mencionar las siguientes:

- ***Moodle*** cuenta recursos ***estáticos e interactivos*** para el

[1] Moodle – http://moodle.org

[2] GPL - http://es.wikipedia.org/wiki/GNU_GPL

desarrollo de los cursos. Entre estos se tiene los siguientes: páginas de texto, páginas Web, tareas, preguntas, diario, lecciones, paneles de discusión, entre otras.

- ***Moodle*** cuenta con actividades de aprendizaje en donde los estudiantes interactúan unos con otros fortaleciendo sus conocimientos en conjunto y desarrollando su parte comunicativa, entre estas actividades se puede mencionar las siguientes: chat, foros, glosarios, wikis, etc.

- ***Moodle*** maneja un conjunto de estándares en lo que respecta a todo su flujo de trabajo, entre estos están: estándares de código, documentación, interfaz etc.

- ***Moodle*** facilita la edición de los archivos de lenguajes directamente, es decir se puede renombrar los textos que aparecen en cada una de las interfaces gracias a la característica de ***internalización*** con la que cuenta

- ***Moodle*** cuenta con el servicio de ***RSS (Really Simple Syndication)***, en el desenvolvimiento de sus cursos.

- ***Moodle*** cuenta con una cantidad de filtros muy interesantes que son de mucha ayuda en el desarrollo de un curso, como los filtros automáticos de enlace, filtros matemáticos, etc.

- ***Moodle*** maneja algunos estándares e-learning como ***SCORM/AICC***[3] los cuales nos permiten interoperabilidad, accesibilidad y reusabilidad de contenido de aprendizaje basado en la Web entre los diferentes LMS.

[3] SCORM http://es.wikipedia.org/wiki/SCORM

1.1.2.3 Estructura de Moodle

Moodle presenta una organización muy buena en cuanto a su estructura, la misma que está definida jerárquicamente en varios directorios de acuerdo a sus funcionalidades. La estructura que se presenta a continuación en la Fig.1.1 es la que tiene ***Moodle*** en su versión ***1.6.3.***

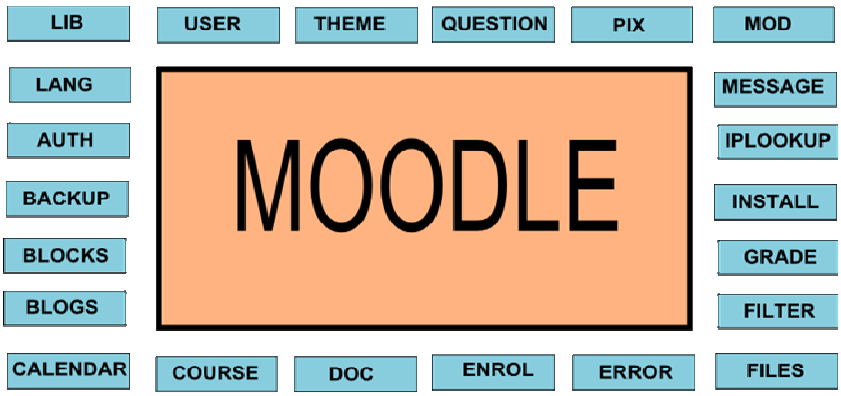

Fig. 1. 1: Estructura de Moodle V.1.6.3

En el directorio ***/lib*** encontramos las librerías necesarias para el manejo de ***Moodle,*** entre éstas tenemos a:

- ***moodlelib.php.-*** la cual es la ***librería principal de Moodle***, donde se maneja las funciones de propósito general por ejemplo aquí se encuentra funciones Web para el manejo de los parámetros, funciones de fechas, funciones de mantenimiento, seguridad etc.

- ***datalib.php.-*** la cual es una librería para la manipulación de la base de datos, donde se encuentran funciones para manejar tanto base de datos transaccionales como no transaccionales. Cabe mencionar que ***Moodle*** por el mismo hecho de manejar una filosofía libre fue diseñado para trabajar con DBMS de esta misma índole como ***MySQL*** y ***Postgresql***[4]**.**

[4] Postgresql - http://www.postgresql.org/

- ***weblib.php.-*** la cual es una librería para la manipulación de la parte Web, es decir todo lo que compete con la generación y manejo de HTML. Aquí se encuentra funciones de formateo de texto, alineaciones, manejo de emails, generación de elementos HTML, etc.

- ***adminlib.php.-*** la cual es una librería utilizada para la administración de Moodle, por lo que solo usuarios administradores deberían utilizarla.

En el directorio ***/lang*** se encuentran los archivos de lenguajes con los que cuenta ***Moodle,*** en cada uno de estos archivos se tiene las definiciones de las cadenas a utilizarse en cada módulo dentro de Moodle.

En el directorio ***/auth*** se encuentran los archivos relacionados con la autentificación dentro de ***Moodle***. Se tiene diferentes métodos de autentificación los cuales utilizan un nombre de usuario y contraseña como patrón, entre estos están: ***Email, Manual, Ldap, Imap, Pop3, Nntp, Db, Ninguna***

El directorio ***/backup*** contiene los archivos de respaldo/recuperación para la opción de administración de ***Moodle***, los cuales son utilizados en caso de que se produzca errores.

En el directorio /***blocks*** se encuentran los archivos referentes a los blocks creados dentro de ***Moodle*** así como la información necesaria para la creación de nuevos blocks, entendiéndose por blocks a los contenedores de elementos que son utilizados para mostrar alguna información.

En el directorio /***blogs*** se encuentran los archivos relacionados con el manejo de blogs, el mismo que fue creado basado en el sistema de blogs

Simplog. ***El Moodle Blog*** como se lo llama dentro de ***Moodle*** es independiente de un curso, por lo que cada usuario puede crear su propio blog si lo requiere.

En el directorio ***/calendar*** se encuentran los archivos relacionados con el manejo del tiempo y fechas dentro de ***Moodle***.

En el directorio ***/course*** se encuentran los archivos necesarios para el manejo de un curso dentro de ***Moodle***, por lo cual es un directorio fundamental dentro de Moodle.

En el directorio ***/doc*** se encuentra la documentación básica necesaria para el manejo de ***Moodle***, como su instalación, configuración etc.

En el directorio ***/enrol*** se encuentran los archivos necesarios para el enrolamiento dentro de un curso en ***Moodle***, es decir la forma de inscripción dentro del mismo. Los cuales pueden ser gratuitos o pagados.

En el directorio ***/error*** se encuentran los archivos necesarios para el manejo de errores generales dentro de ***Moodle*** como las opciones de administración. Ya que cada módulo maneja sus archivos de error.

En el directorio ***/files*** se encuentran los archivos que son subidos a ***Moodle***, para ser utilizados como recursos dentro de un curso. Estos son organizadas en carpetas identificadas por el ID del curso en cuestión.

El directorio ***/filter*** contiene todos los archivos necesarios para el manejo de filtros dentro de ***Moodle***. Los cuales permiten agilitar el proceso de navegación e interacción dentro de Moodle.

En el directorio ***/grade*** se encuentran los archivos necesarios para el manejo de las calificaciones de las diferentes actividades dentro de ***Moodle*** como lecciones, pruebas etc.

En el directorio /***install*** se encuentran los archivos de instalación, los cuales son creados automáticamente luego de la instalación de ***Moodle*** en base a los archivos de lenguaje estándar.

En el directorio /***iplookup*** se encuentran los archivos para el manejo de la información de las direcciones IP.

En el directorio ***/login*** se encuentran los archivos necesarios para controlar el ingreso de los diferentes usuarios dentro de ***Moodle***, entre estos se tiene archivos para cambiar la contraseña, para regenerar una nueva en caso de que nos olvidamos, etc.

En el directorio ***/message*** se encuentran los archivos necesarios para el manejo de los diferentes mensajes de las actividades realizadas dentro de ***Moodle***.

El directorio ***/mod*** se podría decir que es el directorio más importante dentro de ***Moodle*** ya que contiene todos los módulos con sus diferentes archivos de configuración.

En el directorio ***/pix*** se encuentran las imágenes generales de ***Moodle***, exceptuando las imágenes de cada módulo las cuales son manejadas independientemente por cada uno.

En el directorio ***/question*** se encuentran los archivos necesarios para la manejo de las preguntas como formatos, tipos, opciones de importación, exportación etc.

En el directorio /***theme*** se encuentran todos los archivos de configuración necesarios para el manejo de los temas de presentación con los que cuenta ***Moodle***

El directorio ***/user*** contiene todos los archivos necesarios para el manejo de los usuarios dentro de ***Moodle***.

1.1.3 PROBLEMA

Moodle actualmente no cuenta con un módulo de evaluación retroalimentativo que permita a los estudiantes conocer cuales son sus errores y aciertos durante la resolución de un test o lección; ni tampoco cuenta con recursos de ayuda que permitan garantizar el acierto de cada pregunta como ***son las pistas y referencias***.

Moodle dispone actualmente de un módulo de evaluación múltiple, el mismo que para crear un test o lección utiliza un conjunto de preguntas de diferente tipo con predisposición de tiempo, revisión y corrección parametrizables pero todo de una manera estática y en cierta forma cerrada ya que para seguir con la resolución de una pregunta ésta tiene que estar correcta.

Además no existe una forma dinámica de resolverlas ni tampoco presenta un estado que nos muestre como estamos actualmente con el desenvolvimiento de un curso en un punto determinado del mismo.

1.2 PLANTEAMIENTO DE SOLUCIONES

1.2.1 GENERALIDADES

Teniendo claro el enfoque hacia donde nos dirigimos y tomando en cuenta ciertos factores que podrían limitar, retardar, o ayudar al proceso de desarrollo ***como la arquitectura en la cual se desenvuelve Moodle, la experiencia y dominio en el lenguaje con el que fue concebido Moodle, y la difusión de nuevas tecnologías emergentes***; hemos creído conveniente estudiar otras

alternativas de solución con el único afán de encontrar la mejor propuesta que nos brinde los mayores beneficios.

Cabe destacar que estos beneficios no involucran que se cambie el enfoque de ***Moodle***, es decir su arquitectura, características que hacen de él un LMS muy interesante y eficiente se mantienen. Simplemente se busca algo nuevo, algo innovador que le de un toque diferente manteniendo siempre la filosofía del ***Software Libre.***

Cabe destacar que las alternativas que se analizarán a continuación se encuentran dentro el esquema tecnológico aceptado por Moodle en cuanto al desarrollo de nuevos módulos; las cuales se analizarán y de ser el caso se tomarán en cuenta en nuestro desarrollo.

1.2.2 TECNOLOGIA MOODLE

Moodle gracias a la contribución de los desarrolladores de todo el mundo está mejorando constantemente sus versiones, usando tecnologías actuales, mejorando las características de sus herramientas a utilizar, lenguajes etc.

En cuanto a tecnologías Moodle actualmente utiliza **AJAX** para el desarrollo de sus elementos, lo cual tuvo su inició a partir de su distribución 1.6 con el "proyecto ***Chamaleon"***[5]**.**

En cuanto a las herramientas utilizadas por ***Moodle*** en su entorno de desarrollo, éste ya utiliza características actuales de las versiones de ***PHP y MySQL.*** Tal es el caso de que actualmente la versión ***Moodle 1.8***[6] ya soporta las versiones de ***PHP*** y ***MySQL*** iguales o mayores a la ***4.3.0*** y ***4.1.16*** respectivamente.

5 Chameleon Project - http://docs.moodle.org/en/Chameleon
6 Latest Version 1.8.0 - http://download.moodle.org/

1.2.3 AJAX

1.2.3.1 Características

Entre las características más sobresalientes que presenta ***AJAX*** se tiene las siguientes [9] [10]:

- Es intuitivo ya que no se necesita realizar procesos complicados para ejecutar los eventos, es decir se olvida de los clicks.
- Presenta un refrescamiento parcial de pantalla, es decir no se envía al servidor todo el contenido de la página sino simplemente el contenido nuevo a ser actualizado. El cual se ejecuta en el servidor de manera secundaria, con lo que mejoramos el rendimiento notablemente.
- Es independientemente del lenguaje Web utilizado ya que no necesita descargarse o configurarse un plugin para comenzar a trabajar.
- La interfaz es manejada por el cliente, haciendo uso del servidor para proveer los datos únicamente, ganando rendimiento al reducir la carga sobre el servidor
- Presenta una comunicación asíncrona lo cual significa que las operaciones de consulta de datos pueden realizarse independientemente de las operaciones del usuario, sin interrumpirlas.

1.2.3.2 AJAX en Moodle

"***Moodle*** en Julio del 2006 aprobó a la librería de **Yahoo (YUI)** como la librería oficial para el uso de ***AJAX*** dentro de Moodle, sin exceptuar el uso de otras librerías que están en estudio en algunos proyectos como los siguientes:"[7]:

[7] Ajax in Moodle – http://docs.moodle.org/en/Ajax

- ***Tema Chamaleon.-*** Es un tema utilizado por Moodle que utiliza a ***YUI*** para realizar cambios en los estilos de la fachada.
- ***Proyectos de Estudiantes/ Ajax Curso Formato***.- Es un proyecto que actualmente está en desarrollo en el cual se incluyen un estudio de algunas librerías de ***AJAX*** aceptadas por ***Moodle***. En las cuales se especifica características, ventajas y desventajas para cada una con el único objetivo de encontrar la mejor opción para desarrollar un formato de curso utilizando AJAX

1.2.3.3 AJAX en PHP

En lo que respecta con ***PHP***, actualmente ***AJAX*** se utiliza como un complemento que permite mejorar la apariencia de la interfaz mejorando su navegación y manejo a través del toque dinámico que le impone.

Entre las opciones que se tiene para utilizar ***AJAX con PHP*** están desde la creación de los objetos manualmente hasta la utilización de frameworks, librerías como **AjaxCore[8], Xajax[9]** respectivamente. Las cuales facilitan el desarrollo de aplicaciones AJAX permitiéndo la generación del código Javascript.

1.2.4 SELECCIÓN DE ESTRATEGIA

Tomando en cuenta los temas anteriormente tratados y de acuerdo a las aseveraciones mencionadas se puede establecer una estrategia que permita de alguna manera encontrar la forma más óptima de llevar el desarrollo.

Respecto a la utilización ***AJAX como complemento de PHP*** se definió que se hará uso del mismo para realizar todas las mejoras posibles que brinda estas tecnologías, para lo cual utilizaremos la librería **XAJAX** tomando en cuenta

[8] Ajax Core - http://ajaxian.com/archives/ajaxcore-php-ajax-framework

que es una de las librerías aceptadas por ***Moodle*** que están en estudio. Además que presenta una documentación bien fundamentada que permitirá llevar los conceptos de la tecnología a la práctica a través del API que maneja.

En cuanto a las herramientas de desarrollo se utilizará las versiones actuales tanto del lenguaje ***PHP*** como de la base de datos ***MySQL*** en los que se utilizará las nuevas características que nos brindan. Específicamente utilizaremos la versión de ***PHP 5.0.1*** y ***MySQL 5.0.45.***

1.2.5 ANALISIS DE METODOLOGIAS

1.2.5.1 Fundamentos

Para realizar el estudio de las metodologías de desarrollo de software y elegir una de ellas para el seguimiento de nuestro proyecto primero comenzaremos analizando como está constituido ***Moodle*** metódicamente antes de realizar cualquier elección. Para seleccionar las metodologías de estudio se han tomado en cuenta el nivel de aceptación y utilización que presentan actualmente, eligiendo estas tres opciones de estudio:

- ***RUP*** (Rational Unified Process)
- ***MSF*** (Microsoft Solution Framework)
- ***XP*** (Extreme Programming)

1.2.5.2 Proceso Desarrollo de Moodle

Moodle no especifica explícitamente la metodología a seguir en su proceso de desarrollo, pero establece ciertos procedimientos para la creación de nuevos módulos y mantenimiento de los ya existentes que permiten establecer aseveraciones acerca de la metodología a utilizar. Entre estos procedimientos se puede mencionar los siguientes [11]:

---

9 Xajax - http://xajaxproject.org/

- ***Moodle*** define dos perspectivas para el desarrollo de elementos, una para nuevas funcionalidades como código nuevo denominado con el nombre ***Major Development*** y otra para desarrollo pequeños, arreglos de problemas denominados con el nombre ***Minor Development.***

- En cuanto a ***Major Development*** necesitamos tomar en cuenta las siguientes especificaciones:

  - Se debe primero verificar el ***RoadMap***[10] de Moodle para asegurar que no existe esa funcionalidad creada o que si se está trabajando en la misma.

  - Se debe crear la documentación respectiva en ***Moodle*** acerca de la funcionalidad nueva, colocar todos los detalles posibles para ayudar a su entendimiento.

  - Se debe crear un nuevo tema en los foros de Moodle, ***Using Moodle***[11]**,** para despertar el interés en la comunidad por la nueva funcionalidad obteniendo toda la retroalimentación necesaria. Cabe destacar que se debe haber creado una cuenta anteriormente.

  - Se debe crear una entrada en el ***Moodle Tracker***[12], subir las fuentes y añadir subtareas cronológicamente con el objetivo de darle seguimiento a la nueva funcionalidad. Permitiendo a la comunidad apoyar en el desarrollo de estas tareas; una vez registrada la entrada y que el proceso de revisión sea satisfactorio se la debe enlazar a un ***CVS.***

[10] RoadMap Moodle - http://docs.moodle.org/en/Roadmap
[11] Using Moodle - http://moodle.org/login/index.php

- Como paso final se tiene que corregir los ***errores*** que se vayan presentando en la nueva funcionalidad.

- En cuanto a ***Minor Development*** necesitamos tomar en cuenta las siguientes especificaciones:

  - Se debe crear una entrada en el ***Moodle Tracker*** para describir el desarrollo y actuar como un punto de enfoque para todos los debates, el mismo que puede utilizar los foros para su referencia.

  - Si se dispone de código se debería atar al ***Moodle Tracker,*** en ningún caso se debería utilizar los foros para realizar esto.

  - Por último se deber llamar la atención sobre ***el trabajo realizado en los foros de Moodle*** mencionando el número de error o por correo electrónico directamente a los desarrolladores para que sean conscientes de ello.

- Se debe tomar en cuenta que para el desarrollo de una funcionalidad nueva o arreglo de una existente se debe seguir un conjunto de estándares, guías definidos por ***Moodle*** [13] entre los que se tiene a estándares de código, base de datos, interfaces etc.

1.2.5.3 Estudio de las Metodologías

Para el estudio de las metodologías de desarrollo de software como ***RUP, MSF y XP*** se ha realizado tablas funcionales donde se puedan apreciar las características fundamentales de cada una de éstas. Lo cual se lo puede apreciar en la Tabla 1.3, Tabla 1.4, Tabla 1.5 a continuación:

---

[12] Moodle Tracker - http://tracker.moodle.org/secure/Dashboard.jspa
[13] Guidelines Moodle - http://docs.moodle.org/en/Developer_documentation#Guidelines

RUP (Rational Unified Process) – 1998 [12]	
Meta:	Asegurar la producción de un software de alta calidad que satisfaga las necesidades de los usuarios finales dentro de la planificación y presupuesto establecido
Características:	▪ ***RUP*** se centra en la arquitectura basada en componentes ▪ ***RUP*** plantea el desarrollo iterativo e incremental. ▪ ***RUP*** se centra en los casos de uso y utiliza a UML como notación de modelamiento.
Fases:	***RUP*** presenta las siguientes fases: **Inicio, Elaboración, Construcción y Transición.**
Roles:	***RUP*** define grupos de roles, agrupados por participación en actividades relacionadas. Estos grupos son: **Analistas**, **Desarrolladores, Gestores, Apoyo, Especialista en Pruebas**
Disciplinas:	***RUP*** propone una secuencia de actividades realizadas por los diferentes roles, los cuales son conocidos como **flujos o disciplinas de trabajo**. Entre estos se tiene los siguientes: ***Modelado de Negocio, Requisitos, Análisis & Diseño, Implementación, Pruebas, Despliegue, Gestión del Proyecto, Configuración y Control de Cambios, Entorno***

Tabla 1. 3: RUP (Rational Unified Process)

MSF (Microsoft Solution Framework) [13][14]	
Meta:	Proveer un framework que ayude a las organizaciones a implementar soluciones tecnológicas que excederán la visión predefinida y los objetivos del negocio.

Características:	▪ ***MSF*** más que una metodología rígida de administración de proyectos, es una serie de modelos que pueden adaptarse a cualquier proyecto de tecnología de la información. ▪ ***MSF*** es recomendado para proyectos sumamente grandes en donde todos los roles que define se encuentren plenamente identificados. ▪ ***MSF*** se presenta como un marco para un desarrollo fuertemente documentado.
Fases:	***MSF*** presenta cinco fases dentro de su metodología de desarrollo las cuales son: **Visión y Alcance, Planificación, Desarrollo, Estabilización, Implantación**
Roles:	***MSF*** maneja seis roles que corresponden a las metas principales de un proyecto, las cuales pueden estar conformados por una o varias personas siendo importante cada uno de ellos sin poder ser omitidos ninguno para un proyecto efectivo de equipo. Entre los roles se tiene los siguientes: **Product Manager, Program Management, Development, Testing, User Experience, Logistics Management**

Tabla 1. 4: MSF (Microsoft Solution Framework)

XP (Extreme Programming)[15]	
Meta:	Reducir el coste del cambio presentado comúnmente en la ingeniería de software tradicional a través de la combinación de buenas prácticas de programación y tecnología.

Características:	▪ ***XP*** define cuatro variables para cualquier proyecto de software, éstas son: ***coste, tiempo, calidad y alcance*** ▪ ***XP*** contempla un conjunto de principios y prácticas para el desarrollo rápido de software de alta calidad ▪ ***XP*** propone diseños simples ▪ ***XP*** presenta un modelo iterativo e incremental ▪ ***XP*** pone mucho énfasis en el trabajo en equipo, donde el cliente juega un papel importante.
Fases:	***XP*** define cuatro ámbitos o fases los cuales son: **Planificación, Diseño, Codificación y Pruebas**
Roles:	**XP** define un conjunto de roles los cuales son los siguientes: **Programador, Cliente, Encargado de Pruebas, Encargado de Seguimiento, Entrenador, Consultor, Gestor.**

Tabla 1. 5: XP (Extreme Programming)

1.2.5.4 Comparativa

Para comparar las metodologías de desarrollo de software ***RUP, MSF, XP*** anteriormente estudiadas se realizará tomando en cuenta la perspectiva de Moodle frente a éstas, con el objetivo de conocer cuales de los elementos de cada una de las metodologías es aplicado en un mayor grado al utilizado por Moodle. Cabe destacar que no se ha pretendido desmerecer a las metodologías consagradas como Cascada, Espiral, etc. que fueron las impulsadoras en este proceso, simplemente las metodologías fueron elegidas tomando en cuenta el momento que están atravesando y el interés del autor.

1.2.5.4.1 Generalidades

Actualmente las metodologías ágiles son el nuevo enfoque que se le están dando a los proyectos de desarrollo de software de acuerdo a características interesantes que poseen como su desarrollo incremental, cooperativo, directo y adaptativo, siendo puntos claves ***la velocidad y simplicidad.*** Lo cual es una ventaja frente a las metodologías fuertes, en las cuales las exigencias tanto de planeamiento como de control en muchos casos nos brindan resultados que estén acorde con costos en tiempo, complejidad y dinero.

Dentro de este esquema se puede decir que muchas de las veces las estrategias de desarrollo que toman las metodologías fuertes son **predictivas en lugar de adaptativas**, lo que si es contrarestado por la metodologías ágiles debido a que la comunicación dentro del equipo y fuera de él es intensiva haciendo participe al cliente todo el tiempo en el proceso de desarrollo.

Las diferencias entre las metodologías ágiles frente a las fuertes se las puede apreciar en la Tabla 1.6 a continuación:

Metodologías Agiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Tabla 1. 6: Diferencias entre metodologías Ágiles vs. Fuertes [16]

Tanto las metodologías ágiles como las fuertes presentan un marco de desarrollo interesante sobre el cual trabajar, la concepción de que metodología utilizar al momento de adoptar un proyecto dependerá de muchas características propias del proyecto a tomar entre las cuales se puede citar las siguientes:

- La magnitud del proyecto a desarrollar conjuntamente con tiempos de planificación.
- La experiencia y/o afinidad del equipo de desarrollo en alguna metodología en especial.

1.2.5.4.2 Cuadro Comparativo Respecto a Moodle

	FASES	ROLES	PROCESO
MOODLE	***Moodle*** no define una metodología estándar a utilizar pero maneja las fases de desarrollo estándar como son, Planificación, Diseño, Codificación y Pruebas pero en su forma más simple ya que su equipo de trabajo lo conforma gente de todo el mundo que no necesariamente son técnicos.	Se maneja cuatro roles puntuales en el proyecto los cuales son: Líder Proyecto, Administrador Proyecto, Desarrollador y Encargado de Pruebas.	Presenta un proceso iterativo / incremental basado en estándares, prácticas de diseño y codificación. En el cual la comunidad de personas es parte fundamental de su proceso.
RUP (Rational Unified Processs)	***RUP*** maneja cuatro fases y a su vez cada fase se subdivide en iteraciones, las fases de RUP son Inicio, Elaboración, Construcción y Transición	***RUP*** maneja una larga colección de grupos de roles, los cuales son organizados en base a sus disciplinas	Presenta un proceso iterativo / incremental con una serie de ciclos que terminan con la generación del producto para los clientes.

MSF (Microsoft Solution Framework)	***MSF*** tiene cinco fases en el proceso de desarrollo las cuales por ser un modelo prescriptivo solicita demasiada documentación, las fases de MSF son Visión y Alcance, Planificación, Desarrollo, Estabilización y Implantación	***MSF*** maneja seis roles que corresponden a las metas principales de un proyecto.	Es un proceso iterativo más fuerte en donde la iteración de ciclos contiene ciclos más pequeños, hace mucho énfasis en el uso de herramientas de Microsoft.
XP (Extreme Programming)	***XP*** no maneja fases de desarrollo en lugar de ello trabaja con técnicas. Las mismas que se dividen en 4 ámbitos: Planificación, Diseño, Codificación y Pruebas	***XP*** maneja cinco roles puntuales en los cuales el cliente forma uno de ellos.	Es un proceso iterativo / incremental el cual se basa en prácticas y técnicas de desarrollo probadas teniendo como elemento fundamental al cliente.

1.2.5.4.3 Adopción de Metodología

Luego de realizar la comparativa entre las diferentes metodologías de desarrollo de software frente al esquema en que se desenvuelve Moodle se ha decidido utilizar a **XP (Extreme Programming)** como metodología para llevar a cabo el desarrollo de este proyecto. Las razones por las cuales se optó por esta alternativa son las siguientes:

- El proyecto a desarrollarse es pequeño, no existe un equipo de desarrollo, está a cargo de una sola persona, además el tiempo que se dispone es muy corto.

- Los requisitos son volátiles, ya que pueden cambiar dependiendo de la adaptación de los alumnos, y/o basados en nuevas tecnologías.

- Se necesita que los modelos de diseño sean adaptativos más que predictivos.

- Se pretende dar un enfoque diferente a la forma de trabajo actual en el desarrollo de sistemas tratando de enfatizar y utilizar las mejores prácticas de desarrollo adoptadas por las metodologías ágiles.

- Se ha tomado como un punto importante para la elección de XP (Extreme Programming) ***el éxito que ha tenido la misma en proyectos de código abierto***.

1.2.5.5 XP (Extreme Programming) [15]

XP(Extreme Programming) es una nueva metodología en el desarrollo de software que nació a fines de los 90s en un proyecto desarrollado por **Kent Beck,** la cual contempla un conjunto de principios y prácticas

para el desarrollo rápido de software de alta calidad, simplificando el costo del proceso de desarrollo.

XP define cuatro variables para cualquier proyecto de software, éstas son ***coste, tiempo, calidad y alcance***, de las cuales el alcance es la única variable que se la deja libre para fijarla luego de establecer las otras tres por las fuerzas externas del proyecto, clientes y jefes de proyecto [17].

XP propone reducir el coste del cambio presentado comúnmente en la ingeniería de software tradicional a través de la combinación de buenas prácticas de programación y tecnología. Los costes de cambio tanto en la ingeniería de software tradicional frente a XP se pueden apreciar en las figuras Fig. 1.2, Fig. 1.3 respectivamente:

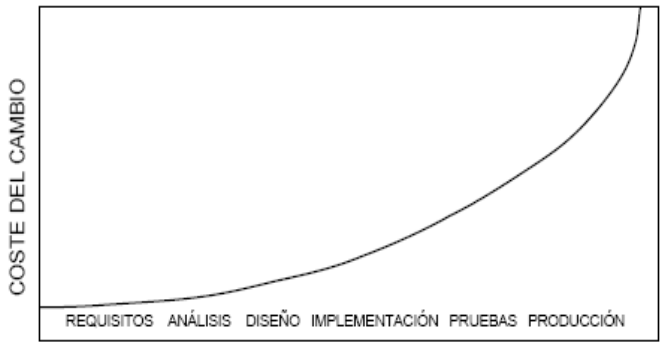

Fig. 1. 2: Ingeniería de Software Tradicional [17]

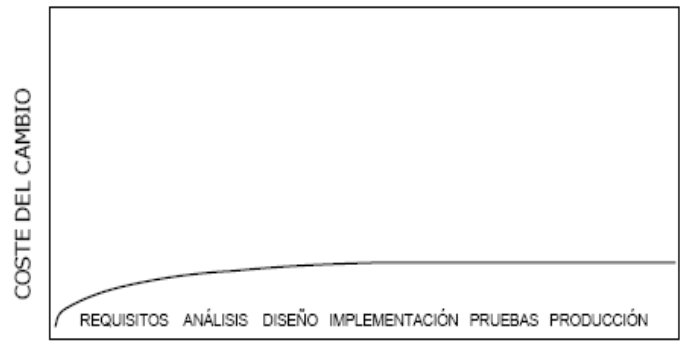

Fig. 1. 3: XP (Extreme Programming) [17]

1.2.5.5.1 Técnicas de XP

Las técnicas de ***XP (Extreme Programming)*** se dividen en cuatro ámbitos: planificación, diseño, codificación y pruebas las que detallaremos a continuación [17] [18]:

- ***La Planificación.-*** XP plantea la planificación como un verdadero diálogo entre la parte empresarial y la parte técnica del proyecto. Entre las técnicas que se sugieren en la planificación están las siguientes:

    - **Historias de Usuarios.-** Representa una descripción informal de las necesidades de los usuarios los cuales representan requerimientos funcionales o no funcionales del sistema.
    - **Sacar Nuevas Versiones con frecuencia**.- Se refiere a la periodicidad de como se manejarán las versiones, es decir con una estricta planificación temporal.
    - **Iteraciones.-** Se refiere a la implementación de forma incremental de nuevas historias de usuario en las sucesivas iteraciones, lo cual se alimenta a través de la retroalimentación objetiva de cada iteración.
    - **Trabajo en Equipo.-** Trata de la cohesión del equipo lo cual se logra a través del proceso rotativo de tareas en el proyecto, es decir conocer todo el proyecto, con el objetivo de disminuir las dependencias personales.

- ***Diseño.- XP*** propone **diseños simples** acordes a las necesidades presentes, no diseños futuristas que pretendan ser dotes adivinatorias para anticiparnos al futuro. Entre las técnicas que sugiere la planificación se tiene las siguientes:

- **Regla Kiss.-** Se refiere a obtener soluciones simples, eficaces nada complejas.
- **Nomenclatura.-** Se refiere a la estandarización dentro del proyecto, tanto en documentación, patrones de diseño etc.
- **Seguir planeación.-** Se realizará solo lo planeado, nada más ni nada menos.
- **Refactoring.-** Consiste en crear soluciones simples, reformando el código constantemente a partir de soluciones ya implementadas.

- ***Codificación.-*** En cuanto a la codificación XP propone técnicas basadas en el trabajo en equipo junto con la estandarización. Entre las técnicas se tiene las siguientes:

  - **Comunicación Usuario.-** Es sumamente importante mantener un contacto con el usuario de tal forma que puede involucrarse con las etapas de desarrollo y ver como se están satisfaciendo sus necesidades.
  - **Estándares Codificación.-** Se debe manejar estándares de codificación públicos que permitan un trabajo de equipo ordenado y colaborativo sobre una base de código común.
  - **Programación en Pareja.-** Es una de las ideas más interesantes de XP, en la cual se utiliza la participación de dos personas para la programación dentro de una misma computadora. En las cuales la persona que no programa tiene una visión analítica – estratégica sobre el código que está observando siendo los roles intercambiables al momento de codificar, la cual tiene el objetivo de promover mejoras y corregir errores.

- **40 Horas Semanales.-** Esta técnica hace referencia al estado físico y mental de los miembros del equipo de desarrollo, tomando en cuenta que el hecho de trabajar más no produce mayor calidad por lo que las horas extras a la semana solo perjudican a la calidad del proyecto teniendo que replanificar las tareas.

- ***Pruebas.-*** Uno de los pilares básicos sobre los que se sustenta XP son las pruebas, las cuales deben especificarse en lo posible antes de escribir el código manteniendo una documentación sustentable que permite verificar la satisfacción del cliente. Entre las técnicas se tiene las siguientes:

  - **Pruebas de Unidad.-** Consiste en realizar pruebas individuales cuyo objetivo es verificar la funcionalidad de una cierta parte del sistema.
  - **Integración Continua.-** Consiste en realizar la integración del sistema completo, para lo cual se necesita de una máquina de integración. La cual será testeada constantemente con cambios unitarios ya probados para garantizar que todo funcione correctamente.

1.2.5.5.2 Roles

Los Roles de XP de acuerdo a la propuesta original de Beck son los siguientes [16]:

- **Programador.-** Es el encargado de producir las pruebas unitarias y de producir el código del sistema.
- **Cliente.-** Es el encargado de especificar las historias de usuario y de diseñar las pruebas funcionales para validar su implementación.

- **Encargado de Pruebas. -** Es el encargado de la ejecución de las pruebas y de proporcionar las herramientas de soporte para pruebas.
- **Encargado de Seguimiento.-** Es el encargado de realizar el seguimiento del proceso en cada iteración, verificando el grado de aciertos en las estimaciones y sus tiempos reales.
- **Entrenador.-** Es el responsable de guiar el proceso global de desarrollo proporcionando al equipo las guías de forma que se aplique las prácticas XP y se siga el proceso correctamente.
- **Consultor.-** Es un miembro externo del equipo, cuya función es la de brindar soporte en algún tema específico en el que puedan surgir inconvenientes.
- **Gestor.-** Es el encargado de realizar las tareas de coordinación entre clientes y programadores en el cumplimiento de sus requerimientos.

1.3 DEFINICIÓN DEL MÓDULO DE EVALUACIÓN

La elección del módulo de Evaluación pasó por un proceso de análisis muy riguroso antes de ser elegido como módulo para implementarse ya que para su elección se tuvo que analizar como estaba constituido el ***LMS de Moodle***, ver sus ventajas, descubrir sus desventajas con el objetivo de encontrar alguna característica faltante que permitiera ayudar a mejorar el desempeño de Moodle como tal.

Y realmente se encontró que carecía de un módulo que permitiera establecer pautas retroalimentativas que garanticen una mejor enseñanza a los estudiantes. Este módulo de Evaluación Abierto será denominado con el nombre **EVAAB**.

1.3.1 CARACTERÍSTICAS

Antes de definir las características de ***EVAAB*** se ha creído conveniente exponer los principios de aprendizaje utilizados en ***EVAAB*** para su definición.

1.3.1.1 Principios de Aprendizaje.

Los principios de aprendizaje aplicativos en el desarrollo de ***EVAAB*** se encuentran ejemplificados en los dos puntos que se expondrán a continuación [19]:

- ***Entrenamiento***.- Los estudiantes no pueden aprender a pensar críticamente, analizar información, comunicar ideas científicas, formular argumentos lógicos, trabajar como parte de un grupo y adquirir otras destrezas deseables a menos que se les permita y anime a realizar dichas tareas una y otra vez en muchos contextos.

- ***Retroalimentación***.- El aprendizaje con frecuencia se lleva a cabo mejor cuando los alumnos tienen oportunidades para expresar ideas y obtener retroalimentación de algún medio. Pero para que ésta sea más útil la retroalimentación debe ser analítica, puntual y llegar en el momento en que los estudiantes están interesados en ella.

1.3.1.2 Características

Una vez que se ha explicado las razones cognitivas en las que se basa ***EVAAB*** para su desarrollo se definirá sus características, entre éstas se tiene las siguientes:

- ***EVAAB*** presentará un enfoque de ***ingreso y resolución dinámico*** con lo que se pretende ganar la atención tanto del profesor como del estudiante en el manejo del cuestionario.

- ***EVAAB*** tendrá elementos de apoyo como ***pistas, referencias*** para cada una de las preguntas facilitando de esta manera su resolución.

- El número de preguntas y el tipo de las mismas conjuntamente con los elementos que están relacionados como las pistas, referencias podrán ser parametrizados.

- En cuanto al manejo de referencias ***EVAAB*** permitirá hacer uso de todos los recursos utilizados por Moodle, facilitando la reutilización de los mismos.

- ***EVAAB*** permitirá generar un reporte acerca del proceso de la evaluación, una vez finalizado su resolución. Lo cual servirá como un elemento retroalimentativo para el estudiante.

- ***EVAAB*** permitirá marcar las preguntas que el estudiante por algún motivo no contestó ya sea porque estén dificultosas o no conozcan su respuesta para posteriormente regresar a completarlas.

- ***EVAAB*** contará con una retroalimentación instantánea para cada pregunta que se vaya resolviendo notificando el estado ya sea correcta, incorrecta e incompleta.

- ***EVAAB*** permitirá controlar el progreso del entrenamiento, mostrando las preguntas correctas, incorrectas e incompletas conjuntamente con el porcentaje de aceptación que se tiene en ese momento para cada una.

- ***EVAAB*** permitirá resolver el cuestionario tantas veces se lo decida con el único objetivo de fortalecer el conocimiento.

1.3.2 LIMITANTES

Todo desarrollo implica riesgos y mucho más sino no es concebido desde un inicio, es decir se tiene que trabajar sobre un esquema constituido por lo que

siempre existirán riesgos en los proyectos a desarrollarse. Entre los limitantes se puede mencionar los siguientes:

- ***Arquitectura Concebida***.- La cual representa un limitante muy fuerte, ya que el desarrollo a realizarse no es desde cero sino sobre una arquitectura ya definida, por lo que se necesitará estudiar muy bien a la misma para tratar de utilizar todas las características y bondades que ésta nos brinda.

- ***Experiencia en el manejo del lenguaje y herramientas de desarrollo***.- El cual representa un problema de adaptación un poco complicado cuando no existe un uso previo del lenguaje conjuntamente con las herramientas.

- ***Conocimientos de las nuevas tecnologías a utilizar.***- Representan al igual que el lenguaje temas que se tienen que aprender y ganar experiencia en el transcurso del proyecto.

1.3.3 HISTORIAS DE USUARIO

Cabe aclarar que las historias de usuario tienen ***el mismo propósito que los casos de uso que es el de especificar los requerimientos de un sistema,*** diferenciándose de éstos por las características como el tratamiento dinámico y flexible que poseen. Un ejemplo de esto es que en la Historia de Usuario se puede establecer una lista de seguimiento con la fecha y estado de las tareas que se tienen pendientes por terminar, además que cada historia de usuario es lo suficientemente comprensible y delimitada para que se puede implementar en un tiempo adecuado.

1.3.3.1 Planificación Inicial de las Iteraciones

Se comenzó definiendo el número de iteraciones a realizar junto con los requerimientos por iteración, los cuales son expresados en **Historias de Usuario**. Para esto se utilizó una herramienta que ayude a llevar el

control conocida con el nombre de **Xplanner[14].**

Se definieron 3 iteraciones importantes dentro de ***EVAAB***, para su estudio se las ha enumerado y nombrado de la siguiente forma:

- **Iteración 1:** Gestión de Elementos de Estudio
- **Iteración 2:** Gestión de Evaluación
- **Iteración 3:** Gestión de Retroalimentación.

1.3.3.2 Desarrollo de las Iteraciones

Cabe destacar que cada iteración presenta un conjunto de historias de usuario, las cuales se han elaborado de acuerdo a la perspectiva inicial pero que se han ido modificando en el avance del proyecto hasta obtener su aceptación completa. El resumen de la planificación de las iteraciones en su punto final se la puede apreciar en la Fig.1.4.

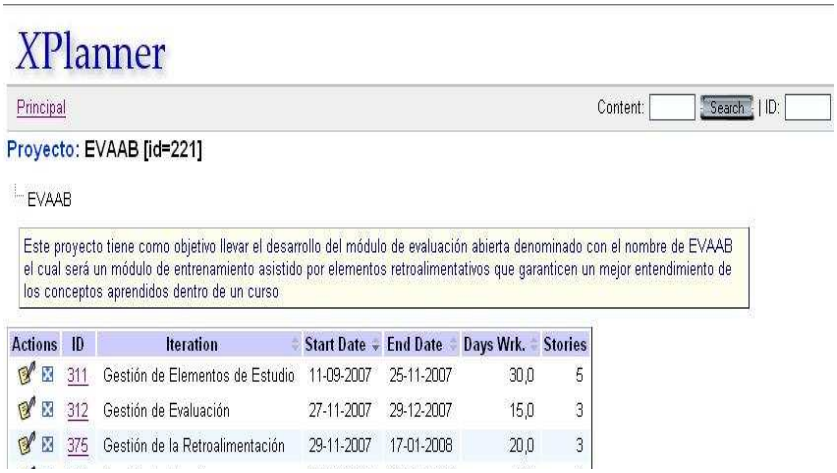

Fig. 1. 4: Planificación de las Iteraciones

[14] XPLANNER - http://xplanner.org/

1.3.3.2.1 Iteración 1: Gestión de Elementos de Estudio

Esta iteración se enfocará en la gestión del cuestionario conjuntamente con los diferentes elementos de estudio que conforman el mismo como son: preguntas, respuestas, pistas y referencias. Los mismos que serán gestionados en sus diferentes operaciones por parte del **Profesor** en lo que refiere al proceso de carga**,** las historias de usuario para esta iteración son las siguientes:

- Gestión del Cuestionario
  - Ingreso y validación del Cuestionario
  - Actualización del Cuestionario
  - Eliminación del Cuestionario
- Gestión de Preguntas
  - Ingreso y validación de Preguntas
  - Actualización de Preguntas
  - Eliminación de Preguntas
- Gestión de Respuestas
  - Ingreso y validación de Respuestas
  - Actualización de Respuestas
  - Eliminación de Respuestas
- Gestión de Pistas
  - Ingreso y validación de Pistas
  - Actualización de Pistas
  - Eliminación de Pistas
- Gestión de Referencias
  - Ingreso y validación de Referencias
  - Actualización de Referencias
  - Eliminación de Referencias

El desarrollo de esta iteración conjuntamente con sus historias de usuario se encuentra esquematizado en la Fig.1.5 que se muestra a continuación:

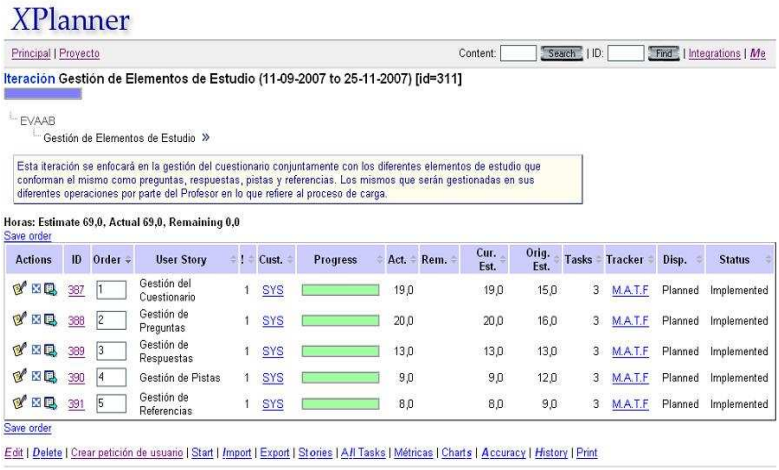

Fig. 1. 5: Desarrollo de la Iteración I

1.3.3.2.2 Iteración 2: Gestión de la Evaluación

Esta iteración se enfocará en el proceso de evaluación de los cuestionarios, es decir la forma como serán calificados, la forma de llevar el proceso de entrenamiento y resolución, la forma de mostrar los elementos de ayuda como pistas y referencias basándose en las características propias de configuración inicial como el tipo de cuestionario a resolver, tipo de preguntas etc. Las historias de usuario para esta iteración son las siguientes:

- Presentar elementos de ayuda para resolución de preguntas
- Evaluar Cuestionario

El desarrollo de esta iteración conjuntamente con sus historias de usuario se muestra a continuación en la Fig.1.6:

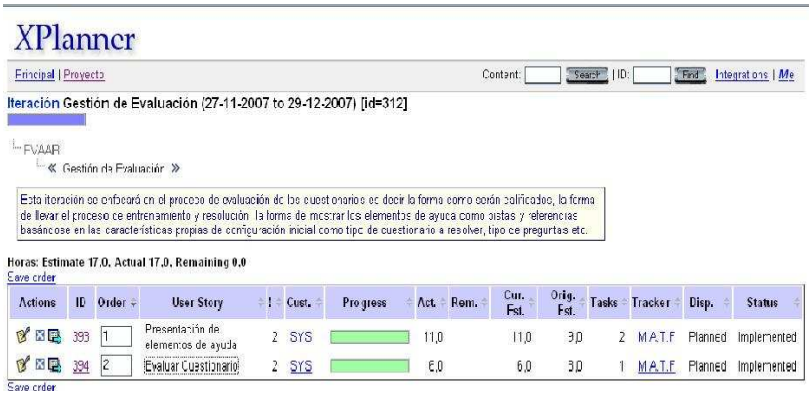

Fig. 1. 6: Desarrollo Iteración II

1.3.2.2.3 Iteración 3: Gestión de la Retroalimentación

Esta iteración se enfocará en dos aspectos sumamente importantes que son ***el seguimiento de la resolución y la generación de informe acerca del mismo***. Los cuales serán los elementos que permitirán establecer pautas retroalimentativas de estudio dentro del proceso de evaluación. Las historias de usuario para esta iteración son las siguientes:

- Seguimiento de resolución de Preguntas
- Mostrar progreso
- Generación de informe de Proceso

El desarrollo de esta iteración conjuntamente con sus historias de usuario se muestra a continuación en la Fig.1.7:

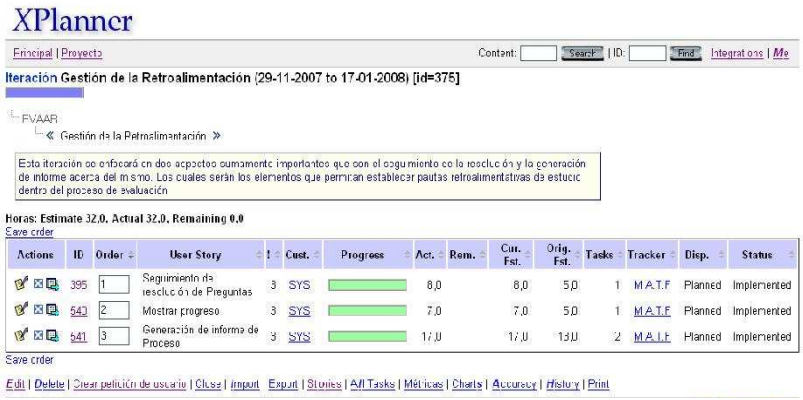

Fig. 1. 7: Desarrollo Iteración III

1.3.3.3 Desarrollo de Historias de Usuario

En el proceso de captura de requisitos inicial se fueron elaborando las primeras historias de usuario, es decir las preliminares las cuales fueron descritas de la manera más simple hasta llegar al punto actual de poderlas explicar con tiempos y estimaciones reales. Cabe mencionar que estas son las historias de usuario finales para cada iteración, las cuales se han modificando hasta tener la aceptación final por parte de los involucrados en el proyecto. Para la realización y seguimiento de las mismas se utilizó **XPLANNER.**

Las historias de usuario que se presentarán a continuación han sido estructuradas en base a iteraciones, las mismas que mantienen prioridades; lo que nos permitirá llevar una secuencia al momento de implementarlas al ser dependientes entre sí. Cada historia de usuario tiene atado tareas, las cuales nos permitirán organizar los requerimientos desde un punto de vista técnico al momento de implementarlos.

1.3.3.3.1 Iteración 1: Gestión de Elementos de Estudio

En esta iteración se agrupó las historias de usuario referentes a los elementos de estudio como son el cuestionario en si, las preguntas conjuntamente con sus respuestas, pistas y referencias.

Las historias de usuario para esta iteración tienen como objetivo gestionar todas las operaciones referentes a los elementos anteriormente mencionados, es decir la **Generación del Cuestionario**, lo cual esta a cargo del Profesor. Se tiene 5 historias de usuario para esta iteración conjuntamente con sus tareas las cuales se detallará a continuación:

Gestión del Cuestionario.- Esta historia de usuario se encarga de gestionar todo lo referente al cuestionario como son el ingreso, actualización y eliminación del mismo como se detalla en la Fig. 1.8

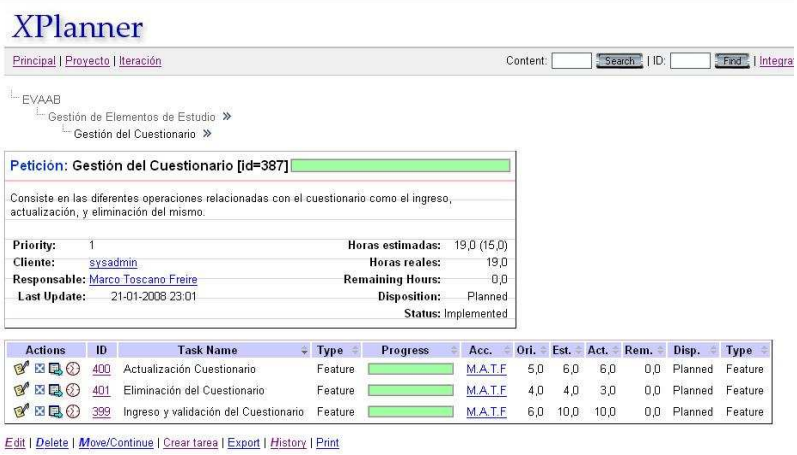

Fig. 1. 8: Gestión Cuestionario

Gestión de Preguntas.- Esta historia de usuario se encarga de gestionar todo lo referente a las preguntas como son el ingreso, actualización y eliminación de las mismas. Cabe destacar que esta historia de usuario tiene una dependencia fuerte con la Gestión de Cuestionario, motivo por el cual presenta una prioridad inferior; esta historia de usuario se la puede aprecia en la Fig. 1.9 a continuación:

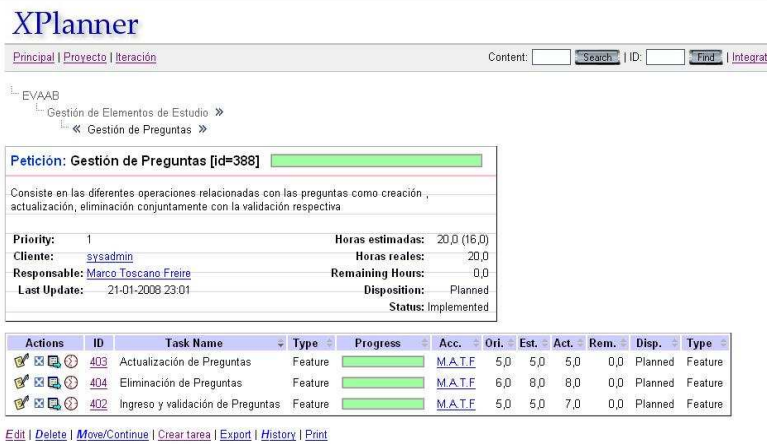

Fig. 1. 9: Gestión de Preguntas

Gestión de Respuestas.- Esta historia de usuario se encarga de gestionar las respuestas para cada pregunta conjuntamente con su validación. Entre las operaciones que se gestionan se tiene el ingreso, actualización y eliminación de las mismas; esta historia de usuario se la puede apreciar en la Fig.1.10 a continuación:

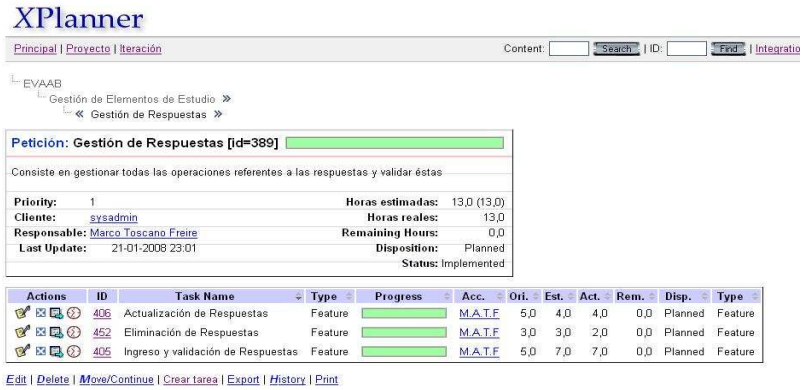

Fig. 1. 10: Gestión de Respuestas

Gestión de Pistas.- Esta historia de usuario se encarga de gestionar uno de los elementos de ayuda con los que cuenta la pregunta para su resolución que son las pistas, conjuntamente con su validación. Entre las operaciones que se gestionan se tiene el ingreso, actualización y eliminación de las mismas como se detalla en la Fig. 1.11 a continuación:

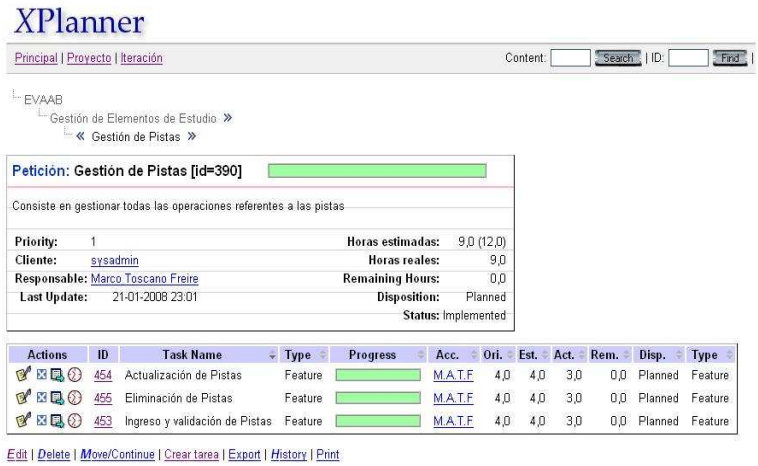

Fig. 1. 11: Gestión de Pistas

Gestión de Referencias.- Esta historia de usuario se encarga de gestionar otro elemento de ayuda con los que cuenta las preguntas diferenciándose del anterior por los recursos que tiene y por la exactitud de los mismos, estas son las referencias. Entre las operaciones que se gestionan se tiene el ingreso, actualización y

eliminación de las mismas conjuntamente con su validación como se detalla en la Fig. 1.12 a continuación:

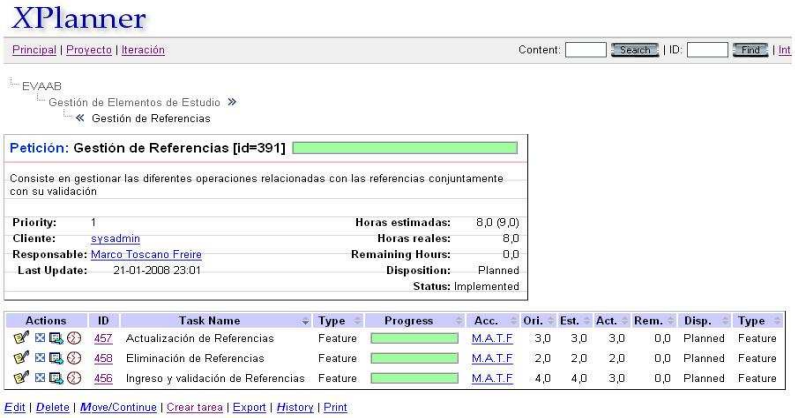

Fig. 1. 12: Gestión de Referencia

1.3.3.3.2 Iteración 2: Gestión de la Evaluación

Esta iteración se enfocará en el proceso de evaluación de los cuestionarios, es decir la forma como serán calificados, la forma de llevar el proceso de entrenamiento y su resolución, la forma de mostrar los elementos de ayuda como pistas y referencias basándose en las características propias de configuración inicial como tipo de cuestionario a resolver, tipo de preguntas etc. Se tiene 2 historias de usuario para esta iteración conjuntamente con sus tareas las cuales se detallarán a continuación:

Presentación Elementos Ayuda: Esta historia de usuario consiste en ver la forma como se presentarán los diferentes elementos de ayuda en el transcurso de la resolución del Cuestionario. La cual se la puede apreciar en la Fig. 1.13 a continuación:

Fig. 1. 13: Presentación Elementos Ayuda

Evaluar Cuestionario: Esta historia de usuario consiste en realizar la evaluación del cuestionario de un estudiante, contabilizando las preguntas correctas, incorrectas e incompletas para luego generar una calificación en base a éstas que muestre cual fue el resultado del Estudiante en la resolución del Cuestionario. La cual se la puede apreciar en la Fig. 1.14 a continuación:

Fig. 1. 14: Evaluar Cuestionario

1.3.3.3.3 Iteración 3: Gestión de la Retroalimentación

Esta iteración se enfocará en dos aspectos sumamente importantes que son el seguimiento de la resolución del Cuestionario y la generación de informe acerca del mismo. Los cuales serán los elementos que permitirán establecer pautas retroalimentativas de estudio dentro del proceso de evaluación.

Seguimiento Resolución Pregunta.- Esta historia de usuario consiste en validar cada pregunta y mostrar el estado correspondiente al momento de ir resolviéndola. La cual se la puede apreciar en la Fig. 1.15 a continuación:

Fig. 1. 15: Seguimiento Resolución Preguntas

Mostrar progreso.- Esta historia de usuario tiene como objetivo mostrar el avance del cuestionario en cualquier momento de su resolución de tal forma que se puede apreciar el porcentaje de aceptación que tiene las preguntas, la cuales se encuentran agrupadas según su estado sea este correcto, incorrecto e incompleto. La cual se la puede apreciar en la Fig. 1.16 a continuación:

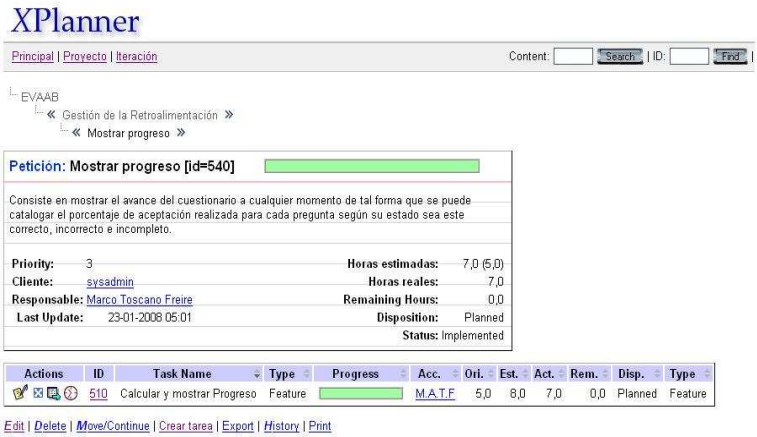

Fig. 1. 16: Mostrar Progreso

Generación de Informe de proceso.- Esta historia de usuario consiste en generar el informe final de la resolución del cuestionario en el que se mostrará un resumen del mismo conjuntamente con todos sus elementos haciendo énfasis en el detalle de las preguntas correctas e incorrectas junto con su aseveración. La cual se la puede apreciar en la Fig. 1.17 a continuación:

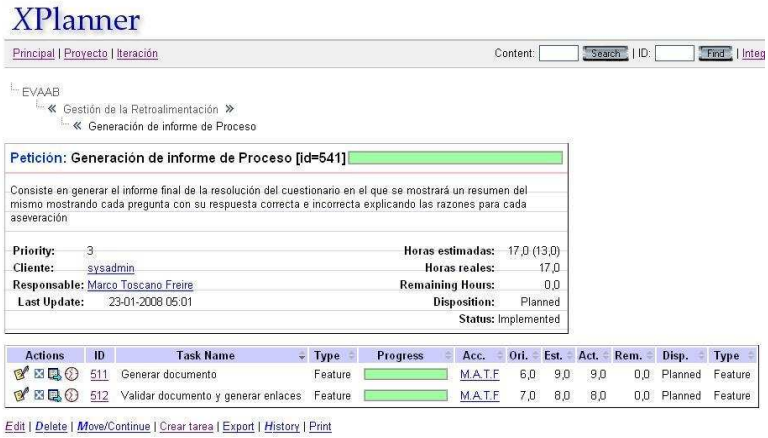

Fig. 1. 17: Generación informe proceso

1.3.3.4 Planificación del Versionamiento

Tomando en cuenta que el versionamiento es un punto fundamental en la metodología ***XP*** se definirá una planificación de versionamiento en nuestro proyecto, el cual determinará el flujo que se deberá llevar a cabo para administrar las diferentes versiones que se vayan obteniendo durante el desarrollo del proyecto. El mismo que se centrará específicamente en los documentos y código generado ya estos representan lo más importante dentro de ***EVAAB,*** la planificación de versionamiento contempla los siguientes puntos:

- Se utilizará la técnica de respaldos, los cuales se generarán periódicamente cada vez que el proyecto sea modificado. Los mismos que estarán almacenados hasta obtener una versión completa que contemple todas las funcionalidades de estos parciales, para luego ser borrados

- En cuanto al aplicativo cada vez que se obtenga una versión estable ésta será almacenada con el siguiente estándar ***evaab<fecha>,*** donde ***<fecha>*** representaba la fecha de finalización de la versión la cual tiene el formato de ***dd/mm/yyyy*** y a partir de ésta se comenzará a trabajar en la siguiente versión.

- En cuanto a la documentación, ésta será almacenada con el siguiente estándar ***evaabDoc<fecha>*** donde ***<fecha>*** representaba la fecha de finalización de la versión la cual tiene el formato de ***dd/mm/yyyy*** y a partir de ésta se comenzará a trabajar en la siguiente versión.

- Para el manejo de los respaldos en cuanto al código se dispone de una ***bitácora*** donde se encuentra detallado las características que posee cada versión, de tal forma de tener un control sobre las mismas y poder regresar a una versión posterior en caso de que amerite su corrección.

CAPÍTULO II

2. DISEÑO

2.1 DISEÑO DE MODELOS

2.1.1 FUNDAMENTOS

Desarrollar un sistema es una tarea compleja que implica una serie de procedimientos a realizar dependiendo de la fase en la que nos encontramos, sea ésta el análisis, diseño, implementación, pruebas e implantación. Pero independientemente de la metodología a utilizar estamos acostumbrados a seguir un proceso de desarrollo muy riguroso en cuanto a la definiciones de roles, actividades, artefactos incluyendo modelado y documentación detallada. Sin tomar en cuenta que este esquema tradicional es efectivo y necesario en proyectos de gran tamaño en cuanto a tiempo y recursos, pero no el más adecuado en proyectos pequeños en los cuales el equipo de desarrollo es reducido, con plazos cortos, requisitos volátiles y/o basados en nuevas tecnologías.

Tal es el caso que con la venida de las metodologías ágiles como **Extreme Programming (XP)** el enfoque relacionado con el diseño de un sistema cambia radicalmente ya que la utilización de modelos complejos de diseño se ven reemplazados mediante la utilización de técnicas propuestas por XP las cuales son **Tarjetas CRC, la Refactorización, Simplicidad [20],** tomando en cuenta que la ***Arquitectura*** definida previamente juega un papel fundamental.

La utilización de estas técnicas deberá seguir un modelamiento diferente al acostumbrado por las metodologías estándar por el mismo hecho de tratarse de una metodología ágil, por lo cual utilizaremos al ***Modelamiento Ágil*** como un proceso de software.

2.1.2 ARQUITECTURA

La arquitectura que utilizaremos para el desarrollo del módulo de evaluación **EVAAB** es una arquitectura complementaria a la ***Moodle,*** la cual utiliza los componentes definidos por Moodle en su arquitectura original conjuntamente con los añadidos por ***EVAAB.*** Esta arquitectura se la puede apreciar a continuación en la Fig.2.1.

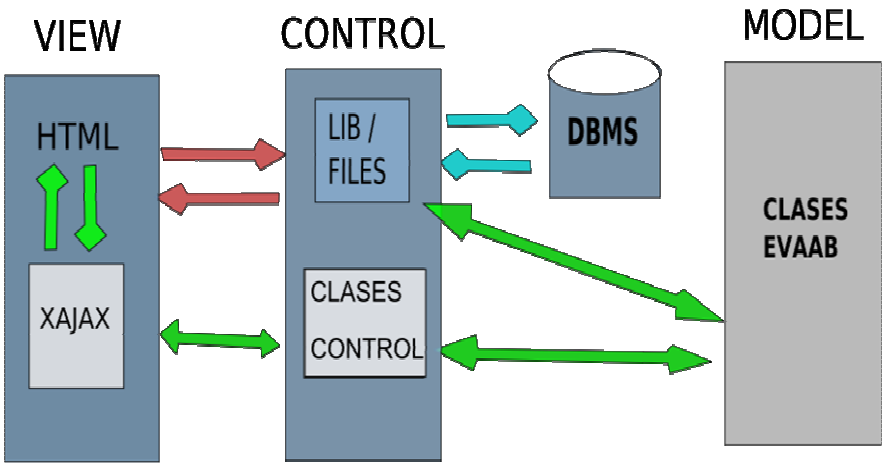

Fig. 2. 1: Arquitectura EVAAB

La arquitectura con la cual trabajará el módulo de evaluación **EVAAB**, está definida de la siguiente manera:

- En lo que respecta a los componentes, los mostrados con ***color azul*** son los cuales están definidos en **Moodle**, mientras que los que presentan un color ***plomo*** son los que han sido añadidos por **EVAAB** a la arquitectura actual.

- En el caso de las flechas se tiene la siguiente notación: los mostrados ***con color rojo*** son los flujos normales de navegación definidos en **Moodle,** los especificados con ***color verde*** son el flujo que presentará **EVAAB** mientras que los mostrados en ***color turquesa*** son los que son comunes para los dos.

- La arquitectura con la cual funciona **Moodle** en una arquitectura formada por dos capas, una de visualización y otra de control siendo ésta última la encargada de comunicarse con el ***DBMS*** para la gestión de los datos.

- La arquitectura con la cual funcionará **EVAAB** es una arquitectura cimentada en la de ***Moodle*** con unas pequeñas variaciones en cuanto a su diseño; entre éstas se puede mencionar las siguientes:

    - Presenta una arquitectura de tres capas, una adicional a las capas definidas en arquitectura de **Moodle,** llamada **modelo.** Esta es la encargada de manejar los objetos persistentes dentro de la ***BDD,*** cabe aclarar que utilizaremos el patrón de diseño ***MVC*** el cual hablaremos en detalle en relación a ***EVAAB*** en el capítulo siguiente.
    - En la parte de la ***Vista*** se ha utilizado la librería **XAJAX** para la utilización de **AJAX**, la cual esta cargada en la parte del cliente como un componente adicional. Para lo cual se ha implementado una serie de ***funciones*** para la comunicación con la librería **XAJAX** para la generación de los elementos dinámicos de HTML mediante **AJAX.**
    - Se implementaron clases ***controladoras*** las cuales manejan la lógica del negocio del módulo de evaluación ***EVAAB,*** las mismas que se comunican con las ***clases del modelo*** para manejar los datos.

2.1.3 MODELAMIENTO AGIL (AM)

2.1.3.1 Fundamentos

El modelamiento ágil (***AM***) es un proceso de software basado en principios y prácticas el cual describe como modelar y documentar de una manera ágil y efectiva. Siendo una de sus metas direccionar como aplicar las técnicas de modelamiento sobre proyectos que manejan

metodologías extremas como ***Extreme Programming (XP), Scrum [21].***

Tomando en cuenta que ***XP*** incluye modelamiento en su fase de diseño como parte de su proceso y que éste no esta descrito tan explícitamente en la forma de cómo realizarlo, se puede hacer uso del ***Modelamiento Ágil (AM)*** donde sea necesario.

2.1.3.2 Principios

El modelamiento ágil (***AM***) define una colección fundamental y suplementaria de principios, los cuales algunos de ellos han sido adoptados de ***XP***, que serán aplicados luego en las prácticas de AM. Entre los principios se tiene los siguientes [21]:

- ***Asumir Simplicidad.-*** La solución más simple es la mejor solución, no se debe gastar el tiempo en añadir características a nuestro modelo las cuales no se necesiten todavía.
- ***Adoptarse al Cambio.-*** Se debe estar consciente que los requerimientos pueden cambiar y la implicación que representará esto en el proyecto.
- ***Cambio Incremental.-*** Comenzar con modelos simples que capturen los requerimientos en ese momento y seguir añadiéndolos incrementalmente.
- ***Modelar con un propósito.-*** Crear modelos que tengan un propósito, es decir identificar porque y para quién se lo esta realizando.
- ***Múltiples Modelos.-***. Crear múltiples modelos para desarrollar software, ya que cada modelo describe a un aspecto simple del software. Un punto importante es que no se necesitará desarrollar todos los modelos todo dependerá del sistema dado.
- ***Calidad de Trabajo.-*** La calidad esta inmersa en el modelamiento.

- ***Rápida Retroalimentación.-*** Tener una retroalimentación rápida por parte de los involucrados en el proyecto que permita rediseñar las soluciones y llegar a la satisfacción total.
- ***Trabajar en el Software es lo más importante.-*** La meta principal de desarrollo de software ***es producir un software de alta calidad*** que satisfaga la necesidad de los clientes; por lo que producir demasiada documentación, artefactos o aún modelos si no contribuyen al proyecto deberían ser cuestionados y evitados a menos que sean justificados a tiempo.
- ***Mantener la Claridad.-*** Cada artefacto que se cree debe ser mantenible en el tiempo.
- ***El Contenido es más importante que la Representación.-*** Este es un principio suplementario que define que un modelo podría tener diferentes formas de representación como por ejemplo: papers, documentos formales, prototipos pero que no necesita ser documentado el punto es tomar ventajas de los beneficios de modelar sin incurrir en los costos de crear y mantener la documentación.
- ***Comunicación Abierta y Honesta.-*** Significa estar libre para receptar sugerencias en el modelamiento, lo que garantiza la calidad de la información.

2.1.3.3 Prácticas

El modelamiento ágil (***AM***) define una colección fundamental y suplementaria de prácticas, las cuales algunos de ellos han sido adoptadas de ***XP.*** Entre las prácticas se tiene las siguientes [21]:

- ***Participación Activa del Cliente.-*** Se refiere a tener una participación constante de los clientes ya sea personal o por algún medio, de tal forma que se pueda recopilar por parte de ellos la información pertinente al sistema que se está construyendo.

- ***Aplicar los modelos correctos.-*** Implica conocer muy bien las fortalezas y desventajas de cada modelo y aplicarlos según nuestras necesidades.
- ***Propiedad Colectiva.-*** Cada persona que esta desarrollando puedo trabajar en un modelo y en efecto en algún artefacto del proyecto.
- ***Crear varios modelos en Paralelo.-*** A causa de que cada modelo tiene sus fortalezas y debilidades ningún modelo simple es suficiente para las necesidades de modelamiento, por lo cual se debería iterar a través de otros artefactos en vez de enfocarse en uno a la vez.
- ***Crear contenido simple.-*** Se debería mantener el contenido de modelos de los requerimientos de análisis, diseño, arquitectura lo más simple posible, es decir no se debería añadir aspectos adicionales a éstos a menos que sean justificables.
- ***Describir los modelos de forma simple-*** Una vez que se considere los potenciales diagramas que se aplicará, se debe realizar éstos colocando las características que estamos intentando entender y describir.
- ***Describir los modelos públicamente.-*** Se refiere a dar a conocer los modelos de una manera pública de tal forma que todos los implicados en el proyecto puedan verlos.
- ***Iterar en otros artefactos.-*** Cuando se este desarrollando artefactos como historias de usuario, CRC, Diagramas de secuencia, entre otros y se encuentre atascado se debería considerar trabajar en otro artefacto. De tal forma de ganar tiempo además que al cambiar de punto de vista se podría encontrar las dificultades presentadas en el otro artefacto.
- ***Modelar en incrementos pequeños.-*** Se incrementa la agilidad para liberar software a los usuarios rápidos.
- ***Simple Recurso de Información.-*** La información debería se almacenada en un y solo un lugar.

- ***Usar las herramientas más simples.-*** Se refiere a utilizar las herramientas más simples para modelar, ya que se podría dibujar los modelos en papel tomando en cuenta que lo importarte es el entendimiento, pero si se necesita mantener y presentar al cliente se debería utilizar una herramienta simple.
- ***Proveer con código.-*** Un modelo debería reflejar el aspecto de lo que se este construyendo.
- ***Aplicar Modelamiento Estándar.-*** La idea básica es que el desarrollador debería seguir un conjunto de modelos estándar sobre un proyecto de software.
- ***Descartar Modelos Temporales.-*** La cantidad de modelos que se crea en la etapa de diseño son temporales, es decir fueron hechos en un momento para cumplir con un propósito. Pero se necesita sincronizar éstos eligiendo los que añadan un valor a nuestro proyecto y descartando los que no lo hacen.

2.1.4 CRC (Class, Responsabilities and Collaborations)

2.1.4.1 Fundamentos

Un modelo CRC es una colección estándar de tarjetas indexadas, las cuales presentan la estructura mostrada a continuación en la Fig. 2.2:

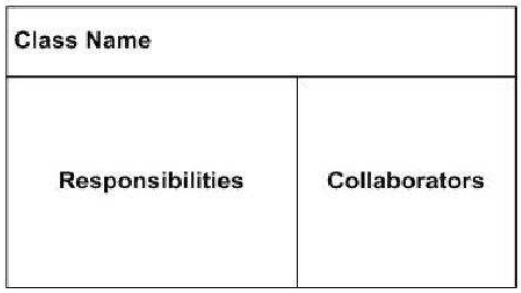

Fig. 2. 2: Tarjeta CRC [21]

Las tarjetas CRC son usadas para diseñar un sistema en ***XP***, una de las características mayores de éstas es que permiten a la gente saltar del

método procedimental al modelo de objetos por el mismo hecho de que fueron originalmente concebidas como una técnica para enseñar los conceptos orientados a objetos [22]. Las tarjetas CRC como pudimos apreciar en la Fig.2.2 presentan tres secciones:

- ***Class Name.-*** Representa una colección de objetos similares un objeto es una persona, cosa, evento o concepto que es relevante al sistema a manejar el cual presenta un estado y un comportamiento.
- ***Responsabilities.-*** Una responsabilidad esta denotada por algo que una clase conoce o realiza.
- ***Collaborators.-*** Esta definida como otra clase con la cual se actúa para cumplir con las responsabilidades, desde dos puntos de vista sea por una petición para obtener información o una petición para hacer algo.

2.1.4.2 Modelamiento CRC

Para realizar el modelamiento CRC se deben seguir algunos pasos fundamentales entre los cuales se tiene los siguientes [22]:

- Identificar las clases que intervienen en el sistema, se podría incluir las UI (User Interface) si fuera necesario. Se recomienda utilizar sola clases de Entidad.
- Identificar las responsabilidades de cada clase tomando en cuenta su funcionamiento conjuntamente con los datos que se desea mantener.
- Identificar los colaboradores de cada clase, que vienen a ser las clases con las cuales se interactúa para cumplir con alguna responsabilidad.
- Luego de haber identificado las clases conjuntamente con sus responsabilidades y colaboradores, debemos colocar la CRC elaboradas en una tabla de tal forma que se puede conocer como están sus dependencias.

La forma más efectiva de mantener un modelo de ***tarjetas CRC*** en **XP (Extreme Programming)** es crear un modelo ***por cada historia de usuario***, ***regla de negocio*** en lugar de crear un modelo general que contemplé todos los requerimientos del sistema siguiendo la práctica del ***Modelamiento Ágil*** llamada ***Modelar en pequeños incrementos.*** En la Fig.2.3 que se presenta a continuación se puede apreciar un modelo de tarjetas CRC [22]:

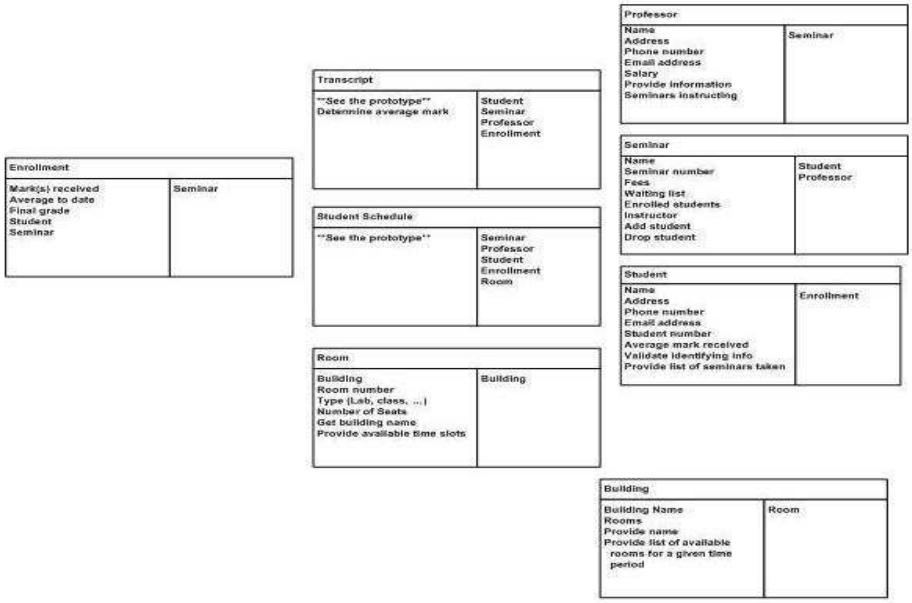

Fig. 2. 3: Modelo CRC

2.1.4.3 Modelo de EVAAB

Para realizar los modelos de las tarjetas CRC se utilizó la práctica del Modelamiento Ágil (AM), ***Modelar en Pequeños Incrementos,*** por lo cual crearemos un ***modelo CRC por cada historia de usuario*** identificada en cada iteración.

Para la realización de cada ***tarjeta CRC*** se tomó en cuenta las ***responsabilidades*** y ***colaboradores*** necesarios en cada historia de usuario, es decir no se definirán todos los elementos en cada sección sino simplemente los que necesitemos particularmente en esa historia de usuario, es decir en los modelos CRC las tarjetas CRC variarán de acuerdo

al enfoque de la historia de usuario.

2.1.4.3.1 Iteración 1: Gestión de Elementos de Estudio

Se diseñó ***5 modelos CRC para esta iteración***, uno por cada historia de usuario, para lo cual se utilizó los pasos definidos en la anterior sección en cuanto al modelamiento con CRC. Las historias de usuario que conforman esta iteración son las siguientes: ***Gestión de Cuestionario, Gestión de Preguntas, Gestión de Respuestas, Gestión de Pistas y Gestión de Referencias***

2.1.4.3.1.1 Gestión de Cuestionario

Para esta historia de usuario se definió 3 tarjetas CRC, ***Cuestionario, Profesor y Tipo Cuestionario,*** las cuales se puede apreciarlas en la Fig.2.4 junto con el modelo de tarjetas CRC.

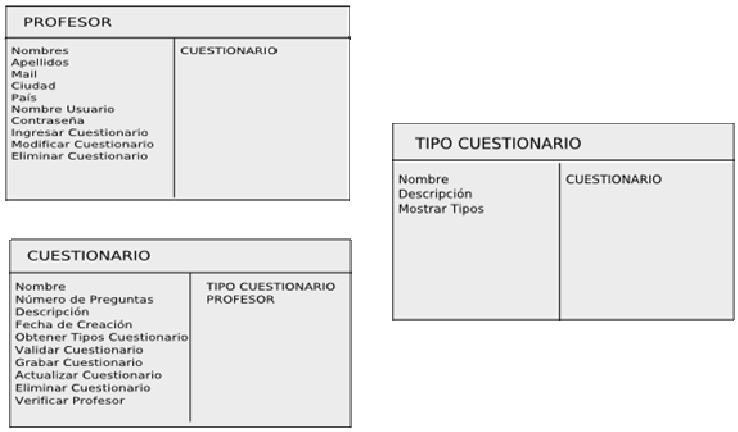

Fig. 2. 4 : Modelo CRC Gestión de Cuestionario

2.1.4.3.1.2 Gestión de Preguntas

Para esta historia de usuario se definió 4 tarjetas CRC, ***Tipo Pregunta, Pregunta, Cuestionario, Profesor***, las cuales se puede apreciar en la Fig.2.5 junto con el modelo de tarjetas

CRC.

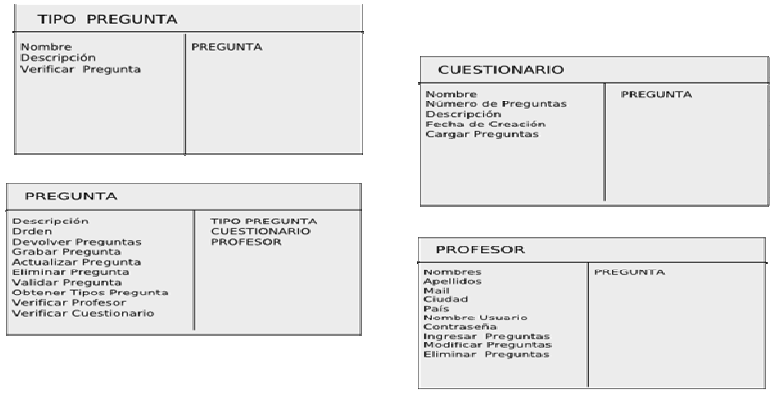

Fig. 2. 5: Modelo CRC Gestión de Preguntas

2.1.4.3.1.3 Gestión de Respuestas

Para esta historia de usuario se definió 4 tarjetas CRC, ***Respuesta, Tipo Pregunta, Pregunta, Profesor***, las cuales se las puede apreciar en la Fig. 2.6 junto con el modelo de tarjetas CRC.

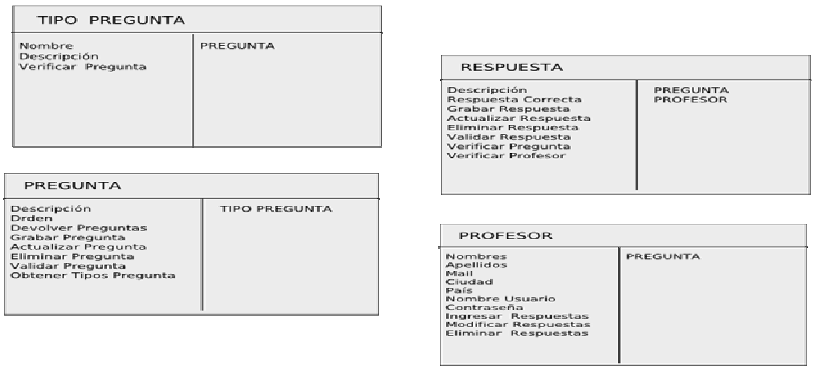

Fig. 2. 6: Modelo CRC Gestión de Respuestas

2.1.4.3.1.4 Gestión de Pistas

Para esta historia de usuario se definió 4 tarjetas CRC, ***Pista, Tipo Pregunta, Pregunta, Profesor***, las cuales se las puede

apreciar en la Fig. 2.7 junto con el modelo de tarjetas CRC.

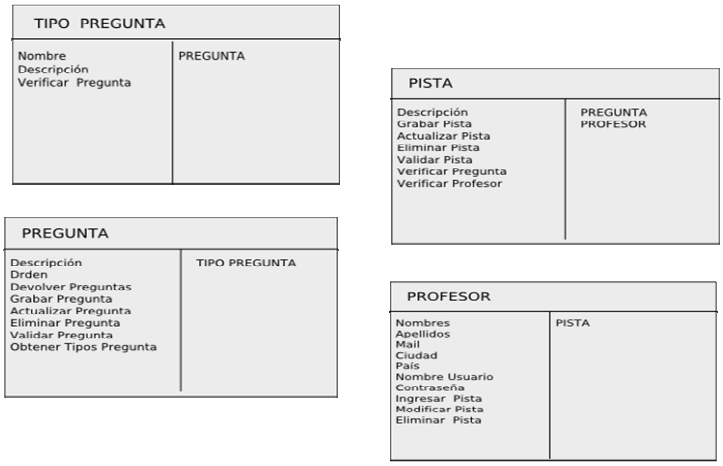

Fig. 2. 7: Modelo CRC Gestión de Pistas

2.1.4.3.1.5 Gestión de Referencias

Para esta historia de usuario se definió 4 tarjetas CRC, ***Referencia, Tipo Pregunta, Pregunta, Profesor*** las cuales se las puede apreciar en la Fig. 2.8 junto con el modelo de tarjetas CRC.

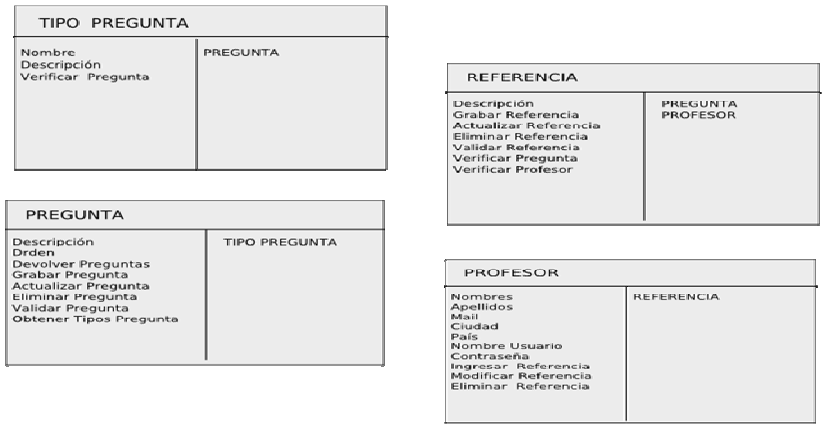

Fig. 2. 8: Modelo CRC Gestión de Referencia

2.1.4.3.2 Iteración 2: Gestión de la Evaluación

Se diseñó 2 modelos CRC para esta iteración, uno por cada historia de usuario, para lo cual se utilizó los pasos definidos en la anterior sección en cuanto al modelamiento con CRC. Las historias de usuario que conforman esta iteración son las siguientes: ***Presentar Elementos Ayuda, Evaluar Cuestionario.***

2.1.4.3.2.1 Presentar Elementos de Ayuda

Para esta historia de usuario se definió 5 tarjetas CRC, ***Cuestionario, Referencia, Pista, Pregunta, Estudiante*** las cuales se las puede apreciar en la Fig.2.9 junto con el modelo de tarjetas CRC.

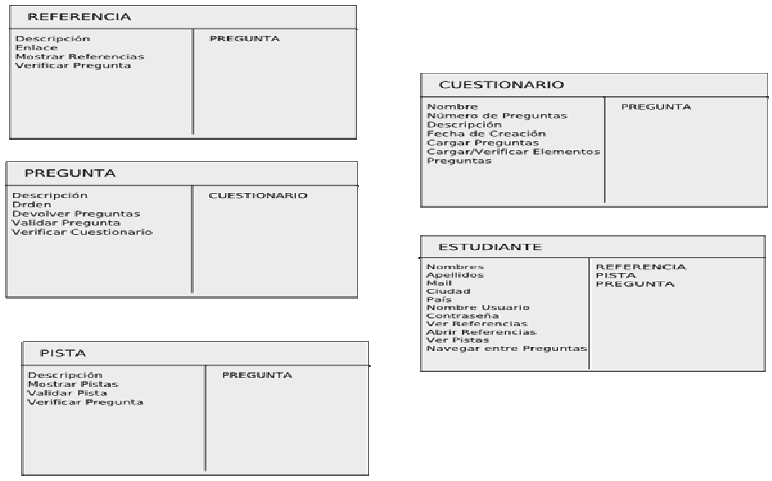

Fig. 2. 9: Modelo CRC Presentar Elementos de Ayuda

2.1.4.3.2.2 Evaluar Cuestionario

Para esta historia de usuario se definió 3 tarjetas CRC, ***Pregunta, Estudiante***, ***Cuestionario*** las cuales se las puede apreciar en la Fig.2.10 junto con el modelo de tarjetas CRC.

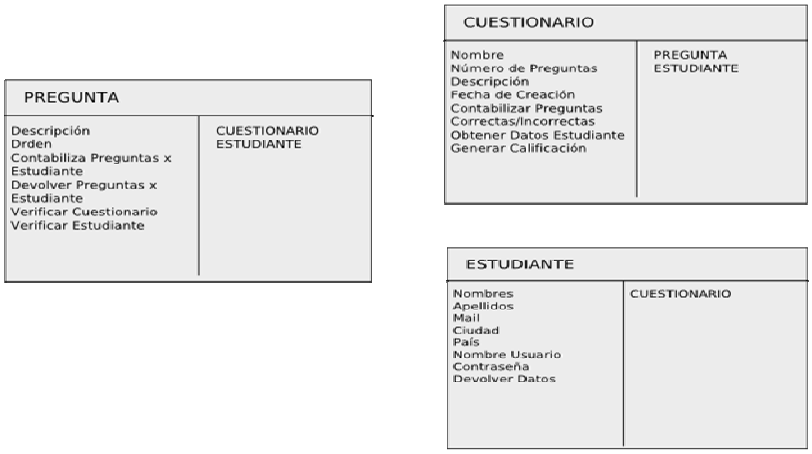

Fig. 2. 10: Modelo CRC Evaluación del Cuestionario

2.1.4.3.3 Iteración 3: Gestión de la Retroalimentación

Se diseñó 3 modelos CRC para esta iteración, uno por cada historia de usuario, para lo cual se utilizó los pasos definidos en la anterior sección en cuanto al modelamiento con CRC. Las historias de usuario que conforman esta iteración son las siguientes: ***Seguimiento de Resolución de Preguntas, Mostrar Progreso, Generación de Informe de Proceso.***

2.1.4.3.3.1 Seguimiento de Resolución de Preguntas

Para esta historia de usuario se definió 5 tarjetas CRC, ***Respuesta, Tipo Pregunta, Pregunta, Estudiante***, ***Cuestionario*** las cuales se las puede apreciar en la Fig. 2.11 junto con el modelo de tarjetas CRC.

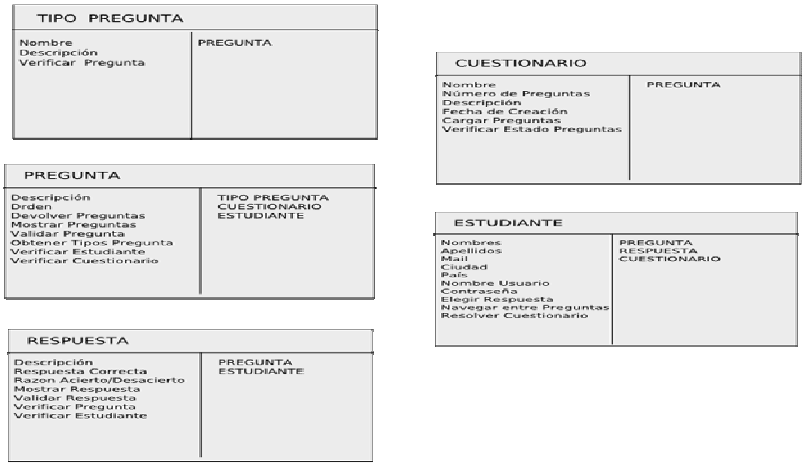

Fig. 2. 11: Modelo CRC Seguimiento de Resolución de Preguntas

2.1.4.3.3.2 Mostrar Progreso

Para esta historia de usuario se definió 4 tarjetas CRC, ***Respuesta, Pregunta, Estudiante***, ***Cuestionario*** las cuales se las puede apreciar en la Fig. 2.12 junto con el modelo de tarjetas CRC.

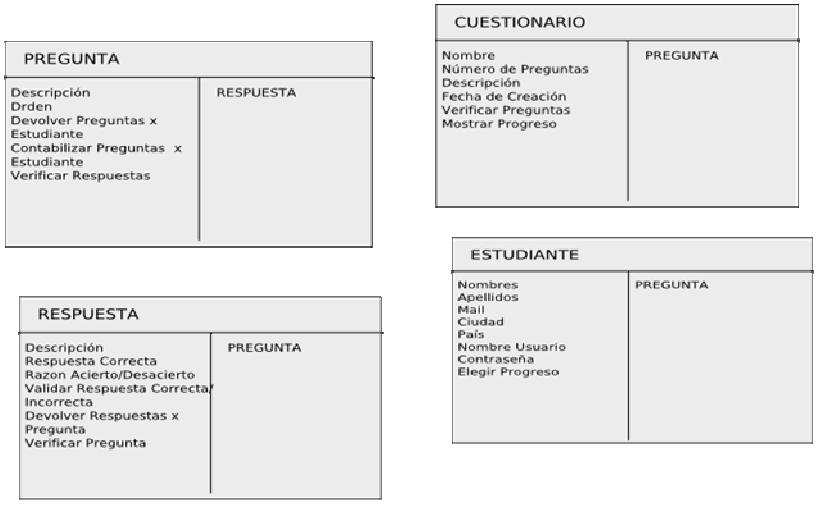

Fig. 2. 12: Modelo CRC Mostrar Progreso

2.1.4.3.3.3 Generación de Informe de Proceso

Para esta historia de usuario se definió 5 tarjetas CRC, ***Referencia, Pregunta, Respuesta, Estudiante***, ***Cuestionario*** las cuales se las puede apreciar en la Fig.2.13 junto con el modelo de tarjetas CRC.

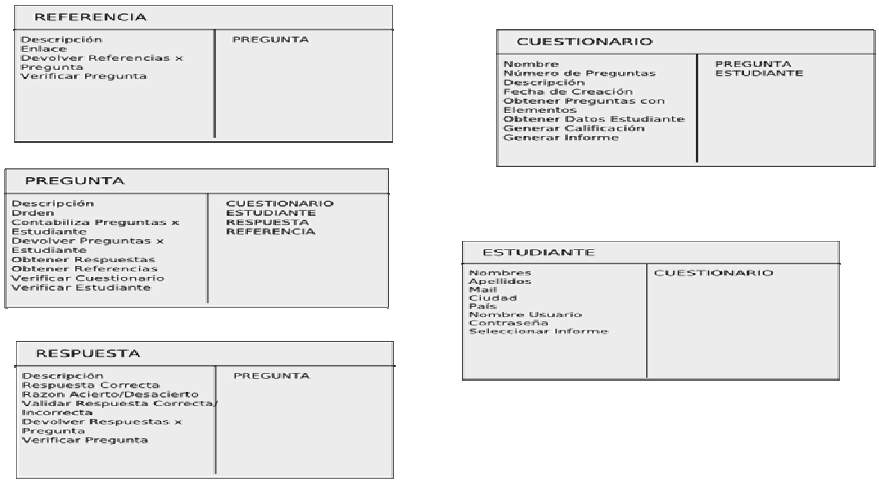

Fig. 2. 13: Modelo CRC Generación de Informe de Progreso

2.1.5 REFACTORIZACIÓN CRC

En ***XP (Extreme Programming)*** se busca siempre optar por la solución más simple tal es el caso de que cuando se dispone de partes comunes entre historias de usuario se debería optar por la utilización de refactorización con el único objetivo de lograr un modelo común que representan a los modelos. Tomando en cuenta esto se ha definido ***modelos de tarjetas CRC comunes*** dentro de cada iteración que garantice un mejor entendimiento, los mismos que se mostrarán a continuación.

2.5.1.1 Iteración 1: Gestión de Elementos de Estudio

Esta iteración esta conformado por las siguientes tarjetas CRC ***Cuestionario, Profesor, Tipo Cuestionario, Tipo Pregunta, Respuesta, Pista, Referencia*** las cuales han sido expresadas en el siguiente modelo CRC.

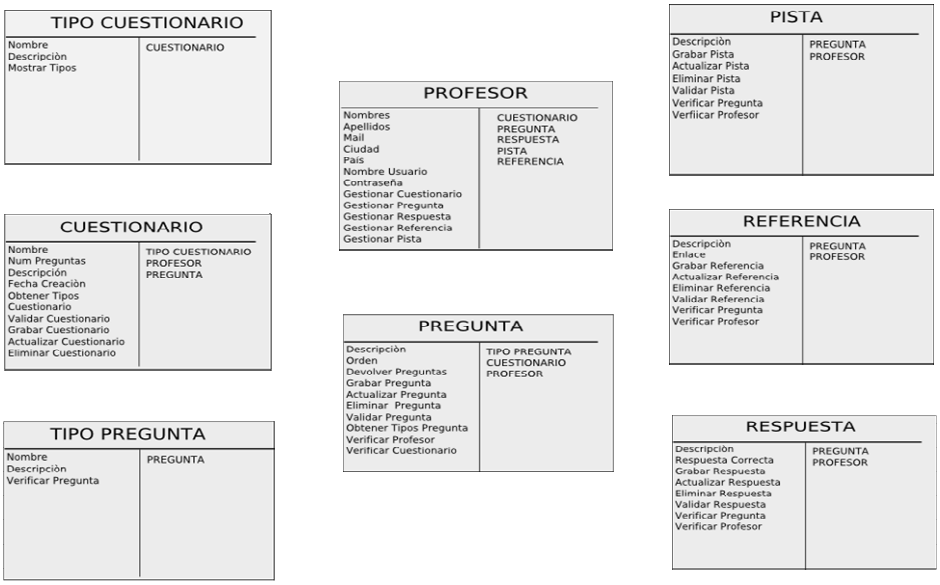

Fig. 2. 14: Modelo CRC para Iteración 1

2.5.1.2 Iteración 2: Gestión de Evaluación

Esta iteración esta conformado por las siguientes tarjetas CRC ***Cuestionario, Pregunta, Estudiante, Pista, Referencia*** las cuales han sido expresadas en el siguiente modelo CRC.

Fig. 2. 15: Modelo CRC para Iteración 2

2.5.1.3 Iteración 3: Gestión de Retroalimentación

Esta iteración esta conformado por las siguientes tarjetas CRC ***Cuestionario, Estudiante, Pregunta, Tipo Pregunta, Respuesta, Pista, Referencia*** las cuales han sido expresadas en el siguiente modelo CRC.

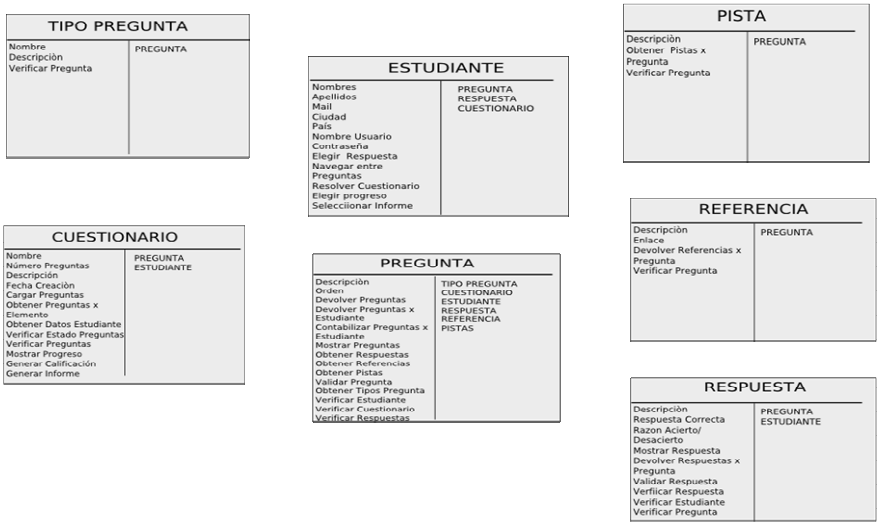

Fig. 2. 16: Modelo CRC para Iteración 3

2.1.6 DIAGRAMAS DE ACTIVIDAD

Una vez definido los ***Modelos CRC*** para las diferentes historias de usuario es necesario realizar los diagramas de actividad involucrados en cada iteración de tal forma que se puede expresar cada una de las historias de usuario de una manera más clara. Se utilizó los principios y prácticas definidas por el modelamiento ágil específicamente: ***Asumir Simplicidad, Modelar con Propósito, Describir los Modelos de Forma Simple, Aplicar Modelamiento Estándar [22]***.

2.1.6.1 Iteración 1: Gestión de Elementos de Estudio

Los diagramas de actividad contemplados en esta iteración representan la gestión de todos los elementos del Cuestionario como son preguntas, respuestas, pistas, referencias y el cuestionario propiamente. Los mismos que contemplan las operaciones de inserción, actualización y eliminación de cada uno de ellos, para los cuales se necesita tener el perfil de Profesor.

2.1.6.1.1 Gestión de Cuestionario

Lo particular en el diagrama de actividad para esta historia de usuario es que cuando se realiza la operación de eliminación se eliminan los elementos relacionados al Cuestionario. Además este es el elemento principal dentro de todo el contexto de ***EVAAB***. El cual se lo puede apreciar en la Fig. 2.17 a continuación:

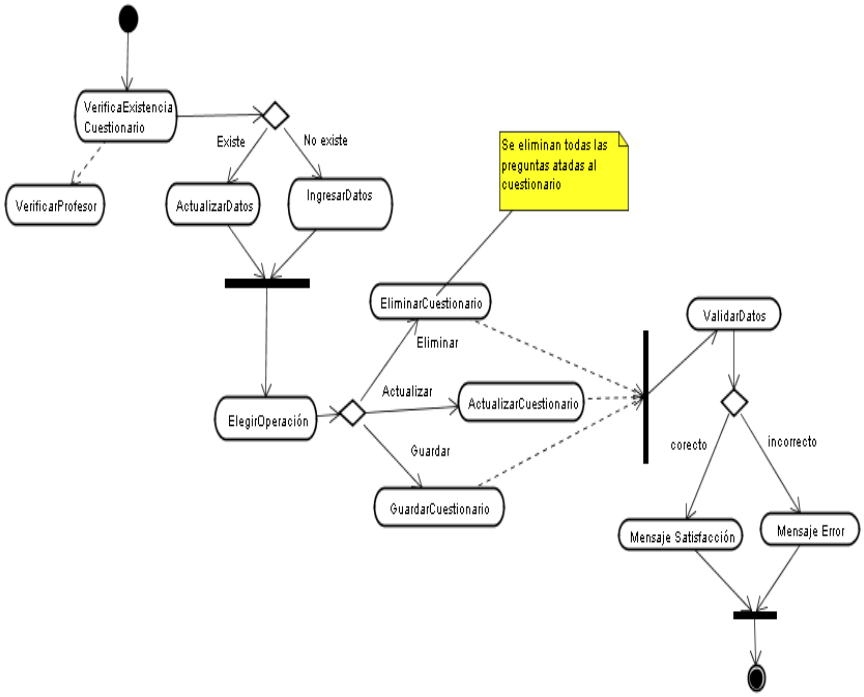

Fig. 2. 17: Diagrama Activad Gestión Cuestionario

2.1.6.1.2 Gestión de Preguntas

El diagrama de actividad para esta historia de usuario presenta una dependencia con el Cuestionario ya que sobre se sobre este se gestionan las preguntas. La operación de eliminación al igual que en la historia de usuario anterior elimina los elementos relacionados, el diagrama de actividad para esta historia de usuario se lo puede apreciar en la Fig. 2.18 a continuación:

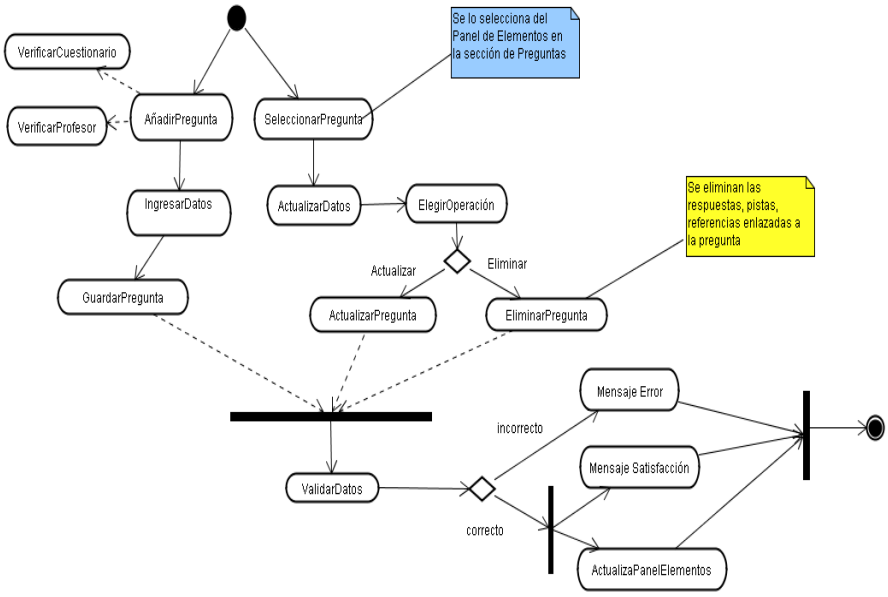

Fig. 2. 18 : Diagrama de Activad Gestión Preguntas

2.1.6.1.3 Gestión de Respuestas

El diagrama de actividad para esta historia de usuario presenta una dependencia con la pregunta ya que sobre se sobre esta se gestionan las respuestas. Cabe aclarar que para poder eliminar una respuesta el número de éstas debe ser mayor que el mínimo que es 2, el diagrama de actividad para esta historia de usuario se lo puede apreciar en la Fig. 2.19 a continuación:

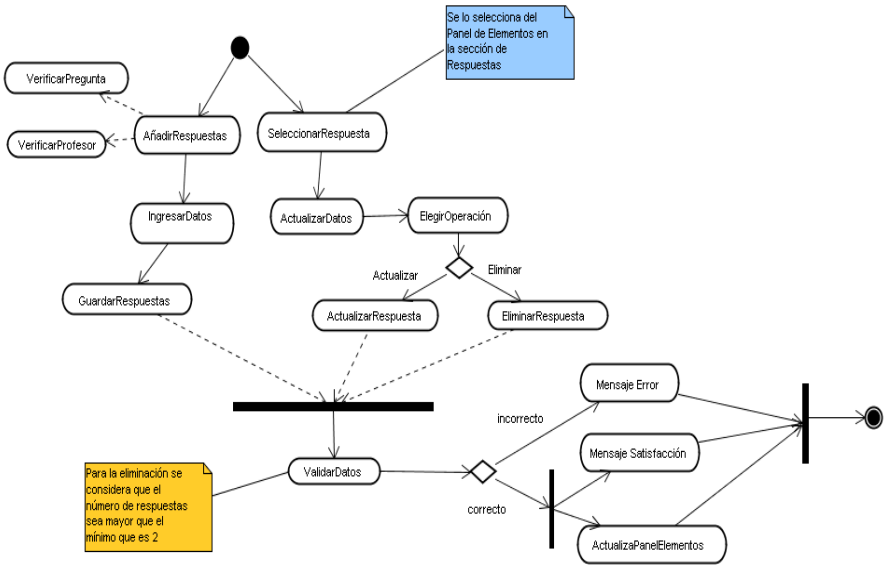

Fig. 2. 19: Diagrama de Actividad Gestión Respuestas

2.1.6.1.4 Gestión de Pistas

El diagrama de actividad para esta historia de usuario presenta una dependencia con la pregunta ya que sobre se sobre esta se gestionan las pistas. La operación de eliminación al igual que en la historia de usuario anterior permite el borrado solamente cuando el número de pistas supera al mínimo de ellas que es 2, el diagrama de actividad para esta historia de usuario se lo puede apreciar en la Fig. 2.20 a continuación:

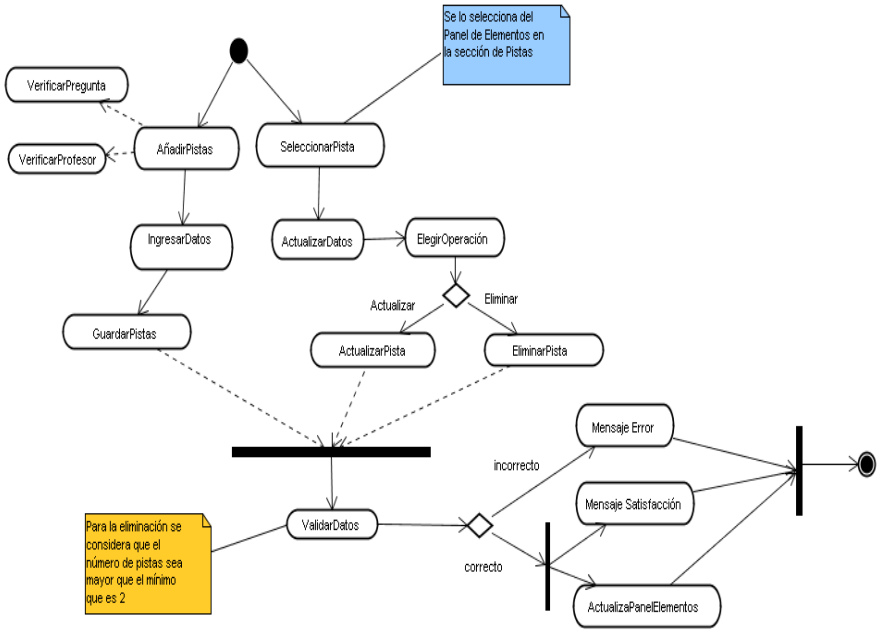

Fig. 2. 20: Diagrama de Actividad Gestión de Pistas

2.1.6.1.5 Gestión de Referencias

El diagrama de actividad para esta historia de usuario presenta una dependencia con la pregunta ya que sobre se sobre esta se gestionan las referencias. La operación de eliminación al igual que en la historia de usuario anterior permite el borrado solamente cuando el número de referencias supera al mínimo de ellas que es 2, el diagrama de actividad para esta historia de usuario se lo puede apreciar en la Fig. 2.21 a continuación:

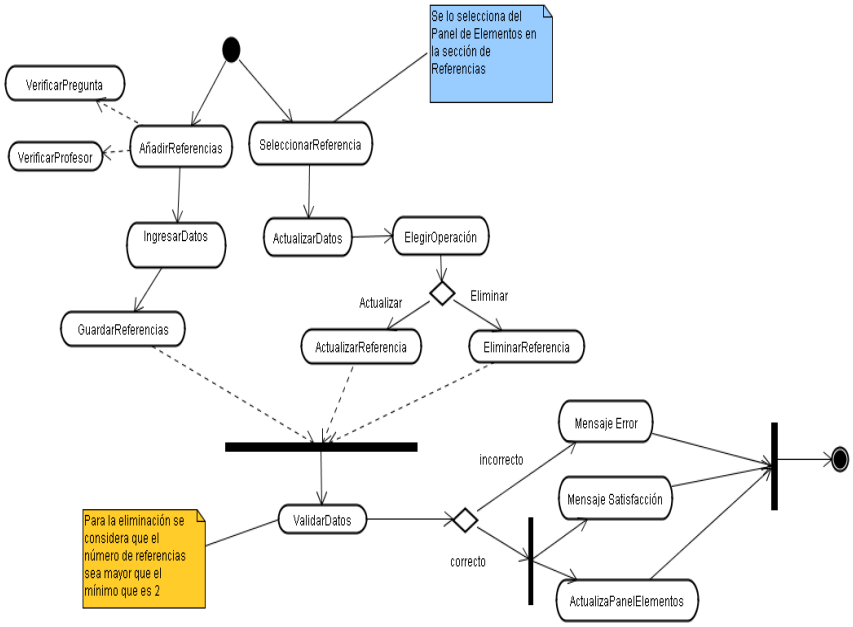

Fig. 2. 21: Diagrama de Actividad Gestión de Referencias

2.1.6.2 Iteración 2: Gestión de Evaluación

Los diagramas de actividad contemplados en esta iteración representan los procesos que involucran la evaluación del Cuestionario comenzando por la visualización de los elementos de ayuda hasta la generación de la calificación. Para los cuales se necesita tener estar logeado con el perfil de Estudiante.

2.1.6.2.1 Presentar Elementos de Ayuda

El diagrama de actividad para esta historia de usuario contempla el proceso que se tiene que seguir para utilizar los elementos de ayuda que presenta el cuestionario al momento de su resolución. Los cuales son las pistas y referencias, el diagrama de actividad para esta historia de usuario se lo puede apreciar en la Fig. 2.22 a continuación:

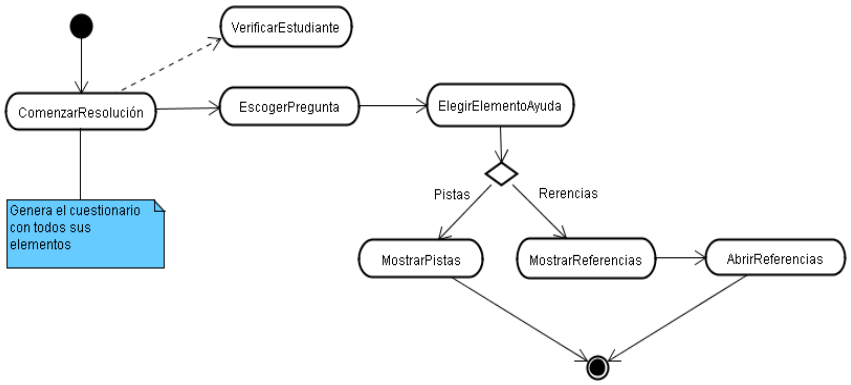

Fig. 2. 22: Diagrama de Actividad Presentar Elementos Ayuda

2.1.6.2.2 Evaluar Cuestionario

El diagrama de actividad para esta historia de usuario representa el proceso que se realiza para evaluar el cuestionario el cual es automático y se genera a través de la contabilización de las preguntas según su estado, sea este correcta, incorrecta o incompleta. El cual se lo puede apreciar en la Fig. 2.23 a continuación:

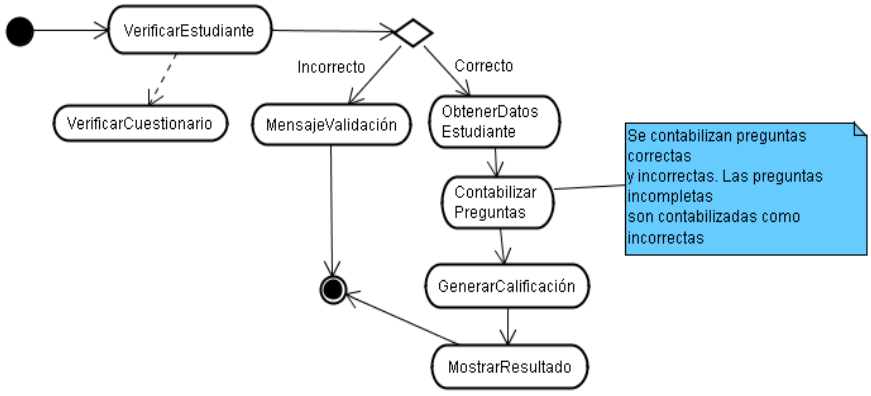

Fig. 2. 23: Diagrama de Actividad Evaluar Cuestionario

2.1.6.3 Iteración 3: Gestión de Retroalimentación

Los diagramas de actividad contemplados en esta iteración representan la parte fundamental del módulo de evaluación ***EVAAB***, es decir la retroalimentación que tiene un Estudiante al resolver un cuestionario permitiendo su mejoramiento.

2.1.6.3.1 Seguimiento de Resolución de Preguntas

El diagrama de actividad para esta historia de usuario denota el proceso de retroalimentación instantánea que tiene una pregunta al momento de su contestación. El cual se la puede apreciar en la Fig. 2.24 a continuación:

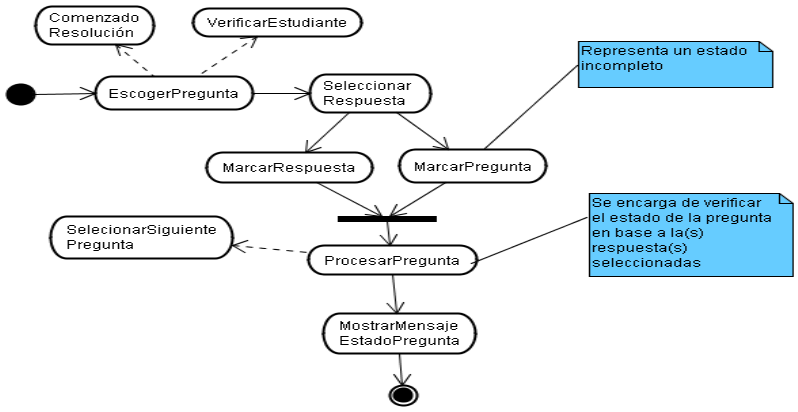

Fig. 2. 24: Diagrama Actividad Seguimiento Resolución Preguntas

2.1.6.3.2 Mostrar Progreso

El diagrama de actividad para esta historia de usuario especifica el proceso realizado para mostrar el progreso que tiene un estudiante en la resolución de un cuestionario en un momento determinado. El cual se la puede apreciar en la Fig. 2.25 a continuación:

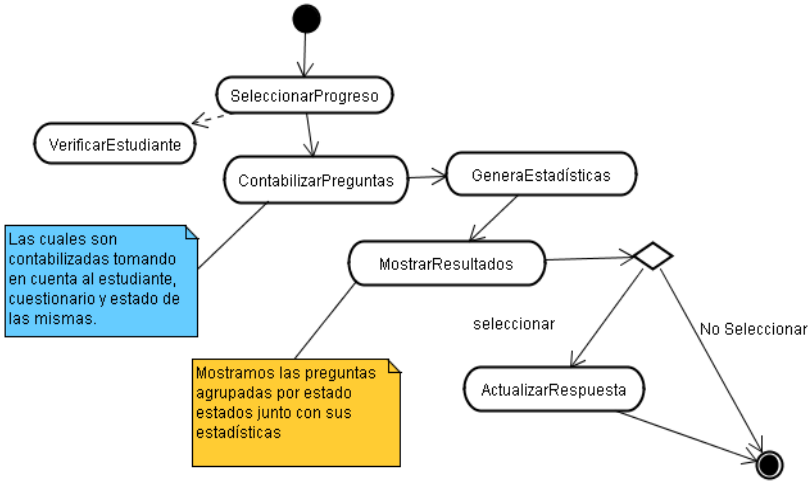

Fig. 2. 25: Diagrama de Actividad Mostrar Progreso

2.1.6.3.3 Generación de Informe de Proceso

El diagrama de actividad para esta historia de usuario muestra el proceso realizado para generar el informe final para el Estudiante, el mismo que muestra un resumen de la evaluación realizada. El cual se la puede apreciar en la Fig. 2.26 a continuación:

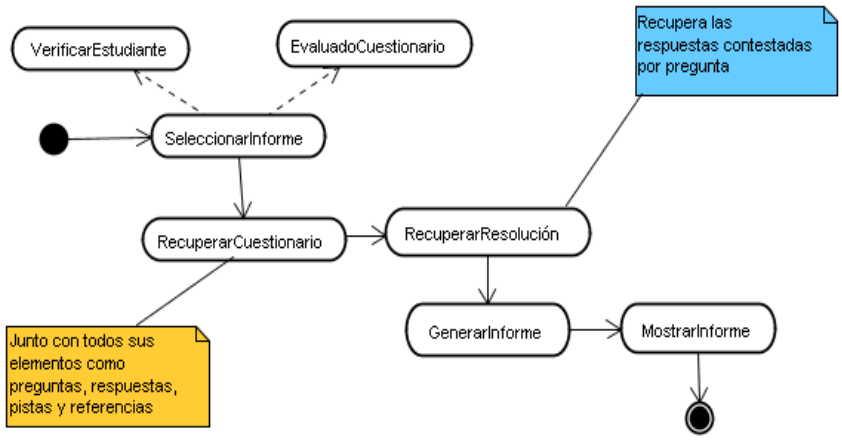

Fig. 2. 26: Diagrama de Actividad Generación Informe Proceso

2.2 DISEÑO DE BASE DE DATOS

2.2.1 GENERALIDADES

Para gestionar los datos se optó por **MYSQL** en su versión **5.0.45** junto con las herramientas de administración denominadas con el nombre de **MySQL GUI Tools 5.0,** se eligió este DBMS ya que esta es la base de datos por omisión definida en ***Moodle***[15]. Además se definió los estándares de ***EVAAB***, los cuales junto con los de ***Moodle*** se utilizarán en la definición de los diferentes elementos de la base de datos.

[15] Database Default - http://docs.moodle.org/en/Database

2.2.2 ESTANDARIZACIÓN

2.2.2.1 Diseño del Estándar EVAAB

Utilizar estándares en el proceso de desarrollo es una tarea sumamente fundamental, la cual debe ser aplicada en cada una de las fases en la que se encuentre. Ya que esto permitirá llevar una mejor organización en todo el proceso.

Tomando en cuenta la importancia que representan los datos en todo sistema se va a comenzar definiendo los estándares propios de ***EVAAB*** en cuanto a la BDD, que se define en los siguientes elementos:

- **Nombres de Entidades.-** El nombre de las entidades se definió utilizando el nombre del prefijo de ***moodle*** seguida del nombre del **módulo** junto con el **nombre de la tabla**, como por ejemplo: ***<PREFIX>_EVAAB_NOMBRETABLA,*** en donde el nombre de la tabla esta definido en su forma completa (no abreviada) de tal forma que exprese el contenido de la misma.

- **Nombres de Atributos.-** El nombre de los atributos se definió utilizando las tres primeras letras de la tabla sin ***tomar el cuenta el prefijo,*** si se tratase de una tabla simple o seis si se tratase de una tabla compuesta ***seguidas del nombre del atributo***. Se utilizó 4 letras cuando las tablas sean semejantes en nombre para diferenciarlas, todos los atributos estarán expresados con letras minúsculas y de forma completa por ejemplo: ***mat_descripcion*** o ***tippre_nombre.***

- **Nombres de Relaciones.-** El nombre de las relaciones se definió utilizando la palabra **REL** como identificador de la relación seguida de las 3 primeras letras del nombre del módulo denominadas con **EVA;** seguida del nombre de las tablas expresado con las tres primeras letras de la misma si es que tratase

de una tabla simple, o seis si se tratase de una tabla compuesta. Además se tomó en cuenta el grado de cardinalidad, de uno a muchos por ejemplo: ***REL_EVA_TIPPRE_PRE*** la cual expresa que la relación de uno a muchos entre las tablas tipo pregunta y pregunta

2.2.2.2 Estándar Moodle

En lo que respecta a Moodle también tiene un estándar a seguir en cuanto a la estructura de la base de datos, los cuales se encuentran definidos junto con otros puntos como **estilo de código, seguridades, reglas generales** en un archivo llamado ***codign.htm***[16]**.** Entre los puntos más relevantes que encontramos en este archivo en cuanto a la estructura de base de datos se puede mencionar los siguientes:

- Cada tabla debe tener un campo ***autoincrement int (10)*** llamado ***id*** como clave primaria, lo cual se ha cumplido en cuanto a la forma del campo y tamaño, pero no en el nombre totalmente ya que también se utilizó el estándar de EVAAB.

- La tabla principal que contiene instancias de cada módulo debería tener el mismo nombre del módulo y contener los siguientes campos como mínimo:

  - ***id***.- Como se describió anteriormente
  - ***course***.- id del curso que cada instancia pertenece
  - ***name***.- nombre completo de cada instancia del módulo

- Las tablas asociadas con el módulo que contengan información acerca de ***"cosas"*** deberían ser nombradas con ***modulo_cosas (note el plural).***

[16] Development: Coding - http://docs.moodle.org/en/Development:Coding

- Los nombres de las columnas deberían estar siempre en ***minúsculas.***

- Los campos booleanos deberían ser implementados como ***smallint*** conteniendo 0 o 1 para permitir una expansión de valores si fuera necesario.

- Cada tabla debería comenzar con el prefijo de la base de datos <***PREFIX***> que se haya definido en la instalación**.**

2.2.3 IDENTIFICACIÓN DE ENTIDADES, ATRIBUTOS Y RELACIONES

Se han identificado un total de **11 Entidades**, **37 Atributos** y **11 relaciones** en el módulo de evaluación **EVAAB,** los cuales han surgido cambios permanentes hasta poder llegar a la definición actual.

2.2.3.1 Descripción de Entidades /atributos.

Para la descripción de las entidades no se ha tomado en cuenta el prefijo denominado por Moodle para la creación de las tablas, **PREFIX,** es decir se le omitió al nombrarla en vez de llamarla **PREFIX_EVAAB** solamente se lo definió como **EVAAB**. Las tablas que maneja EVAAB son las siguientes:

2.2.3.1.1 Evaab

Representa la tabla donde se almacena la información de los cuestionarios a desarrollarse en el trayecto del curso, la misma que es la tabla principal dentro del módulo de evaluación que se integrará con Moodle. Por esta razón algunos de sus atributos tuvieron que ser cambiados en su definición de acuerdo a lo propuesto en la estandarización de Evaab, estos son el **identificador y** el **nombre** los cuales fueron reemplazados con **name y id** respectivamente**.** Entre los atributos que conforman a la entidad

EVAAB están los siguientes:

- ***id.-*** Representa el identificador de cada cuestionario, su tipo de dato es ***int (10)*** de la forma ***autoincrement.***
- ***name.-*** Representa el título del cuestionario, su tipo de dato es un ***varchar(20).***
- ***cue_numpregunta.-*** Representa el número de preguntas a realizar en cada cuestionario, su tipo de dato es ***smallint.***
- ***cue_descripcion.-*** Representa la descripción del cuestionario, la cual puede ser tan simple o explicita como se necesite, su tipo de dato es un ***varchar(100).***
- ***course.-*** Representa el curso dentro de Moodle, el cual es una columna obligatoria que todo módulo debe tener, su tipo de dato es un ***varchar (10).***
- ***cue_fechacre.-*** Representa la fecha de creación del cuestionario, su tipo de dato es un ***datetime***

2.2.3.1.2 Evaab_Estado

Representa la tabla donde se almacenan los diferentes estados que tienen las preguntas, entre estos tenemos correcta, incorrecta, incompleta etc. Entre los atributos que conforman a la entidad ***EVAAB_ESTADO*** están los siguientes:

- ***est_id.-*** Representa el identificador de cada estado, su tipo de dato es ***int (4)*** de la forma ***autoincrement.***
- ***est_nombre.-*** Representa el nombre descriptivo de cada estado el cual puede tomar unas de las opciones mencionadas anteriormente, su tipo de dato es un ***varchar(20).***

2.2.3.1.3 Evaab_Preguntas

Representa una de las tablas más importante dentro de ***EVAAB*** la cual maneja las diferentes preguntas que tendrá el cuestionario, las mismas que tienen un tipo y variarán tanto en el número como tipo de respuesta de acuerdo a él. Entre los atributos que conforman a la entidad ***EVAAB_PREGUNTAS*** están los siguientes:

- ***pre_id.-*** Representa el identificador de cada pregunta, su tipo de dato es ***int (10)*** de la forma ***autoincrement.***
- ***pre_descripcion.-*** Representa la pregunta en sí a realizarse, su tipo de dato es ***varchar(100).***
- ***pre_orden.-*** Representa el orden de las preguntas, lo cual nos servirá para realizar el seguimiento de las mismas en la resolución, su tipo de dato es ***smallint***

2.2.3.1.4 Evaab_Pistas

Representa la tabla donde se almacenan las pistas con las cuales cuenta cada pregunta para su resolución, las mismas que constituyen uno de los elementos de ayuda dentro de ***EVAAB***. Entre los atributos que conforman a la entidad ***EVAAB_PISTAS*** están los siguientes:

- ***pis_id.-*** Representa el identificador de cada pista, su tipo de dato es ***int (10)*** de la forma ***autoincrement.***
- ***pis_descripcion.-*** Representa en si la pista la cual tendrá el detalle de lo que se podría hacer para resolver la pregunta, su tipo de dato es ***varchar (50).***

2.2.3.1.5 Evaab_Referencias

Representa la tabla donde se almacenan las referencias con las cuales cuenta cada pregunta para su resolución, los mismos que

constituyen otro de los elementos de ayuda con los que cuenta ***EVAAB*** diferenciándose de la pista por la información exacta que presenta. Es decir viene a ser un recordatorio de estudio a una pregunta dada más que una ayuda suelta, entre los atributos que conforman a la entidad ***EVAAB_REFERENCIAS*** están los siguientes:

- ***ref_id.-*** Representa el identificador de cada referencia, su tipo de dato es ***int (10)*** de la forma ***autoincrement.***
- ***ref_descripcion.-*** Representa la referencia dada para cada pregunta la cual detallará el tema que se tiene que revisar para resolver la pregunta, su tipo de dato es ***varchar (50).***
- ***ref_enlace.-*** Representa el enlace dado para cada referencia el cual nos permitirá dirigirnos directamente a la referencia para revisarla, su tipo de dato es ***varchar (50).***

2.2.3.1.6 Evaab_Respuestas

Representa la tabla donde se almacenan las respuestas para cada pregunta, las mismas que representan su solución. Entre los atributos que conforman a la entidad ***EVAAB_RESPUESTAS*** están los siguientes:

- ***res_id.-*** Representa el identificador para cada respuesta, su tipo de dato es ***int (10)*** de la forma ***autoincrement.***
- ***res_descripcion.-*** Representa la descripción de la respuesta para cada pregunta formulada, su tipo de dato es ***varchar (50).***
- ***res_correcta.-*** Representa la respuesta correcta en cada pregunta, su tipo de dato es un ***smallint.***
- ***res_razon.-*** Representa la razón por la cual una respuesta es correcta o incorrecta, su tipo de dato es un ***varchar (50).***

2.2.3.1.7 Evaab_Tipocuestionario

Representa la tabla donde se almacena el tipo de cuestionario a resolver el cual puede ser abierto o cerrado, los mismos que se diferencian entre sí por las características que manejan cada uno como son el número de preguntas y los elementos de ayuda que presentan. Entre los atributos que conforman a la entidad ***EVAAB_TIPOCUESTIONARIO*** están los siguientes:

- ***tipcue_id.-*** Representa el identificador para cada tipo de cuestionario, su tipo de dato es ***int (4)*** de la forma ***autoincrement.***
- ***tipcue_nombre.-*** Representa el nombre del tipo de cuestionario el cual puede ser abierto o cerrado, su tipo de dato es **varchar (20).**
- ***tipcue_descripcion.-*** Representa la descripción detallada para cada tipo de cuestionario, su tipo de dato es ***varchar (50).***

2.2.3.1.8 Evaab_Tipopregunta

Representa la tabla donde se almacena los tipos de pregunta a realizar, los cuales pueden ser opción múltiple y V/F (Verdadero/Falso). Los mismos que se diferencian por el número de respuestas y el sentido de resolución que tienen, entre los atributos que conforman a la entidad ***EVAAB_TIPOPREGUNTA*** están los siguientes:

- ***tippre_id.-*** Representa el identificador para cada tipo de pregunta, su tipo de dato es ***int*** **(4)** de la forma ***autoincrement.***
- ***tippre_nombre.-*** Representa el nombre del tipo de pregunta sea este opción múltiple o V/F, su tipo de dato es **varchar (20).**

- ***tippre_descripcion.-*** Representa la descripción detallada para cada tipo de pregunta, su tipo de dato es ***varchar (50).***

2.2.3.1.9 Evaab_Cue_Estu

Representa la tabla donde se almacena el detalle de la resolución del cuestionario, el mismo que involucra al cuestionario y al estudiante. Entre los atributos que conforman a la entidad ***EVAAB_CUE_ESTU*** están los siguientes:

- ***cue_est_id.-*** Representa el identificador para cada cuestionario por estudiante, su tipo de dato es ***int (4)*** de la forma ***autoincrement***.
- ***cue_est_calificacion.-*** Representa la calificación otorgada a cada estudiante por la resolución de un cuestionario, su tipo de dato es ***smallint.***
- ***cue_est_fecha.-*** Representa la fecha en la cual se realizó el cuestionario, su tipo de dato es ***datetime***
- ***cue_est_resoid.-*** Representa el identificador de la resolución realizada, el cual es un atributo referencial a la tabla ***EVAAB_RESOLUCIONES***, su tipo de dato es ***int (10)***.

2.2.3.1.10 Evaab_Opciones

Representa la tabla donde se almacena las diferentes opciones seleccionadas por un estudiante al momento de resolver las preguntas, para poder cuantificar luego su resultado. Entre los atributos que conforman la entidad ***EVAAB_OPCIONES*** están los siguientes:

- ***opc_id.-*** Representa el identificador para cada opción elegida por el estudiante al momento de resolver la

pregunta, su tipo de dato es ***int (10)*** de la forma ***autoincrement.***

- ***opc_userid.-*** Representa el identificador del estudiante que esta resolviendo las preguntas, su tipo de dato es un ***int (10).***
- ***opc_seleccionadas.-*** Representa el conjunto de opciones seleccionadas en la resolución de cada pregunta, su tipo de dato es ***varchar (100).***
- ***opc_estado_sele.-*** Representa el estado de la pregunta al seleccionar una respuesta lo cual es necesario para el informe final, su tipo de dato es un ***smallint.***

2.2.3.1.11 Evaab_Resoluciones

Representa la tabla donde se almacena las diferentes resoluciones que tiene un cuestionario por estudiante. Entre los atributos que conforman la entidad **EVAAB_RESOLUCIONES** están los siguientes:

- ***reso_id.-*** Representa el identificador para cada resolución, su tipo de dato es ***int (10)*** de la forma ***autoincrement***.
- ***reso_fechainireso.-*** Representa la fecha de inicio de resolución del cuestionario, el tipo de dato es ***datetime.***
- ***reso_fechafinreso.-*** Representa la fecha de finalización de resolución del cuestionario, el tipo de dato es ***datetime.***

2.2.3.2 Definición de Relaciones

Se detallarán las relaciones encontradas y la cardinalidad que presentan entre entidades. Las relaciones que presenta **EVAAB** son las siguientes:

2.2.3.2.1 Rel_Eva_Eva_Cue_Estu

Representa la relación entre las entidades ***Evaab*** y ***Evaab_Cue_Estu***, la cual presenta una cardinalidad de uno a muchos respectivamente.

2.2.3.2.2 Rel_Eva_Estu_Cue_Est

Representa la relación entre las entidades ***Prefix_User***, la cual es una tabla de Moodle que servirá para la integración y ***Evaab_Cue_Estu.*** Esta presenta una cardinalidad de uno a muchos respectivamente, pero que solo es referenciada para no afectar la arquitectura.

2.2.3.2.3 Rel_Eva_Eva_Pre

Representa la relación entre las entidades ***Evaab*** y ***Evaab_Pregunta*** la cual presenta una cardinalidad de uno a muchos respectivamente.

2.2.3.2.4 Rel_Eva_Pre_Pis

Representa la relación entre las entidades ***Evaab_Pregunta*** y ***Evaab_Pista*** la cual presenta una cardinalidad de uno a muchos respectivamente.

2.2.3.2.5 Rel_Eva_Pre_Ref

Representa la relación entre las entidades ***Evaab_Pregunta*** y ***Evaab_Referencia*** la cual presenta una cardinalidad de uno a muchos respectivamente.

2.2.3.2.6 Rel_Eva_Pre_Res

Representa la relación entre las entidades ***Evaab_Pregunta*** y ***Evaab_Respuesta*** la cual presenta una cardinalidad de uno a muchos respectivamente.

2.2.3.2.7 Rel_Eva_Tipcue_Eva

Representa la relación entre las entidades ***Evaab_Tipocuestionario*** y ***Evaab*** la cual presenta una cardinalidad de uno a muchos respectivamente.

2.2.3.2.8 Rel_Eva_Tippre_Pre

Representa la relación entre las entidades ***Evaab_Tipopregunta*** y ***Evaab_Pregunta*** la cual presenta una cardinalidad de uno a muchos respectivamente

2.2.3.2.9 Rel_Eva_Est_Opc

Representa la relación entre las entidades ***Evaab_Estado*** y ***Evaab_Opciones*** la cual presenta una cardinalidad de uno a muchos respectivamente.

2.2.3.2.10 Rel_Eva_Pre_Opc

Representa la relación entre las entidades ***Evaab_Pregunta*** y ***Evaab_Opciones*** la cual presenta una cardinalidad de uno a muchos respectivamente.

2.2.3.2.11 Rel_Eva_Eva_Reso

Representa la relación entre las entidades ***Evaab*** y ***Evaab_Resolucion*** la cual presenta una cardinalidad de uno a muchos respectivamente.

2.2.3.2.12 Rel_Eva_Res_Opc

Representa la relación entre las entidades ***Evaab_Resolucion*** y ***Evaab_Opciones*** la cual presenta una cardinalidad de uno a muchos respectivamente.

2.2.3.3 Modelo Entidad – Relación

El modelo entidad – relación tiene una característica especial en su diseño ya que la entidad ***PREFIX_USER*** no forma parte del módulo de evaluación ***EVAAB***, sino que nos sirve para realizar la integración con ***Moodle*** por lo cual esta denotada con un color diferente. El diagrama se lo puede apreciar en la Fig. 2.27 a continuación:

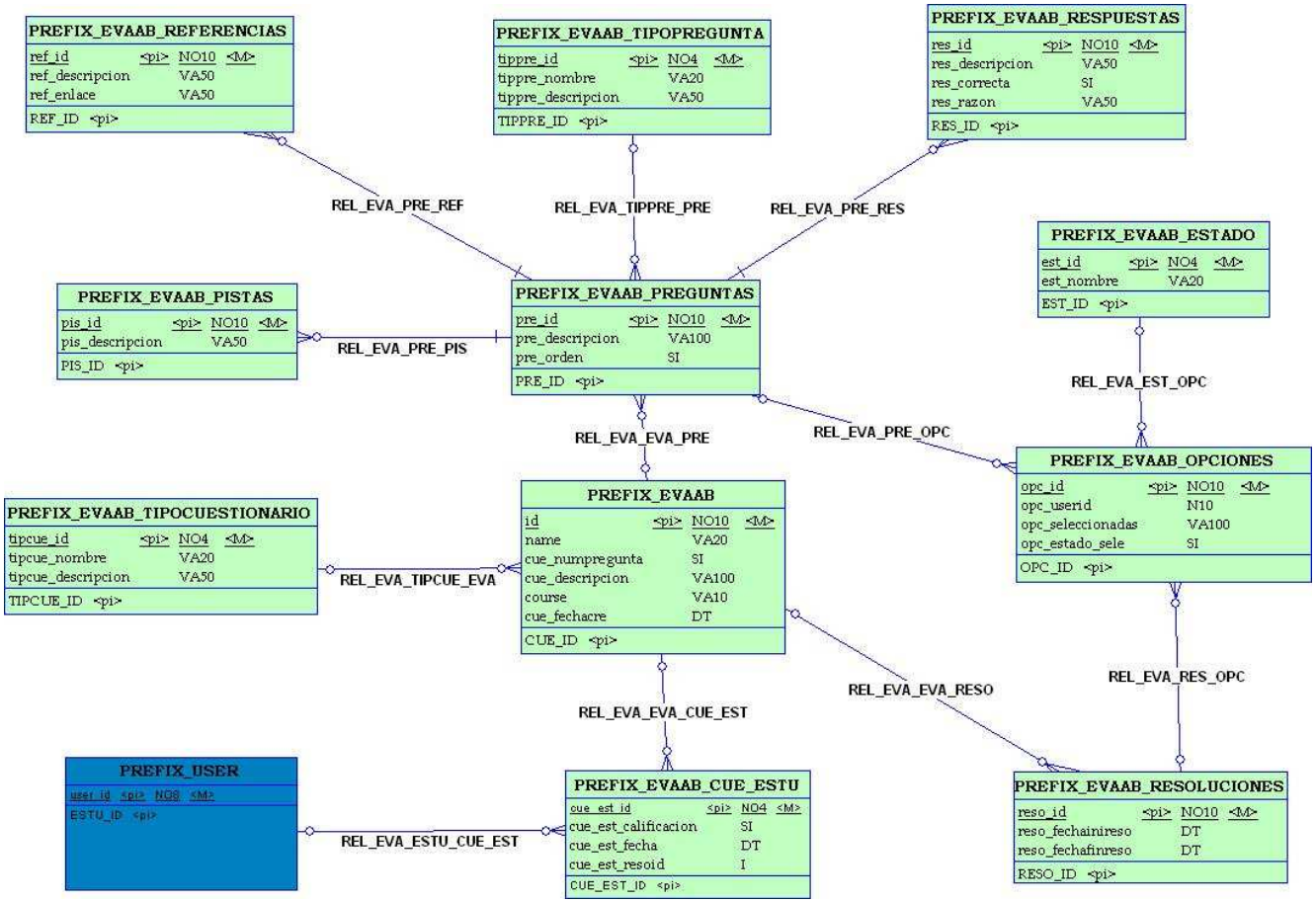

Fig. 2. 27: Modelo Lógico EVAAB

2.2.3.4 Modelo Físico

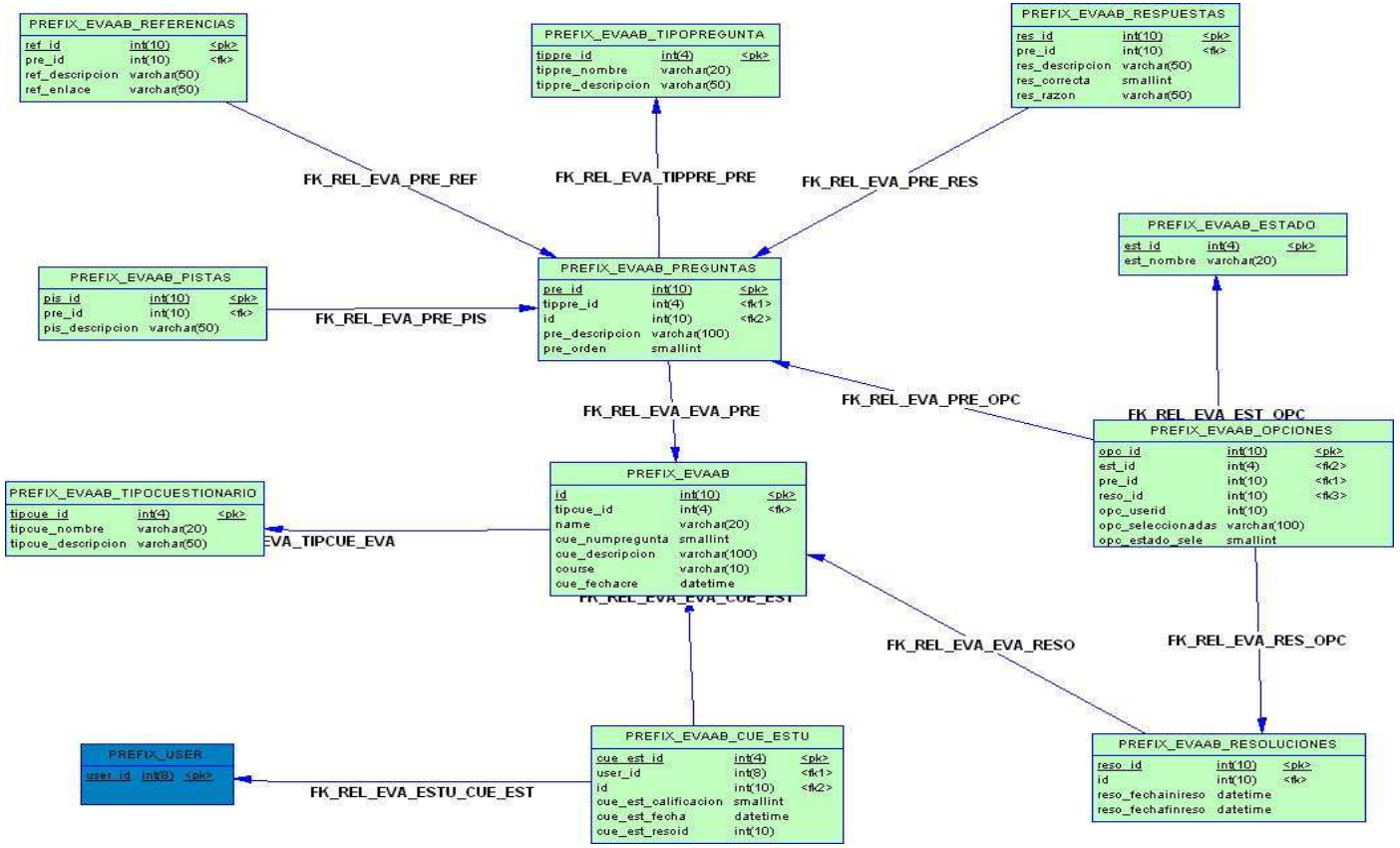

Fig. 2. 28: Modelo Físico EVAAB

2.3 DISEÑO DE INTERFACES

2.3.1 GENERALIDADES

Para el diseño de interfaces se utilizó la estandarización especificada por ***Moodle***[17] conjuntamente con el estándar definido por ***EVAAB***, en los cuales se describe todas las características de diseño generales que se debería tomar en cuenta al momento de la creación de las interfaces. Para nombrar a cada una de las interfaces se utilizó el prefijo ***EVAAB,*** cabe destacar además que las interfaces son generadas dinámicamente utilizando ***AJAX*** mediante la utilización de la librería ***XAJAX.***

2.3.2 ESTANDARIZACION

2.3.2.1 Diseño del Estándar de EVAAB

Para la creación de las interfaces dentro de ***EVAAB*** utilizamos un diseño propio el cual está basado en ***el concepto de componentes***, todo esto gracias a la utilización de ***AJAX.*** Entre las definiciones que se utilizó para la creación de las interfaces están las siguientes:

- Para la creación de los componentes se debe crear una interfaz contenedora la cual contendrá todos los componentes a administrar.
- Para la creación de cada componente se utiliza una ***tabla*** la cual contiene toda la información relacionado con la interfaz a presentar.
- ***La definición de títulos y encabezados*** dentro de la interfaz, deben ser posicionados de una forma centrada.
- La distribución de las interfaces ***presenta un orden jerárquico*** dentro de la página, es decir la interfaz con mayor importancia tiene prioridad de colocación.

[17] Interface Guidelines - http://docs.moodle.org/en/Interface_guidelines

- La navegación dentro de los componentes es ***compacta y dinámica***, es decir se debe administrar y generar de una forma dinámica dentro del mismo contexto.

- Todos los componentes de ***EVAAB*** utilizan la notación estándar en cuanto a colores, formatos y formas de navegación definidos por Moodle.

2.3.2.2 Estándar de Moodle

Moodle para la definición de las interfaces se basa en un archivo denominado con el nombre de ***Interface Guidelines***[18] donde se puede apreciar los puntos que se debe definir en la creación de las nuevas interfaces. A continuación se detalla algunos de los elementos que se ha creído más relevantes:

- Usar las mínimas interfaces requeridas para realizar el trabajo.
- Imprimir las cabeceras de las interfaces con las función estándar de Moodle, ***print_heading.***
- Imprimir las tablas contenedoras a través de la función estándar de Moodle, ***print_simple_box.***
- Usar para la impresión de tablas la función estándar de Moodle, ***print_table*** donde sea posible.
- En cuanto a la navegación dentro de Moodle las páginas deberán llamar a la función ***print_header*** y proporcionar el path correspondiente para ser desplegado. Donde sea posible debería mirarse así: ***COURSE >> INDEX >> INSTANCE >> SUBPAGES***.
- En cuanto a la definición de los ***URL*** se debe definirlas en su forma más corta sin underscore en los nombres de parámetros o nombres de archivos.

2.3.3 DEFINICIÓN DE INTERFACES

2.3.3.1 Evaab_Principal

Esta interfaz describe el comportamiento general del módulo de evaluación, **Evaab,** en el cual se presenta un resumen con todas las características que se manejarán dentro del mismo. Entre estas tenemos la carga de los datos y la ayuda como se lo puede apreciar en la Fig. 2.29:

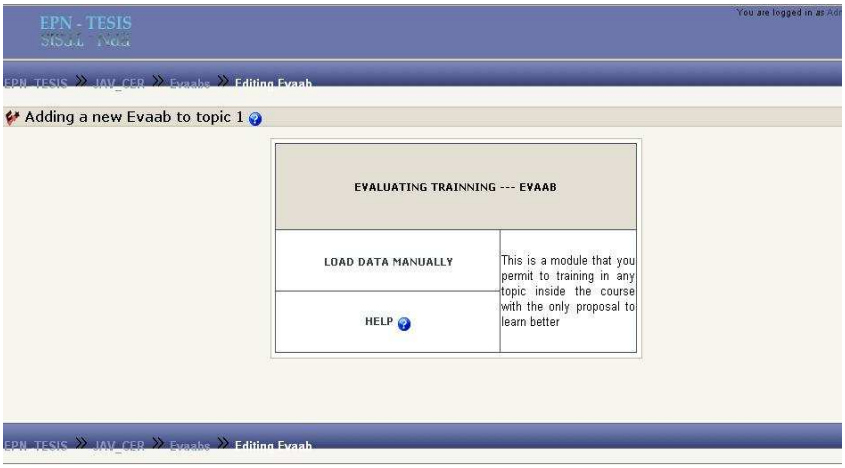

Fig. 2. 29: Pantalla Principal

2.3.3.2 Evaab_Cuestionario

Esta interfaz describe la parte principal del módulo de evaluación **Evaab,** la cual consiste en gestionar todo el cuestionario conjuntamente con los elementos que intervienen en la misma como son **las preguntas, respuestas, pistas y referencias**. Esto esta a cargo del **Profesor,** la interfaz principal se la puede apreciar a continuación en la Fig. 2.30.

---

[18] Interface Guidelines - http://docs.moodle.org/en/Interface_guidelines

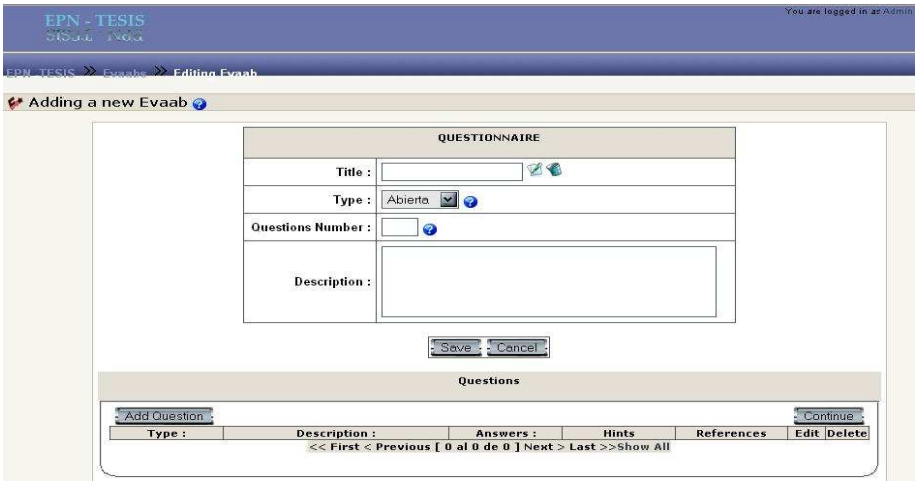

Fig. 2. 30: Evaab_Cuestionario

Cabe destacar que las interfaces relacionadas con los elementos del cuestionario como preguntas, respuestas, pistas y referencias forman un componente dentro de la interfaz contenedora Evaab_Cuestionario. Las cuales se explicarán a continuación.

2.3.3.3 Evaab_Pregunta

Esta interfaz describe la gestión de la preguntas dentro del cuestionario, cabe destacar que la creación de las preguntas nos permitirá continuar con la gestión de los otros componentes como son **respuestas, pistas y referencias** ya que se encuentran ligadas a éstas. La interfaz se la puede apreciar a continuación en la Fig. 2.31.

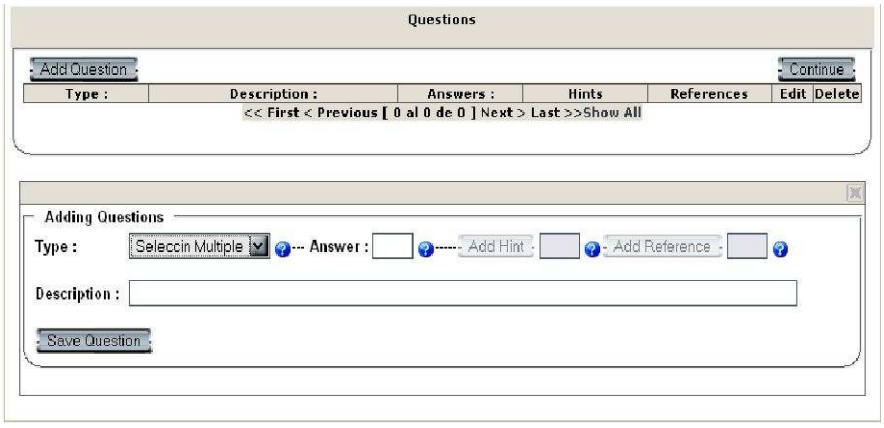

Fig. 2. 31: Pantalla Ingreso de Pregunta

Cada vez que se realice el ingreso de una pregunta se puede observar que la tabla la cual ***visualiza las preguntas*** se incrementa en un registro, denotado por un mensaje enmarcado con color rojo como se muestra en la Fig. 2.32 a continuación:

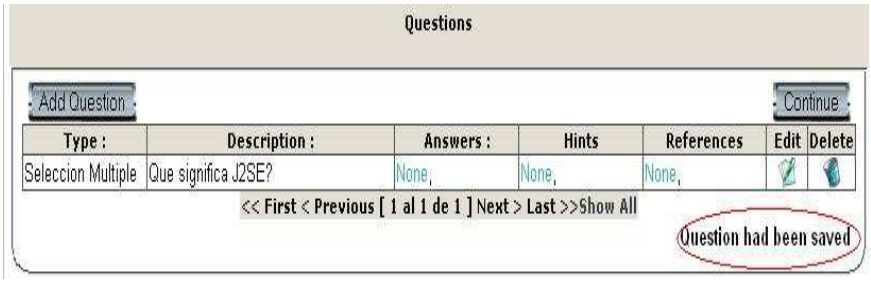

Fig. 2. 32: Pantalla Grabado de Pregunta

2.3.3.4 Evaab_Respuesta

Esta interfaz describe las respuestas dadas para cada pregunta las cuales dependerán del tipo de pregunta a ingresar ya que puede ser selección múltiple o verdadero/falso. La interfaz de respuestas se las puede apreciar en la Fig. 2.33 a continuación:

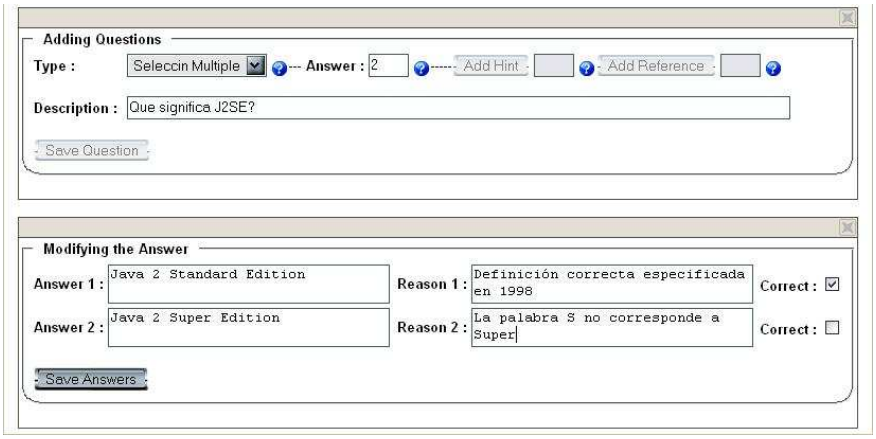

Fig. 2. 33: Pantalla Ingreso de Respuesta

Cada vez que se realice el ingreso de una respuesta se puede observar que la tabla la cual ***visualiza las preguntas*** cambia en la parte de las ***Respuestas,*** denotado por un mensaje enmarcado con color rojo como se muestre en la Fig. 2.34 a continuación:

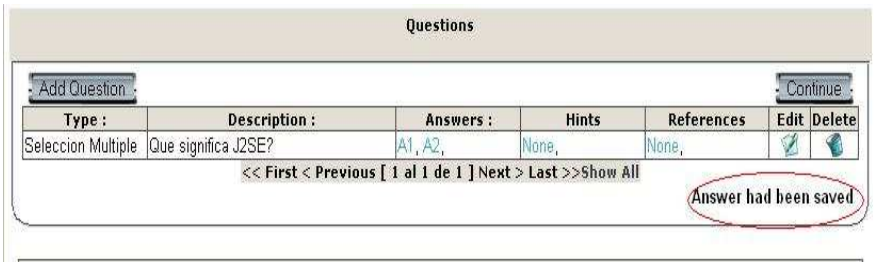

Fig. 2. 34: Pantalla Grabado de Respuesta

2.3.3.5 Evaab_Pista

Esta interfaz describe las ayudas cortas dadas para cada pregunta, las mismas que permitirán facilitar la resolución de cada una de ellas. La interfaz de las pistas se las puede apreciar en la Fig. 2.35 a continuación:

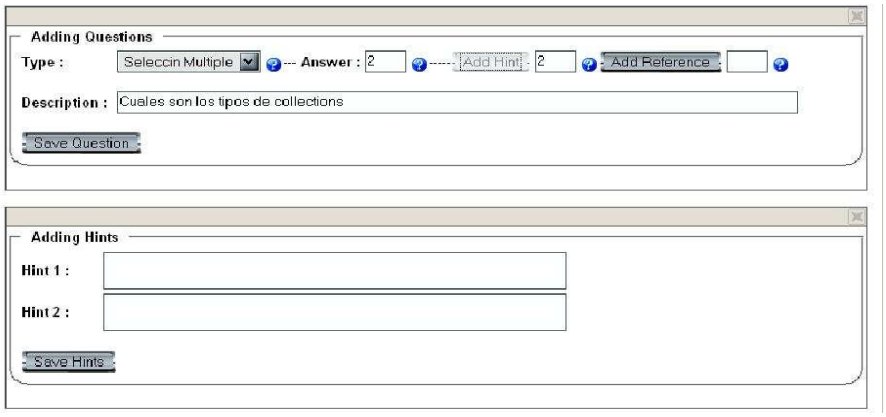

Fig. 2. 35: Pantalla Ingreso de Pista

Cada vez que se realice el ingreso de una pista se puede observar que la tabla la cual ***visualiza las preguntas*** cambia en la parte de las ***Pistas,*** denotado por un mensaje enmarcado con color rojo como se muestra en la Fig. 2.36 a continuación:

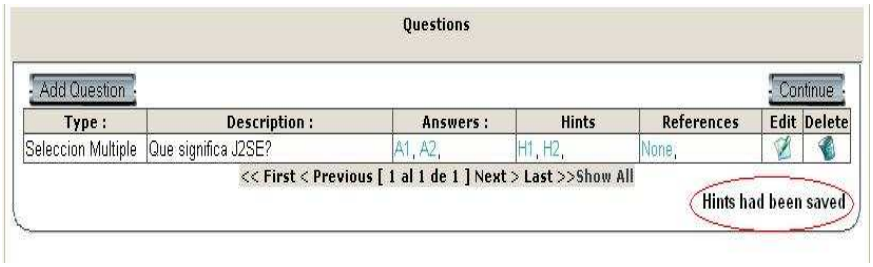

Fig. 2. 36: Pantalla Grabado de Pista

2.3.3.6 Evaab_Referencia

Esta interfaz describe las referencias dadas para cada pregunta las cuales son un elemento de ayuda para la resolución de las mismas que se diferencian de las pistas por la ayuda más exacta que brindan, ya que en éstas se puede utilizar todos lo recursos utilizados por Moodle. La interfaz de las referencias se las puede apreciar en la Fig. 2.37 a continuación:

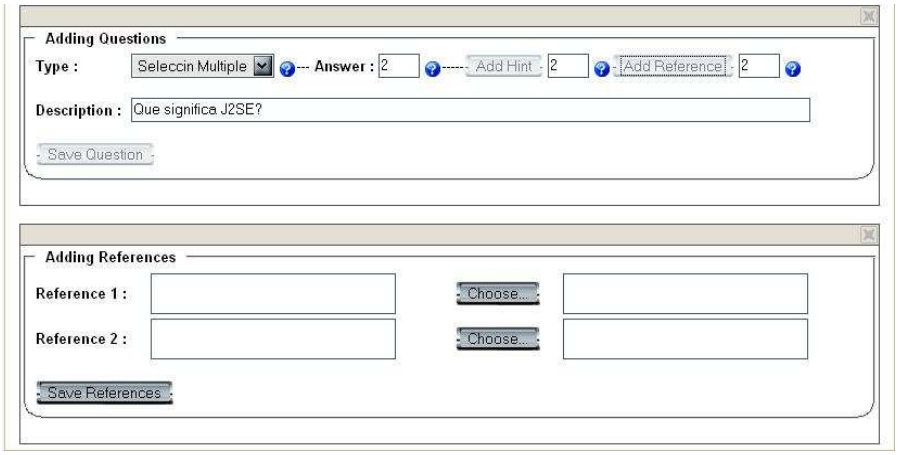

Fig. 2. 37: Pantalla Ingreso de Referencia

Cada vez que se realice el ingreso de una referencia se puede observar que la tabla la cual ***visualiza las preguntas*** cambia en la parte de las ***Referencias,*** denotado por un mensaje enmarcado con color rojo como se muestra en la Fig. 2.38 a continuación:

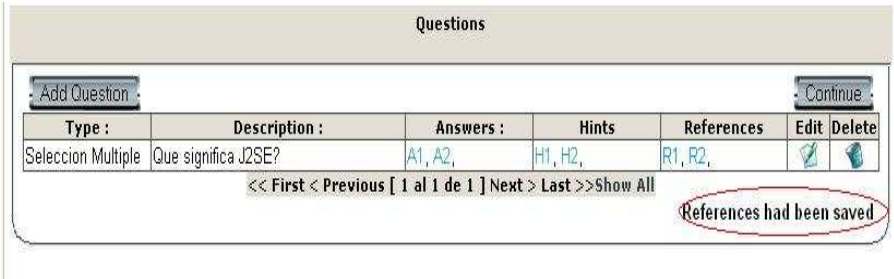

Fig. 2. 38: Pantalla Grabado de Referencia

2.3.3.7 Evaab_Resolucion

Esta interfaz describe la resolución del cuestionario la cual esta a cargo del Estudiante, al igual que ***Evaab_Cuestionario*** presenta los mismos elementos como son las preguntas, respuestas, pistas y referencias pero en una perspectiva diferente la cual es solo de visualización.

La interfaz inicial que se presenta al estudiante antes de comenzar a resolver el Cuestionario se la pude apreciar en la Fig. 2.39 a continuación:

Fig. 2. 39: Pantalla Inicial Resolución

Como se puede apreciar en ésta interfaz se tiene los datos del cuestionario los cuales se presentan en forma de visualización, ya que para modificar se necesita tener el perfil de ***Profesor***. Pero si se intentase realizarlo en el perfil de estudiante se presentará un mensaje con la validación respectiva, la cual se la puede apreciar en la Fig. 2.40 a continuación:

Fig. 2. 40: Mensaje de Validación Perfil

Una vez comenzada la resolución del cuestionario por parte del estudiante se puede apreciar todos los componentes que forman parte del mismo como ***son preguntas, respuestas, pistas y referencias*** pero desde otra perspectiva. La interfaz inicial de resolución se la puede apreciar en la Fig. 2.41 a continuación:

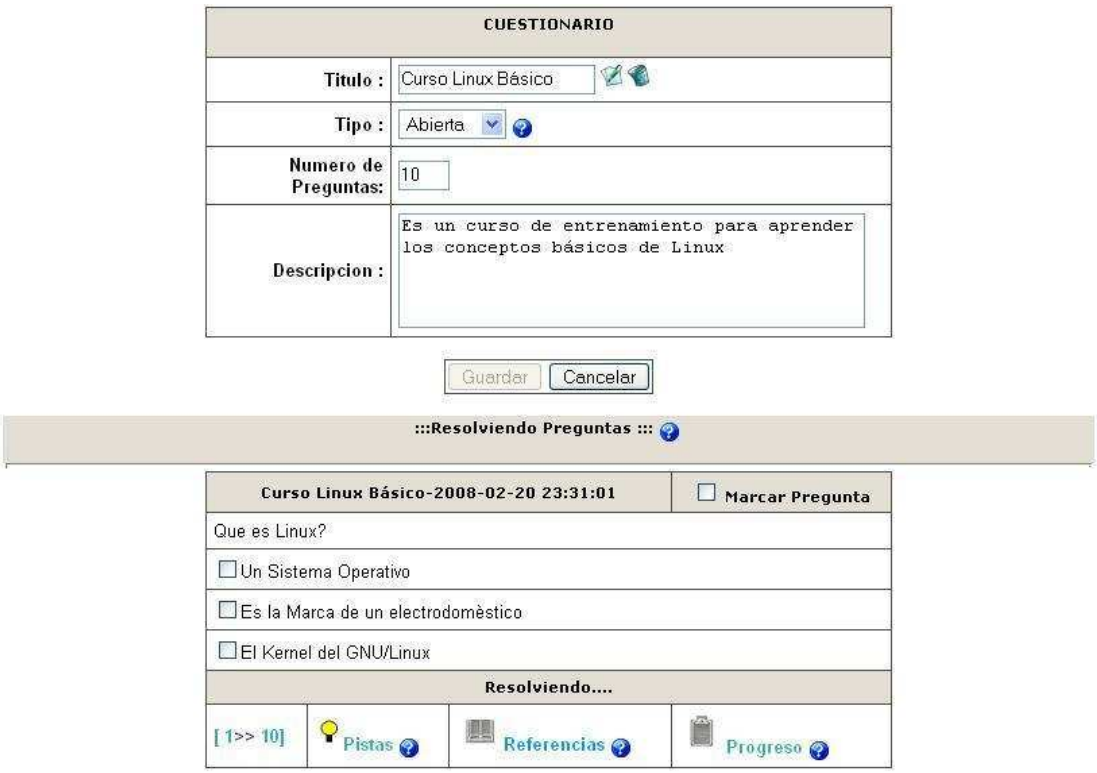

Fig. 2. 41: Pantalla Resolución Cuestionario

Siguiendo el mismo estándar utilizado en el ingreso del Cuestionario se puede decir que los elementos ***como preguntas, respuestas, pistas y referencias*** incluyendo ***el progreso*** también son componentes que forman parte de una interfaz contenedora llamada en esta perspectiva ***Evaab_Resolucion***. Los cuales los detallaremos a continuación.

2.3.3.7.1 Evaab_Resolucion_Pregunta_Respuesta

Esta interfaz define todas las preguntas que forman parte del cuestionario juntamente con sus respuestas, las cuales serán mostradas y resueltas progresivamente una a una. En la Fig. 2.42 se

puede observar a una pregunta conjuntamente con sus respuestas.

Fig. 2. 42: Pantalla Resolución Pregunta

Cabe destacar que cada pregunta al ser contestada será evaluada con el mensaje correspondiente, ya sea que ***este correcta, incorrecta o incompleta.*** Para esta última es necesario marcar el check box el cual se lo puede apreciar marcada con color rojo en la Fig. 2.43 a continuación:

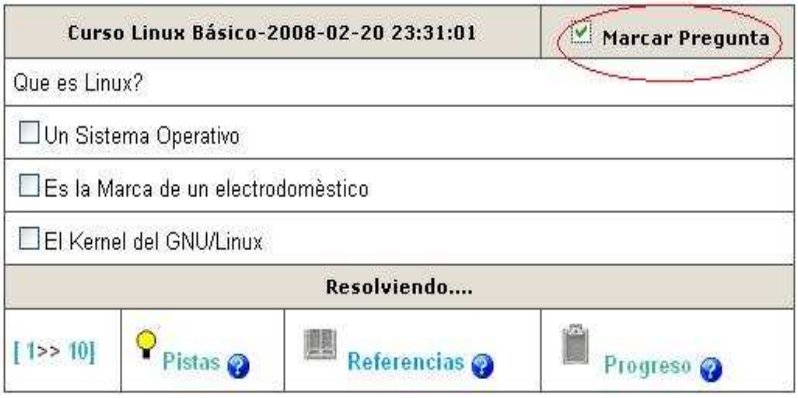

Fig. 2. 43: Pantalla Pregunta Marcada

Sea cual fuese el caso, el mensaje para una pregunta ***correcta, incorrecta e incompleta*** se los puede apreciar en las figuras 2.44, 2.45, 2.46 a continuación:

Fig. 2. 44: Mensaje de Pregunta Correcta

Fig. 2. 45: Mensaje de Pregunta Incorrecta

Fig. 2. 46: Mensaje de Pregunta Incorrecta

2.3.3.7.2 Evaab_Resolucion_Pistas

Esta interfaz describe las ayudas cortas para una pregunta dada, las cuales son elementos de ayuda simples que permiten llegar a la solución más fácilmente. Estas aparecen al estudiante solo con permisos de lectura para que puedan servir en la toma de decisiones ***pero siempre y cuando el cuestionario sea de tipo abierto*** ya que el ***tipo de cuestionario cerrado no contiene elementos de apoyo,*** las mismas que se las puede apreciar en la Fig. 2.47 a continuación:

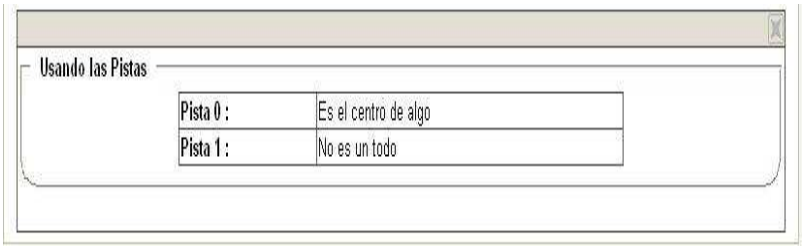

Fig. 2. 47: Pantalla Resolución Pistas

Se puede dar el caso de que no se haya ingresado ninguna pista para esta pregunta, para lo cual se procederá a mostrar el mensaje respectivo; que indica que no existen pistas. El cual se lo puede apreciar en la Fig. 2.48 a continuación:

Fig. 2. 48: Mensaje de Pistas No existentes

2.3.3.7.3 Evaab_Resolucion_Referencias

Esta interfaz describe las referencias para una pregunta dada las cuales al igual que las pistas son elementos de ayuda que permiten llegar a la solución más fácilmente, diferenciándose de las pistas por ser un recurso de ayuda más exacto, También se presenta al estudiante en una forma de solo lectura ***pero siempre y cuando el cuestionario sea de tipo abierto ya que el tipo de cuestionario cerrado no contiene elementos de apoyo***; se la puede apreciar en la Fig. 2.49 a continuación:

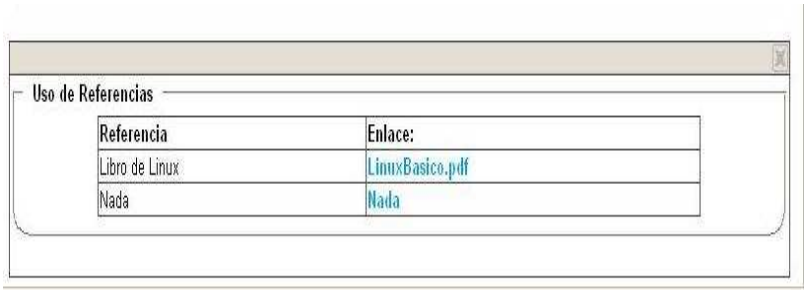

Fig. 2. 49: Pantalla Resolución Referencias

Se puede dar el caso de que no se haya ingresado ninguna referencia para esta pregunta, para lo cual se procederá a mostrar el mensaje respectivo que indica que no existen referencias. Lo cual se lo puede apreciar en la Fig. 2.50 a continuación:

Fig. 2. 50: Mensaje de Referencias No existentes

Además de éstas interfaces también se encuentra la ***interfaz de progreso*** y la ***interfaz de calificación del Estudiante***, las cuales también forman un componente dentro de la interfaz ***Evaab_Resolucion***. Las mismas que se explicarán a continuación.

2.3.3.8 Evaab_Progreso

Esta interfaz describe el progreso que tiene un estudiante en un momento dado en la resolución de un Cuestionario, es decir se presenta un ***resumen general*** donde se puede ***visualizar las preguntas que están correctas, incorrectas e incompletas junto con sus porcentajes de aceptación***. Ésta forma parte de los elementos retroalimentativos con que cuenta el estudiante para la resolución del Cuestionario, el cual se lo puede apreciar en la Fig. 2.51 a continuación:

Progreso	
Respuestas	**Porcentage**
Correcta: 2 , 3 ,	20%
Incorrecta: 1 , 4 ,	20%
Incompleta: 5 , , , , , ,	60%
Ir al Final para validar	

Fig. 2. 51: Pantalla de Progreso

2.3.3.9 Evaab_Calificación

Esta interfaz describe la calificación que tiene un estudiante al momento de finalizar la resolución del Cuestionario, en la cual se puede apreciar datos como el nombre del cuestionario, fecha de resolución, nombre del estudiante y dos elementos sumamente importantes que son ***la nota propiamente y el enlace a la generación del informe***. La cual se la puede apreciar en la Fig. 2.52 a continuación:

RESULTADO-CUESTIONARIO	
Titulo :	Curso Linux Básico
Nombre del Estudiante:	Pepito Cerezo
Fecha:	2008-02-21 22:01:29
Calificacion:	4/(20)
Reporte:	Resultado de Test

Fig. 2. 52: Pantalla de Calificación

2.3.3.10 Evaab_Informe

Como se describió anteriormente una vez que se termina la resolución de un cuestionario se genera un informe del estudiante acerca del proceso en formato ***PDF***, el cual muestra en resumen de como se realizó la resolución del mismo por parte del estudiante mostrando las preguntas junto con sus elementos como respuestas, pistas y referencias. El mismo que denota la respuesta correcta junto con la respuesta seleccionada, lo cual se puede apreciar en la Fig. 2.53 a continuación:

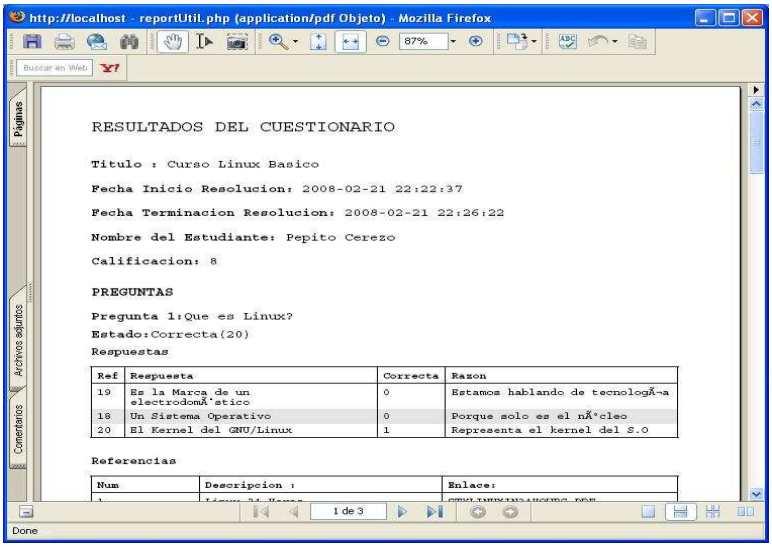

Fig. 2. 53: Pantalla de Calificación

2.3.3.11 Evaab_Ayuda

La interfaz de ayuda representa un punto muy importante dentro del módulo de Evaluación ***EVAAB*** ya que servirá de referencia tanto para los profesores como para los estudiantes para despejar las dudas que se presentan durante el proceso. Una de las interfaces de ayuda se la puede apreciar en la Fig. 2.54 a continuación:

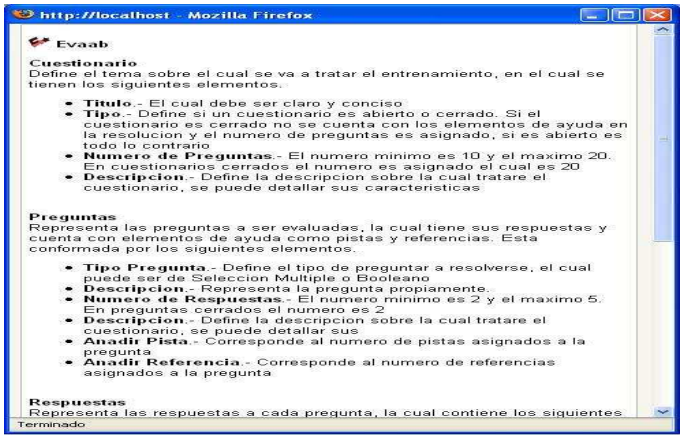

Fig. 2. 54: Pantalla de Ayuda

CAPÍTULO III

3. IMPLEMENTACIÓN Y PRUEBAS

3.1 JUSTIFICACIÓN DEL AMBIENTE DE DESARROLLO

3.1.1 AMBIENTE DE DESARROLLO

Para el desarrollo del módulo de evaluación ***EVAAB*** se utilizó herramientas ***OPEN SOURCE*** para el proceso en su mayoría, de tal forma de seguir con la filosofía manejada por ***Moodle***. El ambiente de desarrollo estuvo conformado por los siguientes elementos:

- LMS Moodle
- Servidores Web
- Base de Datos y Herramienta de Administración
- Herramienta de Diseño
- Herramienta para el Desarrollo con PHP
- Herramienta para seguimiento del proceso de XP
- Utilitarios

3.1.1.1 Definición de Elementos

3.1.1.1.1 LMS Moodle

Uno de los elementos más importantes en el desarrollo de ***EVAAB*** fue el ***LMS Moodle*** ya que éste es la base sobre la cual se realizará el desarrollo de ***EVAAB*** como tal, se utilizó la versión 1.6.3

3.1.1.1.2 Servidores Web

Como servidores Web utilizamos dos, uno para el manejo de ***PHP*** y otro para el manejo de los ***Servlets***, éste último fue necesario para manejar la herramienta de seguimiento de ***XP***.

Para el manejo de PHP se utilizó un ambiente integrado llamado ***XAMPP***[19] en su versión 1.6.6 RC1 el cual incluye un sinnúmero de componentes como: ***Apache, MySQL, PHP, Filezilla*** entre ellos el servidor Web Apache utilizado por Moodle y para el manejo de ***Servlets*** se utilizó a ***Tomcat***[20] en su versión 6.0.

3.1.1.1.3 Base de Datos y Herramienta de Administración

La base de datos utilizada fue la definida por la arquitectura de Moodle, es decir ***MySQL*** en su versión 5.0.21 Para el manejo y administración de la base de datos se utilizó las herramientas propias de MySQL[21], entre éstas tenemos a:

- ***MySQL Browser*** en su versión 1.2.12. en lo que se refiere a la creación, ejecución de queries.
- ***MySQL Administrator*** en su versión 1.2.12 en lo que se refiere a la administración de la base de datos.

3.1.1.1.4 Herramienta de Diseño

Para manejar el diseño del módulo ***EVAAB*** dentro del proceso de desarrollo utilizamos dos herramientas: uno para el diseño de la base de datos llamada ***Power Designer*** [22] en su versión 11.0, la cual es una herramienta comercial y ***JUDE*** [23] en su versión 3.2 la cual la utilizamos para realizar los modelos necesarios para las diferentes iteraciones. Se utilizó a ***Power Designer*** debido a que las herramientas ***Open Source*** que existen actualmente no cumplen con la expectativa de trazabilidad esperada en cuanto a los modelos lógico y físico.

---

[19] Xampp - http://www.apachefriends.org/en/xampp.html
[20] Tomcat - http://tomcat.apache.org/
21 MySQL Tools - http://www.mysql.com/products/tools/
[22] Power Designer - http://www.sybase.com/products/modelingmetadata/powerdesigner
[23] Jude - http://jude.change-vision.com/jude-web/index.html

3.1.1.1.5 Herramienta para el Desarrollo con PHP

Se utilizó como herramienta de desarrollo para PHP al ***IDE ECLIPSE EUROPA*** [24] en su versión 3.3.0. En el cual se tuvo que integrar los plugins correspondientes para ***PHP*** para poder utilizarlo, el cual tiene el nombre de ***PHPECLIPSE*** [25]

3.1.1.1.6 Herramienta para seguimiento de XP

Tomando en cuenta que la metodología de desarrollo, ***Extreme Programming***, es una metodología ágil el cual su principal característica es reducir tiempos de desarrollo podríamos no contemplar algún elemento fundamental dentro de la misma. Razón por la cual se vio la necesidad de utilizar una herramienta que sirva de ayuda para gestionar los procesos dentro ***XP*** en lo que se refiere a la planificación, para lo cual se utilizó a ***Xplanner***[26].

3.1.1.1.7 Utilitarios

En cuanto a los utilitarios que se utilizó para el desarrollo del módulo ***EVAAB*** están los siguientes:

- **Editor de Textos.-** Se utilizó a ***OpenOffice***[27] en su versión 2.0 para la creación de la documentación, pero se tuvo que migrar a ***Microsoft Office 2003***[28] en la parte final del documento por compatibilidad.
- **Editor de Imágenes.-** Se utilizó a ***Inkscape***[29] en su versión 0.44 para la elaboración de los gráficos dentro del módulo ***EVAAB***.

[24] Eclipse Europa - http://www.eclipse.org/europa/
[25] PHPEclipse - http://www.phpeclipse.de/tiki-view_articles.php
[26] XPlanner - http://xplanner.org/
[27] OpenOffice - http://www.openoffice.org/
[28] Microsoft Word - http://office.microsoft.com/en-us/word/default.aspx
[29] Inkscape - http://www.inkscape.org/

3.1.2 JUSTIFICACIÓN

El ambiente de desarrollo fue elegido de acuerdo a afinidades y experiencia en el proceso de desarrollo de software. Las herramientas utilizadas para la consecución del proyecto son herramientas en su mayoría de ***código abierto,*** las cuales han sido utilizadas en proyectos afines al mismo otorgando muy buenos resultados.

En lo que respecta específicamente a las herramientas de desarrollo se puede mencionar que se revisó algunas opciones, pero al final se optó por ***PHP-ECLIPSE*** por la experiencia que se tenía en el manejo de este IDE en otros proyectos.

3.2 IMPLEMENTACIÓN DEL MÓDULO DE EVALUACIÓN

3.2.1 ESTÁNDAR DE CODIFICACIÓN PARA EVAAB

En lo que respecta al desarrollo del código dentro de ***EVAAB*** se utilizó las siguientes convenciones:

- En cuanto a los comentarios éstos tendrán las siguientes notaciones:
  - Todas las clases a implementarse deberán empezar con ***comentario cabecera*** en el cual se establecerán todos los datos principales de la clase como fecha y hora de creación, autor entre otros. El formato que se utilizó es el siguiente:

```
/**
 * Created on dd/mm/yyyy
 * ESCUELA POLITECNICA NACIONAL - E.P.N
 * Quito - Ecuador
 * Author: Marco Antonio Toscano Freire
 * Hora Creación: hh:mm:ss
 * Descripción Clase:*/
```

  - Cada método dentro de una clase tendrá un comentario que estará conformado por la descripción del método junto con cada uno de los parámetros que lo conforman más el tipo de retorno. El formato que se utilizó es el siguiente:

```
/**
 * Method Description
 * @param param1: description
 * @param param2: description
  * @return param3: description
*/
```

- El nombre de las diferentes clases dentro de ***EVAAB*** presenta un sufijo para identificar a la sección a la que pertenece, el cual puede ser uno de los siguientes ***Model***, ***View***, ***Controller***, ***Util***.
- Para el nombre de los métodos y variables se utilizó la ***notación de Camello***, la cual consiste en describir un nombre utilizando la primera palabra con minúsculas seguida de la segunda palabra con la primera letra mayúscula y lo demás con minúscula así sucesivamente para las siguientes palabras si existiesen.
- Se utilizó una identación de un ***TAB*** para cada elemento que se vaya creando en orden jerárquico.
- Todos los mensajes o cadenas a mostrarse dentro del aplicativo deberán llamarse a través del método estándar de Moodle, ***get_string()***, para lo cual deberán definirse anteriormente en los archivos de internalización.

3.2.2 DEFINICIÓN DE EVAAB

Para poder realizar la implementación de ***EVAAB*** se tuvo que primeramente realizar un estudio profundo del e-learning ***Moodle***, lo cual implicó conocer como está estructurado el mismo, conocer la arquitectura, conocer las formas de programación utilizadas, estándares aplicados etc. Cabe destacar que este punto fue el proceso más largo en el desarrollo de ***EVAAB.***

Con fundamentos sólidos acerca del e-learning ***Moodle*** en cuanto a los puntos anteriormente definidos se procedió a realizar la implementación del módulo de evaluación ***EVAAB*** para lo cual primero definiremos su ***Estructura,*** para luego continuar con la definición de sus ***Componentes*** y todo el proceso de implementación.

3.2.2.1 Estructura de EVAAB

La estructura que presenta ***EVAAB*** definida jerárquicamente se la puede apreciar en la Fig. 3.1 a continuación:

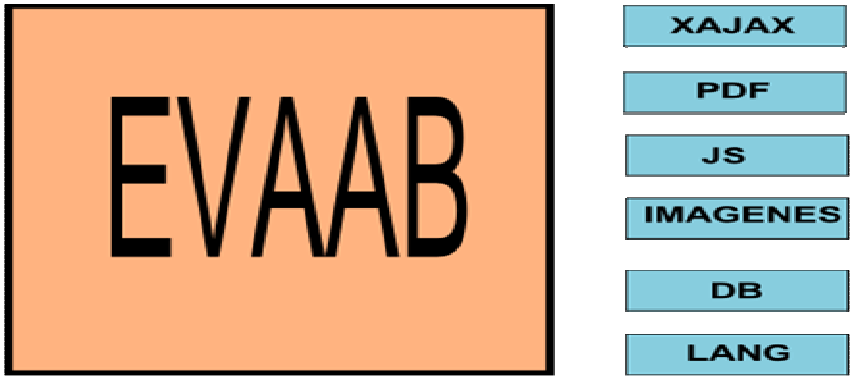

Fig. 3. 1: Estructura de EVAAB

El directorio ***/xajax*** es uno de los directorios más importante dentro del módulo de evaluación ***EVAAB,*** el cual contiene todos los archivos necesarios para manejar ***AJAX*** dentro de ***EVAAB.***

El directorio ***/pdf*** contiene todos los archivos relacionados con la utilización y generación de ***PDF,*** los cuales son necesarios para la generación de los ***informes*** una vez realizado la resolución de un cuestionario. Además encontramos una carpeta ***fonts como subdirectorio*** donde se encuentran todas las fuentes respectivas que son utilizadas en el informe.

El directorio ***/js*** contiene todos los archivos relacionados con ***JavaScript,*** los cuales me permiten realizar las validaciones de los elementos HTML dentro del módulo de evaluación ***EVAAB.***

El directorio ***/images*** contiene todas las imágenes utilizadas en ***EVAAB***, como imágenes para botones, cabecera etc.

El directorio ***/db*** contiene los archivos relacionados con la base de datos, tomando en cuenta que ***EVAAB*** fue implementado utilizando ***MySQL*** solo encontramos los archivos relacionados a éste.

En el directorio ***/lang*** encontramos los archivos de ayuda así como los archivos de internalización para las diferentes cadenas utilizadas tanto para los lenguajes de español e inglés bajo las carpetas ***en_utf8*** y ***es_utf8*** respectivamente.

Cabe destacar que el módulo de evaluación ***EVAAB*** dentro de la estructura de ***Moodle*** se encuentra ubicado en la carpeta referente a los módulos que tiene el nombre de ***mod,*** lo cual se puede apreciar en la Fig. 3.2 a continuación.

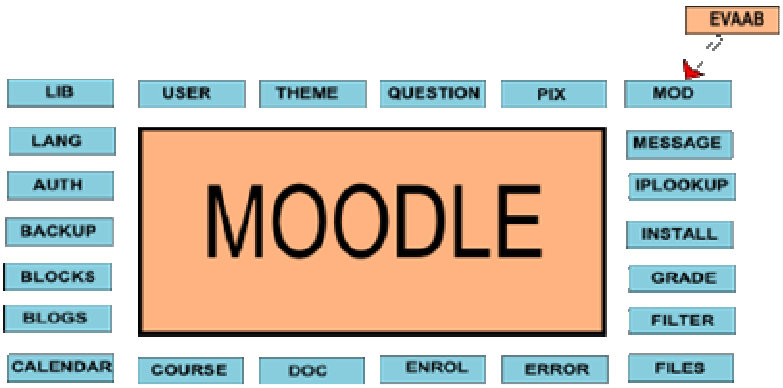

Fig. 3. 2: Evaab dentro de Moodle

Además de las carpetas mencionadas anteriormente dentro del módulo de evaluación ***EVAAB*** que representan los recursos dentro del mismo también encontramos los archivos propios de ***EVAAB,*** los cuales representan el ***CORE*** del aplicativo que serán explicados posteriormente.

3.2.2.2 Capas de EVAAB

EVAAB fue construido utilizando el concepto de un patrón de diseño muy conocido en el mundo de objetos denominado con el nombre de ***MVC (Model – View – Controller)*** [30] que tiene como finalidad separar los componentes de un aplicativo en tres capas las cuales son ***capa de datos, capas de vista (interfaz de usuario)*** y ***capa de Control*** de tal forma que cada capa maneje sus tareas específicas conjuntamente con responsabilidades con las otras capas.

Basado en este patrón se construyó el módulo de evaluación ***EVAAB*** específicamente el ***CORE,*** se tomo en cuenta las siguientes especificaciones:

- La capa referente al ***modelo*** dentro de ***EVAAB*** se encuentra representada por las clases persistentes conjuntamente con las librerías de ***Moodle.*** Las cuales son utilizadas para manejar las diferentes operaciones de la base de datos**,** especificadas puntualmente en su librería llamada ***datalib.php*** para lo que se crearon funciones propias de ***EVAAB*** que hacían uso de estas para el manejo de los datos.

- La capa referente a la ***vista*** dentro de ***EVAAB*** que se refiere a las interfaces de usuario se encuentra representado por las páginas html definidas por ***Moodle*** conjuntamente con algunos archivos php de ***EVAAB,*** incluidos los archivos para manejo de ***Ajax.***

- La capa referente al ***controlador*** dentro de ***EVAAB*** se encuentra representada por las clases que manejan la lógica del negocio, conformados por las archivos que manejan cada uno de los elementos del Cuestionario como son preguntas, respuestas, pistas, referencias y el cuestionario propiamente.

3.2.2.3 Definición de Componentes Complementarios a EVAAB

Antes de explicar la parte más importante de ***EVAAB*** que es el ***CORE*** es necesario y fundamental revisar los componentes complementarios que hacen de ***EVAAB*** un módulo diferente dentro del contexto acostumbrado por ***Moodle*** para desarrollo de módulos enfocados en un curso. Entre estos componentes están los siguientes:

---

[30] MVC - http://en.wikipedia.org/wiki/Model-view-controller

3.2.2.3.1 XAJAX

XAJAX [31] es una librería enfocada directamente al manejo de ***AJAX*** a través de ***PHP***, actualmente es una de los proyectos que se encuentran en estudio dentro de ***Moodle.***

EVAAB utiliza esta librería para el manejo de sus interfaces en todo su contexto excluyendo las interfaces que forman parte propia de ***Moodle***. Entre las características que se podría mencionar acerca de ***XAJAX*** están las siguientes:

- Es un producto ***Open Source***, compatible con la mayoría de navegadores .
- ***XAJAX*** tiene compatibilidad con PHP 4.3.x o PHP 5.x
- Se puede llamar a funciones php del Backend dentro del ambiente del cliente. Es decir funciones del Backend se encuentren disponibles en el FrontEnd
- ***XAJAX*** genera el 99% de javascript
- ***XAJAX*** tiene un API donde se define cada una de las clases con sus respectivas funciones a utilizar.
- ***XAJAX*** presenta una buena documentación.

3.2.2.3.2 CPdf y Cezpdf [32]

Este componente adicional a ***EVAAB*** es un proyecto ***OPEN SOURCE*** que nos permite generar documentos en formato ***PDF***, sus elementos tienen licencia pública siguiendo con la filosofía del Software Libre para que todas las personas que deseen puedan disfrutar de ellas. Su autor tiene el nombre de ***Wayne Munro*** y actualmente estamos utilizando la versión ***0.09***

[31] XAJAX - http://xajaxproject.org/
[32] CPdf y Cezpdf - http://manual.phpdoc.org/HTMLSmartyConverter/HandS/Cpdf/Cpdf.html

Este componente es el que nos permite generar el informe en formato ***PDF***, el mismo que está conformado por dos clases php relacionadas entre sí conjuntamente con una carpeta de fuentes que es la que determina el formato de letra a utilizar. Los elementos que conforman este componente son los siguientes:

- ***class.pdf.php.-*** Esta es la ***clase principal*** donde se encuentran definidos todas las variables y funciones para el tratamiento de documentos en formato PDF. Aquí encontramos funciones para el manejo del tipo de documento a imprimir, funciones para manejar la información general del documento, funciones para manejar las imágenes etc.

- ***class.ezpdf.php.-*** Esta es una clase que extiende de la clase principal ***class.pdf.php***, la cual utiliza las funcionalidades básicas de ésta y crea funciones más útiles y/o entendibles con el único objetivo que el usuario no tenga que conocer todos los asuntos involucrados en la implementación del PDF y su generación sea más sencilla. Cabe destacar que para la implementación de esta clase existió la colaboración de ***Nicola Asuni.***

- ***fuentes.-*** Es una carpeta que contiene todos los archivos relacionados con las diferentes fuentes de letras a utilizarse por las clases para la generación del documento ***PDF.***

3.2.2.4 Definición del CORE de EVAAB

Luego de haber explicado la estructura que presenta ***EVAAB*** junto con las capas de cómo está distribuido y los componentes adicionales que lo conforman es momento de explicar la parte fundamental del módulo de evaluación abierta ***EVAAB***, el cual tiene la denominación de ***CORE.*** El

mismo que lo conforman todos los archivos del módulo ***EVAAB*** excepto los definidos propiamente por ***Moodle*** y los archivos de los componentes adicionales como ***PDF y XAJAX*** explicados anteriormente.

3.2.2.4.1 Clases Modelo

Las ***clases Modelo*** definidas en ***EVAAB*** como se explicó anteriormente representan las clases que manejan los datos persistentes, es decir éstas clases se caracterizan por las funciones que poseen para manejar las operaciones con la BDD como guardar, actualizar, eliminar, buscar y todo lo relacionado con la manipulación de datos.

Cabe destacar que las funciones definidas en las clases ***Modelo*** no utilizan un mecanismo diferente para el manejo de datos del utilizado por ***Moodle***, simplemente hacen uso de éste a través de llamadas a sus funciones las cuales se encuentran definidas en las librerías manejadas por ***Moodle*** específicamente en ***datalib.php***

La definición de las clases que se utilizaron como ***Modelo*** son las clases más representativas dentro ***EVAAB*** en cuanto al manejo de los datos, es decir todos los elementos propios del cuestionario como son: ***preguntas***, ***respuestas***, ***pistas***, ***referencias*** y el ***cuestionario*** propiamente teniendo de esta forma actualmente en la versión inicial de ***EVAAB*** 5 clases Modelo.

Citaremos como ejemplo a la clase ***QuestionModel*** la cual expone lo anteriormente indicado en la Fig.3.3 mostrada a continuación:

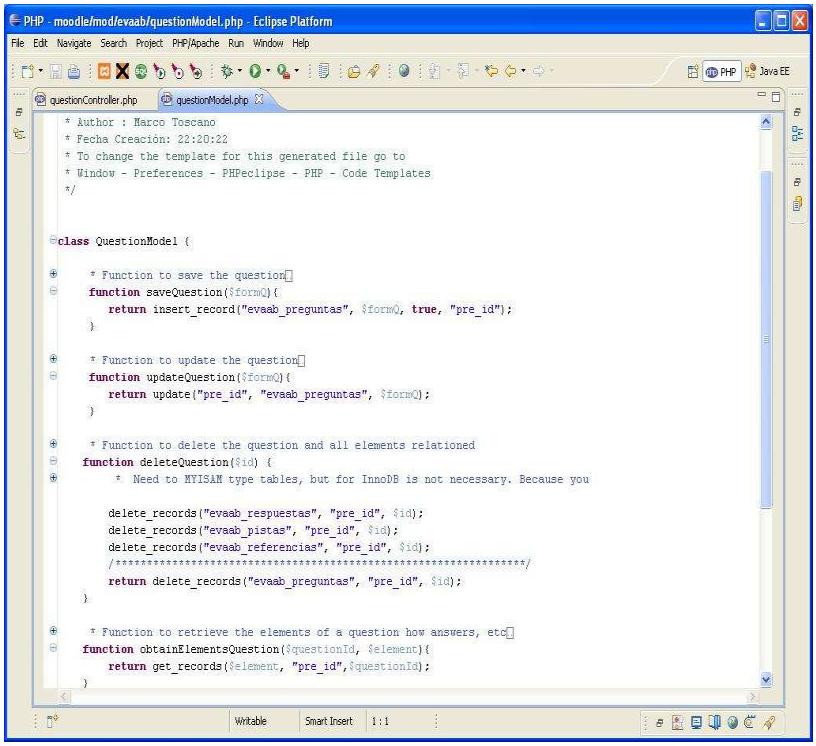

Fig. 3. 3: Ejemplo de Clase Modelo

3.2.2.4.2 Clases Controladoras

Las clases ***Controladoras*** como fue explicado en la arquitectura la constituyen las clases que contienen ***la lógica del negocio***, es decir aquí se encuentran los procesos propios del módulo de evaluación ***EVAAB*** los cuales son el componente intermedio que habilita la comunicación entre la interfaz y los datos.

Estas clases controladoras manejan adicional a la lógica del negocio, la ***generación dinámica de los elementos html*** a razón de que todas las interfaces son generadas automáticamente con ***XAJAX*** a través de la utilización de ***AJAX.*** Actualmente en la versión inicial de ***EVAAB*** tenemos 6 clases controladoras.

Citaremos como ejemplo a la clase ***QuestionController*** la cual expone lo anteriormente indicado en la Fig.3.4 mostrada a continuación:

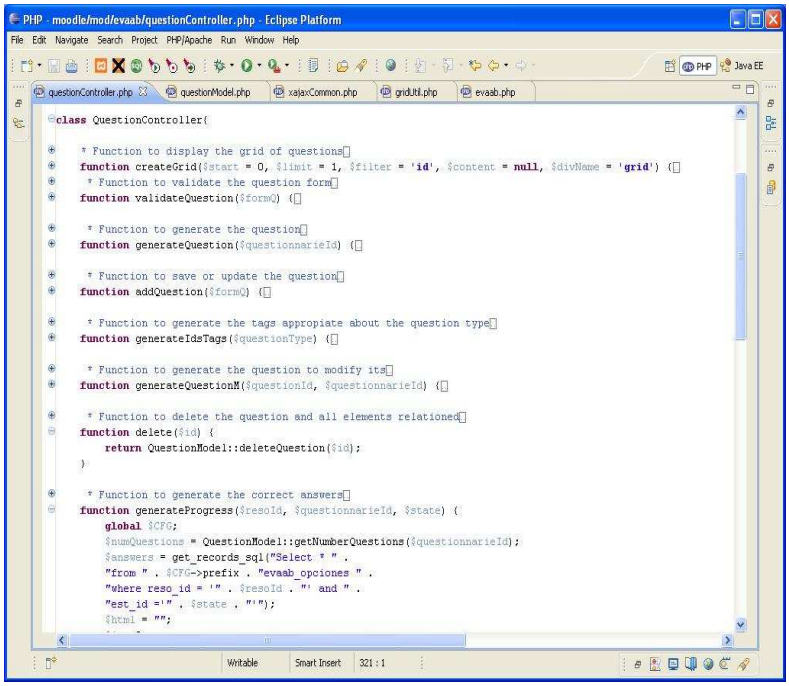

Fig. 3. 4: Ejemplo de Clase Controladora

3.2.2.4.3 Clases Vista

Las clases ***Vista*** definidas en ***EVAAB*** están representadas propiamente por los archivos que hacen uso de las características de ***AJAX*** a través de la utilización de la librería ***XAJAX*** , a los cuales se suman adicionalmente los archivos definidos propiamente por ***Moodle*** como esqueleto para el manejo de interfaces como son : ***view.php, mod.html, index.php.***

Cabe destacar que el tratamiento para ***las clases vista es dinámico*** razones por las cuales ***no existen interfaces físicas*** para todos los elementos del cuestionario que se gestionan sino simplemente para las clases contenedoras que manejan a estos elementos como componentes dentro de la misma a través de la utilización de ***XAJAX***. Actualmente disponemos en la versión inicial de ***EVAAB*** 4 clases vista.

Citaremos como ejemplo a los archivos ***view.php, loadDataManuView.php*** los cuales expone lo anteriormente indicado en las figuras 3.5, 3.6 mostradas a continuación

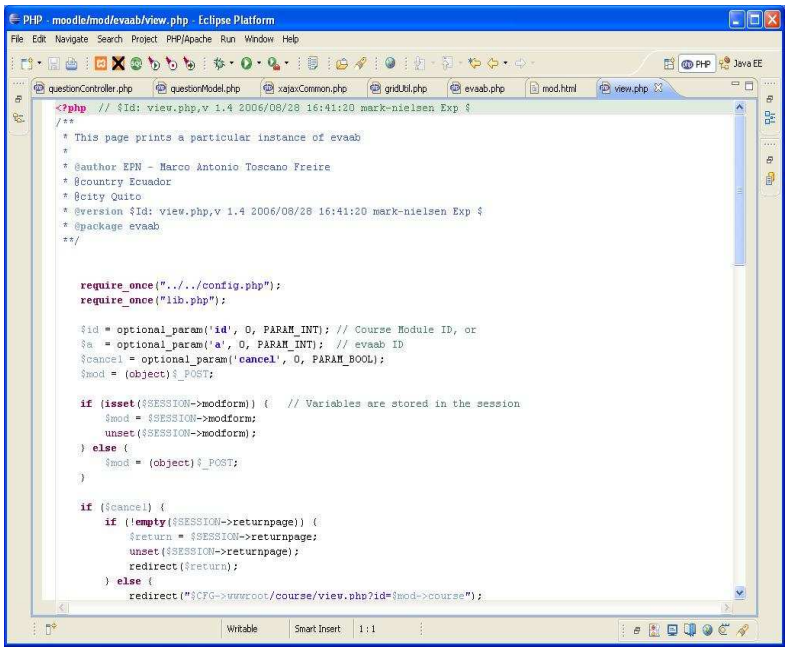

Fig. 3. 5: Clase Vista Moodle view.php

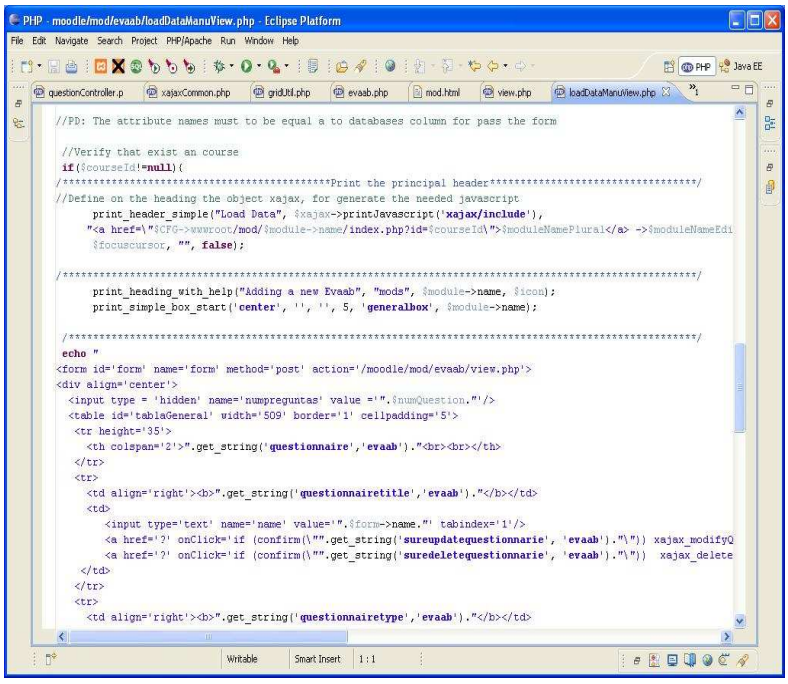

Fig. 3. 6: Ejemplo de Clase Vista EVAAB

Dentro de las clases vista la más importante dentro de este contexto y dentro de todo el módulo de evaluación ***EVAAB*** es la clase ***functionAjaxView.php*** la cual es la encargada de manejar la parte dinámica del módulo a través de la utilización de ***AJAX*** a través de ***XAJAX***. Esta clase se la puede observar a continuación en la Fig.3.7.

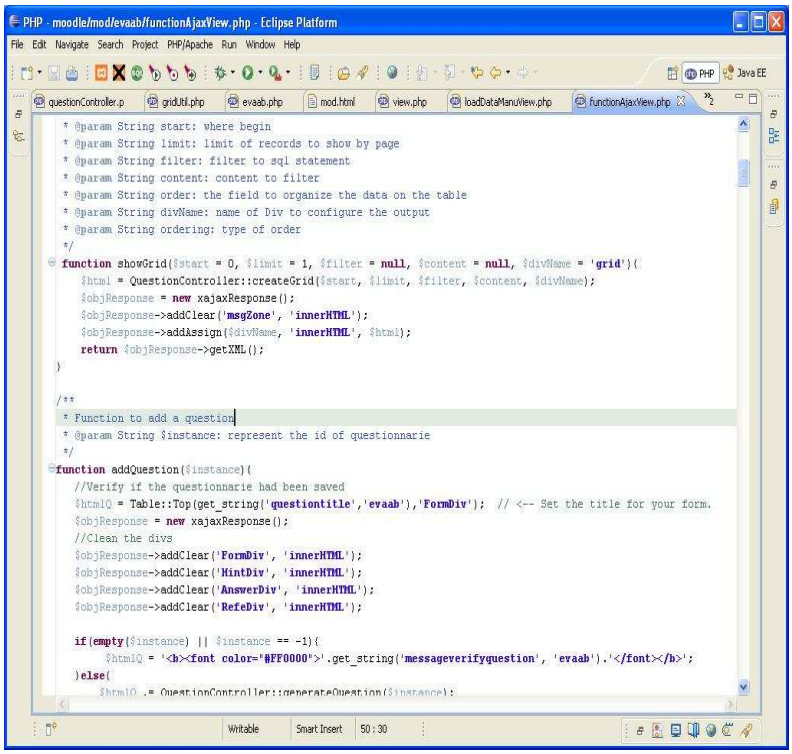

Fig. 3. 7: Ejemplo de Clase Modelo EVAAB- AJAX

3.2.3 VERSIONAMIENTO EN EVAAB

Tomando en cuenta todas las consideraciones especificadas en el ***primer capítulo*** en cuanto a la planificación del versionamiento actualmente se dispone de la siguiente información:

- En lo que respecta al código dentro módulo de evaluación ***EVAAB*** actualmente se dispone de 15 versiones, las cuales se encuentran empaquetadas de acuerdo al estándar definido ***evaab<fecha>*** donde ***<fecha>*** tiene el formato de ***dd/mm/yyyy.*** Siendo la última versión la

definida con el nombre ***evaab08042008.rar,*** todos los detalles de los cambios de versión se encuentran definidos en un archivo de texto denominado con el nombre de ***bitácora.***

- En lo que respecta a la documentación dentro de ***EVAAB*** actualmente disponemos de 19 versiones, las cuales se encuentran almacenadas de acuerdo al estándar definido ***evaab<fecha>*** donde ***<fecha>*** tiene el formato de ***dd/mm/yyyy.*** Siendo la última versión la definida con el nombre ***evaabDoc20042008.doc.***

3.2.4 COMUNICACIÓN CON EL USUARIO

Este punto dentro del desarrollo de ***EVAAB*** es muy importante ya que para subir a ***EVAAB*** dentro de los módulos de Moodle se tiene que entrar en un proceso de revisión, para lo cual se tiene que lograr establecer una comunicación con la gente que lidera las contribuciones y mejoras de ***Moodle.***

Cabe destacar que no se puede iniciar ningún trámite si no se cuenta con algo concreto, razón por la cual no se pudo comunicarse anteriormente hasta acabar el desarrollo de ***EVAAB***. Además que hay muchas contribuciones y mejoras en espera que pueden pasar mucho tiempo ahí hasta que la gente que lidera el proyecto asuma su dirección razón por la cual se estuvo intentando contactarse con los líderes principales de ***Moodle*** de tal forma que se puede agilitar el trámite, específicamente se contacto con ***Martin Dougiamas*** quien es el creador y director de proyecto de todo el sistema Moodle y ***Anthony Borrow*** quien es la persona que lidera todas las contribuciones de ***Moodle.***

Pasaron algunas semanas hasta tener una respuesta de una de las personas anteriormente mencionadas específicamente de ***Anthony Borrow*** quien asumió la dirección de la revisión del proyecto y con el cual estamos actualmente trabajando.

Las conversaciones que se tuvo con ***Anthony Borrow*** de manera general se las puede apreciar a continuación en la Fig. 3.8.

Fig. 3. 8: Pantalla Comunicación Líder Proyecto

Actualmente la última versión de ***EVAAB*** se encuentra subida al Internet dentro del sistema de seguimiento que utiliza ***Moodle*** llamado ***Traker*** para el análisis y seguimiento del mismo. Se encuentra identificado por la siguiente especificación ***CONTRI-355*** para lo cual se necesita tener una cuenta en ***Moodle,*** como se puede apreciar en la Fig. 3.9 a continuación:

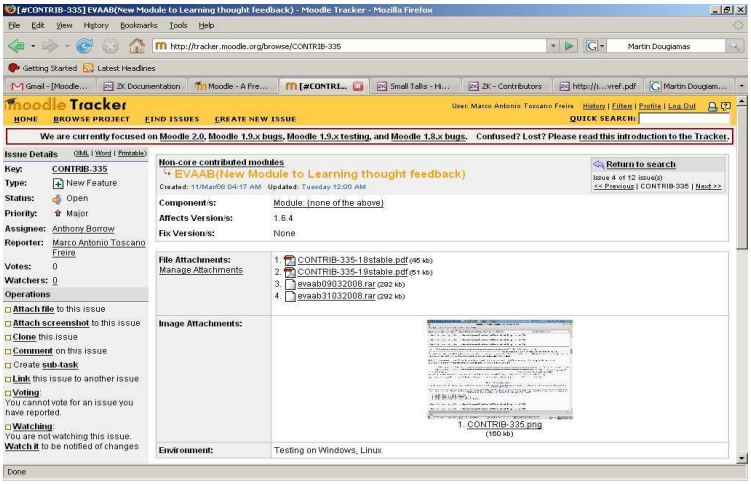

Fig. 3. 9: EVAAB dentro de Moodle Traker

3.3 PRUEBAS

3.3.1 DEFINICIÓN DE AMBIENTE DE PRUEBAS

Para definir el Ambiente de Prueba a utilizar se tuvo que elaborar primeramente un curso dentro de ***Moodle*** para a partir de este poder probar nuestro módulo de Evaluación ***EVAAB,*** para lo cual también se tiene que crear una instancia de éste. Luego se definió los casos de prueba y se verificó resultados, todos estos puntos se detallarán a continuación.

3.3.1.1 Definición de Curso

Como paso inicial se tuvo que definir un curso dentro de Moodle ya que sobre este se desarrollan todos los módulos y ***EVAAB*** no es la excepción, nuestro curso tiene las siguientes características:

- Se creó un curso con un formato basado en temas sin ninguna modalidad de grupo y sin ninguna restricción de acceso. El tema principal sobre el cual se realizará el aprendizaje es el ***Software Libre.***

En la Fig.3.10 mostrada a continuación se puede mirar la pantalla inicial del curso de ***Moodle*** la cual ha sido modificada con el único propósito de hacer referencia a la ***Escuela Politécnica Nacional E.P.N***

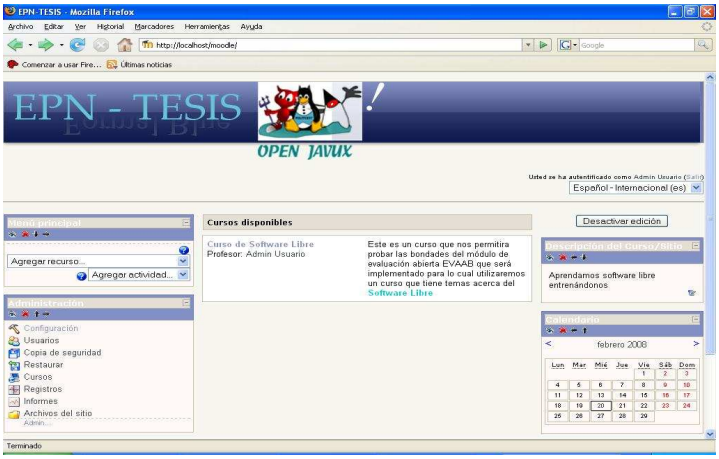

Fig. 3. 10: Pantalla Inicial Moodle – E.P.N

- Se crearon dos usuarios con los roles de ***Profesor y Estudiante*** para mostrar las opciones que se maneja en cada uno dentro del módulo de evaluación ***EVAAB***. Cabe destacar que el rol de Profesor fue asignado al usuario Administrador.
- Características como duración, fechas de inicio, fecha de fin, número de ítems en el tema, restricciones entre otras no fueron tomadas en cuenta ya que para este caso no son importantes ya que solamente nos enfocaremos en los elementos y características de ***EVAAB*** por lo cual se manejaron las por omisión..

3.3.1.2 Definición de Casos de Prueba

Para la definición de los Casos de Prueba se tiene que crear una instancia del módulo de Evaluación Abierta ***EVAAB*** dentro de Moodle, es decir crear un ***Cuestionario y gestionarlo***. Esto involucra administrar el Cuestionario conjuntamente con todos los elementos que lo conforman y una vez listo éste proceder a resolverlo.

Las operaciones involucradas en estos procesos son realizados por el ***Profesor y Estudiante*** respectivamente por lo que los casos de prueba se definirán por perfiles.

3.3.1.2.1 Casos Prueba Profesor

Los ***casos de prueba del profesor*** tienen relación con la gestión de de toda la carga del ***Cuestionario*** juntamente con sus respectivos elementos como son ***preguntas, respuestas, pistas y referencias***. Tomando en cuenta esto se definieron los siguientes casos de prueba generales:

3.3.1.2.1.1 Gestionar Cuestionario

En este caso de prueba general se definieron los siguientes casos de prueba específicos:

Nombre:	**Guardar Cuestionario**
Propósito:	Almacenar el Cuestionario
Prerrequisitos:	Se debe haber logeado con el perfil de Profesor y haber agregado la actividad EVAAB
Datos Prueba	Entre los datos tenemos los siguientes: Nombre (String) Tipo Cuestionario (String) = Abierto,| Cerrado Número Preguntas (int) = Max 20, Min 10 Descripción (String)
Pasos	Se deben seguir los siguientes pasos: 1. Ingresar todos los datos del formulario como son el nombre, el tipo, número de preguntas y descripción. 2. Hacer clic en el botón Guardar 3. Verificar que se haya desplegado e mensaje
Resultados:	**Correcto:** Si todos los datos fueron ingresados correctamente se despliega un mensaje informando el éxito de la operación. **Incorrecto:** Si se olvido de ingresar algún dato el mensaje correspondiente al elemento olvidado o incorrecto es desplegado

Tabla 3. 1: Caso de Prueba Guardar Cuestionario

Nombre:	**Actualizar Cuestionario**
Propósito:	Actualizar el Cuestionario que fue ingresado anteriormente.

Prerrequisitos:	Se debe haber logeado con el perfil de Profesor y se debe haber seleccionado el cuestionario correspondiente
Datos Prueba	Entre los datos de prueba tenemos: Nombre (String) Tipo Cuestionario (String) = Abierto, Cerrado Número Preguntas (int) = Max 20, Min 10 Descripción (String)
Pasos	Se deben seguir los siguientes pasos: 1. Modificar los datos del formulario que se vaya a actualizar 2. Hacer clic en la imagen 3. Verificar que se haya desplegado el mensaje informativo
Resultados:	**Correcto:** Si se pudo modificar el cuestionario aparece un mensaje informando el éxito de la operación. **Incorrecto:** Si no se pudo modificar el cuestionario un mensaje de falla es desplegado.

Tabla 3. 2: Caso de Prueba Actualizar Cuestionario

Nombre:	**Eliminar Cuestionario**
Propósito:	Eliminar el Cuestionario con todos los elementos que lo conforman como son preguntas, respuestas, pistas y referencias
Prerrequisitos:	Se debe haber logeado con el perfil de Profesor y se debe haber seleccionado el cuestionario correspondiente
Datos Prueba	Ninguno

Pasos	Se deben seguir los siguientes pasos: 1. Hacer clic en la imagen 2. Verificar que se haya desplegado e mensaje informativo
Resultados:	**Correcto:** Si se pudo eliminar el cuestionario junto con sus elementos aparece un mensaje informando el éxito de la operación y quedan limpios todos los campos. **Incorrecto:** Si no se pudo eliminar el cuestionario un mensaje de falla es desplegado.

Tabla 3. 3: Caso de Prueba Eliminar Cuestionario

3.3.1.2.1.2 Gestionar Preguntas

En este caso de prueba general se definieron los siguientes casos de prueba específicos:

Nombre:	**Guardar Pregunta**
Propósito:	Almacenar la Pregunta
Prerrequisitos:	Se debe haber logeado con el perfil de Profesor y se debe haber guardado el Cuestionario.
Datos Prueba	Entre los datos tenemos los siguientes: Tipo Pregunta (String) = Selección Múltiple, Booleana Numero Respuestas (int) = Max 5, Min 2 Descripción (String)
Pasos	Se deben seguir los siguientes pasos: 1. Elegir el tipo de pregunta, llenar el número de respuestas y la

	descripción. 2. Hacer clic en el botón Guardar Pregunta. 3. Verificar que se haya desplegado el mensaje correspondiente.
Resultados:	**Correcto:** Si todos los datos fueron ingresados correctamente se despliega un mensaje informando el éxito de la operación y se procede a generar las respuestas. **Incorrecto:** Si se olvido de ingresar algún dato el mensaje correspondiente al elemento olvidado o incorrecto es desplegado.

Tabla 3. 4: Caso de Prueba Guardar Pregunta

Nombre:	**Actualizar Pregunta**
Propósito:	Actualizar la Pregunta que fue ingresada anteriormente.
Prerrequisitos:	Se debe haber logeado con el perfil de Profesor y se debe haber guardado el Cuestionario.
Datos Prueba	Entre los datos tenemos los siguientes: Tipo Pregunta (String) = Selección Múltiple, Booleana Numero Respuestas (int) = Max 5, Min 2 Descripción (String)
Pasos	Se deben seguir los siguientes pasos: 1. Elegir la pregunta que se va a modificar. 2. Hacer clic en la imagen 3. Modificar los datos que necesiten ser actualizados

	4. Seleccionar el botón Actualizar Pregunta. 5. Verificar que se haya desplegado e mensaje informativo
Resultados:	**Correcto:** Si se pudo modificar la pregunta aparece un mensaje informando el éxito de la operación. **Incorrecto:** Si no se pudo modificar la pregunta un mensaje de falla es desplegado.

Tabla 3. 5: Caso de Prueba Actualizar Pregunta

Nombre:	**Eliminar Pregunta**
Propósito:	Eliminar la pregunta con todos los elementos que lo conforman como son respuestas, pistas y referencias
Prerrequisitos:	Se debe haber logeado con el perfil de Profesor y se debe haber seleccionado la pregunta correspondiente
Datos Prueba	Ninguno
Pasos	Se deben seguir los siguientes pasos: 1. Hacer clic en la imagen correspondiente a la pregunta a eliminar 2. Verificar que se haya desplegado e mensaje informativo
Resultados:	**Correcto:** Si se pudo eliminar pregunta junto con sus elementos aparece un mensaje informando el éxito de la operación y se elimina un registro en el tablero de preguntas. **Incorrecto:** Si no se pudo eliminar pregunta

	un mensaje de falla es desplegado.

Tabla 3. 6: Caso de Prueba Eliminar Pregunta

3.3.1.2.1.3 Gestionar Respuestas

En este caso de prueba general se definieron los siguientes casos de prueba específicos:

Nombre:	**Guardar Respuesta**
Propósito:	Almacenar Respuesta
Prerrequisitos:	Se debe haber logeado con el perfil de Profesor y se debe haber guardado la pregunta
Datos Prueba	Entre los datos tenemos los siguientes: Respuesta (String) Razón (String) Correcto(Booleano)
Pasos	Se deben seguir los siguientes pasos: 1. Ingresar el número de respuestas el cual será generado una vez guardada la pregunta o se puede seleccionar el link ***Ninguno*** en la sección de Respuesta en el Tablero de Preguntas si es que se olvido de generarlas. Cabe destacar que el máximo número es 5 y el mínimo 2. 2. Ingresar la respuesta, la razón y marcarla como correcta o incorrecta. 3. Hacer clic en el botón Guardar Respuestas.

	4. Verificar que se haya desplegado el mensaje correspondiente.
Resultados:	**Correcto:** Si todos los datos fueron ingresados correctamente se despliega un mensaje informando el éxito de la operación **Incorrecto:** Si se olvido de ingresar algún dato el mensaje correspondiente al elemento olvidado o incorrecto es desplegado.

Tabla 3. 7: Caso de Prueba Guardar Respuesta

Nombre:	**Actualizar Respuesta**
Propósito:	Actualizar la Respuesta que fue ingresada anteriormente.
Prerrequisitos:	Se debe haber logeado con el perfil de Profesor y se debe haber guardado la respuesta.
Datos Prueba	Entre los datos tenemos los siguientes: Respuesta (String) Razón (String) Correcto(Booleano)
Pasos	Se deben seguir los siguientes pasos: 1. Elegir la respuesta que se va a modificar. 2. Modificar los datos que necesiten ser actualizados 3. Hacer clic en el botón Actualizar Respuesta 4. Verificar que se haya desplegado el mensaje informativo
Resultados:	**Correcto:** Si se pudo modificar la respuesta aparece un mensaje informando el éxito de

	la operación. **Incorrecto:** Si no se pudo modificar la respuesta un mensaje de falla es desplegado.

Tabla 3. 8: Caso de Prueba Actualizar Respuesta

Nombre:	**Eliminar Respuesta**
Propósito:	Eliminar la respuesta que fue ingresada
Prerrequisitos:	Se debe haber logeado con el perfil de Profesor y se debe haber guardado la pregunta referenciada.
Datos Prueba	Ninguno
Pasos	Se deben seguir los siguientes pasos: 1. Elegir la respuesta a eliminar. 2. Hacer clic en el botón eliminar respuesta. 3. Verificar que se haya desplegado e mensaje informativo
Resultados:	**Correcto:** Si se pudo eliminar la respuesta aparece un mensaje informando el éxito de la operación y se actualiza en el tablero de preguntas en la sección de respuestas. **Incorrecto:** Si no se pudo eliminar la respuesta un mensaje de falla es desplegado.

Tabla 3. 9: Caso de Prueba Eliminar Respuesta

3.3.1.2.1.4 Gestionar Pistas

En este caso de prueba general se definieron los siguientes casos de prueba específicos:

Nombre:	**Guardar Pista**
Propósito:	Almacenar Pista
Prerrequisitos:	Se debe haber logeado con el perfil de Profesor y se debe haber guardado la pregunta
Datos Prueba	Entre los datos tenemos los siguientes: Pista (String)
Pasos	Se deben seguir los siguientes pasos: 1. Ingresar el número de pistas o seleccionar el link ***Ninguno*** en la sección Pistas en el Tablero de Preguntas si es que se olvido de generarlas. Cabe destacar que el máximo número es 5 y el mínimo 2. 2. Ingresar la pista. 3. Hacer clic en el botón Guardar Pistas. 4. Verificar que se haya desplegado el mensaje correspondiente.
Resultados:	**Correcto:** Si todos los datos fueron ingresados correctamente se despliega un mensaje informando el éxito de la operación **Incorrecto:** Si se olvido de ingresar algún dato el mensaje correspondiente al elemento olvidado o incorrecto es desplegado.

Tabla 3. 10: Caso de Prueba Guardar Pista

Nombre:	**Actualizar Pista**
Propósito:	Actualizar la Pista que fue ingresada anteriormente.
Prerrequisitos:	Se debe haber logeado con el perfil de

	Profesor y se debe haber guardado la pista.
Datos Prueba	Entre los datos tenemos los siguientes: Pista (String)
Pasos	Se deben seguir los siguientes pasos: 1. Elegir la pista que se va a modificar. 2. Modificar el dato de Pista. 3. Hacer clic en el botón Actualizar Pista 4. Verificar que se haya desplegado el mensaje informativo
Resultados:	**Correcto:** Si se pudo modificar la pista aparece un mensaje informando el éxito de la operación. **Incorrecto:** Si no se pudo modificar la pista un mensaje de falla es desplegado.

Tabla 3. 11: Caso de Prueba Actualizar Pista

Nombre:	**Eliminar Pista**
Propósito:	Eliminar una pista
Prerrequisitos:	Se debe haber logeado con el perfil de Profesor y se debe haber guardado la pregunta referenciada.
Datos Prueba	Ninguno
Pasos	Se deben seguir los siguientes pasos: 1. Elegir la pista a eliminar. 2. Hacer clic en el botón eliminar pista. 3. Verificar que se haya desplegado e mensaje informativo
Resultados:	**Correcto:** Si se pudo eliminar la pista aparece un mensaje informando el éxito de la operación y se actualiza en el tablero de

	preguntas en la sección de pistas. **Incorrecto:** Si no se pudo eliminar la pista un mensaje de falla es desplegado.

Tabla 3. 12: Caso de Prueba Eliminar Pista

3.3.1.2.1.5 Gestionar Referencias

En este caso de prueba general se definieron los siguientes casos de prueba específicos:

Nombre:	**Guardar Referencia**
Propósito:	Almacenar Referencias
Prerrequisitos:	Se debe haber logeado con el perfil de Profesor y se debe haber guardado la pregunta
Datos Prueba	Entre los datos tenemos los siguientes: Referencia (String) Recurso (String)
Pasos	Se deben seguir los siguientes pasos: 1. Ingresar el número de referencias o seleccionar el link ***Ninguno*** en la sección Referencias en el Tablero de Preguntas si es que se olvido de generarlas. Cabe destacar que el máximo número es 5 y el mínimo 2. 2. Ingresar la referencia y seleccionar el recurso explorando los recursos de Moodle. 3. Hacer clic en el botón Guardar Referencias. 4. Verificar que se haya desplegado el mensaje correspondiente.

Resultados:	**Correcto:** Si todos los datos fueron ingresados correctamente se despliega un mensaje informando el éxito de la operación **Incorrecto:** Si se olvido de ingresar algún dato el mensaje correspondiente al elemento olvidado o incorrecto es desplegado.

Tabla 3. 13: Caso de Prueba Guardar Referencia

Nombre:	**Actualizar Referencia**
Propósito:	Actualizar la Referencia que fue ingresada anteriormente.
Prerrequisitos:	Se debe haber logeado con el perfil de Profesor y se debe haber guardado la referencia.
Datos Prueba	Entre los datos tenemos los siguientes: Referencia (String) Recurso (String)
Pasos	Se deben seguir los siguientes pasos: 1. Elegir la referencia que se va a modificar. 2. Modificar los datos que sean necesarios. 3. Hacer clic en el botón Actualizar Referencia. 4. Verificar que se haya desplegado el mensaje informativo
Resultados:	**Correcto:** Si se pudo modificar la referencia aparece un mensaje informando el éxito de la operación. **Incorrecto:** Si no se pudo modificar la pista un mensaje de falla es desplegado.

Tabla 3. 14: Caso de Prueba Actualizar Referencia

Nombre:	**Eliminar Referencia**
Propósito:	Eliminar una Referencia
Prerrequisitos:	Se debe haber logeado con el perfil de Profesor y se debe haber guardado la pregunta referenciada.
Datos Prueba	Ninguno
Pasos	Se deben seguir los siguientes pasos: 1. Elegir la referencia a eliminar. 2. Hacer clic en el botón eliminar referencia. 3. Verificar que se haya desplegado e mensaje informativo
Resultados:	**Correcto:** Si se pudo eliminar la referencia aparece un mensaje informando el éxito de la operación y se actualiza en el tablero de preguntas en la sección de referencias. **Incorrecto:** Si no se pudo eliminar la referencia un mensaje de falla es desplegado.

Tabla 3. 15: Caso de Prueba Eliminar Referencia

3.3.1.2.2 Casos Prueba Estudiante

Los ***casos de prueba del Estudiante*** tienen relación con la ***resolución del Cuestionario*** propiamente donde están involucrados la visualización de los elementos de ayuda, progreso, junto con ***generación de calificaciones y presentación de informe*** Tomando en cuenta esto se definieron los siguientes casos de prueba:

3.3.1.2.2.1 Resolución Cuestionario

En este caso de prueba general se definieron los siguientes casos de prueba específicos:

Nombre:	**Seguimiento Preguntas**
Propósito:	Validar el seguimiento de la resolución de las preguntas del Cuestionario.
Prerrequisitos:	Se debe haber logeado con el perfil de Estudiante, creado el cuestionario y haber comenzado la Resolución.
Datos Prueba	Preguntas, Respuestas, Pistas y Referencias
Pasos	Se deben seguir los siguientes pasos: 1. Seleccionar el link ***Comenzar Resolución.*** 2. Seleccionar la Pregunta que se vaya a contestar. 3. Seleccionar la respuesta o respuestas que piense que son correctas. 4. Dar un clic en ***Siguiente.***
Resultados:	**Correcto:** Aparece un mensaje que notifica que la pregunta es correcta, incorrecta o que está incompleta. **Incorrecto:** Aparece un mensaje que notifica que no existen datos para seguir.

Tabla 3. 16: Caso de Prueba Seguimiento Preguntas

Nombre:	**Visualización de Pistas**
Propósito:	Validar que las pistas se pueden visualizar de tal forma que pueden servir de ayuda en el momento de la resolución del Cuestionario.
Prerrequisitos:	Se debe haber logeado con el perfil de Estudiante, creado el cuestionario, las preguntas, pistas y haber comenzado la Resolución.

Datos Prueba	Preguntas, Respuestas, Pistas
Pasos	Se deben seguir los siguientes pasos: 1. Se debe dar un clic en la imagen referente a las pistas 1. Verificar que las pistas se muestren correctamente.
Resultados:	**Correcto:** Se visualiza en la parte inferior un bloque que contiene las pistas para la pregunta en cuestión. **Incorrecto:** Aparece un mensaje que notifica que no existen pistas para mostrar.

Tabla 3. 17: Caso de Prueba Visualización de Pistas

Nombre:	**Visualización de Referencias**
Propósito:	Validar que las Referencias se pueden visualizar de tal forma que pueden servir de ayuda en el momento de la resolución del Cuestionario.
Prerrequisitos:	Se debe haber logeado con el perfil de Estudiante, creado el cuestionario, las preguntas, referencias y haber comenzado la Resolución.
Datos Prueba	Preguntas, Respuestas, Referencias
Pasos	Se deben seguir los siguientes pasos: 1. Se debe dar un clic en la imagen referente a las referencias. 2. Verificar que las referencias se muestren correctamente.
Resultados:	**Correcto:** Se visualiza en la parte inferior

	un bloque que contiene las referencias para la pregunta en cuestión. **Incorrecto:** Aparece un mensaje que notifica que no existen referencias para mostrar.

Tabla 3. 18: Caso de Prueba Visualización de Referencias

Nombre:	**Mostrar Progreso**
Propósito:	Verificar que cada pregunta procesada haya sido contabilizada según sea su estado correcto, incorrecto e incompleto y que se puede observar en un resumen.
Prerrequisitos:	Se debe haber logeado con el perfil de Estudiante, creado el cuestionario, las preguntas, respuestas y haber comenzado la Resolución.
Datos Prueba	Preguntas, Respuestas
Pasos	Se deben seguir los siguientes pasos: 1. Se debe dar un clic en la imagen referente al progreso. 2. Con lo cual se mostrará un resumen de todas las preguntas realizadas agrupadas por sus estados sean estas correctas, incorrectas e incompletas. Además se nos mostrará los porcentajes de aceptación. 3. Adicionalmente a este resumen, tenemos la opción de modificar cualquiera de las preguntas mostradas.
Resultados:	**Correcto:** Se visualiza el resumen de las

	preguntas. **Incorrecto:** No se visualizará el resumen y de hacerlo este no contendrá ninguna pregunta.

Tabla 3. 19: Caso de Prueba Mostrar Progreso

Nombre:	**Generar Informe**
Propósito:	Generar un informe de la resolución del cuestionario realizado por estudiante, donde se contemple el cuestionario junto con todos los elementos que lo conforman.
Prerrequisitos:	Se debe haber logeado con el perfil de Estudiante y se debe haber terminado de resolver el cuestionario
Datos Prueba	Ninguno
Pasos	Se deben seguir los siguientes pasos: 1. Dar un clic en el enlace ***Resultado del Test*** 2. Verificar que el informe se haya desplegado
Resultados:	**Correcto:** Se visualiza el informe con el resumen de la resolución **Incorrecto:** No se visualiza el informe o se visualiza sin datos.

Tabla 3. 20: Caso de Prueba Generar Informe

3.3.2 RESULTADOS

Antes de detallar los resultados obtenidos cabe destacar que las pruebas realizadas en base a los casos de prueba fueron realizadas integrando el módulo de evaluación ***EVAAB*** dentro de ***Moodle***, por lo que las pruebas unitarias se

verificaron dentro de la integración. Cabe destacar que se realizó de esta manera para ocupar todos los recursos, funciones que cuenta Moodle ya que de otra forma hubiera sido muy complejo realizarlo y luego integrarlo.

A continuación se presentará figuras concretas donde se demuestra que se cumplió con éxito la ejecución de las pruebas, lo cual se lo realizó de esta manera con el objetivo de enfocarnos en las partes fundamentales de EVAAB.

3.3.2.1 Cuestionario General

En la Fig.3.11 se puede apreciar que el Cuestionario ha sido creado con éxito junto con todos sus elementos como son las preguntas, respuestas pistas y referencias.

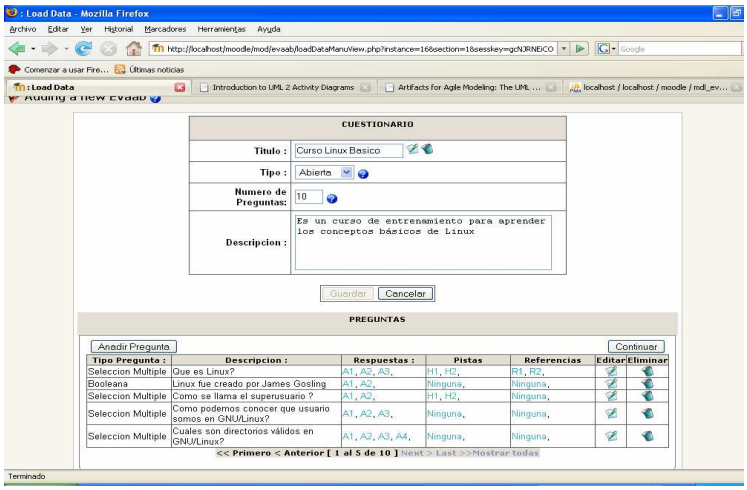

Fig. 3. 11: Pantalla Comprobación Cuestionario Ingreso

3.3.2.2 Resolución Cuestionario

En la Fig.3.12 se puede apreciar a ***EVAAB*** desde la perspectiva del Estudiante al momento de comenzar la resolución del Cuestionario, donde se puede constatar que los elementos fueron ingresados correctamente.

Fig. 3. 12: Pantalla Comprobación Resolución

3.3.2.3 Generación Calificación

En la Fig.3.13 se puede apreciar que la generación de la calificación fue realizada correctamente, la cual toma en cuenta la contabilización de las preguntas correctas, incorrectas e incompletas para definir una nota

RESULTADO-CUESTIONARIO	
Titulo :	Curso Linux Básico
Nombre del Estudiante:	Pepito Cerezo
Fecha:	2008-02-21 22:01:29
Calificacion:	4/(20)
Reporte:	Resultado de Test

Fig. 3. 13: Pantalla Comprobación Calificación

3.3.2.3 Generación del Informe

En la Fig.3.14 se puede apreciar el informe generado para el Estudiante luego de que ha finalizado la resolución del Cuestionario, el cual contiene un resumen de todo el proceso.

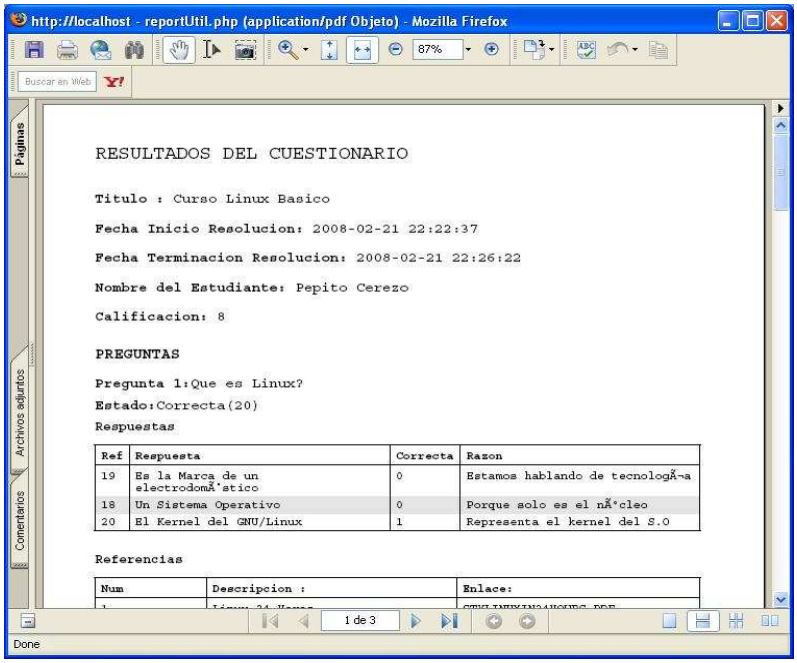

Fig. 3. 14: Pantalla Comprobación Informe

CAPÍTULO IV

4. INTEGRACIÓN Y EVALUACIÓN

4.1 INTEGRACIÓN CON MOODLE

4.1.1 GENERALIDADES

Moodle presenta una forma de organización muy ordenada en cuanto al manejo de sus módulos, permitiendo realizar la integración de módulos nuevos de una forma muy sencilla. Los módulos de ***Moodle*** se encuentran organizados mediante carpetas y archivos, los cuales deben seguir ciertas reglas o estandarizaciones que permitan garantizar el buen desenvolvimiento del mismo. Entre estas restricciones o reglas se tiene las siguientes:

- Los módulos de ***Moodle*** deben encontrarse ubicados en el path relativo: ***..\moodle\mod\nombre_modulo***, donde el nombre del módulo representa la carpeta que contiene todos los archivos relacionados con este.
- Los módulos de ***Moodle*** deben contener ciertos archivos propios de configuración, los cuales cumplen con funciones específicas. Entre los archivos que deben tener los módulos tenemos los siguientes[33]:

    - ***mod.html.-*** Permite instalar y actualizar instancias del módulo
    - ***version.php.-*** Define información de versión y provee elementos de actualización
    - ***index.php.-*** Página para listar todas las instancias de un curso.
    - ***view.php.-*** Página para ver una instancia en particular.
    - ***lib.php.-*** Define todas las funciones definidas en el módulo.
    - ***icon.gif.-*** Icono correspondiente al módulo.

[33] Readme.txt - http://download.moodle.org/plugins/mod/NEWMODULE.zip

- Para manejar los scripts de base de datos ***Moodle*** utiliza una carpeta dentro de cada módulo llamada ***db*** en la cual se tienen que colocar los archivos correspondientes a la creación de las tablas del módulo. Generalmente existen ***cuatro archivos*** de acuerdo la versión de Moodle 1.6.3 para cada módulo, los cuales se encuentran definidos en base a los DBMS soportados como ***MySQL*** y ***Postgress,*** los archivos que se tiene en este directorio son los siguientes:

    - ***mysql.sql***.- Este archivo contiene los scripts para la creación de las tablas del módulo para ***MySQL***
    - ***postgress7.sql***.- Este archivo contiene los scripts para la creación de las tablas del módulo para ***Postgress***
    - ***mysql.php***.- Este archivo contiene las funciones necesarias para realizar las actualizaciones de las tablas del módulo para ***MySQL*** de acuerdo a las versiones de Moodle
    - ***postgres7.php***.- Este archivo contiene las funciones necesarias para realizar las actualizaciones de las tablas del módulo para ***Postgress*** de acuerdo a las versiones de Moodle

4.1.2 EVAAB DENTRO DE MOODLE

La integración del módulo de evaluación abierta ***EVAAB*** dentro de ***Moodle*** en su versión ***1.6.3*** se basó en una plantilla propia de Moodle para la creación de módulos adicionales, la cual se ve reflejada en los puntos mencionados anteriormente.

La integración ***EVAAB dentro de Moodle*** fue un proceso progresivo, ya que se fue trabajando desde un inicio conjuntamente con todo el e-learning por lo que al finalizar el desarrollo del mismo los problemas o asuntos de integración fueron mínimos. Se realizó de esta manera porque era necesario utilizar todos los beneficios y complementos que tiene Moodle, el proceso de integración se lo puede definir en los siguientes puntos:

- Lo primero que se realizó fue descargarse la plantilla definida por ***Moodle*** para la creación de módulos adicionales, la cual contenía todos los archivos de configuración mencionados anteriormente. Los mismos que contenían las configuraciones estándar que debía ser parametrizada en base a nuestras necesidades.
- Se configuró el contenido de los archivos estándar de Moodle para que hagan referencia al módulo de evaluación ***EVAAB*** y se los colocó conjuntamente con todos los archivos de nuestro módulo dentro una carpeta llamada ***evaab***.
- Se creo dos archivos de configuración con el nombre de ***evaab.php*** para definir todas las cadenas a utilizarse en el módulo ***EVAAB***, lo cual es necesario para cada definición de lenguaje a utilizar. En nuestro caso creamos un archivo para el ***lenguaje inglés y otro para el lenguaje español*** los mismos que fueron colocados localmente en nuestro directorio ***evaab*** bajo el subdirectorio ***/lang*** dentro de las carpetas ***en_utf8 y es_utf8*** para los lenguajes de inglés y español respectivamente.
- Se colocó el script de la base de datos relacionado con la creación de las tablas de ***EVAAB*** dentro del directorio ***/evaab/db.***

Luego de realizar las tareas anteriormente mencionadas se procedió a instalar nuevamente ***Moodle 1.6.3*** con el objetivo de verificar que la integración final esta correcta y constatar que nuestro módulo ***EVAAB*** cumple con todos los estándares.

Entre las verificaciones a tomar en cuenta en la integración de ***EVAAB*** dentro de ***Moodle*** tenemos: la creación de las tablas de nuestro módulo de evaluación ***EVAAB*** conjuntamente con la verificación de su funcionamiento, los cuales serán evaluados a continuación.

4.2 EVALUACIÓN DE RESULTADOS

Para realizar la evaluación de la integración de ***EVAAB*** dentro de ***Moodle*** se consideró una instalación totalmente nueva de ***Moodle*** conjuntamente con ***EVAAB,*** la cual consistió en los siguientes puntos:

- Se eliminó la base de datos de ***moodle*** para vaciar todas las configuraciones, luego se creó ***la base de datos moodle*** lo cual es necesario antes de proseguir con la instalación de Moodle.
- Se instaló nuevamente la herramienta ***XAMPP*** la cual contiene el servidor Web ***Apache*** y la base de datos ***MySQL.***
- Se cargo la versión de ***Moodle 1.6.3*** y se integró a ***evaab*** en el directorio correspondiente a los módulos dentro de Moodle, ***mod,*** para luego colocarlo en el directorio correspondiente a apache para su funcionamiento.
- Se procedió a instalar ***Moodle.***
- Se procedió a instalar nuestro tema propio de ***Moodle***, lo cual es algo opcional.

Cabe destacar que se realizó estos pasos con la finalidad de simular un ambiente real que sucederá cuando alguna persona se descargue ***Moodle*** conjuntamente con ***EVAAB*** para instalarlo.

4.2.1 FACTORES A EVALUAR

Para poder determinar que factores serán evaluados en la integración de ***EVAAB dentro de Moodle*** se verificó ciertos puntos desde ***la perspectiva de Moodle***, exceptuando el funcionamiento como tal de ***EVAAB*** que fue ya analizado en el capítulo anterior. Tomando en cuenta esto se puede citar los siguientes puntos a evaluar:

- Las tablas definidas para el módulo de evaluación ***EVAAB*** se deben crear automáticamente al momento de la instalación de ***Moodle***.
- ***EVAAB*** se debe visualizar como una actividad más dentro de ***Moodle***.

- Una vez creada una instancia de ***EVAAB*** ésta pueda ser administrada desde la pantalla principal, es decir pueda ser ***actualizada y eliminada***.
- ***EVAAB*** debe aparecer en la lista de módulos definidos en la administración de ***Moodle***.

4.2.2 RESULTADOS

4.2.2.1 Evaab como Tabla

Para realizar la verificación de la generación de los scripts relacionados con la base de datos que conforman ***EVAAB*** es necesario realizar la instalación de ***Moodle*** para poder constatarlo. El script se lo puede apreciar a continuación en la Fig.4.1.

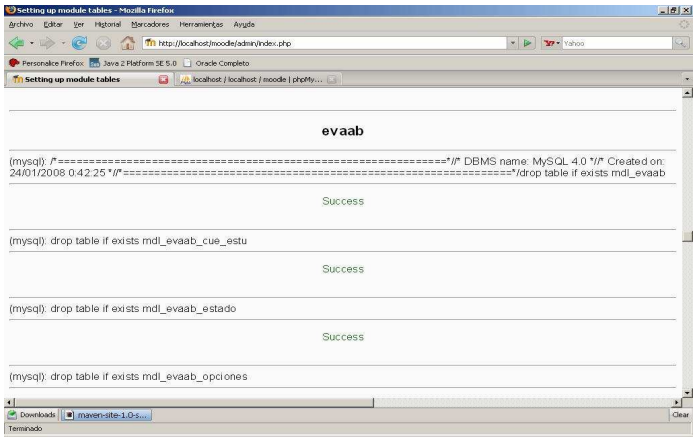

Fig. 4. 1: Pantalla Tablas EVAAB

4.2.2.2 Evaab como Actividad

Para realizar la verificación de que efectivamente ***EVAAB*** forma parte de una de las actividades dentro de ***Moodle*** para el desenvolvimiento de un curso se tiene primero que ingresar a ***Moodle*** con el perfil de ***Profesor*** para tener acceso a las diferentes opciones de Administración de un Curso, como son la creación de actividades y recursos.

Dentro del combo destinado para la creación de nuevas actividades se encuentra a ***EVAAB,*** con lo que se puede concluir que forma parte de

una de las actividades que conforman ***Moodle.*** El mismo que se puede apreciar a continuación en la Fig.4.2.

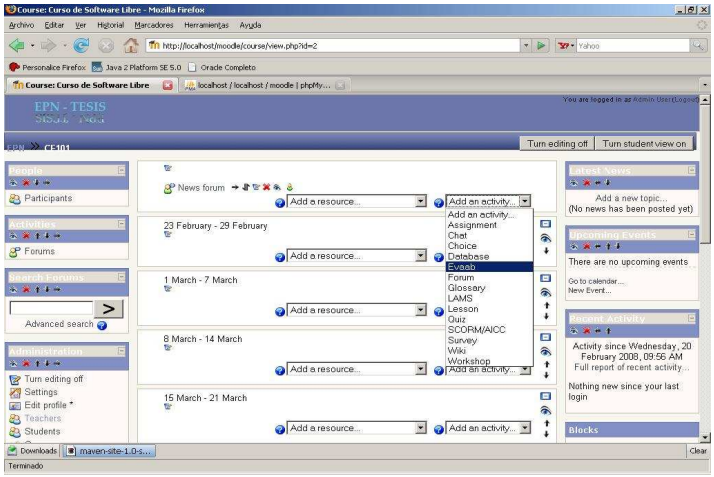

Fig. 4. 2: Pantalla Actividad

4.2.2.3 Evaab como Instancia

Para realizar la verificación de ***EVAAB*** como instancia dentro de ***Moodle*** se verificó que las opciones de ***actualización*** y ***eliminación*** para ***EVAAB*** funcionarán correctamente desde la interfaz principal de Moodle en cuanto a la gestión del curso. A continuación se muestra la opción de eliminación ya que este detalle de mejor forma el tema tratado, el cual se lo puede apreciar a continuación en la Fig.4.3.

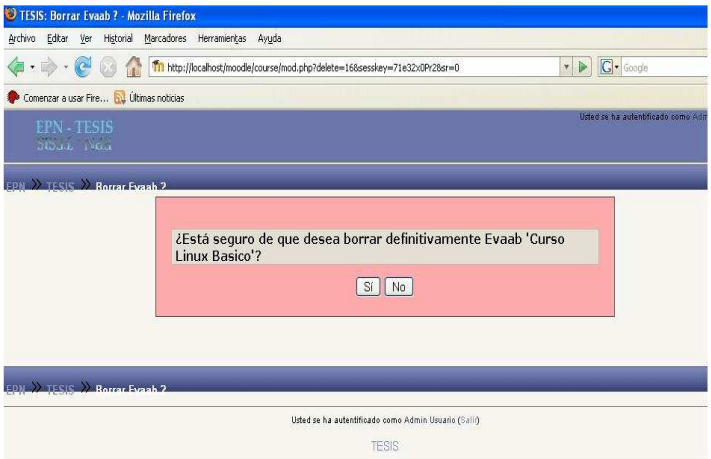

Fig. 4. 3: Pantalla Eliminación Instancia

4.2.2.4 Evaab como Módulo

Para realizar la verificación de ***EVAAB*** como un módulo de ***Moodle*** debemos dirigirnos a la parte de administración de ***Moodle*** y elegir la opción ***Configuración – Administrar Módulos,*** el cual presenta una pantalla con todos los módulos disponibles en donde se puede apreciar a ***EVAAB*** como se muestra en la Fig.4.4 a continuación:

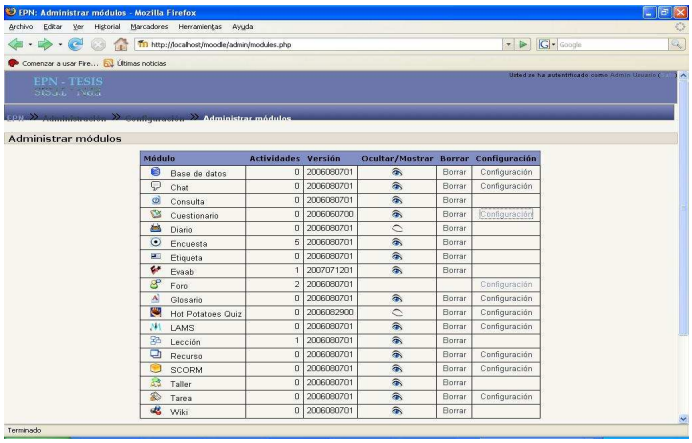

Fig. 4. 4: Pantalla Módulo

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

5.1.1 METODOLOGÍA DESARROLLO

En lo que respecta a la metodología de desarrollo utilizada, la cual fue ***Extreme Programming*** se puede concluir lo siguiente:

- La forma como se organiza y maneja un proyecto requiere de un control sumamente preciso, ya que por tratarse de un concepto diferente en la forma de llevar el proceso de desarrollo mucha de las veces confundimos el concepto de ágil por fácil.

- ***Extreme Programming (XP)*** es una metodología de desarrollo muy acentuada a la realidad de un proyecto ya que se preocupa por obtener un proyecto de calidad en tiempos cortos, algo que se pueda mostrar y refinar con el tiempo sin preocuparse de temas formalistas que muchas de las veces nos quitan tiempo

- No existen muchas herramientas que permitan llevar todo el proceso de ***Extreme Programming***, actualmente las que existen como por ejemplo ***XPlanner*** que fue utilizada en el proyecto solo permiten manejar el juego de la planificación. Aunque existen otras alternativas pagadas como ***VersionOne*** que fue diseñada para manejar el proceso de metodologías ágiles pero de una manera muy genérica.

5.1.2 EVAAB

En lo que respecta a ***EVAAB*** como un módulo adicional al E-learning ***Moodle*** se puede concluir lo siguiente:

- El desarrollo de ***EVAAB*** es un proceso progresivo que no terminó con el desarrollo del módulo en si, sino que se mantiene vigente a través del autor y la comunidad para realizar los cambios que se crean conveniente en versiones posteriores a ***Moodle***.

- La utilización de estándares, especificaciones de ***Moodle*** utilizados dentro del desarrollo de ***EVAAB*** fueron sumamente imprescindibles para llevar un proceso organizado y garantizar el mantenimiento y flexibilidad en el tiempo.

- ***EVAAB*** está desarrollado de una forma modular, es decir su núcleo cumple con todas las especificaciones, estándares de ***Moodle*** de tal forma que a pesar de que fue desarrollado basado en la ***versión 1.6*** su integración con últimos releases actualmente en su ***versión 1.9*** funciona correctamente.

5.2 RECOMENDACIONES

Las recomendaciones van en el sentido de desarrollar módulos dentro de un Sistema ya concebido, de tal forma que la persona o personas que vayan a desarrollar en este ámbito conozcan cuales son las pautas que se debería tomar cuenta antes de comenzar en el proceso. Tomando en cuenta que si cumple con la filosofía de ***Software Libre*** no será mantenido ni modificado por los creadores necesariamente sino por toda la comunidad que desee hacerlo. Entre estas recomendaciones se puede mencionar las siguientes:

- Antes modificar un proyecto existente se tiene que investigar algunas características importantes de tal forma que en el transcurso no existan inconvenientes fuertes que no pueden ser solucionados. Entre estas características se tiene las siguientes:

  - Verificar como está estructurado, es decir donde deben colocarse los archivos dependiendo de su función de tal forma de no omitir los estándares.
  - Verificar la arquitectura que utiliza para ver si se puede implementar o mejorar algunas características actuales.
  - Verificar si se tiene el soporte adecuado en caso de alguna eventualidad, de tal forma que no se pierda tiempos y se pueda avanzar en el proceso.

- Se debe seguir con todos los estándares aplicados en el proyecto padre de tal forma de garantizar el mantenimiento y la flexibilidad en las diferentes versiones o actualizaciones que se vayan realizando.

- Comunicarse directamente con los líderes de proyecto de tal forma que se pueda captar la atención necesaria que necesita nuestro proyecto, ya que existen un sinnúmero de proyectos de terceros que tienen prioridades superiores al nuestro en tiempo y personas.

BIBLIOGRAFIA

[1] GIOUPM, Introducción al E-learning [en línea], Avda. Complutense s/n - informacion@gioupm.com, [fecha consulta: 21 Marzo 2007], Capítulo I, Disponible en: http://tecnologias.gio.etsit.upm.es/elearning/introduccion-al-e-learning-27.asp

[2] GIOUPM, LMS (Learning Management System) [en línea], Avda. Complutense s/n - informacion@gioupm.com, [fecha consulta: 21 Marzo 2007], Capítulo VII, Disponible en: http://tecnologias.gio.etsit.upm.es/elearning/lms--learning-management-system--sistema-de-gestion-de-aprendizaje--25.asp

[3] GIOUPM, Plataformas E-learning [en línea], Avda. Complutense s/n - informacion@gioupm.com, [fecha consulta: 21 Marzo 2007], Capítulo VI, Disponible en: http://tecnologias.gio.etsit.upm.es/elearning/plataformas-de-e-learning-33.asp

[4] Sistemas de Administración de Aprendizaje [en línea], [fecha consulta: 13 Agosto 2007] Disponible en: http://www.biblioweb.dgsca.unam.mx/libros/repositorios/la_web.htm#24

[5] FOIX Cristian, ZAVANDO Sonia, Estándares E-Learning, Elementos E-learning [en línea], 7 de Octubre del 2002, Versión 1, [fecha consulta: 26 de Marzo del 2007], Disponible en: http://empresas.sence.cl/documentos/elearning/INTEC%20-%20Estandares%20e-learning.pdf

[6] GESEM, Estándares E-learning [en línea], Barcelona - España, [fecha consulta: 21 Marzo 2007], Disponible en: http: //www.formacionaempresas.com/modules.php?op=modload&name=New

[7] GREGORI Álvaro, Una introducción al e-learning para diseñadores Web, Estándares E-learning [en línea], 25 de Agosto 2005, [fecha de consulta: 15 de Marzo 2007], Disponible en: http://www.wikilearning.com/estandares_en_e_learning-wkccp-3944-5.htm

[8] RICE IV William H, Moodle E-leaning Course Development, Olton - Birmingham , Packt Publishing Ltd, 2006, 252 p.

[9] EGUÍLUZ PÉREZ Javier, Introducción a Ajax [en línea], Barcelona - España, [fecha consulta: 26 Octubre 2007], Disponible en: http://www.librosweb.es/ajax

[10] HOLZNER Steven, WILEY John, Ajax Bible, Indianapolisis – Indiana, Wiley Publishing Inc, 2007, 251 p.

[11] MOODLE, Development Process [en línea], [fecha consulta: 16 Enero 2008], Disponible en: http://docs.moodle.org/en/Development:Overview#Development_processes

[12] UNIVERSIDAD POLITÉCNICA DE VALENCIA, RUP, Capítulos I - V [en línea], 25 de Septiembre 2006, [fecha de consulta: 8 de Mayo 2007], Disponible en: https://pid.dsic.upv.es

[13] GPI CONSULTORES, Disciplina de Administración de Proyectos MSF [en línea], [fecha de consulta: 24 Abril 2007], Disponible en: http://www.gpicr.com/msf.aspx

[14] ECHOES, Microsoft Solution Framework an Introduction [en línea], 14 de Noviembre 2006, [fecha de consulta: 24 Abril 2007], Disponible en: http://www.echoes.com/msf/index.html

[15] BECK K, Extreme Programming Explained Embrace Change, Addison Wesley Longman, 2000, 224 p.

[16] CANÓS José, LETELIER Patricio, PENADÉS Carmen, Metodologías Ágiles en el Desarrollo de Software [en línea], Madrid 2002, [fecha consulta: 24 Abril del 2007], Disponible en: http://www.willydev.net/descargas/prev/TodoAgil.Pdf

[17] ACEBAL César, CUEVA Juan, Universidad Oviedo, Extreme Programming (XP): Un nuevo método de desarrollo de Software [en línea], 30 Agosto del 2001, [fecha consulta: 6 Junio del 2007], Disponible en:
http://www.cesaracebal.com/docencia/asignaturas/arquitectura-software/workspace/material/acebal-xp.pdf

[18] DAPENA José, Desarrollo en Comunidad con Extreme Programming [en línea], [fecha consulta: 14 Agosto del 2007], Disponible en: http://www.igalia.com

[19] FUTURA, Metodología de la Formación: Principios y Elementos de Aprendizaje [en línea], [fecha de consulta: 12 Noviembre del 2007], Disponible en:
http://www.cfnavarra.es/inam/futura/metmet.htm

[20] XP Site, The Rules and Practice of Extreme Programming [en línea], 1999, [fecha consulta: 24 Abril 2007], Disponible en: http://www.extremeprogramming.org/rules.html

[21] ABRAHAMSSON P, SALO O, Agile Software Development Methods Review And Analysis, VTT Publications, 2002, 263 p.

[22] SCOUT W, Agile Model Driven Development with UML 2, Cambridge University Press, 2004, 572 p.

ANEXOS

MANUAL DE INSTALACIÓN DE EVAAB

Para realizar la instalación del módulo de evaluación abierto ***EVAAB*** tenemos dos alternativas la primera es realizar la instalación ***conjuntamente con la instalación de Moodle*** y la otra realizar ***la instalación una vez instalado Moodle***, las cuales explicaremos en detalle a continuación.

Conjuntamente con Moodle

La instalación de ***EVAAB*** conjuntamente con la instalación de ***Moodle*** es la instalación más sencilla a seguir para lo cual se tiene que realizar los siguientes pasos.

1. Descargar el plugin referente a ***EVAAB*** que actualmente la fecha de este trabajo ya se encuentra en proceso de revisión en el ***Tracker de Moodle*** referenciado con la especificación ***CONTRIB-335***. Por lo que próximamente lo podremos descargar de la página oficial de plugins de ***Moodle***[34].
2. Descomprimir el plugin bajado y copiarlo en el directorio de ***Moodle*** referente a los módulos el cual es ***/moodle/mod***
3. Realizar la instalación de ***Moodle***, la cual en su proceso de instalación en uno de sus pasos instalará todos los módulos que lo conforman incluyendo a ***EVAAB.***
4. Una vez realizado esto la instalación de ***EVAAB*** ha terminado, para la comprobación se tiene que crear una instancia de ***EVAAB*** una vez que se haya definido un curso.

Después de la Instalación de Moodle

El proceso de instalación luego de tener instalado y configurado ***Moodle*** se diferencia del anterior porque se tiene que crear las estructuras de la base de datos manualmente. Para lo cual se tiene que adicionar a los puntos anteriores 1,2 los siguientes pasos:

[34] Modules and Plugins Moodle -http://moodle.org/mod/data/view.php?id=6009

1. Se debe ingresar un registro de ***EVAAB*** en la tabla de módulos de ***Moodle*** la cual tiene el nombre de ***<prefix>_modules***, donde prefix es el prefijo con el cual se realizó la instalación de Moodle.
2. Luego de tener registrado a ***evaab*** dentro de los módulos de ***Moodle*** se tiene que correr los script de ***EVAAB,*** los mismos que tienen todas las estructuras de base de datos necesarias para su funcionamiento. Estos se encuentran localizados en el directorio ***/evaab/db/*** en un archivo denominado con el nombre de ***mysql.sql*** en el cual se tiene que realizar los cambios respectivos en su contenido de la palabra ***prefix*** por el prefijo utilizado por ***Moodle*** en su instalación.
3. Una vez realizado esto la instalación de ***EVAAB*** ha terminado, para la comprobación se tiene que crear una instancia de ***EVAAB*** una vez que se haya definido un curso.

MANUAL DE USUARIO DE EVAAB

Para describir el funcionamiento de ***EVAAB*** lo haremos tomando en cuenta la perspectiva del ***Profesor*** y la del ***Estudiante*** ya que cada uno maneja opciones diferentes.

PROFESOR

Para comenzar a utilizar el módulo de evaluación abierto ***EVAAB*** se debe primeramente agregar una actividad al curso, lo cual se lo puede apreciar en la siguiente figura.

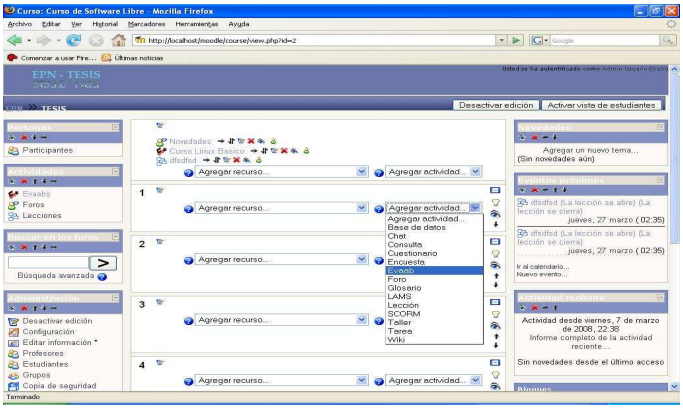

Fig. A 1: Agregando Actividad EVAAB

Luego de esto se nos presenta la pantalla principal de ***EVAAB***, la cual nos presenta dos opciones una para comenzar con la creación del cuestionario denotada por el texto ***CARGAR DATOS*** y otra para presentar la ayuda denotada por el texto ***AYUDA.*** Esto se lo puede apreciar en la siguiente figura.

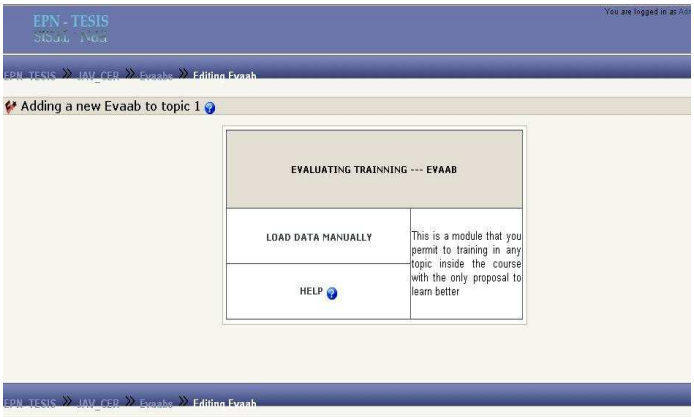

Fig. A 2: Opciones Profesor

Al seleccionar la opción primera se nos presenta una interfaz donde procederemos a gestionar el cuestionario junto con los elementos a creerse, la descripción de cada elemento de la interfaz se lo puede apreciar en la figura a continuación.

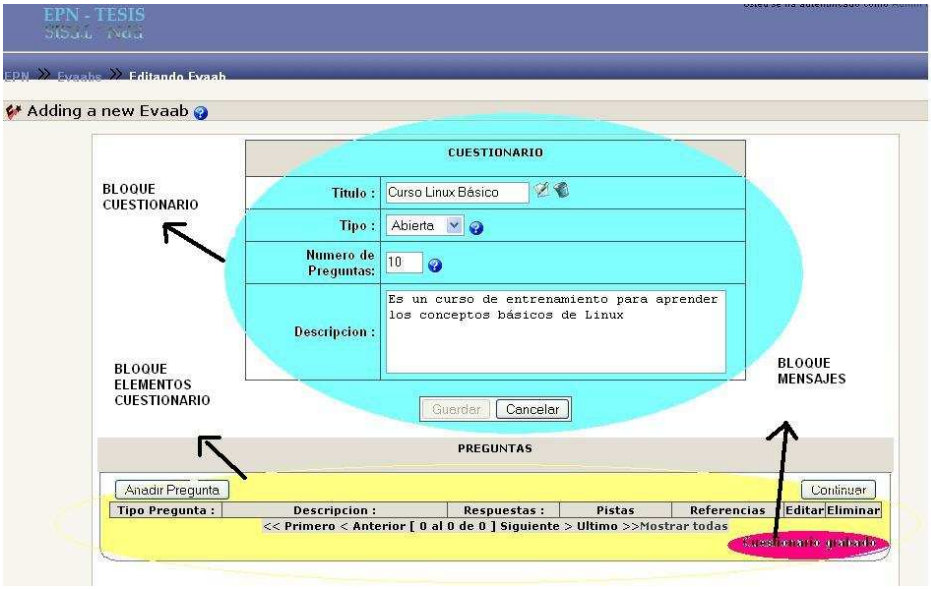

Fig. A 3: Elementos Interfaz Profesor

Como se puede apreciar en el gráfico tenemos tres bloques de trabajo, los cuales explicaremos a continuación:

- ***Bloque Cuestionario.-*** En este bloque se gestionará todas las operaciones relacionadas con el elemento principal del módulo de evaluación ***EVAAB*** que es el cuestionario. El cual debe ser creado para poder proseguir con la gestión de los siguientes elementos como son las preguntas, respuestas, pistas y referencias.

- ***Bloque Elementos Cuestionario.-*** Este bloque representa un cuadro de resumen de todas las operaciones realizadas a los demás elementos del cuestionario como son ***las preguntas, respuestas, pistas y referencias.*** Aquí podremos gestionar todos los elementos con sus diferentes operaciones y observar su comportamiento de una forma dinámica.

- ***Bloque de Mensajes.-*** En este bloque se presentará todos los mensajes informativos en cuanto a operaciones para los diferentes elementos de ***EVAAB***, exceptuando las validaciones de los elementos dentro del formulario.

Cuestionario

Primero debemos crear el cuestionario para lo que tenemos dos tipos, cada uno con sus características definidas entre estas tenemos:

- Un ***cuestionario tipo abierto*** el cual se caracteriza porque en su resolución permite interactuar con los elementos de ayuda como pistas y referencias, además que el número de preguntas es parametrizable.

- Un ***cuestionario tipo cerrado*** el cual se caracteriza porque en su resolución no cuenta con los elementos de ayuda como pistas y referencias, además que el número de preguntas es un número fijo.

Una vez llenado los datos del cuestionario entre estos el tipo se procede a grabar, si la grabada fue exitosa se nos presenta el mensaje correspondiente de lo contrario se nos presenta un cuadro informativo indicando los errores. A continuación en las figuras se puede apreciar lo expuesto anteriormente.

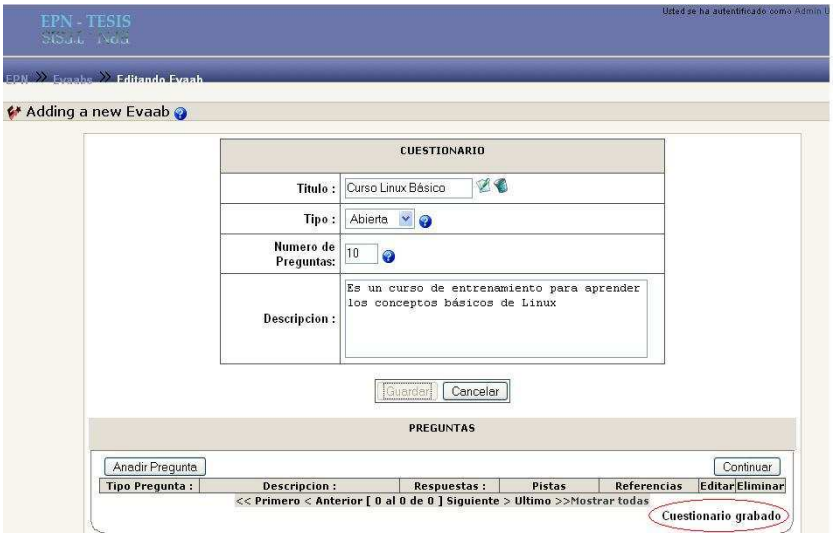

Fig. A 4: Cuestionario Grabado Exitosamente

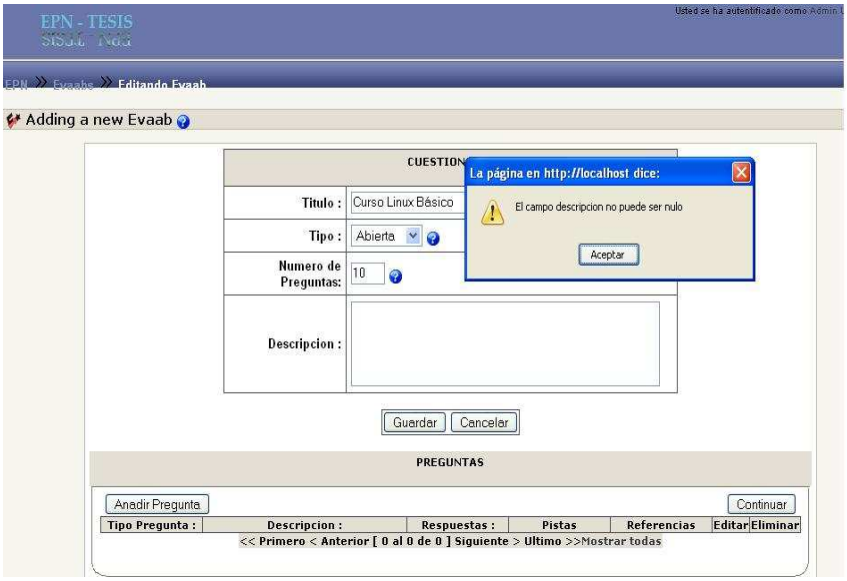

Fig. A 5: Grabada de Cuestionario No Exitosa

Una vez grabado el cuestionario éste puede ser ***actualizado o eliminado*** para lo cual se debe seleccionar los iconos correspondientes a cada operación que se encuentran ubicados a lado derecho del título , se debe tener en cuenta cuando se realice la eliminación del cuestionario ***los elementos enlazados al cuestionario como preguntas conjuntamente con sus respuestas y elementos de ayuda también serán borrados***.

Cabe destacar que las operaciones de actualización y eliminación del cuestionario siguen el mismo patrón de comportamiento de la operación de inserción del cuestionario, es decir que si la operación fue realizada satisfactoriamente se presenta el mensaje de satisfacción, caso contrario se presentará el mensaje de error.

Pregunta

Una vez grabado el cuestionario se debe proseguir con la creación de las preguntas, para lo cual se debe presionar el botón denotado por el texto **Añadir Pregunta**. El mismo que solo realizará su ejecución cuando se haya creado previamente un cuestionario de lo contrario nos presentará el mensaje correspondiente, el cual se lo puede apreciar en la figura a continuación:

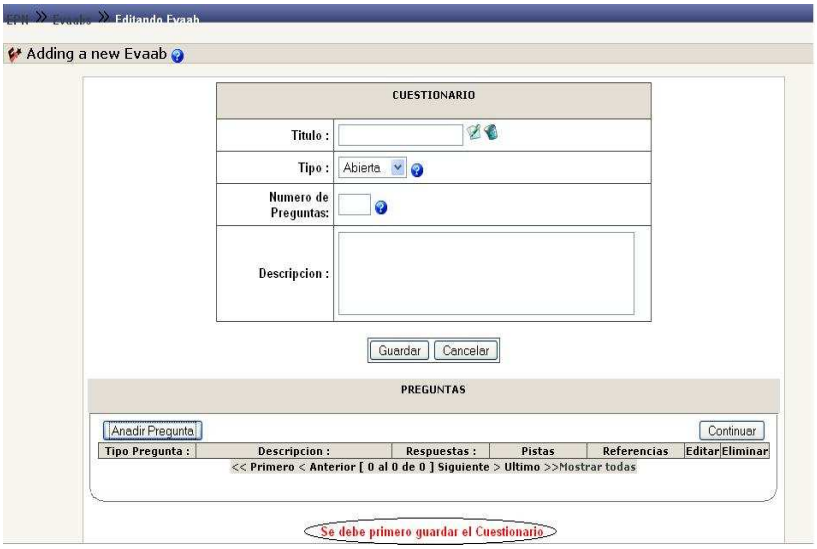

Fig. A 6: Creación de Pregunta Errada

Cuando el cuestionario ya ha sido creado se debe proceder a crear las preguntas para lo cual se debe elegir el tipo de pregunta, su descripción y su número de respuestas. Si todos los datos están correctos se procederá ***a grabar la pregunta, crear sus respuestas y actualizar el bloque de elementos del Cuestionario en las secciones respectivas*** de lo contrario se nos presentará un cuadro informático indicando los errores, esto se lo puede apreciar en las figuras a continuación.

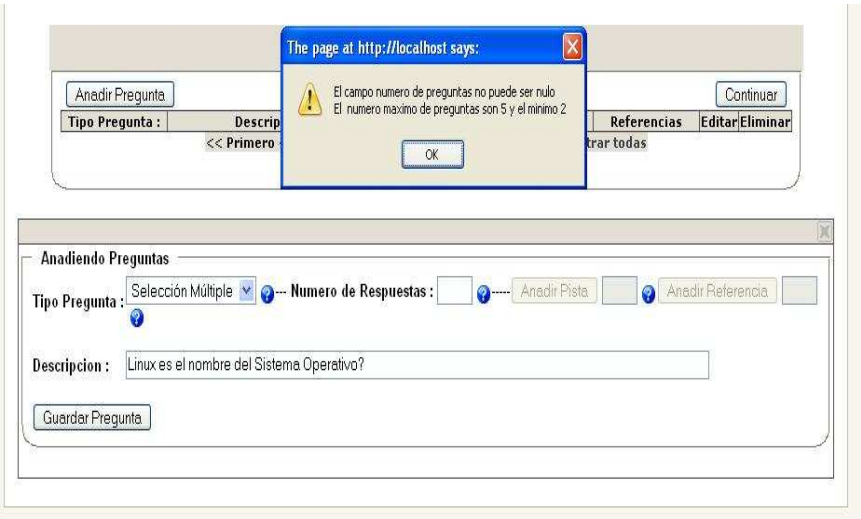

Fig. A 7: Grabada de Pregunta No Exitosa

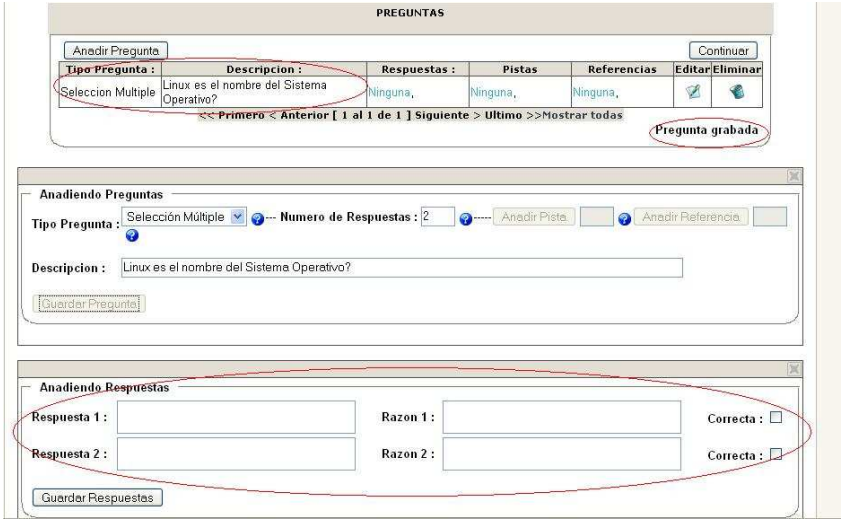

Fig. A 8: Pregunta Grabada Exitosamente

Respuestas

La gestión de respuestas como vimos en la figura anterior va ligada ***directamente con la pregunta*** en el momento de guardar la misma, por lo que luego de guardar la pregunta se debe proceder a llenar sus respuestas ingresando su descripción, razón (motivo por el cual es correcta o incorrecta) y estado para cada una de ellas. Cabe destacar que el rango de respuestas esta entre los valores de ***2 – 5***, mínimo y máximo respectivamente.

Si algún campo de las respuestas esta incorrecto se nos presenta un cuadro informativo indicando los errores, caso contrario se realiza ***la grabación de las respuestas y se actualiza el bloque de elementos de cuestionario en la sección correspondiente***. Lo expuesto se lo puede apreciar en las siguientes figuras mostradas a continuación.

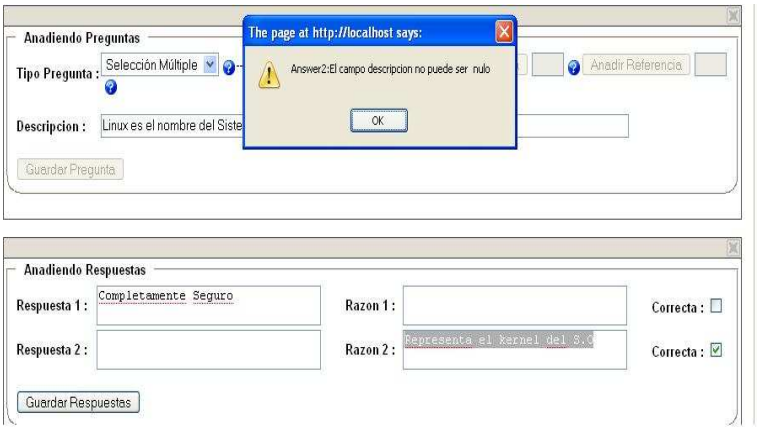

Fig. A 9: Grabada de Respuesta No Exitosa

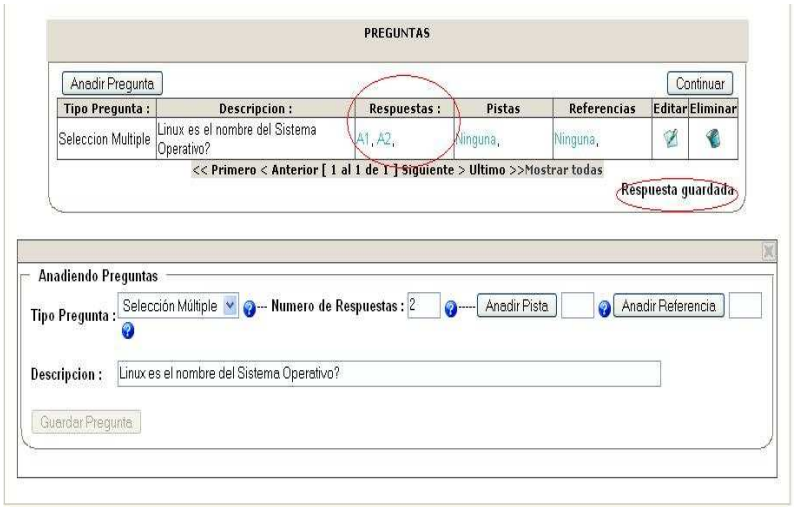

Fig. A 10: Respuestas Grabada Exitosamente

Pistas

La gestión de las pistas presenta también una relación con la pregunta pero menos fuerte ya que esta representa una ayuda a la resolución de la misma, lo cual no quiera decir que no sea importante ya que este elemento representa un ***elemento fundamental en la resolución del cuestionario ya que proporciona retroalimentación.***

Para la generación de las pistas se debe ingresar un número entre los valores de ***2 – 5***, mínimo y máximo respectivamente en el cuadro referente a las pistas en el bloque de las preguntas, luego presionar el botón ***Añadir Pistas*** para su generación. Una vez generado las pistas debemos ingresar los datos para lo que si alguno de sus elementos esta incorrecto se nos presenta un cuadro informativo indicando los errores, caso contrario se realiza ***la***

grabación de las pistas y se actualiza el bloque de elementos de cuestionario en la sección correspondiente. Lo expuesto se lo puede apreciar en las siguientes figuras mostradas a continuación.

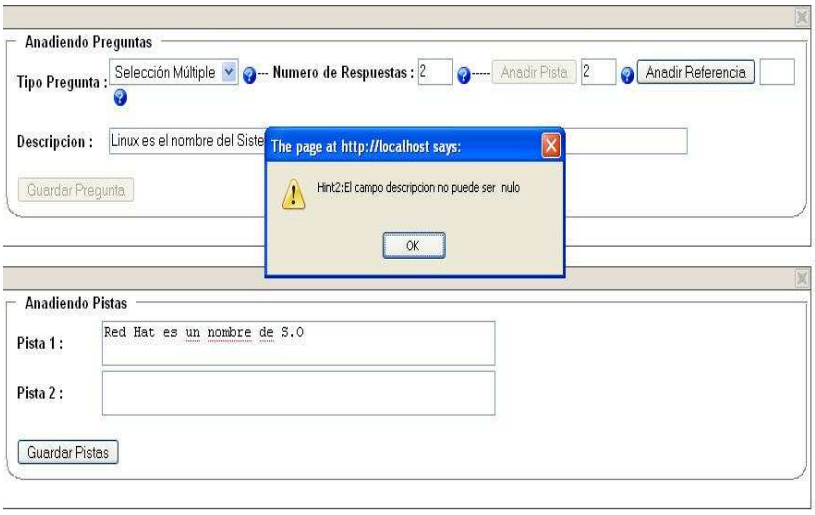

Fig. A 11: Grabada de Pista No Exitosa

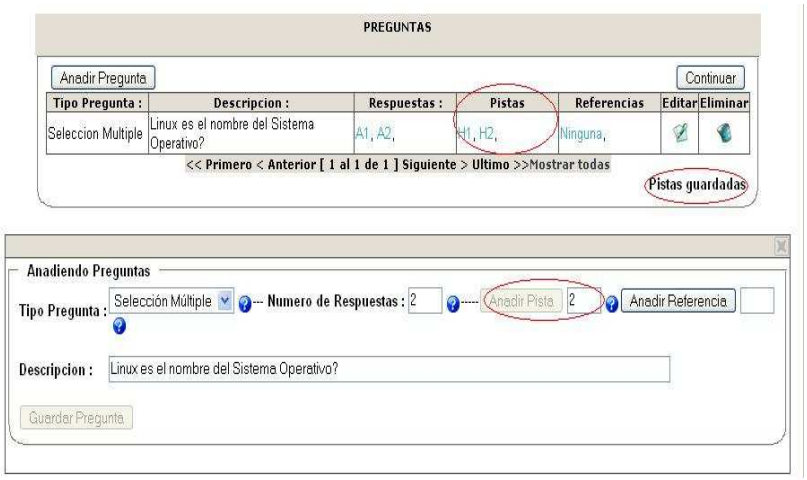

Fig. A 12: Pistas Grabadas Exitosamente

Referencias

La gestión de las referencias presenta al igual que las pistas una relación con la pregunta que ayuda a la resolución de la misma, representa también un ***elemento fundamental en la resolución del cuestionario*** pero se diferencia de las pistas porque presenta una ayuda más exacta a través de un sinnúmero de recursos como pueden ser libros digitales, videos etc.

Para la generación de las referencias se debe ingresar un número entre los valores de ***2 – 5***, mínimo y máximo respectivamente en el cuadro referente a las referencias en el bloque de las preguntas, luego presionar el botón ***Añadir Referencias*** para su generación. Una vez generado las referencias debemos ingresar los datos para lo que si alguno de sus elementos esta incorrecto se nos presenta un cuadro informativo indicando los errores, caso contrario se realiza ***la grabación de las referencias y se actualiza el bloque de elementos de cuestionario en la sección correspondiente***. Lo expuesto se lo puede apreciar en las siguientes figuras mostradas a continuación.

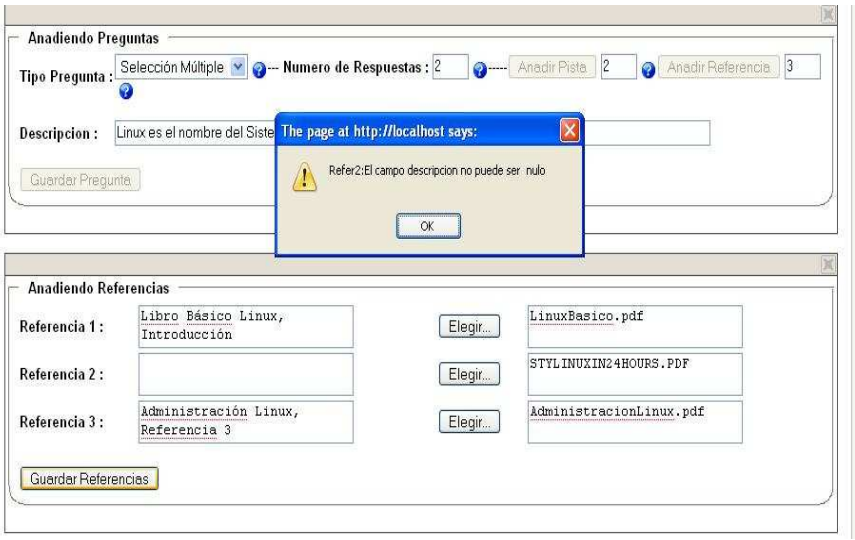

Fig. A 13: Grabada de Referencia No Exitosa

Fig. A 14: Referencias Grabadas Exitosamente

Como algo adicional se puede mencionar que para ***la elección de los recursos en las referencias utilizamos los recursos proporcionados por Moodle***, los cuales han sido adoptados a ***EVAAB*** para su utilización. Lo cual se lo puede apreciar en la siguiente figura:

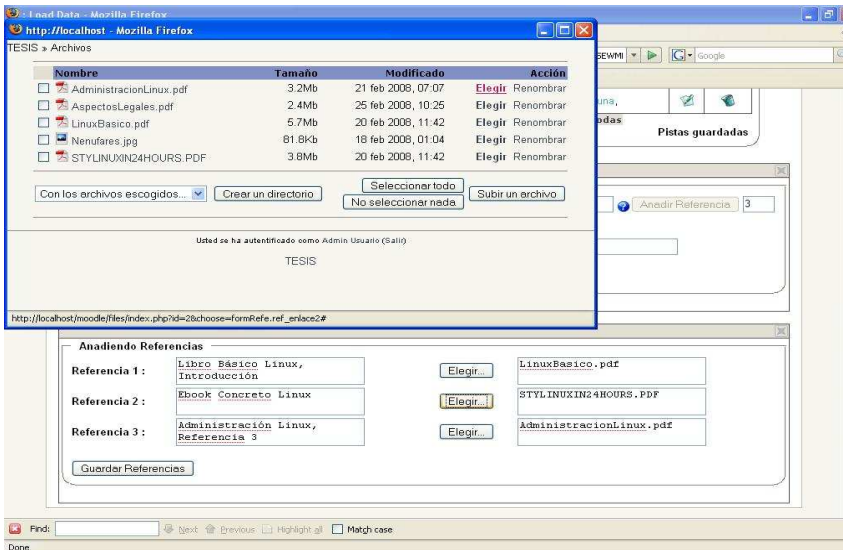

Fig. A 15: Pantalla Recursos Referencias

Luego de haber ingresado todas las preguntas ***definidas en el Cuestionario junto con sus elementos esta listo para ser resuelta por el Estudiante***. La interfaz final con todos los elementos se los puede apreciar a continuación en la siguiente figura.

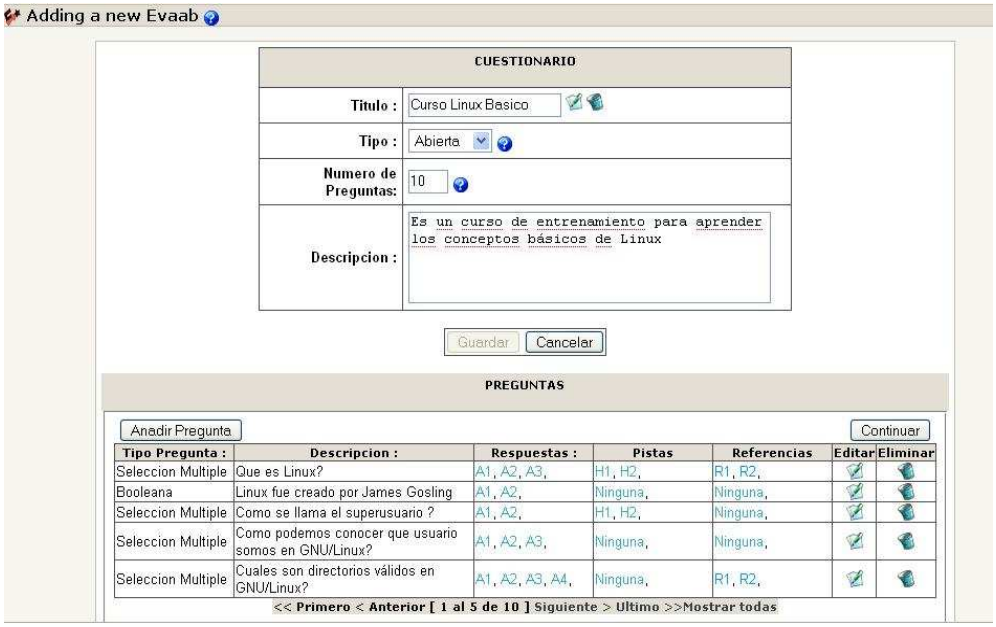

Fig. A 16: Pantalla Final Cuestionario

ESTUDIANTE

Una vez que se ha realizado el Cuestionario el siguiente paso es proceder a resolverlo, lo cual esta a cargo del Estudiante, para lo cual el paso inicial luego de elegir la instancia del cuestionario es dar un clic en el enlace ***Resolver Cuestionario*** lo que genera los elementos del cuestionario dando inicio a la resolución del mismo. La interfaz principal junto con los elementos se los puede apreciar a continuación en la siguiente figura:

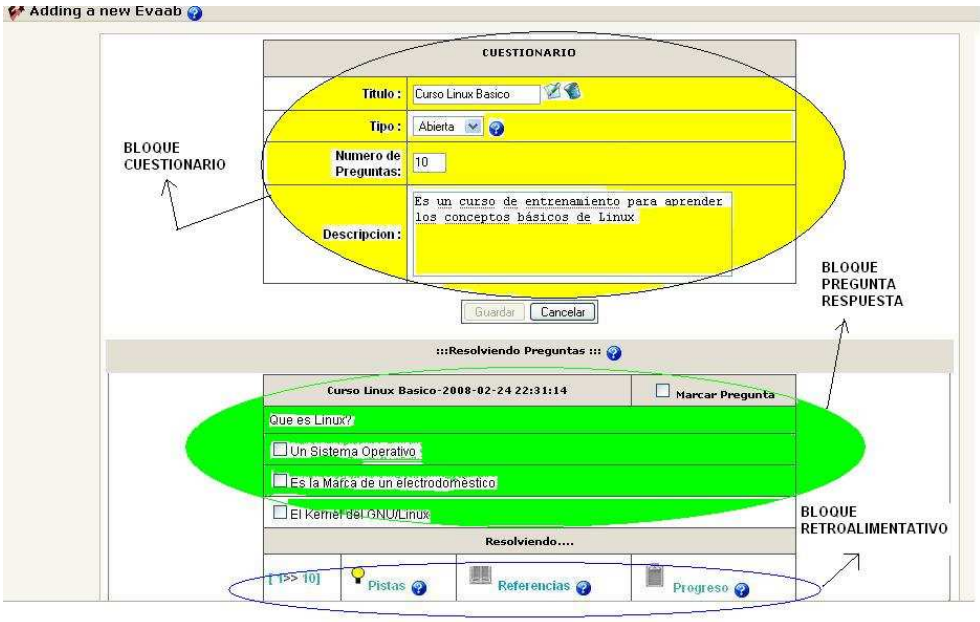

Fig. A 17: Elementos Interfaz Estudiante

Como se puede apreciar en el gráfico tenemos tres bloques de trabajo, los cuales explicaremos a continuación:

- ***Bloque Cuestionario.-*** En este bloque se previsualiza la cabecera del cuestionario, el cual tiene elementos fundamentales como el número de preguntas y el tipo de cuestionario
- ***Bloque Pregunta Respuesta.-*** Este bloque contiene ***las preguntas y respuestas*** del cuestionario las cuales irán apareciendo de acuerdo al orden que fueron ingresadas.
- ***Bloque Retroalimentativo.-*** Este bloque contiene uno de los elementos fundamentales en la resolución del cuestionario que son las ***pistas y referencias***.

Retroalimentación Instantánea

La retroalimentación instantánea consiste uno de los elementos claves en la resolución del cuestionario ya que nos permite conocer a la marcha el estado de la pregunta contestada, el cual puede ser ***correcto***, ***incorrecto.*** Funciona automáticamente cuando ***se prosigue con la resolución de la siguiente pregunta*** mostrando en un mensaje informativo uno de los estados anteriormente expuestos según la contestación de la pregunta, lo cual se lo puede apreciar en la siguiente figura a continuación.

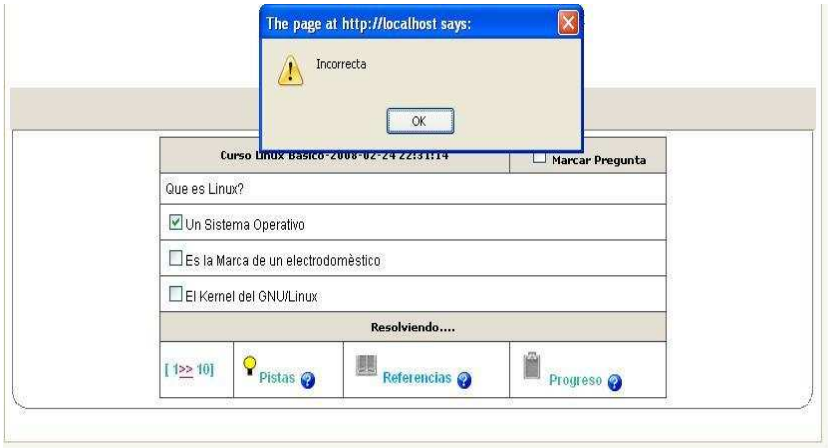

Fig. A 18: Pantalla Retroalimentación Instantánea

Además tenemos la ***opción de contestar luego la pregunta en el caso de que no estemos seguros de su respuesta***, para lo cual necesitamos marcar la pregunta con un visto en el check box como que se muestra en la figura a continuación.

Fig. A 19: Pantalla Pregunta Postergada

Visualización De Elementos De Ayuda

En la resolución de cada pregunta contamos con dos elementos de ayuda los cuales son las ***pistas y referencias*** que pueden mostrarse siempre y cuando ***hayan sido gestionados en el ingreso del cuestionario y además el cuestionario sea de tipo abierto***. Ya que caso contrario no serán habilitados, tanto la visualización de las pistas como de las referencias se los puede apreciar en las siguientes figuras a continuación.

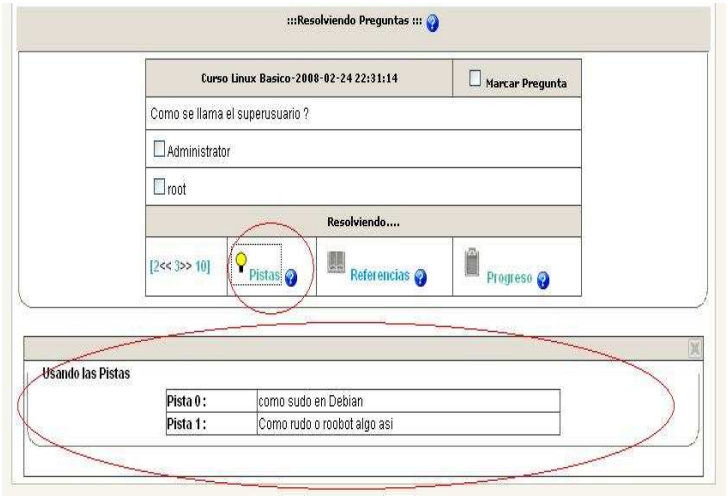

Fig. A 20: Pantalla Visualización Pistas

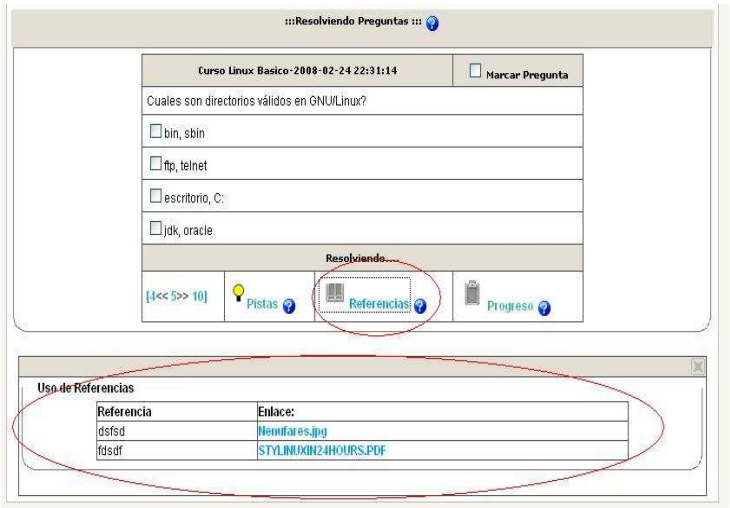

Fig. A 21: Pantalla Visualización Referencias

Una vez que se haya desplegado las referencias se puede consumir cada uno de estos recursos dando un clic sobre el que necesitamos, a lo que se nos presentará el tipo de recurso seleccionado como se muestra en la siguiente figura:

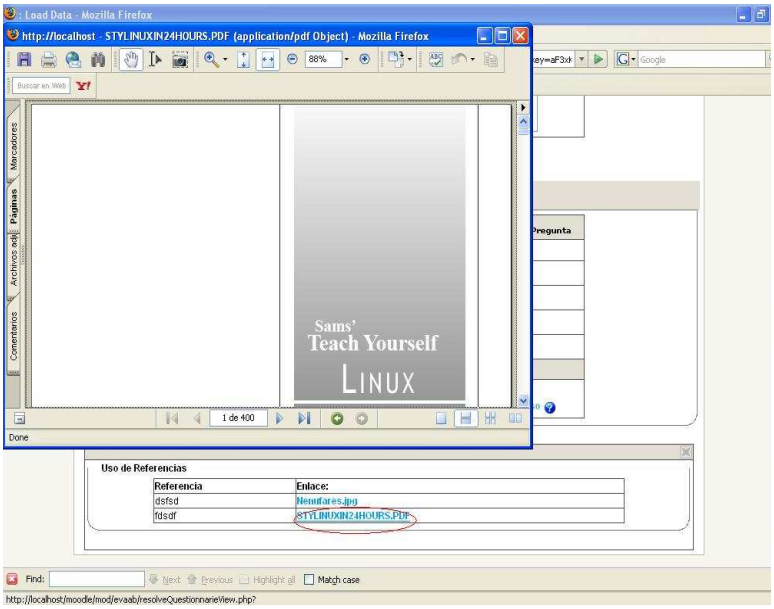

Fig. A 22: Mostrando Recurso Referencia

Seguimiento De Progreso

En cualquier punto de la resolución del cuestionario, es decir independientemente del número de pregunta que nos encontremos resolviendo se puede conocer en que estado nos encontramos actualmente en la resolución del Cuestionario a través del ***Progreso***. El cual nos muestra un resumen de las preguntas con sus diferentes estados y porcentajes, el cual se lo puede apreciar en la siguiente figura.

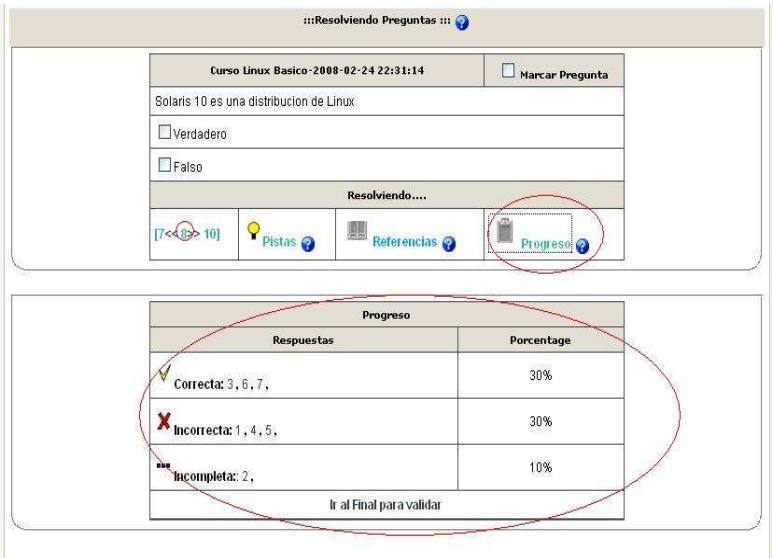

Fig. A 23: Pantalla de Progreso Resolución

Generación De Calificación

Luego de terminar con la resolución de todas las preguntas tenemos la opción de revisar el estado de todas las preguntas a través del ***Progreso*** y actualizarlas si fuera el caso de lo contrario se puede generar la calificación. Para lo cual debemos seleccionar el enlace ***Validar*** el mismo que nos muestra un mensaje que nos pregunta si estamos seguros o no de generar la calificación como se puede apreciar en la figura a continuación.

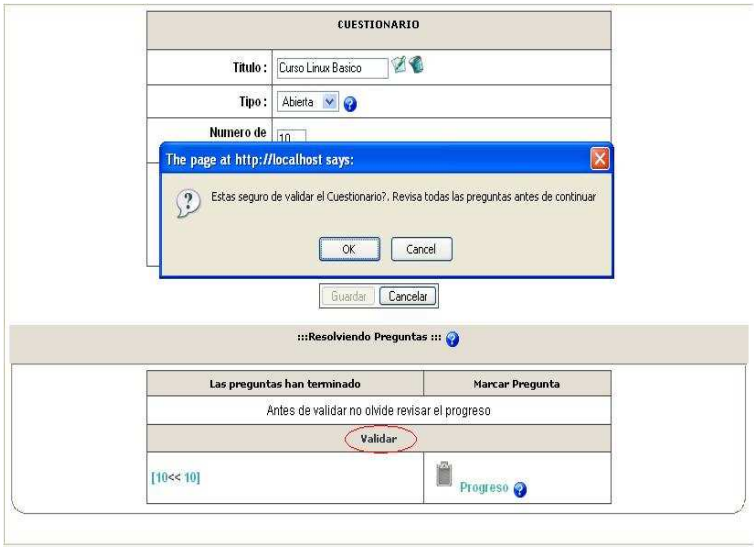

Fig. A 24: Pantalla Preliminar Generación Calificación

Si estamos seguros de nuestra resolución y procedemos a afirmar el mensaje anterior se procederá a generar la calificación correspondiente del Estudiante en curso, la misma que se puede apreciar en la figura a continuación:

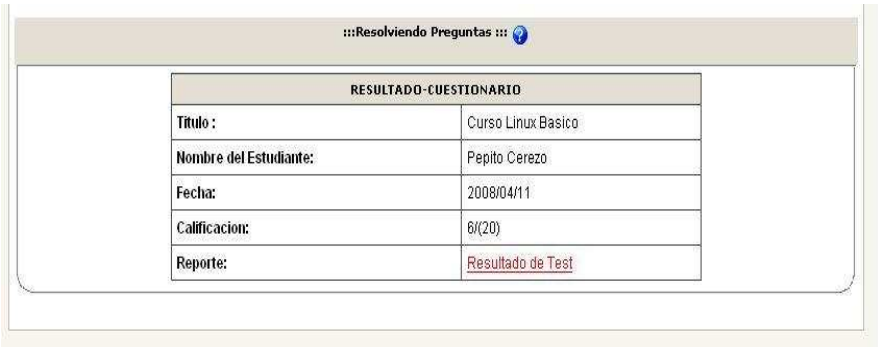

Fig. A 25: Pantalla Generación Calificación

Generación de Informe

Una vez generada la calificación correspondiente del Estudiante el último paso que tenemos y ***uno de los más importantes en el proceso de resolución del Cuestionario es el informe final que se genera***, el cual contiene toda la información del Cuestionario junto con las pautas retroalimentativas como son detalle equivocación, respuesta correcta etc. Este informe se puede apreciar en la figura a continuación:

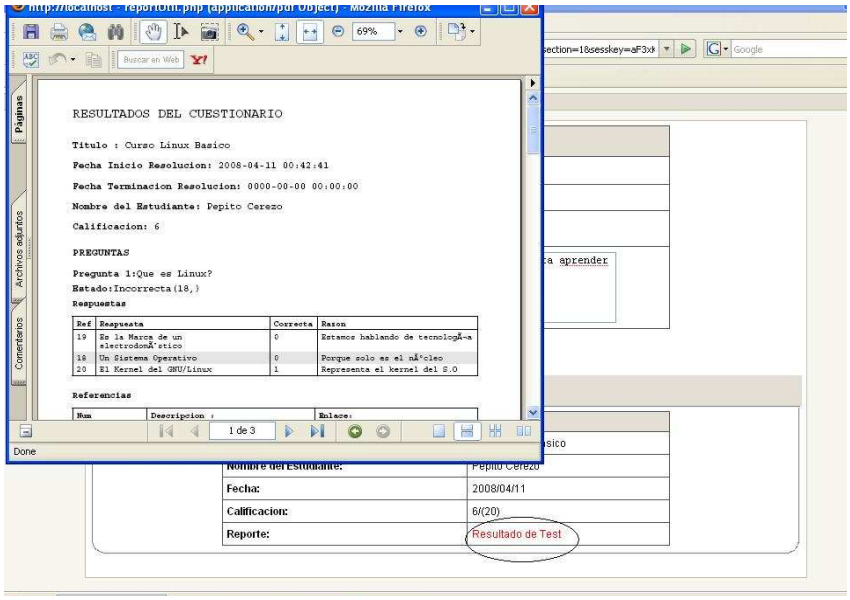

Fig. A 26: Pantalla Generación Informe