

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

OPCIÓN I.- TESIS

TRABAJO PROFESIONAL

"PLATAFORMA PARA LA GESTIÓN DE
EVALUACIÓN DE EXÁMENES".

QUE PARA OBTENER EL GRADO DE:
MAESTRO EN SISTEMAS
COMPUTACIONALES

PRESENTA:

I.S.C. Israel Colohua Cruz

DIRECTOR DE TESIS:

Mtro. Luis Ángel Reyes Hernández

CODIRECTOR DE TESIS:

Dr. Hilarión Muñoz Contreras



ORIZABA, VERACRUZ, MÉXICO.

MAYO 2017



"Año del Centenario de la Promulgación de la Constitución Política de los Estados Unidos Mexicanos"

FECHA: 17/05/2017
DEPENDENCIA: POSGRADO
ASUNTO: Autorización de Impresión
OPCIÓN: I

C. ISRAEL COLOHUA CRUZ
CANDIDATO A GRADO DE MAESTRO EN:
SISTEMAS COMPUTACIONALES

De acuerdo con el Reglamento de Titulación vigente de los Centros de Enseñanza Técnica Superior, dependiente de la Dirección General de Institutos Tecnológicos de la Secretaría de Educación Pública y habiendo cumplido con todas las indicaciones que la Comisión Revisora le hizo respecto a su Trabajo Profesional titulado:

"PLATAFORMA PARA LA GESTION DE EVALUACION DE EXAMENES".

Comunico a Usted que este Departamento concede su autorización para que proceda a la impresión del mismo.

A T E N T A M E N T E


M.C. MA. ELENA GARCÍA REYES
JEFE DE LA DIV. DE ESTUDIOS DE POSGRADO

C.A. TITULACIÓN



**SECRETARIA DE
EDUCACIÓN PÚBLICA
INSTITUTO
TECNOLÓGICO
DE ORIZABA**

ggc





"Año del Centenario de la Promulgación de la Constitución Política de los Estados Unidos Mexicanos"

FECHA : 27/04/2017

ASUNTO: Revisión de Trabajo Escrito

C. M.C. MA. ELENA GARCÍA REYES
JEFE DE LA DIVISION DE ESTUDIOS
DE POSGRADO E INVESTIGACION.
P R E S E N T E

Los que suscriben, miembros del jurado, han realizado la revisión de la Tesis del (la) C. :

ISRAEL COLOHUA CRUZ

la cual lleva el título de:

"PLATAFORMA PARA LA GESTION DE EVALUACION DE EXAMENES".

Y concluyen que se acepta.

A T E N T A M E N T E

PRESIDENTE : **MTRO. LUIS ANGEL REYES HERNANDEZ**

SECRETARIO : **DR. HILARION MUÑOZ CONTRERAS**

VOCAL : **MCE BEATRIZ ALEJANDRA OLIVARES**

VOCAL SUP. : **DR.. ULISES JUAREZ MARTINEZ**

FIRMA

FIRMA

FIRMA

FIRMA

EGRESADO(A) DE LA MAESTRIA EN **SISTEMAS COMPUTACIONALES**

OPCION: **I Tesis**



Agradecimientos

A mis padres el Sr. **Luciano Colohua Colohua** y la Sra. **Herminia Cruz Rivera** que me dieron la vida y su cariño, además de enseñarme a ser siempre sincero y humilde. A mi hermano **Juan Manuel** por el tiempo que hemos compartido en los momentos buenos y malos que hemos estado. A la vida que me enseñó que no importa los retos, siempre se logran las cosas a pesar de las dificultades que se nos presentan.

A la **L.C Carmen Celeste Conguillo Nieto** por el apoyo moral y los ánimos de seguir adelante, además de confiar en las decisiones que he tomado, así como agradecerle todas las atenciones que me ha brindado en cada descuido que doy.

A mi director de tesis, el **M.S.C Luis Ángel Reyes Hernández** por el apoyo durante la elaboración de la tesis, además de agradecerle la paciencia que me tuvo, tolerar mis acciones negativas durante la maestría y guiarme para culminar con éxito este trabajo.

A la **M.C.E Beatriz Olivares Zepahua** por su gran paciencia que me tuvo al guiarme durante las materias de la licenciatura, residencias, propedéutico y maestría, agradezco sus conocimientos en programación, gracias a ellos, le agarre cariño a la carrera.

Al **Dr. José Luis Sánchez Cervantes** por el apoyo y atención brindada durante la etapa de estancias, además de ser una gran persona, le agradezco el tiempo disponible que me brindo para la elaboración de mi artículo.

A mis revisores el **Dr. Ulises Juárez López** y al **Dr. Hilarión Muñoz Contreras** por sus observaciones y comentarios que permitieron que este trabajo de tesis tenga mejoras.

A todos mis amigos que me han apoyado, les agradezco su tiempo que me brindado y por su valiosa amistad, en especial a **Kevin Uriel H.V** por brindarme su apoyo.

Al **Ing. Víctor Manuel Torres Morales** por su apoyo en cada momento, además de ser una gran persona que ha confiado en todo lo que me he propuesto, el cual lo considero como un amigo en el que puedo confiar.

Al Consejo Nacional de Ciencia y Tecnología (**CONACyT**), por el apoyo económico otorgado para seguir con mis estudios.

Índice

| | |
|--|------|
| Índice de Imágenes | i |
| Índice de Tablas..... | iv |
| Índice de Listados..... | vi |
| Resumen | vii |
| Abstract..... | viii |
| Introducción..... | 1 |
| Capítulo 1. Antecedentes..... | 2 |
| 1.1 Marco teórico..... | 2 |
| 1.1.1 Lenguaje Web | 2 |
| 1.1.2 Framework | 2 |
| 1.1.3 Ontología..... | 3 |
| 1.1.4 Methontology | 4 |
| 1.1.5 Sistema de gestión de contenido | 5 |
| 1.1.6 Quiz (Evaluación) | 5 |
| 1.1.7 MCQ (Evaluación de Opción Múltiple)..... | 6 |
| 1.1.8 Shuffle | 6 |
| 1.1.9 E-learning..... | 7 |
| 1.1.10 Ingeniería <i>Web</i> | 7 |
| 1.1.11 SPARQL..... | 8 |
| 1.2 Planteamiento del problema..... | 8 |
| 1.4 Objetivos | 9 |
| General | 9 |
| Específicos | 9 |
| 1.5 Justificación | 9 |
| Capítulo 2. Estado de la práctica | 11 |
| 2.1 Trabajos relacionados | 11 |
| 2.2 Análisis comparativo | 17 |
| 2.3 Propuesta de solución | 20 |
| 2.4 Justificación de la solución seleccionada..... | 21 |

| | |
|---|----|
| Capítulo 3. Aplicación de la metodología | 22 |
| 3.1 Formulación | 22 |
| 3.2 Planificación | 23 |
| 3.3 Análisis | 23 |
| 3.3.1 Caso de uso “Gestionar Cursos” | 24 |
| 3.3.2 Caso de uso “Gestionar Temas” | 24 |
| 3.3.3 Caso de uso “Gestionar Grupos” | 25 |
| 3.3.4 Caso de uso “Gestionar Alumnos” | 26 |
| 3.3.5 Caso de uso “Gestionar Preguntas” | 27 |
| 3.3.6 Caso de uso “Gestionar exámenes” | 28 |
| 3.3.7 Caso de uso “Ver concentrado” | 29 |
| 3.3.8 Especificación y conceptualización de la ontología <i>itoTest</i> | 30 |
| 3.3.9 Codificación en <i>Protégé</i> | 37 |
| 3.4 Ingeniería | 41 |
| 3.4.1 Diseño arquitectónico | 41 |
| 3.4.2 Vista Lógica | 42 |
| 3.4.3 Vista de Desarrollo | 43 |
| 3.4.4 Vista física | 43 |
| 3.4.5 Diseño del contenido | 44 |
| 3.4.6 Acceso a la ontología | 44 |
| 3.4.6 Narrativas de Caso de Uso | 45 |
| 3.5 Generación de páginas | 54 |
| 3.6 Etapa de pruebas | 65 |
| 3.7 Evaluación y aceptación del cliente | 65 |
| Capítulo 4. Resultados | 66 |
| 4.1 Agregar Profesor | 66 |
| 4.2 Agregar curso y temas | 72 |
| 4.3 Asignar cursos | 74 |
| 4.4 Consultar cursos asignados | 76 |
| 4.4 Registrar un alumno | 77 |

| | |
|---|----|
| 4.5 Registrar un reactivo | 80 |
| Capítulo 5. Conclusiones y trabajos a futuro..... | 82 |
| 5.1 Conclusiones | 82 |
| 5.2 Trabajos a futuro..... | 83 |
| Glosario de Términos | 84 |
| Productos académicos..... | 85 |
| Artefactos..... | 86 |
| Bibliografía..... | 87 |

Índice de Imágenes

| | |
|--|----|
| Fig. 3.1 Diagrama de casos de usos extendido de alto nivel. | 23 |
| Fig.3.2 Diagrama de casos de usos “Gestionar Curso” detallado. | 24 |
| Fig. 3.3 Diagrama de casos de usos “Gestionar Tema” detallado. | 25 |
| Fig. 3.4. Diagrama de casos de usos “Gestionar Grupos” detallado. | 26 |
| Fig. 3.5 Diagrama de casos de usos “Gestionar Alumnos” detallado. | 27 |
| Fig. 3.6 Diagrama de casos de usos “Gestionar Preguntas” detallado. | 28 |
| Fig. 3.7 Diagrama de casos de usos “Gestionar Exámenes” detallado. | 29 |
| Fig. 3.8 Taxonomía de conceptos. | 32 |
| Fig. 3.9 Diagrama de relaciones binarias | 33 |
| Fig. 3.10 Creación de clases del dominio de <i>E-learning</i> en <i>Protégé</i> | 37 |
| Fig. 3.11 Relaciones binarias del dominio <i>E-learning</i> | 38 |
| Fig. 3.12 Instancias agregadas a la ontología del dominio <i>E-learning</i> | 38 |
| Fig. 3.13 Resultado de la consulta sobre los profesores. | 39 |
| Fig. 3.14 Resultado de la consulta de los cursos | 40 |
| Fig. 3.15 Resultado de la consulta de buscar los temas del curso 1 | 40 |
| Fig. 3.16 Patrón Arquitectónico MVC | 41 |
| Fig. 3.17 Diagrama de clases..... | 42 |
| Fig. 3.18 Distribución del patrón arquitectónico de JSF | 43 |
| Fig. 3.19 Diagrama de despliegue | 44 |
| Fig. 3.20 Prototipo de la interfaz para el administrador. | 54 |
| Fig. 3.21 Prototipo de interfaz responsiva para el alumno. | 55 |
| Fig. 3.22 Prototipo de interfaz para la gestión de profesores. | 56 |
| Fig. 3.23 Prototipo responsivo de interfaz para la gestión de profesores. | 56 |

| | | |
|-----------|--|----|
| Fig. 3.24 | Prototipo de interfaz para la gestión de curso. | 60 |
| Fig. 3.25 | Prototipo de interfaz para la gestión de temas..... | 60 |
| Fig. 3.26 | Formulario para almacenar un reactivo..... | 61 |
| Fig. 3.27 | Interfaz del listado de reactivos..... | 61 |
| Fig. 3.28 | Interfaz para la gestión de alumnos..... | 62 |
| Fig. 3.29 | Interfaz para la gestión de grupos. | 62 |
| Fig. 3.30 | Interfaz para la agregar un alumno a un grupo..... | 63 |
| Fig. 3.31 | Interfaz para agregar un alumno a un grupo. | 63 |
| Fig. 3.32 | Interfaz para la gestión de evaluaciones..... | 64 |
| Fig. 3.33 | Interfaz para agregar reactivos al examen..... | 64 |
| Fig. 3.34 | Interfaz del menú para el alumno. | 65 |
| Fig. 4.1 | Interfaz para acceder al formulario de sesión..... | 67 |
| Fig. 4.2 | Interfaz para iniciar sesión. | 67 |
| Fig. 4.3 | Interfaz para iniciar sesión. | 68 |
| Fig. 4.3 | Menú para la gestión de profesores..... | 68 |
| Fig. 4.4 | Formulario para dar de alta un profesor. | 69 |
| Fig. 4.5 | Opciones de exportación de la lista de profesores. | 69 |
| Fig. 4.6 | Interfaz responsiva para la gestión de profesores..... | 70 |
| Fig. 4.7 | Interfaz responsiva para eliminar un docente..... | 71 |
| Fig. 4.8 | Interfaz responsiva que muestra otro atributo de profesor..... | 71 |
| Fig. 4.9 | Interfaz para gestionar cursos y temas. | 72 |
| Fig. 4.10 | Interfaz para dar de alta un curso. | 73 |
| Fig. 4.11 | Interfaz para visualizar los temas de un curso..... | 73 |
| Fig. 4.12 | Interfaz para visualizar la gestión de cursos..... | 74 |

| | | |
|-----------|--|----|
| Fig. 4.13 | Interfaz que visualiza que los campos son requeridos. | 75 |
| Fig. 4.14 | Interfaz que visualiza el formulario con los datos requeridos..... | 75 |
| Fig. 4.15 | Interfaz para editar el profesor asignado. | 76 |
| Fig. 4.16 | Interfaz para consultar los grupos asignados..... | 77 |
| Fig. 4.19 | Lista de alumnos del grupo “1g1a”. | 77 |
| Fig. 4.20 | Interfaz para la gestión de alumnos. | 78 |
| Fig. 4.21 | Interfaz del formulario para un nuevo alumno. | 78 |
| Fig. 4.22 | Edición de la información de un alumno. | 79 |
| Fig. 4.23 | Confirmación eliminar la información de un alumno. | 79 |
| Fig. 4.24 | Opción para ingresar a la gestión de reactivos. | 80 |
| Fig. 4.25 | Formulario para agregar un nuevo reactivo. | 81 |
| Fig. 4.26 | Diálogo para mostrar la imagen del reactivo. | 81 |

Índice de Tablas

| | |
|---|----|
| Tabla 2.1 Comparativa de artículos relacionados con el término <i>E-learning</i> | 17 |
| Tabla 2.2 Continuación de la comparativa de artículos relacionados con el <i>E-learning</i> | 18 |
| Tabla 2.3 Comparativa de artículos relacionados que implementaron una ontología..... | 18 |
| Tabla 2.4 Continuación de la Comparativa de artículos relacionados que implementaron una ontología..... | 19 |
| Tabla 3.1 Lista de usuarios y actividades para la plataforma..... | 22 |
| Tabla 3.2 Descripción del caso de uso “Gestionar Cursos” | 24 |
| Tabla 3.3 Descripción del caso de uso “Gestionar Cursos” | 25 |
| Tabla 3.4 Descripción del caso de uso “Gestionar Grupos”..... | 26 |
| Tabla 3.5 Descripción del caso de uso “Gestionar Alumnos”..... | 27 |
| Tabla 3.6 Descripción del caso de uso “Gestionar Preguntas”..... | 28 |
| Tabla 3.7 Descripción del caso de uso “Gestionar Exámenes”..... | 29 |
| Tabla 3.8 Especificaciones de la ontología ITOTest..... | 30 |
| Tabla 3.9 Requerimientos funcionales y no funcionales de la ontología ITOTest..... | 31 |
| Tabla 3.10 Extracto del glosario de términos..... | 32 |
| Tabla 3.11 Diccionario de conceptos. | 34 |
| Tabla 3.12 Detalles de las relaciones binarias..... | 34 |
| Tabla 3.13 Atributos de instancia de las clases <i>Student</i> e <i>Item</i> | 35 |
| Tabla 3.14 Atributos de clase <i>Quiz</i> | 35 |
| Tabla 3.15 Axiomas Formales..... | 36 |
| Tabla 3.15 Reglas de la ontología. | 36 |
| Tabla 3.16 Narrativa del caso de uso “Registrar Profesor” | 46 |
| Tabla 3.17 Narrativa del caso de uso “Registrar Curso” | 47 |
| Tabla 3.18 Narrativa del caso de uso “Asignar Grupo” | 48 |

| | |
|---|----|
| Tabla 3.19 Narrativa del caso de uso “Registrar Alumno” | 50 |
| Tabla 3.20 Narrativa del caso de uso “Registrar Reactivo” | 51 |
| Tabla 3.21 Narrativa del caso de uso “Registrar Examen” | 52 |
| Tabla 3.21 Narrativa del caso de uso “Agregar Reactivo al Examen” | 53 |

Índice de Listados

| | |
|---|----|
| Listado 3.1 Buscar los datos de los profesores..... | 39 |
| Listado 3.2 Buscar los datos de los cursos | 39 |
| Listado 3.3 Buscar los temas del curso “Course_1” | 40 |
| Listado 3.4 Acceso a la ontología..... | 45 |
| Listado 3.5 Método para buscar a todos los profesores..... | 57 |
| Listado 3.6 Método para eliminar a un profesor de la ontología..... | 58 |
| Listado 3.7 Método para insertar un profesor a la ontología..... | 59 |

Resumen

En la actualidad existen diversos medios electrónicos que proporcionan materiales de apoyo para el aprendizaje a través de contenidos académicos, herramientas de evaluación y actividades de retroalimentación, un ejemplo de ello son plataformas en línea, su objetivo principal consiste en aplicar y difundir el *E-learning* (aprendizaje electrónico), que se considera como una de las estrategias actuales que ayuda a resolver diversos problemas con enfoque educativo como el proceso de evaluación para elementos matemáticos, el cual es tedioso y extenso.

Las plataformas educativas que existen cuentan con una gran variedad de tareas para evaluar y retroalimentar temas, además de proporcionar material didáctico, pero se limitan porque sólo permiten utilizar algunas herramientas básicas. Para elaborar un examen, el docente tiene acceso a elementos estáticos, los cuales no cambian de valor durante su uso, sin embargo se necesita un examen donde se aplique barajeos en preguntas y respuestas que permita implementar filtros de búsqueda en los reactivos para mejorar su construcción y motive al alumno para tener un mayor desempeño al momento de resolver un ejercicio de cualquier disciplina técnico-científica, como matemáticas y otras ciencias exactas.

El Instituto Tecnológico de Orizaba (ITO) utiliza la plataforma *Moodle* para apoyar a los docentes a divulgar material didáctico, proponer actividades y evaluar un tema, pero no contiene la opción de generar cuestionarios dinámicos ni aplicación de filtros adecuados para construir evaluaciones, hasta el momento no se cuenta con otra herramienta que permita estas opciones. Este trabajo presenta el desarrollo de una plataforma *Web* enfocada a la gestión y evaluación de exámenes de las materias técnico-científicas que se imparten en el ITO, dichas evaluaciones se construyen de forma dinámica mediante la aplicación de filtros que el docente determine y se aplican de manera implícita los barajeos en preguntas y respuestas con la finalidad de generar reactivos aleatorios para que el alumno se vea forzado a desarrollar el ejercicio de forma manual y lograr una retroalimentación de los temas vistos en determinado momento. Los resultados de las evaluaciones se almacenan en una ontología especializada que sirve como banco de preguntas y reactivos para que se utilicen posteriormente.

Abstract

Nowadays there are different electronics media to provide support materials for E-learning through academics contents, assessment tools and some activities for feedback as online platforms, its main goal is to implement and disseminate E-learning, that is considered like one strategy that help to resolve different problems with approach educative as process of assessing for mathematical elements.

There are educative platforms with a wide variety of activities to evaluate and feedback to topics, besides providing material, but limited to user because just allow to use some basic tools. When elaborates a quiz, the teacher has access to static elements, however is necessary do a test where applies a shuffle in questions and answers that allow do a search filters to its construction and allow to student have greater dedication at time of resolving an exercise of some technical-scientific discipline as mathematics and other sciences.

The Instituto Tecnológico de Orizaba (ITO) uses the Moodle platform to support the teachers, where they propose activities and is assesses a topic, but does not have the option to generate dynamic questionnaires or application of filters to do assessments, until now there is not other tool that apply these options. This document presents the development of a Web platform focused to management and evaluate of technical-scientific courses that are taught at ITO, assessments are did dynamically through the application of filters that the teacher choose and is applied implicitly a shuffle in questions and answers with the purpose of the student is forced to do the exercise manually so like feedback of others similar topics. The results of the assessments are stored in a specialized ontology that is used as a bank of questions and answers for use later.

Introducción

El uso de plataformas en línea o entornos virtuales de enseñanza como también se les conoce simplifican el trabajo para la mayoría de los docentes que se apoyan en estas herramientas porque permiten proveer conocimiento sobre temas educativos, además de facilitar el proceso de enseñanza y retroalimentación del conocimiento, sin olvidar que los elementos multimedia como los videos e imágenes también son de gran apoyo para proporcionar información al usuario. La mayoría de estas plataformas se adquieren por un precio elevado y las que cuentan con licencia libre se limitan en sus módulos o requieren un costo adicional para adquirir nuevos elementos.

La evaluación de las materias técnico-científicas que se imparten en el Instituto Tecnológico de Orizaba requiere de un tiempo prolongado porque consiste en aplicar técnicas matemáticas a las áreas de ingeniería que involucran problemas de optimización, aproximación, variación numérica y procedimientos de rendimiento. Además se resalta que en la mayoría de los ejercicios poseen como solución una respuesta numérica exacta, por lo cual no hay variación alguna con otro resultado. Por lo tanto el trabajo de tesis presenta el desarrollo de una plataforma *Web* que gestione a profesores, alumnos y grupos de las materias técnico-científicas, además de generar cuestionarios dinámicos que permita de manera implícita el barajeo en preguntas y respuestas, incluyendo las opciones de un filtrado especial para que el docente cuente con una opción adicional al momento de elaborar una evaluación y así calificar los temas vistos en clase. La información gestionada en la plataforma se almacena en una ontología especializada además de que también funciona como un banco de preguntas y respuestas.

El presente trabajo se detalla a continuación: en el capítulo I se muestran los objetivos y justificación, así como los conceptos básicos que ayudan a comprender mejor la tesis, en el capítulo II se presentan los trabajos relacionados sobre el uso de ontologías y herramientas para el *E-learning*, así como un análisis comparativo de las investigaciones, el capítulo III describe la metodología utilizada para el desarrollo del trabajo de tesis, en el capítulo IV se muestra el caso de estudio de la plataforma *Web* y por último el capítulo V muestra las conclusiones y trabajos a futuro.

Capítulo 1. Antecedentes

En este capítulo se explican los conceptos básicos relacionados con el tema de tesis, el planteamiento del problema, se describe el objetivo general y los objetivos específicos, terminando con la justificación del mismo.

1.1 Marco teórico

En esta sección se mencionan los conceptos básicos que comprenden el tema de tesis.

1.1.1 Lenguaje Web

En [1] se define como lenguaje de programación aquella estructura con una base sintáctica y semántica que se diseña para realizar procesos computacionales y posee determinadas instrucciones para su implementación. Por otra parte la programación es el procedimiento de escritura del código fuente de un *Software*, de esta manera se indica al programa informático qué acción se lleva a cabo y como concretarla.

La *World Wide Web* (WWW) o simplemente *Web* se define como un espacio de información donde los elementos de interés denominados recursos se identifican a través de URI's (*Uniform Resource Identifier*, Identificador de recursos uniforme), con esta definición se afirma que un lenguaje de programación *Web* se estructura con una base sintáctica y semántica que permite indicar instrucciones a un programa en la *Web*.

1.1.2 Framework

En [2] se menciona que el concepto de *Framework* (marco de trabajo) se emplea en muchos ámbitos del desarrollo de sistemas, incluido el de aplicaciones *Web*. En general, éste término se refiere a una estructura de *Software* compuesta de elementos personalizables e intercambiables para el desarrollo de aplicaciones. En otras palabras, se considera como una estructura genérica e incompleta para configurar y añadir otros componentes para construir una aplicación concreta.

Además en [2] se menciona que al utilizar los *Frameworks*, estos permiten acelerar el proceso de desarrollo, reutilizar código que ya existe y promover buenas prácticas de desarrollo como el uso de patrones de diseño. Por lo tanto se considera como un conjunto de elementos (por ejemplo clases en *Java* y archivos de configuración en *XML*) que permiten construir un diseño reutilizable para facilitar y agilizar el desarrollo de sistemas *Web*.

1.1.3 Ontología

En 1993, Thomas Gruber definió a una ontología como una especificación explícita de una conceptualización, en otras palabras se considera como una visión abstracta y simplificada de la realidad (objetos y sus relaciones) que se desea representar con un propósito particular, donde cada base, sistema o agente del conocimiento se comprende con alguna conceptualización de forma explícita o implícita. Este último término se define como una estructura compuesta de un dominio D y un conjunto de relaciones sobre D denominado rango R , por lo que las ontologías computacionales son un medio para modelar la estructura formal de un sistema [3].

Para el contexto de los sistemas de información, las ontologías se utilizan en el campo de la Inteligencia Artificial para describir con detalle la semántica de un dominio en particular, como un área de conocimiento (física, medicina, etc.) o de interés específico a partir de los conceptos y relaciones entre ellos. Además en [3] se menciona que las ontologías se componen de clases, conceptos, instancias y atributos o propiedades, donde una instancia se considera como un caso concreto de una clase y ésta define a un concepto.

En [4] y [5] se menciona que las ontologías se elaboran por los siguientes elementos que sirven para representar el conocimiento de algún dominio en específico.

1. **Clases:** Describen los conceptos del dominio y se organizan en taxonomías y por lo general se les aplican mecanismos de herencia.

2. **Relaciones:** Representan las interacciones entre los conceptos del dominio, contienen relaciones binarias. El primer elemento de la relación se le conoce como dominio y el segundo el rango.
3. **Funciones:** Representan un tipo concreto de relación, donde se identifica un elemento mediante el cálculo a partir de otros elementos que se encuentren en la ontología.
4. **Instancias:** Representan objetos determinados de un concepto.
5. **Taxonomía:** Conjunto de conceptos organizados jerárquicamente.
6. **Axiomas:** Modelan sentencias que siempre se consideran ciertas y permiten, junto con la herencia, inferir conocimiento que no esté indicado explícitamente en la taxonomía de conceptos.
7. **Propiedades (slots):** Representan características o atributos que describen a los conceptos.

Por lo tanto, las ontologías se forman de un conjunto de conceptos y relaciones que están ordenados en jerarquías con propiedades asociadas.

1.1.4 Methontology

En [6] se menciona que la metodología *Methontology* permite el desarrollo de ontologías que provee actividades y guías de especificación, así como un ciclo de vida para la misma.

Los principales pasos de *Methontology* son:

- **Especificación:** Representa la meta, alcance y granularidad de la ontología.
- **Conceptualización:** Ayuda a organizar y estructurar el conocimiento adquirido (modelo conceptual) mediante el uso de representaciones independientes de los lenguajes de implementación.
- **Implementación:** Consiste en formalizar y construir el modelo conceptual mediante lenguajes formales.
- **Evaluación:** Considera la verificación y validación de la ontología, por ejemplo validar el modelo del dominio en específico.

Además en [7] se menciona que esta metodología permite construir ontologías a nivel de conocimiento y sus actividades se identifican en el proceso de desarrollo de *Software* propuesto por la organización IEEE y otras metodologías de ingeniería de conocimiento.

1.1.5 Sistema de gestión de contenido

En [8] se menciona que un sistema de gestión de contenido (*CMS, Content Management System*) permite la administración de elementos electrónicos que se utilizan en aplicaciones *Web*, estos sistemas cuentan con una interfaz para la creación, gestión, publicación y presentación de contenidos a través de formularios para acceder a los elementos alojados en el sitio mediante un visualizador *Web*.

Los gestores de contenidos que controlan una o varias bases de datos permiten crear una serie de roles como administrador, editor o participante sin necesidad de un nuevo formato para los contenidos que se vayan añadiendo, estos gestores permiten facilitar y generar contenidos pre-determinados.

1.1.6 Quiz (Evaluación)

En [9] se menciona que la plataforma *Moodle* utiliza el término *Quiz* para referirse a una prueba sencilla que evalúe un contenido, tema o avance de alguna lectura a integrantes de un grupo de estudio o aula de clase. A diferencia de un examen, la evaluación se caracteriza por ser menos formal y la asignación del puntaje se da inmediatamente, que se utiliza para reforzar conocimiento porque en él, se incluyen temas y objetivos de aprendizaje.

Las técnicas para elaborar un *Quiz* varían de acuerdo al objetivo con el que se desarrolla, porque al elaborar una pregunta de un tema en particular, el resultado especifica un tipo de conocimiento previamente enseñado, mientras que un *Quiz* elaborado al azar sólo se muestra el interés por presentar dicha evaluación.

1.1.7 MCQ (Evaluación de Opción Múltiple)

En [10] se menciona que la Evaluación de Opción Múltiple (MCQ, *Multiple Choice Question*.) se considera la técnica más utilizada en la evaluación de conocimiento dónde se plantea una pregunta o problema, denominado reactivo, que consta de un enunciado y una serie de respuestas llamadas opciones, entre ellas se encuentra la solución correcta y otras incorrectas, conocidas como distractores.

Además, se recomienda que cada pregunta cuente con tres opciones de respuesta pero no siempre se utilizan todas las buenas prácticas pedagógicas para generar este tipo de exámenes de forma sistematizada, por ejemplo que se muestren las opciones del reactivo de forma aleatoria. En [11] se menciona que el uso apropiado de distractores u opciones incorrectas, se describe como una importante faceta en la elaboración y análisis de los reactivos de opción múltiple ya que requieren mayor esfuerzo por parte de los elaboradores de *ítems*, estos elementos determinan la calidad de las preguntas para los sustentantes a lo largo del proceso de evaluación.

Durante la elaboración de reactivos se identifican a los distractores de una evaluación porque su comportamiento permite mejorar los estándares en los exámenes, especialmente en aquellos de largo procedimiento, ya que reducen el periodo para responderlos y permiten ampliar los contenidos a evaluar, igualmente se considera relevante para cuestiones administrativas como el tiempo de construcción de los recursos y/o materiales.

1.1.8 Shuffle

En [12] se menciona que *Shuffle* (barajeo) es una técnica, la cual, permite aplicar un orden aleatorio en preguntas y respuestas. Al crear evaluaciones al azar, se toma en cuenta que se construyan de forma diferente para cada usuario y el orden de presentación para los reactivos se muestre de manera distinta en cada *Quiz* elaborado, en caso de la plataforma *Moodle*, esta opción está limitada.

Además en [11] se desarrolló un algoritmo computacional que permite al docente crear un examen a través de una configuración, la cual cuenta con los algoritmos *Shuffle Option* y *Shuffle Question*, es decir que preguntas y respuestas se barajan para generar la evaluación.

1.1.9 E-learning

En [13] se menciona que *E-learning* (aprendizaje electrónico) engloba a las aplicaciones y servicios que se orientan a facilitar el proceso de enseñanza-aprendizaje y que además utilizan las Tecnologías de la Información y la Comunicación (TIC).

La implementación del *E-learning* en la educación va más allá de la utilización de las TIC para el proceso evaluativo, porque su propósito se refleja en la interacción y continuidad de los principios pedagógicos, los cuales proponen un modelo donde el alumno toma una mayor responsabilidad en su aprovechamiento, contribuyendo al desarrollo de la eficiencia en el proceso de enseñanza-aprendizaje para la mejora cualitativa del modelo educativo. Además se centra en las concepciones, intenciones y decisiones del profesor para obtener recursos y así ofertar cursos y servicios.

1.1.10 Ingeniería Web

En [14] se menciona que la Ingeniería Web (IWeb) aplica los principios científicos de ingeniería y de administración con enfoques disciplinados y sistemáticos para el desarrollo, despliegue y mantenimiento exitoso de sistemas y aplicaciones basados en *Web*.

La evolución y crecimiento continuo de las aplicaciones *Web* lleva a un proceso incremental y evolutivo que permite que al usuario involucrarse activamente, facilitando el desarrollo del producto final. Los métodos de la Ingeniería *Web* definen las etapas y actividades necesarias para llevar a cabo la construcción completa de una aplicación *Web*. El principio básico para desarrollar una aplicación *Web* parte de una descripción en forma de un esquema conceptual que se transforma en un *Software* mediante un conjunto de componentes para la *Web* como diagramas, esquemas, herramientas de diseño y para código fuente.

1.1.11 SPARQL

En [15] se menciona que RDF (*Resource Description Framework*) se considera como un modelo estándar para el intercambio de datos en la *Web* provee un modelo general para definir tripletas, independientemente del lenguaje (turtle, N3, RDF/XML, RDFa, etc.).

Por lo tanto en [16] menciona que el lenguaje de consultas SPARQL (*Protocol and RDF Query Language*) está diseñado para elaborar consultas sobre datos representados para RDF el cual contiene funciones de consultas requeridas y opcionales para ser utilizadas al momento de expresar los resultados a través de diversas fuentes de datos y forma parte de las recomendaciones de la W3C, Además al utilizar este lenguaje semejante a *SQL* permite trabajar con valores estructurados y semiestructurados, incluyendo la exploración los datos consultando relaciones desconocidas de diversas fuentes.

1.2 Planteamiento del problema

En las instituciones de nivel medio superior y superior se imparten asignaturas técnico-científicas como matemáticas, física o química, en el Instituto Tecnológico de Orizaba se ofertan ingenierías que contemplan materias que estudian el conocimiento, manejo y dominio de las matemáticas y otras ciencias, las cuales se aplican para el desarrollo de tecnologías, así como para el manejo eficiente de recursos en beneficio de la sociedad.

Si bien, estas asignaturas al no ser fácil de evaluar, tampoco se elabora de manera sencilla un ejercicio que garantice el conocimiento adquirido de un tema en específico. Sin embargo, el Instituto Tecnológico de Orizaba no cuenta con una plataforma apropiada con un repositorio de preguntas para evaluar materias de este tipo, donde se generen distractores en preguntas y respuestas, también de no contar con una herramienta para evaluar temas específicos, el docente lo cubre con alguna tarea o trabajo, pero no se llega a calificar mediante un ejercicio práctico.

1.3 Objetivos

En esta sección se menciona el objetivo general y a continuación los objetivos específicos del proyecto de tesis.

General

Desarrollar una plataforma *Web* que utilice una ontología que contenga filtros adecuados para la generación de reactivos, así como aplicar barajeos en preguntas y respuestas que permitan generar versiones aleatorias de un examen.

Específicos

- Analizar el estado del arte de los trabajos relacionados.
- Estudiar, revisar y comparar herramientas similares al tema propuesto.
- Seleccionar la ontología que sustente el desarrollo de la plataforma.
- Aplicar algoritmos *Shuffle Question* y *Shuffle Option*.
- Aplicar filtros adecuados para la generación de reactivos (por orden de utilización, por docente, los más o menos utilizados y por nivel de complejidad).
- Aplicar una metodología para el desarrollo del tema de tesis.
- Determinar el caso de estudio.

1.4 Justificación

Hoy en día las grandes editoriales en el mundo ofrecen diversos apoyos electrónicos y multimedia junto con los libros de texto para facilitar el aprendizaje de los contenidos académicos. Dentro de estas herramientas de apoyo se encuentran los llamados *Companion Websites*. En [17], [18], [19] el *E-learning* presenta estrategias formativas que resuelven muchos de los problemas educativos como la evaluación de cuestionarios, soporte a actividades adicionales y proveer material didáctico a los alumnos. El profesor que adopta un libro de texto para su curso, tiene

por lo general acceso a bancos de reactivos estáticos (la respuesta correcta para un reactivo dado siempre es la misma), por lo que la introducción de elementos aleatorios en la generación de cuestionarios, se considera una característica valorada por sus autores porque toma mayor valor para el caso de las disciplinas técnico-científicas como las matemáticas, física o química. En estas asignaturas se permiten generar cuestionarios dinámicos con un enunciado que permiten al estudiante ejercitar sus habilidades.

Existen en [20] y [21] algunos cuestionarios dinámicos que bien, se limitan en contenido o representan un gasto adicional para los alumnos al momento de adquirirlos. Además, muchos de estos apoyos están en inglés, lo cual resulta contraproducente, sobre todo para los alumnos de bachillerato o de nivel profesional donde existen materias que son del área técnico-científica. Por eso se desarrolló una plataforma *Web* en español que permita la generación y evaluación de exámenes en línea propia del Instituto Tecnológico de Orizaba. Este producto va a crecer en contenido si los profesores lo retroalimentan de ejercicios y de compartirse entre otras instituciones educativas de nivel medio y superior.

Por lo cual, se desarrolló una plataforma *Web* que gestione alumnos, profesores y grupos además de realizar la generación de exámenes y que evalúe en línea dichos cuestionarios. En [22], [23] y [24], se enfoca a asignaturas que se adapten a este tipo de evaluación como: Cálculo Diferencial, Cálculo Integral, Análisis Numérico, Probabilidad y Estadística, por citar algunas impartidas en las carreras ofertadas en el Instituto Tecnológico de Orizaba, con el fin de reforzar sus conocimientos y dar otra posibilidad de evaluar estas disciplinas. La plataforma *Web* se elaboró dinámicamente porque los reactivos y parámetros de entrada que se generan de manera aleatoria (dentro de ciertos rangos previamente establecidos), de modo que la respuesta correcta siempre estará en diferente orden en cada una de las versiones de examen. La plataforma evaluará cada reactivo mediante un algoritmo *ad hoc*, como los que utiliza *Wiris* en [24], [25] y que los resultados se almacenen en una ontología.

Capítulo 2. Estado de la práctica

En este capítulo se da a conocer el estado de la práctica del tema de tesis, el cual menciona los trabajos relacionados que desarrollan sistemas informáticos que facilitan el proceso de evaluación de exámenes y reactivos utilizando estrategias y herramientas para su desarrollo porque consideran que el *E-learning* permite aumentar las capacidades del alumno, así como trabajos donde implementan ontologías para el diseño del dominio de *E-learning*.

2.1 Trabajos relacionados

En [10] se desarrolló una aplicación *Web* con el *Framework JavaServer Faces (JSF)* para el lenguaje de programación *Java* que permite crear exámenes mediante una configuración previa que consiste en barajar las preguntas y opciones de respuesta, así como elegir aleatoriamente un conjunto de elementos para generar una evaluación (*Quiz*), la principal diferencia de la aplicación desarrollada con respecto a otras herramientas similares, radicó que al momento del barajeo en las respuestas se siguió un criterio de orden, la cual se generó por una sugerencia pedagógica. Dicha aplicación utilizó una ontología especializada para el dominio de *E-learning*, que funcionó como un banco de reactivos para lograr construir cuestionarios dinámicos en español, esta aplicación le pertenece al Instituto Tecnológico de Orizaba (ITO) y se enfoca al tema de Aplicación de Derivadas para la materia de Cálculo Diferencial que se imparte en todas las carreras ofertadas en el ITO con el propósito de que los alumnos del Tecnológico la utilicen con el fin de reforzar sus conocimientos en Cálculo Diferencial demostrando su habilidad para interpretar y resolver problemas específicos de esta disciplina.

En [26] se presentó el desarrollo de una plataforma de evaluación electrónica para ser aplicada en la universidad de Ahmadu Bello, Zaria, Nigeria, esta herramienta se utilizó como un campo de pruebas en 2008. Actualmente el uso de *Internet* se considera un medio fiable para aplicar evaluaciones a estudiantes mediante ésta tecnología. Por lo que se diseñó y desarrollo una aplicación *Web*, la cual proponía diversas pruebas en múltiples formatos de elección para su solución y se logró calificar de manera inmediata.

La aplicación *Web* desarrollada se basó en la tecnología de *Microsoft* como lenguaje de programación se desarrolló con la tecnología *ASP.NET* e implementó *C#*, para interactuar con la Base de Datos (BD) relacional se utilizó *ADO.NET* y para la gestión de los datos se ocupó *Microsoft SQL Server*. Este trabajo empleó *Visual Studio Express* como entorno de desarrollo para la aplicación y a “*Internet Explorer*” como visualizador *Web*. Además se puso a prueba en una *Intranet* para garantizar la conectividad y resolver el problema de la evaluación en las instituciones de enseñanza superior en Nigeria.

En [24] se presentó el diseño de *Numbas 2004*, compatible con el estándar *SCORM* (*Sharable Content Object Reference Model*, Modelo de Referencia de Objetos de Contenido Compartible), un nuevo código abierto compatible para un sistema de evaluación multi-plataforma para el aprendizaje electrónico, desarrollado y utilizado en la Universidad de Newcastle, Inglaterra. El enfoque principal de la aplicación contemplaba una amplia variedad de exámenes para diversas materias como matemáticas y estadística, su funcionalidad y diseño puso en juego la plena capacidad y recursos de *Internet*. Se utilizó para numerosas disciplinas en materia de educación y formación, aprovechó y empleó modelos de desarrollo en la educación superior. *Numbas* se consideró como parte de los sistemas de evaluación de código abierto de gran alcance dirigida a disciplinas del área técnica. Además, el estándar *SCORM* facilita su implementación, porque mediante la documentación disponible en www.scorm.com, se describe como un sistema de gestión de aprendizaje (*Learning Management System*, *LMS*). El acceso al sistema *Numbas* se encarga de la administración de usuarios, sesiones y almacenamiento de datos de seguimiento e informes, de modo que se ahorra tiempo.

En [27] se indicó que el *E-learning* (aprendizaje electrónico) se considera un elemento básico para complementar con éxito cualquier acción formativa. Este artículo presenta un *Software* genérico, flexible, interoperable y reutilizable para las evaluaciones de diseño y su aplicación en el Sistema de Gestión de Aprendizaje. Este sistema se basó en el lenguaje de esquemas *XML Schema* para los documentos *XML* y se definen entidades básicas como lo son: alumnos, profesores, contenidos y evaluaciones, este último término es el grado de asimilación de conocimientos y habilidades logrado por el alumno. El objetivo principal de este *Software* define que el

diseño de exámenes se construya de manera sencilla y eficiente. Al implementarse adecuadamente se logró reducir el tiempo necesario para configurar las actividades de evaluación.

En [28] se mencionaron algunas áreas que utilizan una ontología como herramienta para generar, organizar y personalizar algún dominio. En contexto, la noción de ontología se implementa para diferentes propósitos, uno de ellos se utiliza como un medio para proporcionar una estructura y diseño en los contenidos de *E-learning*, principalmente de las evaluaciones. A pesar del esfuerzo realizado en este sentido, aún queda trabajo por hacer ya que la mayoría de los enfoques proponen ontologías ligeras que modelan una parte del dominio de la evaluación sin tener en cuenta los aspectos pedagógicos. Este artículo presentó una ontología que conceptualiza el *E-learning* e incluye un conjunto de reglas que representan los principios pedagógicos a seguir.

En [29] se mencionó que los Sistemas Tutoriales Inteligentes (STI), se clasifican como sistemas de *E-learning* diseñados para apoyar y mejorar el proceso de aprendizaje-enseñanza, en particular, los sistemas de *E-learning* ofrecen entornos de creación de elementos de evaluación para los usuarios que se enfocan a impartir ese dominio, de tal forma que se representan mediante diferentes especificaciones, principalmente proponen plantillas para la generación dinámica de preguntas para una prueba formalizada. Además, se menciona que la ontología implementada se describe como un lenguaje que forma parte del dominio donde aplica el conocimiento en una evaluación.

En [30] se comentaron los temas de fiabilidad en las evaluaciones electrónicas donde las preguntas marcadas se seleccionan al azar a partir de un banco de *Items*. En Reino Unido, el sector de la educación superior utilizó las evaluaciones para fines médicos, pero los elementos seleccionados no lograron la atención por parte de los estudiantes, profesores e instituciones. Los *Items* se basaron en el análisis de pruebas para que se adapten posteriormente a una computadora, las evaluaciones permitieron la utilización de reactivos almacenados y propuso al método *OSIRIS* (*objective standardisation in random item selection*, estandarización objetiva para

selección al azar de elementos), para mejorar las calificaciones del estudiante, además se basaron en el nivel de dificultad de las preguntas que elaboraron y se demostró con datos que tuvo gran uso para la evaluación en línea. Como resultado, el método propuesto aplicó las mejores prácticas para garantizar la calidad en el banco de *Items*, el cual seleccionó un *Item* al azar y aumentó la imparcialidad y fiabilidad de las evaluaciones en lugar de ser vistas como un riesgo.

En [31] se utilizó “*Linked Open Data*” para la generación de elementos de evaluación en el sector educativo que describe un medio para crear variables y probar modelos de elección simple utilizando el estándar IMS-QTI (*IMS-Question Test Interoperability Especificación*), el cual permite el intercambio de elementos de prueba y resultados, así como banco de reactivos y herramientas de diseño entre los sistemas de aprendizaje. Se implementó para apoyar la generación de preguntas en una evaluación formativa, en particular, para los problemas de calidad e identificación de sub-gráficos. Los exámenes generados se importaron en una plataforma de evaluación. Se probaron cinco modelos, los cuales permitieron la identificación de los principales retos para mejorar la fuente de datos.

En [32] se comentó que la plataforma *Moodle*, se enfoca a la educación virtual y posibilita la evaluación de alumnos a través de cuestionarios. Sin embargo, en la creación de reactivos se encuentran las siguientes dificultades: al ingresar un número considerable de preguntas tediosas se logra incurrir en errores de digitalización porque no posee un editor de *LaTeX* que se considera como un sistema de composición de texto plano, sólo generan preguntas aleatorias previamente descritas en texto. Por ello *Scientific WorkPlace* ofrece ventajas respecto a la creación de preguntas, brindando la posibilidad de crear una plantilla en la cual se generan cuestionarios específicos pero no logran ser compatibles con los formatos admitidos por *Moodle*. Este trabajo propuso un algoritmo para convertir evaluaciones realizadas en *Scientific WorkPlace* a formato *GIFT* (*General Import Format Technology*, Tecnología de formato general de importación) para *Moodle*, además se presentó un prototipo llamado *tex2Gift* para validar el algoritmo propuesto y finalmente se planteó un protocolo que generó preguntas aleatorias para *Moodle* utilizando *GIFT*.

En [33] se indicó que programar sistemas informáticos se considera un proceso difícil para estudiantes que empiezan a desarrollar por lo que se convierte en un desafío para los docentes. Este artículo presentó una plataforma de evaluación sistematizada que apoya el proceso de enseñanza-aprendizaje en cursos introductorios de programación de computadoras para estudiantes de ingeniería en la Universidad de Concepción, Concepción, Chile. Esta plataforma utiliza mecanismos que combinan análisis estático-dinámico aplicando métodos para la comprensión y análisis en línea, permitiendo una retroalimentación personalizada para los alumnos. Este trabajo presentó los criterios usados para crear secuencias didácticas de problemas y la forma de aplicarlas mediante la evaluación sistematizada. Los resultados obtenidos después de seis años de aplicación mostraron un efecto positivo de motivación, desempeño y autoeficacia por parte de los alumnos. La metodología utilizada para estudiar la efectividad de la herramienta incluye un análisis cualitativo y cuantitativo.

En [34] se propuso que las ontologías representan una gran oportunidad teniendo grandes ventajas para los sistemas de *E-learning*. Su aplicación se ve como una mejor solución para organizar y visualizar el conocimiento didáctico, se reutilizo por diferentes aplicaciones educativas. Este trabajo tuvo como objetivo proponer un modelo que se centra en la integración de los principios ontológicos con los estándares de *E-learning*, se elaboró un prototipo que se integró con una ontología para representar la semántica, también añadió anotaciones para cada recurso y previamente se definió el vocabulario necesario con el propósito de que los estudiantes conceptualizarán el módulo de aprendizaje. Además se implementó otra ontología, la cual proporcionó materiales didácticos que se encuentra en los metadatos del sistema. Aquí también se incluyeron las opciones de acceso al sistema y los resultados de registro.

En [35] se evaluó a *DQGen*, una herramienta que genera preguntas de opción múltiple para poner a prueba la comprensión de un niño mientras hace la lectura de un texto dado. A diferencia de los métodos tradicionales, genera diferentes tipos de distractores diseñados para diagnosticar la insuficiencia de comprensión en pruebas, no para una sola sentencia sino que evaluó la calidad de las preguntas generales y distractores individuales. Los resultados, errores y los comentarios de 8 jueces revelaron una sugerencia para la forma de abordar dicho diagnóstico.

En [36] se presentó una ontología para ayudar a los sistemas de aprendizaje que contienen las relaciones entre puntos de conocimiento y materiales de los mismos. Se aplicó la construcción de una ontología centrada en un curso de gramática japonesa existente. Por otra parte, el sistema de idiomas personalizable, construido para manipular la ontología, se centró en proporcionar una interfaz para la disposición de los objetos de aprendizaje que muestra la representación visual de puntos de conocimiento y sus relaciones. La intención del desarrollo del sistema animó a los instructores para orientar sus materiales de enseñanza a puntos de conocimiento específicos. Con estas orientaciones, el sistema proporcionó un entorno, en el que los alumnos distinguieron fácilmente el tipo de contenido. Por último, sobre la base del resultado de una evaluación preliminar, se llevó a cabo un estudio para explorar el impacto de los estilos de aprendizaje y los hábitos en el rendimiento del mismo para evaluar aún más el sistema de apoyo que se basó en la ontología.

En [37] se mencionó que la evaluación en los entornos virtuales comparte algunas estrategias realizadas con las del modo presencial pero no se les asigna el mismo valor. Aunque el propósito es similar, el proceso cuenta con variantes importantes tanto en el tiempo de elaboración, la calidad de los reactivos entre otros aspectos. Para lograr la sistematización se requiere de técnicas e instrumentos que permitan obtener información antes, durante y después del proceso de enseñanza-aprendizaje en la conducta y habilidades del estudiante. Es por eso que la ontología propuesta permitió la especificación del lenguaje para aspectos de evaluaciones virtuales para el dominio de *E-learning* y su desarrollo se llevó a cabo con el editor de ontologías *Protégé* basado en el lenguaje de ontologías para el proceso evaluativo en la educación superior.

En [38] se menciona que en la actualidad hay un gran número de ontologías que se utilizan para varios dominios, incluyendo el de la salud (genética, proteínas, componentes celulares, anatomía y enfermedades específicas, entre otros). Sin embargo, en algunos casos, la definición y la estructura de estas ontologías se consideran suficientes para utilizarse en dominios concretos. Este artículo presenta el diseño de un conjunto de ontologías de asistencia médica para su utilización directa en el diagnóstico de los sistemas de apoyo a la toma de decisiones el cual sugieren una

solución para estructurar pequeños dominios que forman parte de uno en general y la alternativa se basó en la arquitectura modular donde la ontología principal (raíz) se creó para definir las principales relaciones que se aplican a un conjunto de sub-ontologías para cada sub-dominio.

2.2 Análisis comparativo

A continuación se muestra la comparación de los trabajos presentados, la Tabla 2.1 menciona los artículos relacionales con el término *E-Learning*, la Tabla 2.2 muestra la continuación de dichos artículos, la Tabla 2.3 muestra a aquellos trabajos que implementaron una ontología para resolver la problemática del *E-Learning* y la Tabla 2.4 la continuación de la misma.

Tabla 2.1 Comparativa de artículos relacionados con el término *E-learning*

| Artículo | Herramientas | Propuesta | Estado | Año |
|--|---------------|--|---------------|------|
| <i>Development of an E-Assessment Platform for Nigerian Universities</i> [26] | .NET | Evaluación de múltiples formatos de elección y se calificarse de manera inmediata | Terminado | 2010 |
| <i>A completely client-side approach to e-assessment and e-learning of mathematics and statistics</i> [39] | <i>Numbas</i> | Ampliar la variedad de evaluación electrónica y aprendizaje, entre matemáticas y estadística. | No específica | 2012 |
| <i>Assessment Design: A Step Towards Interoperability</i> [27] | XML | Hacer que el diseño de evaluación permita la interoperabilidad entre los sistemas de una manera sencilla y eficiente. | Terminado | 2011 |
| <i>In search of Osiris random item selection fairness and defensibility in high stakes e-assessment</i> [30] | No específica | Usar un método para modificar las calificaciones del estudiante basado en la selección al azar a partir de un banco de ítems, ara generar preguntas. | Terminado | 2011 |

Tabla 2.2 Continuación de la comparativa de artículos relacionados con el *E-learning*

| Artículo | Herramientas | Propuesta | Estado | Año |
|---|----------------------------|---|-----------|------|
| <i>Generating educational assesment items from Linked Open Data the case of DBPEdia [31]</i> | DBPEdia | Mejorar los problemas de calidad de datos y la identificación de sub-gráficos. | Prototipo | 2011 |
| <i>Generation of Random Questions with Multiple Choice Single Answer for Moodle. [32]</i> | Scientific-Workplace, | Crear una plantilla de preguntas, de la cual se generan cuestionarios específicos usando <i>ScientificWorkplace</i> , | Prototipo | 2011 |
| Una plataforma de evaluación automática con una metodología efectiva para la enseñanza / aprendizaje en programación de computadoras [33] | Sistema Web en PHP y MySQL | Combinar un análisis estático/dinámico y aplicar una evaluación de comprensión/análisis en línea, permitiendo una retroalimentación personalizada a los alumnos | Terminado | 2011 |
| <i>Generating diagnostic multiple choice CLOZE question comprehension [35]</i> | No especifica | Generar diferentes tipos de distractores diseñados para diagnosticar diferente tipos de insuficiencia de comprensión y pruebas. | Terminado | 2012 |

Tabla 2.3 Comparativa de artículos relacionados que implementaron una ontología

| Artículo | Herramientas | Propuesta | Estado | Año |
|---|--------------|---|-----------|------|
| Entorno Web para la generación dinámica de reactivos en la evaluación de aplicación de derivadas [10] | JSF | Desarrollar de una aplicación Web para el Instituto Tecnológico de Orizaba para la evaluación de la asignatura de Cálculo Diferencial | Terminado | 2014 |

Tabla 2.4 Continuación de la Comparativa de artículos relacionados que implementaron una ontología

| Artículo | Herramientas | Propuesta | Estado | Año |
|---|-----------------|---|-----------|------|
| <i>EAOnto A Heavyweight Ontology for Supporting eAssessment Generation</i> [28] | <i>EAOnto</i> , | Presentar la ontología <i>EAOnto</i> , que conceptualiza el dominio de evaluación que incluye un conjunto de reglas que representan los principios pedagógicos. | Prototipo | 2012 |
| <i>Dynamic test generation over ontology-based knowledge representation in authoring shell</i> [29] | No especifica | Usar Ontologías <i>Web</i> (OWL) para realizar pruebas del conocimiento para lidiar con el problema de la generación de ensayos y presentación de las pruebas. | Prototipo | 2009 |
| <i>On Integration of Ontologies into E-Learning Systems</i> [34] | No especifica | Buscar una mejor solución para organizar y visualizar el conocimiento didáctico, y por este medio, ser compartida y reutilizada por diferentes aplicaciones educativas. | Prototipo | 2014 |
| <i>A language learning support system using course-centered ontology and its evaluation</i> [36] | No especifica | Proporcionar un entorno en el que los alumnos distinguieron fácilmente entre puntos de conocimiento relacionadas. | Terminado | 2014 |
| Ontología para el proceso evaluativo en la educación superior [37] | <i>Protégé</i> | Proponer una ontología para llevar a cabo el proceso de evaluación para la educación superior | Terminado | 2014 |
| <i>Towards an Ontology to Support Semantics Enabled Diagnostic Decision Support Systems</i> [38] | No especifica | Proponer una estructura que facilite la visualización de las relaciones de una ontología y sus sub-ontología. | Terminado | 2012 |

En las Tablas 2.1, 2.2, 2.3 y 2.4 se mostraron las comparativas de los trabajos relacionados donde indican que el proceso de evaluación es importante, sin embargo algunos proponen el uso de ontologías para ser utilizadas como bancos de reactivos, de los cuales se obtiene información,

en otros casos se mencionan que al utilizarlas muestran mejor diseño y estructura para visualizar de manera sencilla el dominio para el cual se crearon los sistemas. En otros casos, las estrategias implementadas mejoraron el aprendizaje electrónico y la retroalimentación, algunos artículos se basan en el uso de sistemas *Web* para su aplicación.

2.3 Propuesta de solución

Se desarrolló una plataforma *Web* para la gestión de profesores, alumnos, grupos y la generación dinámica de exámenes para las asignaturas técnico-científicas implementando los algoritmos *Shuffle Option/Question* propuestos en [10], además de elaborar las opciones de filtrado múltiple para que el docente cuente con la opción de elegir reactivos de otro profesor o seleccionar un *top* de preguntas utilizadas con mayor o menor frecuencia durante la aplicación de otros exámenes previamente creados, además de asignar un puntaje a los reactivos para determinar el nivel de dificultad del mismo con el propósito de que al momento de elaborar un examen, su contenido se forme de diversas fuentes y se consideren otros puntos de vista para elaborar un reactivo y alimentar el banco de reactivos. Todo esto se codifica con el entorno de desarrollo *NetBeans* y para el lenguaje de programación se utilizó el marco trabajo *JavaServer Faces* para facilitar y agilizar el desarrollo de la plataforma. También para la función de banco de reactivos se elaboró una ontología especializada para el dominio de *E-Learning* diseñada con la metodología de ontologías *Methodology* y se construyó en *prótege*, contemplando que para las consultas requeridas a la ontología se aplicó el lenguaje SPARQL (*Protocol and RDF Query Language*) cuyo diseño permite para elaborar consultas sobre datos representados para RDF (*Resource Description Framework*), el cual contiene funciones de consultas requeridas y opcionales para que se utilicen al momento de expresar los resultados. El proyecto de tesis se elaboró con la metodología de desarrollo de aplicaciones *Web IWeb* (Ingeniería *Web*), esta propuesta de solución complementa el trabajo descrito en [10], el cual se desarrolló por un alumno egresado de la Maestría en Sistemas Computacionales.

2.4 Justificación de la solución seleccionada

Para el proyecto de tesis se ocupó el entorno de desarrollo *NetBeans* porque este IDE cuenta con asistentes para la creación y configuración de distintos proyectos, incluida la elección de algunos *Frameworks* además ayuda a comprobar código fuente de manera sintáctica y semántica, facilita el uso de plantillas y herramientas de refactorización, también permite la comunicación con el contenedor *Web* de *Servlets Tomcat*.

Se eligió *JavaServer Faces* porque da soporte a varios elementos utilizados en *Web* con él se crean las vistas parecidas a HTML estándar, lo que se utiliza facilita su desarrollo, además permite utilizar validaciones, conversiones, mensajes de error e internacionalización (i18n).

Para el proceso de desarrollo, la metodología *IWeb* se consideró apropiada porque en sus fases involucran elementos que se utilizan para la elaboración de aplicaciones *Web* mientras que para la construcción de la ontología se aplicó *Methontology* porque permite la extensión y mejora de la ontología base propuesta en [10] a través de las actividades de especificación, conceptualización, formalización, implementación y mantenimiento de la misma, además se hace la reutilización de conceptos y relaciones ya propuestos que se identificaron en las actividades de gestión (planificación, control y aseguramiento de la calidad), y de soporte (adquisición de conocimientos, integración, evaluación, documentación y gestión de la configuración).

Capítulo 3. Aplicación de la metodología

El presente capítulo describe la metodología utilizada durante el desarrollo de la tesis, bajo el modelo para establecer y utilizar principios científicos, de ingeniería y gestión con enfoques sistemáticos y disciplinados para el éxito del desarrollo, manejo y mantenimiento del sistema, por lo cual se implementó *IWeb*.

3.1 Formulación

En esta etapa se identifican los requerimientos, propósitos y metas para la construcción de la plataforma *Web*, el propósito general consiste en aplicar mecanismos de barajeo y filtrado múltiple para la gestión de exámenes en el Instituto Tecnológico de Orizaba, por lo cual en la Tabla 3.1 muestra la función de las personas involucradas en el proceso de evaluación (lado izquierdo) y la lista de actividades que permite realizar la plataforma (lado derecho), cabe mencionar que este trabajo extiende de uno elaborado por un alumno egresado de la Maestría en Sistemas Computacionales, por lo tanto se toman en cuenta las funciones actuales y las que fueron definidas con anterioridad en [10]. Esta etapa sólo se generalizan los objetivos que se requieren obtener y quienes participan en las tareas definidas.

Tabla 3.1 Lista de usuarios y actividades para la plataforma

| Usuario | Tareas |
|---------------|-----------------------------------|
| Estudiante | 1. Iniciar sesión |
| | 2. Gestionar Exámenes |
| | 3. Gestionar Preguntas |
| | 4. Salir del sistema. |
| | 5. Gestionar Alumnos. |
| Docente | 6. Gestionar Profesores |
| | 7. Aplicar Filtros múltiples |
| | 8. Gestionar Grupos. |
| | 9. Gestionar Temas. |
| | 10. Aplicar Algoritmos de barajeo |
| Administrador | 11. Ver calificación. |
| | 12. Ver Concentrado |
| | 13. Presentar Exámenes |
| | 14. Gestionar Cursos. |
| | 15. Descargar Reportes |

3.2 Planificación

En esta etapa se estima el costo global del proyecto y se evalúan los riesgos asociados al desarrollo de la aplicación, por lo tanto como se trata de la extensión del sistema *Quadelito 8.0*, el cual es de licencia libre, esta plataforma extendida para el Instituto Tecnológico de Orizaba seguirá con la misma licencia para los usuarios que la utilicen y para evitar riesgos en los tiempos de desarrollo, se tuvo la guía del director de tesis para supervisar las funciones y garantizar la terminación de la misma.

3.3 Análisis

Para esta fase de la metodología se establecen los requerimientos para la aplicación *Web*, la identificación de los elementos del contenido y los requisitos de diseño gráfico, para visualizar mejor los requerimientos de la plataforma, la Fig. 3.1 muestra el diagrama de casos de usos, en el cual los óvalos sin relleno representan los casos de uso general que fueron propuestos en [10] y los que están rellenos de color gris representan la prolongación para el proyecto de tesis, de los cuales se hace una breve descripción. Los casos de uso rellenos de color naranja se modificaron para adaptar las nuevas modificaciones.

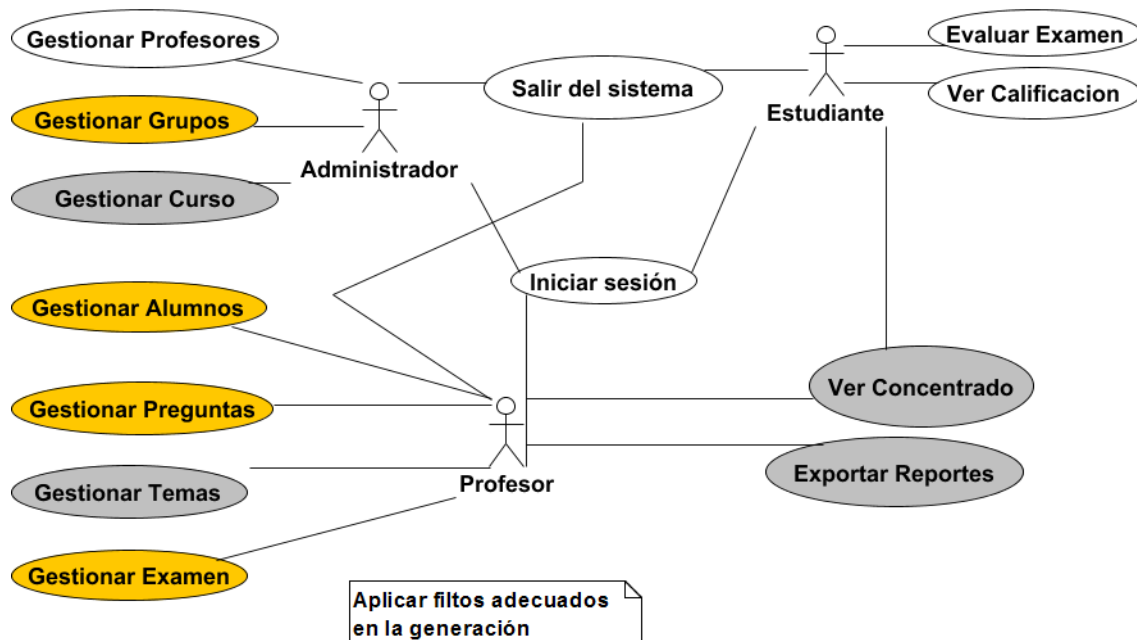


Fig. 3.1 Diagrama de casos de usos extendido de alto nivel.

3.3.1 Caso de uso “Gestionar Cursos”

El caso de uso “Gestionar Cursos” implica administrar el módulo de los cursos impartidos en el Tecnológico a las diferentes carreras ofertadas, la Fig. 3.2 muestra el diagrama de caso de uso detallado y en la Tabla 3.2 la descripción breve del mismo.

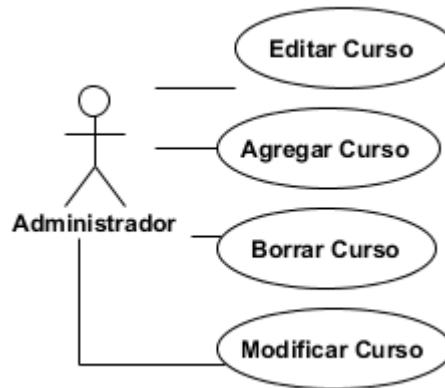


Fig.3.2 Diagrama de casos de usos “Gestionar Curso” detallado.

Tabla 3.2 Descripción del caso de uso “Gestionar Cursos”

| Clave | Caso de Uso | Descripción |
|-------|-----------------|--|
| CU01 | Editar Curso | El administrador edita la información de un curso |
| CU02 | Agregar Curso | El administrador agrega la información de un curso |
| CU03 | Borrar Curso | El administrador borra la información de un curso |
| CU04 | Modificar Curso | El administrador modifica la información de un curso |

3.3.2 Caso de uso “Gestionar Temas”

El caso de uso “Gestionar Temas” está asociado con “Gestionar Cursos”, porque se considera que para los exámenes se evalúa un tema en específico que el docente determine, por lo cual los cursos que se imparte en el Tecnológico contienen temas que se evalúan por un determinado grupo. En la Fig. 3.3 se muestra el diagrama de caso de uso detallado y en la Tabla 3.3 se visualiza la descripción del mismo.

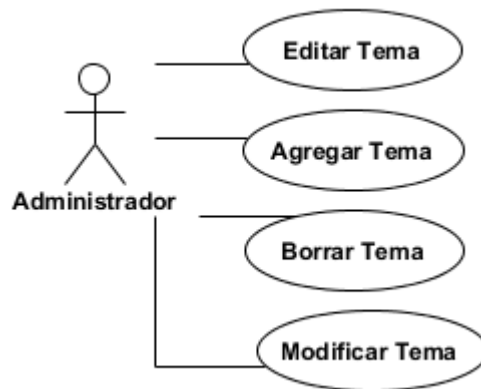


Fig. 3.3 Diagrama de casos de usos “Gestionar Tema” detallado.

Tabla 3.3 Descripción del caso de uso “Gestionar Cursos”

| Clave | Caso de Uso | Descripción |
|-------|----------------|---|
| CU05 | Editar Tema | El administrador edita la información de un tema |
| CU06 | Agregar Tema | El administrador agrega la información de un tema |
| CU07 | Borrar Tema | El administrador borra la información de un tema |
| CU08 | Modificar Tema | El administrador modifica la información de un tema |

3.3.3 Caso de uso “Gestionar Grupos”

Este caso de uso se resalta de color naranja en la Fig. 3.1, el cual indica que hay adaptaciones que se implementaron para extenderlo, el cual consiste en administrar al conjunto de grupos de los diferentes cursos que se imparte en el ITO, por lo que una evaluación ya no pertenece a un solo grupo de alumnos de una carrera sino que la plataforma permite que se aplique el examen del curso a una especialidad diferente o incluso a otro grupo de la misma sin olvidar que se necesita administrar los grupos de acuerdo al semestre en el que se están impartiendo. La Fig. 3.4 muestra el diagrama de caso de uso detallado y en la Tabla 3.4 la descripción del mismo.

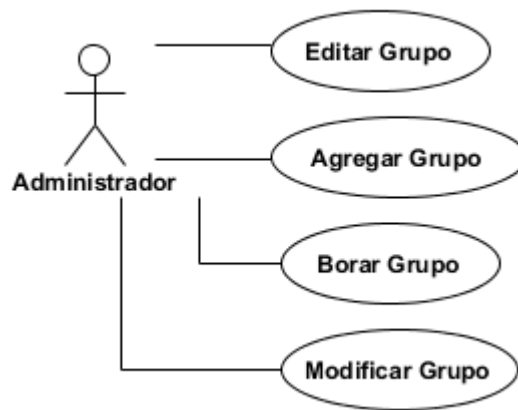


Fig. 3.4. Diagrama de casos de usos “Gestionar Grupos” detallado.

Tabla 3.4 Descripción del caso de uso “Gestionar Grupos”

| Clave | Caso de Uso | Descripción |
|-------|-----------------|--|
| CU09 | Editar Grupo | El administrador edita la información de un grupo por semestre y por profesor |
| CU10 | Agregar Grupo | El administrador agrega la información de un grupo de determinado semestre y profesor |
| CU11 | Borrar Grupo | El administrador borra la información de un grupo |
| CU12 | Modificar Grupo | El administrador modifica la asignación del profesor que imparte un curso a determinado grupo. |

3.3.4 Caso de uso “Gestionar Alumnos”

Como se mencionó, a este caso de uso también se le implementó una mejora donde al alumno se le asigna un grupo previamente creado y haciendo énfasis a la modificación, existen diversos grupos de estudiantes que corresponden a los diferentes cursos que se imparten en el Tecnológico e incluso al alumno se le permite tomar dos o más cursos de su especialidad por semestre, la actualización del caso de uso se muestra en el diagrama de la Fig. 3.5 y en la Tabla 3.5 la descripción breve del mismo.

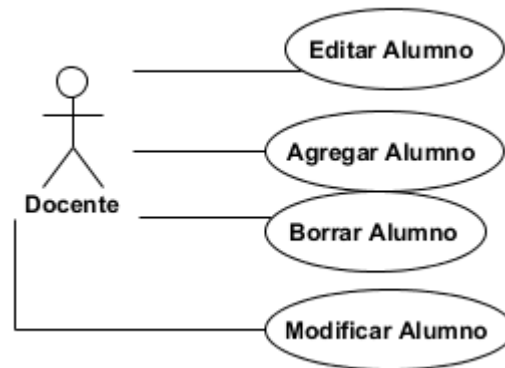


Fig. 3.5 Diagrama de casos de usos “Gestionar Alumnos” detallado.

Tabla 3.5 Descripción del caso de uso “Gestionar Alumnos”

| Clave | Caso de Uso | Descripción |
|-------|------------------|--|
| CU13 | Editar Alumno | El profesor edita la información de un alumno que está en determinado grupo y semestre. |
| CU14 | Agregar Alumno | El profesor agrega la información necesaria de un alumno a determinado grupo que imparte. |
| CU15 | Borrar Alumno | El profesor borrar la información necesaria de un alumno de un determinado grupo que imparte. |
| CU16 | Modificar Alumno | El profesor modifica la información necesaria de un alumno que pertenece a un grupo que imparte. |

3.3.5 Caso de uso “Gestionar Preguntas”

El caso de uso “Gestionar Preguntas” también se extendió, porque se permite reutilizar una pregunta en diferentes exámenes si el profesor así lo requiere, el caso de uso propuesto en [10] sólo contempla una pregunta para uso de la misma en la asignatura de Cálculo Diferencial, por lo tanto la plataforma desarrollada admite la reutilización de la pregunta, además de asignarle un nivel para determinar el grado de dificultad y así contar con una opción más al momento de crear un examen. La Fig. 3.6 muestra el diagrama de caso de uso y en la Tabla 3.6 la descripción breve del mismo.

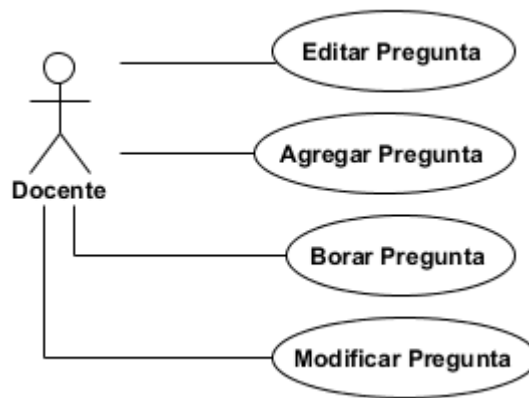


Fig. 3.6 Diagrama de casos de usos “Gestionar Preguntas” detallado.

Tabla 3.6 Descripción del caso de uso “Gestionar Preguntas”

| Clave | Caso de Uso | Descripción |
|-------|--------------------|--|
| CU17 | Editar Pregunta | El profesor edita la información necesaria de una pregunta. |
| CU18 | Agregar Pregunta | El profesor agrega la información necesaria de una pregunta para ser implementada en una evaluación, además si se necesita una imagen se agrega, así mismo elige a que tema corresponde. |
| CU19 | Borrar Pregunta | El profesor borra la información de una pregunta |
| CU20 | Modificar Pregunta | El profesor modifica la información de una pregunta |

3.3.6 Caso de uso “Gestionar exámenes”

Este módulo ampliado permite la creación de una evaluación con opciones de filtrados especiales al momento de construirla, la cual consiste en elegir preguntas del banco de reactivos, que forman parte de un top que el docente considere mediante la búsqueda de reactivos, dichos reactivos cuentan con una descripción del nivel de complejidad que el docente considere al momento de su elaboración o que pertenezca al grupo de las más o menos utilizadas en diferentes evaluaciones, cabe resaltar que las configuraciones de barajeo que propuso el alumno egresado de la Maestría en Sistemas Computacionales se omiten porque se aplicarán de manera implícita al momento de mostrar un examen al estudiante. A continuación se muestra un listado del tipo de filtro que visualiza el docente al momento de crear la evaluación.

1. Top de los reactivos más usados.
2. Top de reactivos menos usados.
3. Top de reactivos por docente.
4. Combinaciones de reactivos de diversos docentes.
5. Por nivel de complejidad.

La Fig. 3.7 muestra el diagrama de caso de uso y en la Tabla 3.7 la descripción breve del mismo.

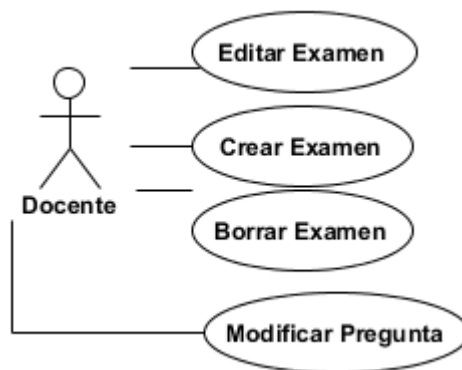


Fig. 3.7 Diagrama de casos de usos “Gestionar Exámenes” detallado.

Tabla 3.7 Descripción del caso de uso “Gestionar Exámenes”

| Clave | Caso de Uso | Descripción |
|-------------|------------------|---|
| CU21 | Editar Examen | El profesor edita la información de un examen, es decir, quitar los reactivos de un examen. |
| CU22 | Crear Examen | El profesor crea un examen aplicando los filtros mencionados si así lo desea. |
| CU23 | Borrar Examen | El profesor borra las preguntas y los datos de un examen. |
| CU24 | Modificar Examen | El profesor modifica la información de un examen. |

3.3.7 Caso de uso “Ver concentrado”

Este caso de uso permite visualizar un concentrado de calificaciones al alumno y profesor para llevar un control del puntaje obtenido al evaluarse un tema en particular, el tema se permite

evaluarse por diversos grupos y además el alumno tiene la posibilidad de resolver uno o varios exámenes de diferentes cursos. El resto de los casos de uso que están sin relleno, se mantienen y su definición propuesta se encuentra en [10].

3.3.8 Especificación y conceptualización de la ontología *itoTest*

Es necesario especificar la función y el contenido de la ontología extendida, de esta manera en la Tabla 3.8 se hace la descripción del objetivo del diseño de la ontología *itoTest* y la Tabla 3.9 muestra los requerimientos funcionales, además toma como referencia algunas funciones de la ontología *Assessment* propuesta en el sistema *Quadelito 8.0*.

Tabla 3.8 Especificaciones de la ontología ITOTest

| Especificación de la ontología <i>ITOTest</i> |
|---|
| Propósito |
| La ontología ITOTest se desarrolló con el propósito de extender el proceso de evaluación de la ontología <i>Assessment</i> , la cual solo se basa en la parte de los reactivos y desarrollo de los algoritmos de barajeo, sin embargo se necesita complementar la ontología para gestionar los temas, grupos y cursos que se imparten en el Tecnológico de Orizaba. |
| Alcance |
| La ontología se limita para el proceso de evaluación de las asignaturas técnico-científicas del Tecnológico, para diferentes temas de diversos cursos |
| Lenguaje de implementación |
| <i>Ontology Web Language (Owl)</i> |
| Usuarios |
| <ol style="list-style-type: none"> 1. Profesor 2. Alumno |
| Módulos a manejar |
| <ol style="list-style-type: none"> 1. Gestión de profesores 2. Gestión de alumnos 3. Gestión de temas 4. Gestión de cursos 5. Gestión de evaluaciones |

Tabla 3.9 Requerimientos funcionales y no funcionales de la ontología ITOTest

| Requerimientos | |
|--|--|
| Funcionales | No Funcionales |
| <ol style="list-style-type: none"> 1. Definir los atributos de un profesor 2. Definir los atributos de un curso 3. Definir los atributos de los temas de los cursos 4. Definir los atributos de los alumnos 5. Almacenar la información de profesores, alumnos, cursos, etc. 6. Almacenar la frecuencia de la utilización de los reactivos 7. Almacenar el grado de dificultad de los reactivos 8. Almacenar los resultados de los examen 9. Definir las relaciones entre los conceptos | <p>Considerar estándar <i>IMS QTI</i> de un modelo de datos para el dominio de <i>E-Learning</i> limitado para el proceso de evaluación en el ITO.</p> |

El diseño de la ontología *itoTest* se basa en las tareas definidas por la metodología *Methontology* porque permite cubrir los requerimientos propios del *E-learning*, por ejemplo motivar al estudiante para procesar la información mientras se trabaja con los materiales de aprendizaje y que los medios de comunicación faciliten el procesamiento de la información para que el alumno obtenga el conocimiento requerido, sin embargo la extensión de la misma hizo que se reestructuraran los elementos que la conformaban. A continuación se describen las once tareas de conceptualización de la ontología.

— **Tarea 1: Glosario de Términos**

Este glosario está constituido por los términos de interés del dominio *E-learning* determinados para el proceso de evaluación de las materias técnico-científicas del ITO. La Tabla 3.10 muestra los elementos (conceptos y atributos) más importantes. Cabe mencionar que hay términos que se entienden como metadatos del estándar de Especificación de Interoperabilidad de Preguntas y Pruebas de IMS (*IMS Question & Test Interoperability Specification*) que menciona el intercambio de temas, contenidos y resultados de datos entre herramientas de edición, bancos de reactivos, plataformas de aprendizaje y de evaluación. No se detectaron sinónimos para los conceptos.

Tabla 3.10 Extracto del glosario de términos

| Nombre | Tipo | Descripción |
|-----------------------|----------|---|
| <i>Text</i> | Atributo | Es el enunciado que forma parte del <i>Item</i> . |
| <i>Distractor</i> | Atributo | Son aquellas respuestas que presentan errores de pensamiento que ocurren con frecuencia y que parecen verdaderas. |
| <i>Correct Answer</i> | Atributo | Es la respuesta correcta a un <i>Item</i> . |
| <i>Result</i> | Atributo | Es el puntaje obtenido del <i>Quiz</i> . |
| <i>Item</i> | Concepto | Es una pregunta estructurada que forma parte de un <i>Quiz</i> . |
| <i>Course</i> | Concepto | Es la asignatura que imparte un maestro. |
| <i>Topic</i> | Concepto | Los temas que están diseñados con base en los contenidos de las asignaturas. |
| <i>Student</i> | Concepto | Persona que presenta un <i>Quiz</i> . |
| <i>Quiz</i> | Concepto | Es el conjunto de <i>Item</i> 's para ser evaluados. |
| <i>Concentrated</i> | Concepto | Es el conjunto de resultados de varios <i>Quiz</i> que presenta el <i>Student</i> . |
| <i>Group</i> | Concepto | Es el conjunto de <i>Student</i> 's de una asignatura que presentan un <i>Quiz</i> . |
| <i>Teacher</i> | Concepto | Es la persona responsable de impartir una materia (<i>Course</i>) y elaborar un <i>Quiz</i> . |

— Tarea 2: Taxonomía de conceptos

En esta tarea se observa la jerarquía de los conceptos de acuerdo a la tarea anterior. Véase Fig. 3.8

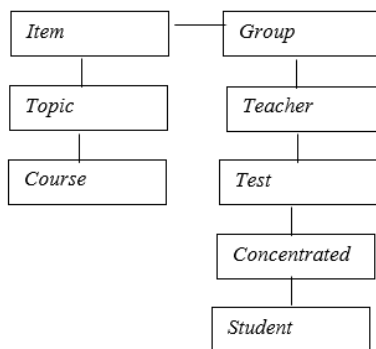


Fig. 3.8 Taxonomía de conceptos.

-Tarea 3: Diagrama de relaciones binarias

En esta tarea se muestra el diagrama que establece los tipos de relaciones entre los términos antes definidos. Fig. 3.9

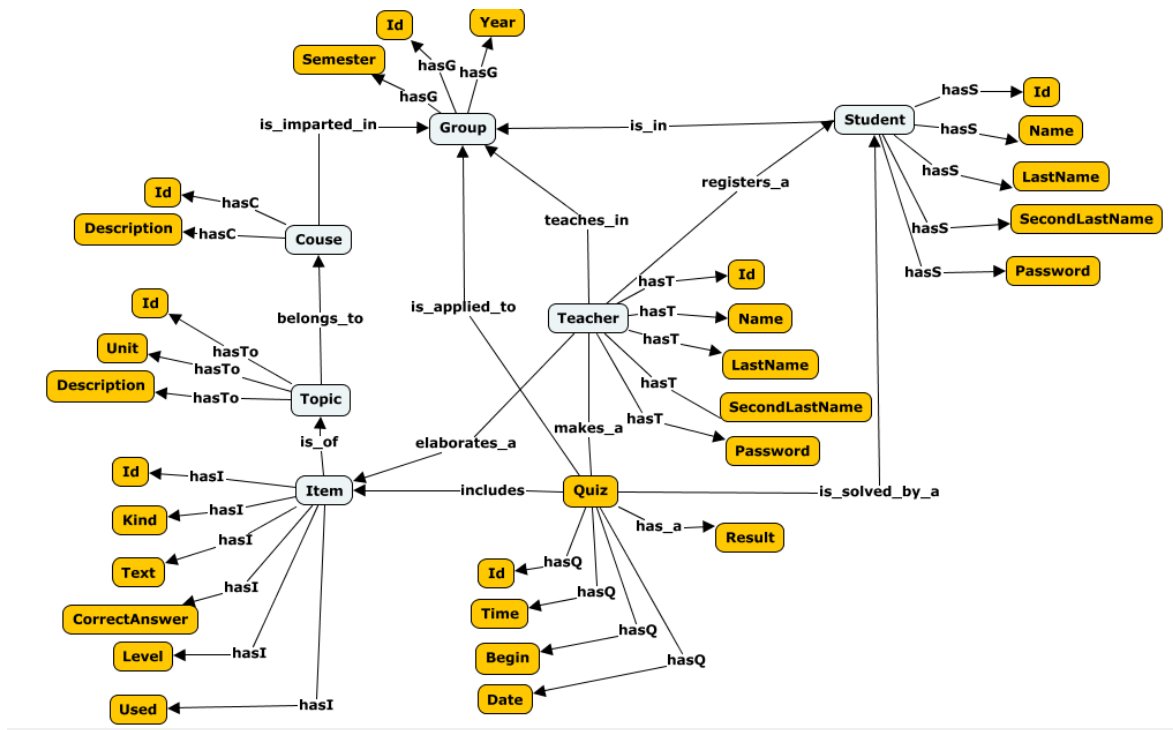


Fig. 3.9 Diagrama de relaciones binarias

La Fig. 3.9 permite visualizar conceptos y relaciones tomando en cuenta los elementos que se necesitan para el proceso de evaluación en las asignaturas técnico-científicas, por lo cual se agregaron los conceptos y atributos que son: *Teacher*, *Course*, *Topic*, *Result*, *Text*, *Student*, *Image* y *Correct Answer*, *Quiz* cuya descripción se encuentra en la Tabla 3.10.

— Tarea 4: Construir el diccionario de conceptos

En la Tabla 3.11 se muestran los conceptos que se determinaron como los más relevantes en el proceso de *E-learning*. Estos conceptos se consideran los de mayor importancia porque en la dinámica correspondiente al proceso de elaboración de evaluaciones, los actores principales son

los profesores, ya que ellos definen y agregan reactivos a los exámenes, sin embargo, esto conlleva otras actividades anteriores, por ejemplo, crear los grupos y los exámenes de acuerdo con los contenidos de las asignaturas que hayan impartido. Por otra parte, el *Quiz* es por sí mismo, un elemento fundamental, ya que se trata del medio o la herramienta mediante la cual se logra la evaluación.

Tabla 3.11 Diccionario de conceptos.

| Concepto | Relaciones |
|----------------|--|
| <i>Teacher</i> | <i>Teaches_in</i> -> <i>Group</i> <i>elaborates_a</i> -> <i>Item</i> <i>makes_an</i> -> <i>Quiz</i> |
| <i>Quiz</i> | <i>is_applied_to</i> -> <i>Group</i> <i>is_solved_by_a</i> -> <i>Student</i> <i>includes</i> -> <i>Item</i> |

– **Tarea 5: Detallar relaciones binarias**

Para cada relación se especifica el nombre, conceptos fuente (A) y destino (B), cardinalidad del dominio de *E-learning*. Véase Tabla 3.12

Tabla 3.12 Detalles de las relaciones binarias.

| Nombre | Concepto A | Concepto B | Cardinalidad |
|-----------------------|----------------|----------------|-----------------|
| <i>teaches_in</i> | <i>Teacher</i> | <i>Course</i> | 1: 1...* |
| <i>is_impacted_to</i> | <i>Course</i> | <i>Group</i> | 1...*: 1 |
| <i>creates_a</i> | <i>Teacher</i> | <i>Group</i> | 1: 1...* |
| <i>makes_an</i> | <i>Teacher</i> | <i>Quiz</i> | 1: 1...* |
| <i>elaborates_a</i> | <i>Teacher</i> | <i>Item</i> | 1: 1...* |
| <i>Includes</i> | <i>Quiz</i> | <i>Item</i> | 1 : * |
| <i>is_applied_to</i> | <i>Quiz</i> | <i>Group</i> | 1: 1...* |
| <i>is_solved_by_a</i> | <i>Quiz</i> | <i>Student</i> | 1...*: 1...* |
| <i>is_of</i> | <i>Item</i> | <i>Topic</i> | 1: 1 |
| <i>belongs_to</i> | <i>Topic</i> | <i>Course</i> | 1...*: 1 |
| <i>is_in_a</i> | <i>Student</i> | <i>Group</i> | 1...*: 1...*: 1 |

– **Tarea 6: Definir los atributos de instancias**

Aquí se definen algunos atributos de instancias. Todas las clases tienen por lo menos una instancia, a continuación, en la Tabla 3.13 se muestran los más relevantes. *Name* representa un

atributo de *Student*, el cual es el nombre del alumno, *Id* representa la matrícula del estudiante, los demás atributos que pertenecen a *Item* esta descritos en la Tabla 3.10.

Tabla 3.13 Atributos de instancia de las clases *Student* e *Item*.

| Atributo | Con-cepto | Tipo de Valor | Rango | Cardinalidad |
|-----------------------|----------------|---------------|----------------|--------------|
| <i>Name</i> | <i>Student</i> | <i>String</i> | ilimitado | 1 |
| <i>Id</i> | <i>Student</i> | <i>String</i> | ilimitado | 1 |
| <i>LastName</i> | <i>Student</i> | <i>String</i> | ilimitado | 1 |
| <i>SecondLastName</i> | <i>Student</i> | <i>String</i> | ilimitado | 1 |
| <i>Correct Answer</i> | <i>Item</i> | <i>int</i> | números reales | 1 |
| <i>Text</i> | <i>Item</i> | <i>String</i> | ilimitado | 1 |

– Tarea 7: Detallar Atributos de clase

La Tabla 3.14 muestra los atributos de clase, los cuales representan características generales de un concepto, es decir, todas las instancias del concepto tendrán el mismo valor para ese atributo.

Tabla 3.14 Atributos de clase Quiz.

| Atributo | Concepto | Tipo de Valor | Rango de Valores |
|--|-------------|---------------|------------------|
| Mínimo de <i>Item</i> ’s por <i>Quiz</i> . | <i>Quiz</i> | <i>Item</i> | 3 |

– Tarea 8: Detallar Constantes

Para cada constante se especifica el nombre, tipo de valor, valor y unidad de medida. Esta tarea no aplica para el dominio del *E-learning*.

– Tarea 9: Detallar Axiomas Formales

Los axiomas describen restricciones del modelo, así como las características que se cumplen para algún concepto. En la Tabla 3.15 se muestra el nombre, descripción, expresión lógica, conceptos o relaciones y variables de los axiomas que intervienen.

Tabla 3.15 Axiomas Formales

| Nombre | Descripción | Expresión Lógica | Conceptos/ Relaciones | Variables |
|-----------------|---|--|--------------------------------------|-----------|
| <i>NoAnswer</i> | Un distractor es una respuesta que no es correcta | NO [EXISTE(<i>Item</i> (?x) Y <i>?x.Correct Answer=false</i> ;)] | <i>Item</i> <i>Correct Answer</i> | ?x |
| <i>Anwser</i> | De una pregunta solo hay una respuesta correcta | EXISTE(<i>Item</i> (?x) Y <i>?x.Correct Aswer =true</i> ;) | <i>Item</i> <i>Correct Answer</i> | ?x |

– Tarea 10: Definir reglas

En esta tarea se definen las reglas necesarias para la ontología de *E-learning*, se definen en la Tabla 3.15, recordando que para cada regla, se especifica el nombre, descripción, expresión y conceptos.

Tabla 3.15 Reglas de la ontología.

| Nombre | Descripción | Expresión | Conceptos |
|--|---|--|----------------------------|
| Barajear opciones | Se barajan las opciones siempre y cuando el atributo no se haya seleccionado. | Si [<i>Quiz</i> (?x) Y <i>Item</i> (?y) Y <i>?x include ?y</i>] ENTONCES [<i>chose</i> (?x, ?y) <i>shuffe</i> (?x, ?y)] | <i>Quiz</i> <i>Item</i> |
| Mostrar <i>Item</i> 's más frecuentes. | Muestra los <i>Item</i> 's utilizados con mayor frecuencia. | Si [<i>Quiz</i> (?x) Y <i>Item</i> (?y)Y <i>chose</i> (?x, ?y)] ENTONCES <i>?y.masUsu=?y. masUsu + 1</i> | <i>Quiz</i> <i>Item</i> |
| Mostrar <i>Item</i> 's menos frecuentes. | Muestra los <i>Item</i> 's utilizados con menor frecuencia. | Si [<i>Quiz</i> (?x) Y <i>Item</i> (?y)Y <i>chose</i> (?x, ?y)] ENTONCES <i>?y.menUsu=?y.menUsu + 1</i> | <i>Quiz</i> <i>Item</i> |

– Tarea 11: Definir instancias

En esta tarea se definen las instancias de conceptos de la ontología, cada una de ellas cuenta con nombre, el tipo concepto al que pertenece y los valores de sus atributos de instancia, si se conocen. Esta tarea no aplica por el momento hasta la creación de la ontología del dominio de *E-learning*.

3.3.9 Codificación en *Protégé*

El siguiente paso fue la codificación de la ontología con la herramienta *Protégé*, porque permite generar código de un lenguaje ontológico. La Fig. 3.10 muestra el panel de clases, la cual, permite definir las, establecer la jerarquía y restricciones. En la Fig. 3.11 se visualiza la creación de algunas relaciones binarias.

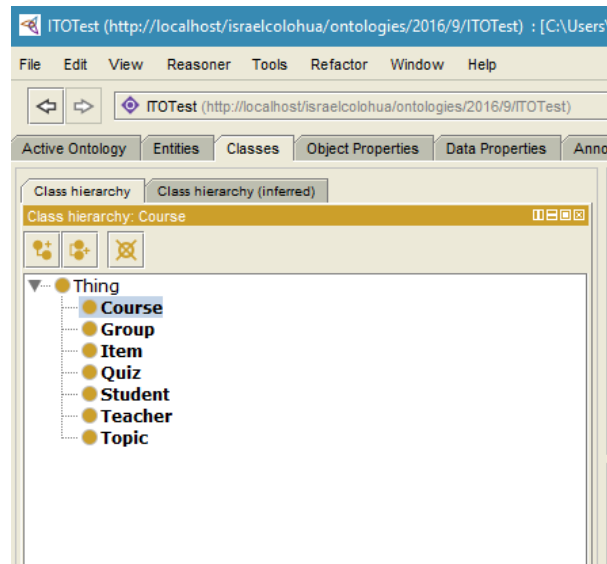


Fig. 3.10 Creación de clases del dominio de *E-learning* en *Protégé*

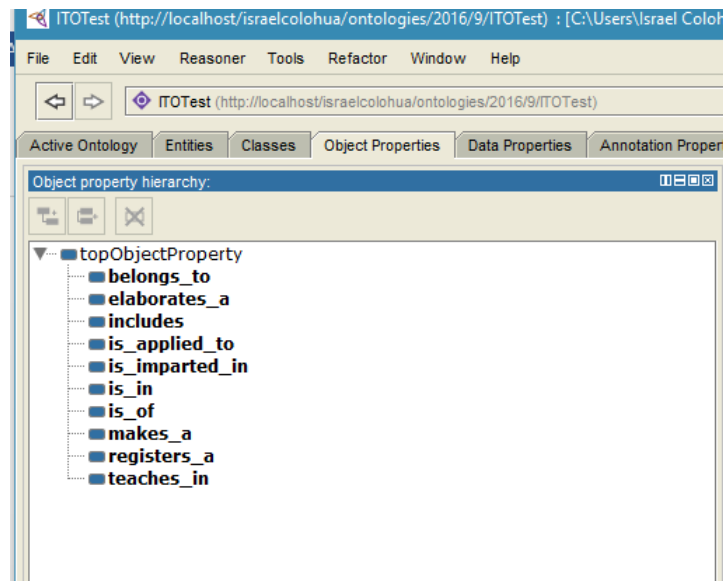


Fig. 3.11 Relaciones binarias del dominio *E-learning*

Para comprobar el diseño de la ontología se agregaron algunos individuos (instancias) para comprobar que las relaciones y las clases estén correctamente diseñadas. En la Fig. 3.12 se observan algunos individuos de los conceptos de *Teacher*, *Topic*, *Group* y *Course* con sus respectivos atributos de instancia que se les proporcionaron para realizar algunas consultas basadas en SPARQL.

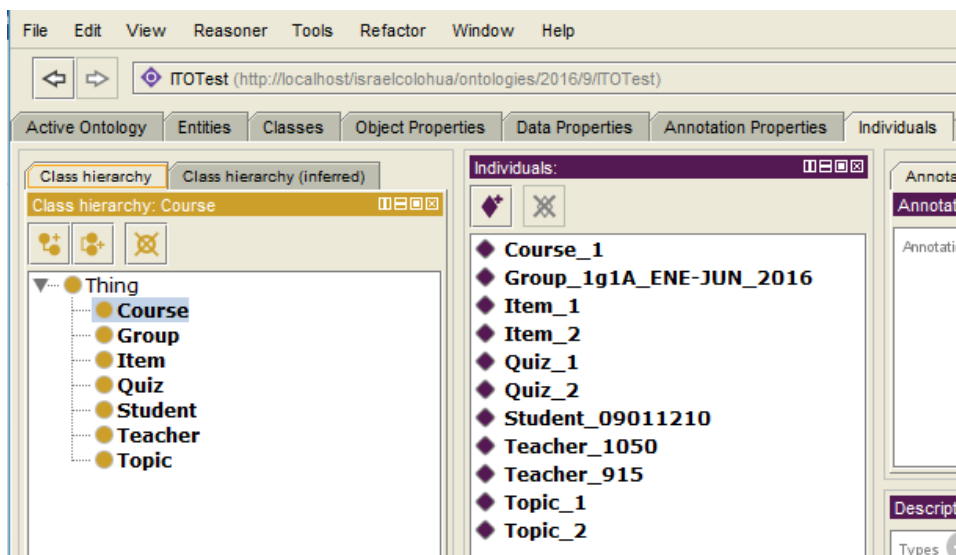


Fig. 3.12 Instancias agregadas a la ontología del dominio *E-learning*

A continuación, en la Fig. 3.13, Fig. 3.14 y Fig. 3.15 se muestran los resultados de algunas consultas que se elaboraron sobre la ontología y de igual manera se muestran los listados que representan su respectivo código escrito en *SPARQL*. Es importante mencionar que se han omitido los prefijos en las consultas basadas en *SPARQL* para optimizar espacio en la presente tesis.

Listado 3.1 Buscar los datos de los profesores.

```

1 SELECT ?id ?name ?lastName ?secondLastName
2     WHERE {
3         ?x rdf:type itoTest:Teacher.
4         ?x itoTest:hasTId ?id.
5         ?x itoTest:hasTName?name.
6         ?x itoTest:hasTLastName?lastName.
7         ?x itoTest:hasTSecondLastName ?secondLastName
8     } ORDER BY ASC (?id)

```

| id | name | lastName | secondLastName |
|--------|--------------|----------|--|
| "1050" | "Luis Ángel" | "Reyes" | "Hernández"^^<http://www.w3.org/2001/XMLSchema#string> |

Fig. 3.13 Resultado de la consulta sobre los profesores.

Listado 3.2 Buscar los datos de los cursos

```

1 SELECT ?id ?description
2     WHERE {
3         ?x rdf:type itoTest:Course.
4         ?x itoTest:hasCId ?id.
5         ?x itoTest:hasCDescription ?description
6     }ORDER BY ASC (?id)
7
8

```

```

SELECT ?id ?description
WHERE {
    ?x rdf:type itoTest:Course.
    ?x itoTest:hasCId ?id.
    ?x itoTest:hasCDescription ?description
}ORDER BY ASC (?id)

```

| id | description |
|-----|---|
| "1" | "Cálculo Integral"^^<http://www.w3.org/2001/XMLSchema#string> |

Fig. 3.14 Resultado de la consulta de los cursos

Listado 3.3 Buscar los temas del curso “Course_1”

```

1 SELECT ?id ?unit ?description
2     WHERE {
3         ?x rdf:type itoTest:Topic.
4         ?x itoTest:hasToId ?id.
5         ?x itoTest:hasToUnit ?unit.
6         ?x itoTest:hasToDescription ?description.
7         ?x itoTest:belongs_to itoTest:Course_1
8     }ORDER BY ASC (?unit)

```

```

SELECT ?id ?unit ?description
WHERE {
    ?x rdf:type itoTest:Topic.
    ?x itoTest:hasToId ?id.
    ?x itoTest:hasToUnit ?unit.
    ?x itoTest:hasToDescription ?description.
    ?x itoTest:belongs_to itoTest:Course_1
}ORDER BY ASC (?unit)

```

| id | unit | description |
|-----|------|---|
| "1" | "1" | "Antiderivadas"^^<http://www.w3.org/2001/XMLSchema#string> |
| "2" | "2" | "Métodos de Integración"^^<http://www.w3.org/2001/XMLSchema#string> |

Fig. 3.15 Resultado de la consulta de buscar los temas del curso 1

3.4 Ingeniería

Esta etapa se compone de secuencias paralelas de tareas que consisten en elaborar el diseño de la arquitectura e interfaz de usuario.

3.4.1 Diseño arquitectónico

La plataforma *Web* implementa el patrón arquitectónico MVC (Modelo-Vista-Controlador), el cual permite separar en capas los datos, la lógica de negocio y la interfaz de usuario.

La Fig. 3.16 muestra el patrón arquitectónico MVC de la plataforma desarrollada que separa los elementos antes mencionados en capas: en la vista están las etiquetas de *PrimeFaces* y los archivos *.xhtml* que se comunicarán con *FacesServlet*, el cual administra las peticiones de los recursos en la plataforma y muestra los resultados en la capa de la vista, el conjunto de *ManageBean's* se comunica con la clases de modelo, las cuales representan la lógica de negocio y por último existe una clase que se llama *AccesoOnto.java* que accede a los datos de la ontología mediante el API de Jena, cabe mencionar que todos estos elementos se ejecutan por el contenedor *Web Tomcat*.

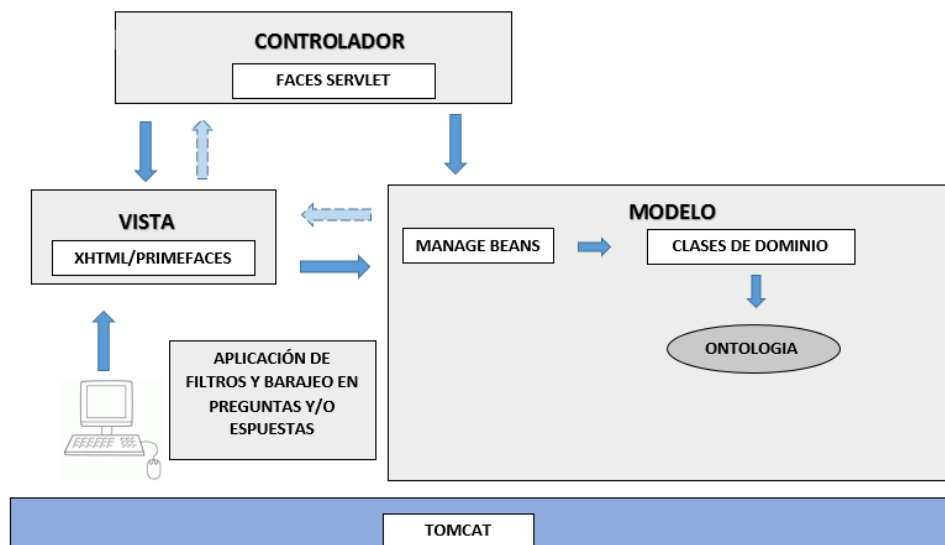


Fig. 3.16 Patrón Arquitectónico MVC

3.4.2 Vista Lógica

En la Fig. 3.17 se muestra el diagrama de clases que se elaboró para la plataforma *Web*. Las relaciones descritas se ocuparon para el desarrollo de la ontología, además se aclara que cada clase cuenta con las operaciones descritas que hacen que funcione el prototipo de interfaz.

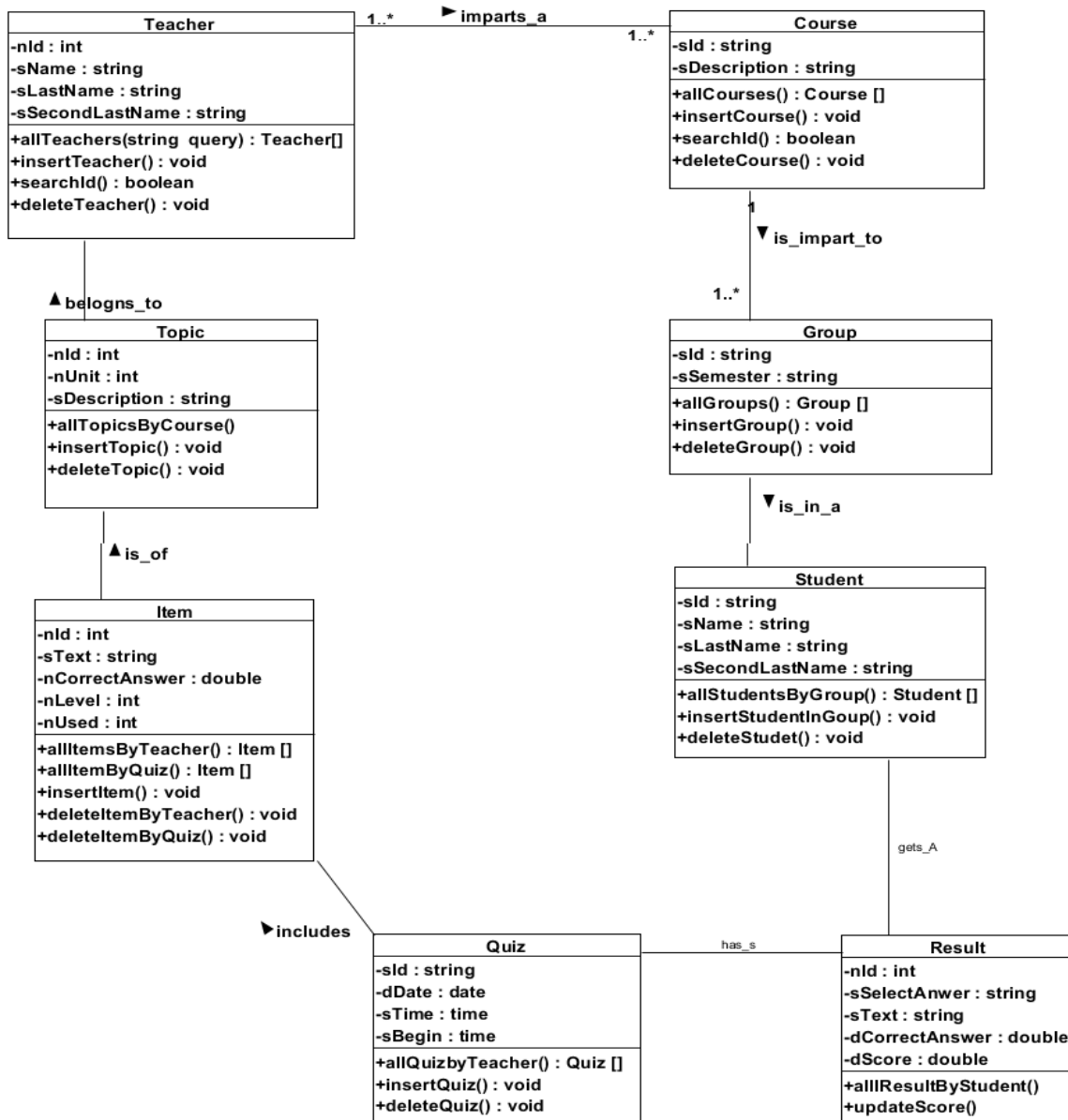


Fig. 3.17 Diagrama de clases

Nota: Cada clase equivale a un concepto de la ontología *ITOTest*.

3.4.3 Vista de Desarrollo

Al implementar el patrón arquitectónico MVC a la plataforma *Web* permite separar los elementos que conforman, es decir, las clases codificadas en *Java* se comunican entre sí, los archivos de la vista (*.xhtml*) interactúan con los *ManageBeans* y estos con las clases de modelo. La Fig. 3.18 muestra el diagrama de paquetes de los archivos que componen a la plataforma *Web*.

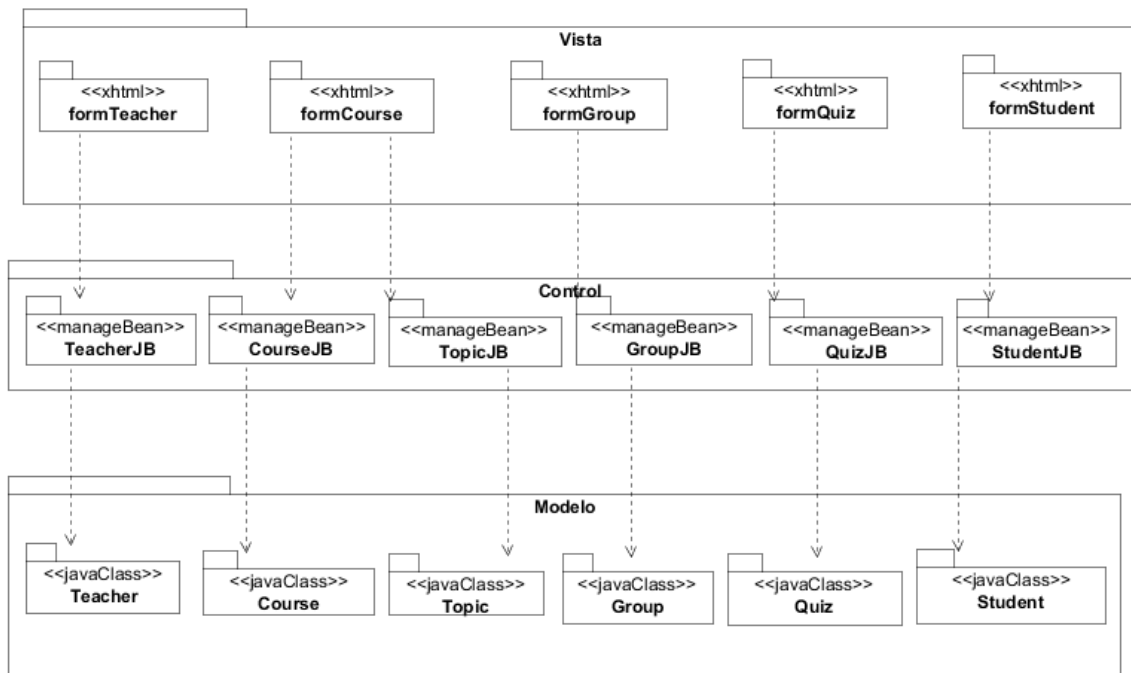


Fig. 3.18 Distribución del patrón arquitectónico de JSF

3.4.4 Vista física

En la Fig. 3.19 se muestra la vista física de la plataforma, la cual se hace énfasis porque se realizaron cambios diferentes a la propuesta en [10]. El manejo de toda la información se obtendrá de la ontología, por lo que la base de datos queda descartada para el siguiente diagrama de despliegue y como contenedor *Web* se utilizó *Apache Tomcat*, porque ejecuta menos componentes que *Glassfish*, La plataforma *Web* accede a la ontología *ITOTest* mediante consultas *SPARQL*.

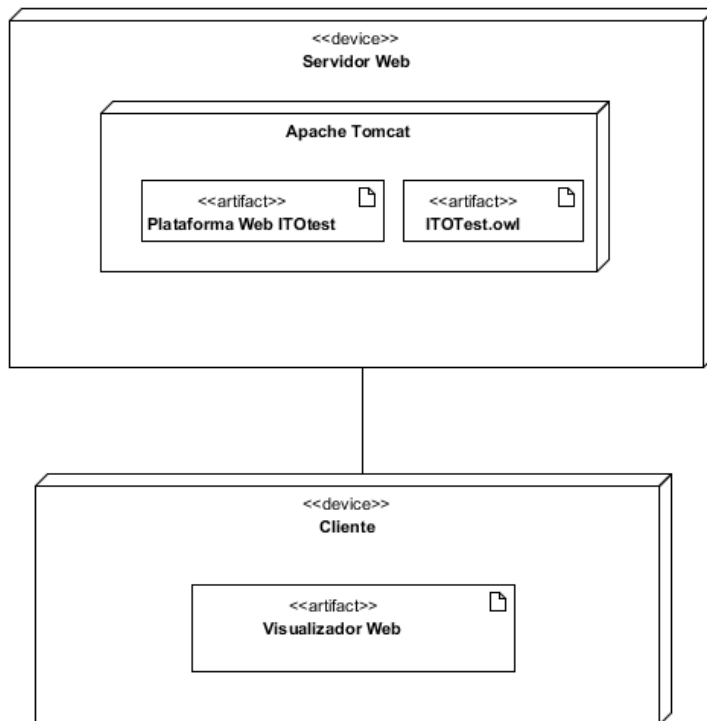


Fig. 3.19 Diagrama de despliegue

.4.5 Diseño del contenido

En esta sección de la metodología se describe las narrativas de los casos de uso sobre el contenido de la información que se presentará así como descripción de código relevante.

3.4.6 Acceso a la ontología

Para la aplicación *Web*, se necesita contar con una clase en *Java* que se conecte a la ontología, la cual mediante la API de *Jena* permite crear y leer grafos RDF, serializa la información usando formatos *RDF/XML* o *Turtle*. Además *Jena* permite ejecutar consultas en *SPARQL* para obtener los datos, en la presente tesis se ocupó una clase llamada `AccesoOnto.java`. A continuación en el Listado 3.4 se muestra el método que acceder a la ontología y ejecutar una consulta (líneas 6-39).

Listado 3.4 Acceso a la ontología.

```

1  public class AccesoOnto implements Serializable{
2  private String sServ = "";
3  private String sNomArch = "";
4  private String sLenguaje="RDF/XML";
5  public ArrayList<ArrayList> ejecutarConsultaOnto(String sQuery)
6      throws Exception {
7  ArrayList<ArrayList> arr=null;
8  Query q=null;
9  QueryExecution qe=null;
10  ResultSet rs=null;
11  OntModel modOwl=null;
12      if ((getNomArch()==null || getNomArch().equals("")) ||
13          (sQuery==null || sQuery.equals("")))
14          throw new Exception
15              ("AccesoOnto.ejecutarConsultaOnto:faltan datos");
16      else{
17          try{
18              modOwl = ModelFactory.createOntologyModel
19                  (OntModelSpec.OWL_MEM_MINI_RULE_INF);
20              modOwl.read("file:"+getNomArch(), sLenguaje);
21              q = QueryFactory.create(sQuery);
22              qe = QueryExecutionFactory.create(q, modOwl);
23              rs = qe.execSelect();
24              if (rs != null)
25                  arr = convierteResultSet(rs);
26          }catch(Exception e){
27              throw e;
28          }finally{
29              if (qe != null)
30                  qe.close();
31              if (modOwl != null)
32                  modOwl.close();
33          }
34      }
35      return arr;
36  }
37  }

```

Como se muestra en el listado 3.4, este código permite leer un archivo *.owl* y manipular la información, para el presente proyecto de tesis facilitó la administración de los datos que se encuentran en la ontología.

3.4.6 Narrativas de Caso de Uso

Como parte de la fase de Ingeniería de *IWeb*, menciona que el diseño de contenido se muestra de forma lineal, a continuación se van a visualizar las narrativas de casos de usos más relevantes donde se extendió su funcionalidad o simplemente se cambió el caso para determinado actor.

Capítulo III. Aplicación de la metodología.

El caso de uso “Registrar docente”, le pertenece al rol de administrador, porque se concluyó que es mejor que un usuario administre este módulo. La Tabla 3.16 muestra la narrativa de este caso de uso.

Tabla 3.16 Narrativa del caso de uso “Registrar Profesor”

| | | | |
|-------------------------|---|---|---|
| CU001 | Registrar Profesor | | |
| Versión | V 1.0 | | |
| Fecha | 30/08/16 | | |
| Autor | Israel Colohua Cruz | | |
| Descripción | El sistema almacena la información de un docente en la ontología. | | |
| Precondición | Administrador autenticado. | | |
| Post-condición | Almacenar los datos del docente en la ontología. | | |
| Flujo de Eventos | | Entrada del actor | Respuesta del Sistema |
| | 1 | El caso de uso inicia cuando el administrador presiona del menú la opción “Profesores”. | |
| | 2 | | Muestra el formulario correspondiente para llenar la clave, nombre, apellido paterno, apellido materno y una contraseña de un profesor. |
| | 3 | El administrador llena los campos que solicita el formulario. | |
| | 4 | | Valida que todos los campos estén llenos y que la clave del profesor no esté repetida, además de que contenga 4 dígitos y verifica que los campos del nombre y apellidos no admitan algún número, en caso contrario el sistema muestra mensajes de error. |
| | 5 | Rectifica si los datos que ingresó son correctos. | |
| | 6 | | Almacena la información del profesor en la ontología y muestra una lista actualizada con los datos llenados. |
| | 7 | Visualiza los datos almacenados en una lista actualizada. | |

Capítulo III. Aplicación de la metodología.

En la Tabla 3.17 se muestra la narrativa de caso de uso “Registrar Curso”. Se determinó que el administrador también lo gestione para que no haya algún problema de letras mayúsculas o minúsculas o que alguna palabra le falte acento.

Tabla 3.17 Narrativa del caso de uso “Registrar Curso”

| | | | |
|-------------------------|---|---|---|
| CU002 | Registrar Curso | | |
| Versión | V 1.0 | | |
| Fecha | 30/08/16 | | |
| Autor | Israel Colohua Cruz | | |
| Descripción | El sistema almacena la información de un curso y los temas en la ontología. | | |
| Precondición | Administrador autenticado. | | |
| Post-condición | Almacenar los datos del curso y temas en la ontología. | | |
| Flujo de Eventos | | Entrada del actor | Respuesta del Sistema |
| | 1 | El caso de uso inicia cuando el administrador presiona del menú la opción “Cursos”. | |
| | 2 | | Muestra el formulario correspondiente para llenar el nombre del curso y la descripción de cada uno de los temas. |
| | 3 | El administrador llena los campos que solicita el formulario. | |
| | 4 | | Valida que todos los campos estén llenos y verifica que los valores del nombre y la descripción de los temas no admitan algún número, en caso contrario el sistema muestra mensajes de error. |
| | 5 | Rectifica si los datos que ingresó son correctos. | |
| | 6 | | Almacena la información del curso y los temas en la ontología y muestra una lista actualizada con los datos llenados. |
| 7 | Visualiza los datos almacenados en una lista actualizada. | | |

Capítulo III. Aplicación de la metodología.

La Tabla 3.18 muestra la narrativa del caso de uso “Asignar Grupo”. El administrador gestiona a todos los grupos que se imparten en el Tecnológico, se menciona que un grupo es ofertado en determinado semestre y año en las diferentes especialidades del ITO.

Tabla 3.18 Narrativa del caso de uso “Asignar Grupo”

| | | | |
|-------------------------|--|---|---|
| CU003 | Asignar Grupo | | |
| Versión | V 1.0 | | |
| Fecha | 30/08/16 | | |
| Autor | Israel Colohua Cruz | | |
| Descripción | El sistema almacena la información de la asignación de grupos en la ontología. | | |
| Precondición | Administrador autenticado. | | |
| Post-condición | Almacenar la asignación de grupos en la ontología. | | |
| Flujo de Eventos | | Entrada del actor | Respuesta del Sistema |
| | 1 | El caso de uso inicia cuando el administrador presiona del menú la opción “Grupos”. | |
| | 2 | | Muestra el formulario correspondiente para llenar la clave, semestre, año, curso y profesor que impartirá la materia en el grupo. |
| | 3 | El administrador llena los campos que solicita el formulario. | |
| | 4 | | Valida que todos los campos estén llenos y verifica no este registrado la clave del grupo en el mismo semestre y año, además de que la clave contenga 4 caracteres, en caso contrario el sistema muestra mensajes de error. |
| | 5 | Rectifica si los datos que ingresó son correctos. | |
| | 6 | | Almacena la información de la asignación de grupos en la ontología y muestra una lista actualizada con los datos llenados. |
| 7 | Visualiza los datos almacenados en una lista actualizada. | | |

Capítulo III. Aplicación de la metodología.

En la Tabla 3.19 describe la narrativa del caso de uso “Registrar Alumno” que permite saber al grupo en el que está inscrito un alumno, además existe la posibilidad de que un estudiante curse diferentes materias en el mismo semestre.

En la Tabla 3.20 describe la narrativa del caso de uso “Registrar Reactivo”, en este caso el profesor formula los reactivos que permitirán reutilizarse en los diferentes cuestionarios para las materias técnico-científicas, donde se indican el tema al que pertenecen determinada pregunta.

Tabla 3.19 Narrativa del caso de uso “Registrar Alumno”

| | | | |
|-------------------------|--|---|--|
| CU004 | Registrar Alumno | | |
| Versión | V 1.0 | | |
| Fecha | 30/08/16 | | |
| Autor | Israel Colohua Cruz | | |
| Descripción | El sistema almacena la información de un alumno en la ontología. | | |
| Precondición | Profesor autenticado. | | |
| Post-condición | Almacenar los datos del alumno en la ontología. | | |
| Flujo de Eventos | | Entrada del actor | Respuesta del Sistema |
| | 1 | El caso de uso inicia cuando el profesor presiona del menú la opción “Alumnos”. | |
| | 2 | | Muestra el formulario correspondiente para llenar la matricula, nombre, apellido paterno, apellido materno y una contraseña de un alumno, además de visualizar los grupos que imparte el profesor para asignar al estudiante. |
| | 3 | El profesor llena los campos que solicita el formulario. | |
| | 4 | | Valida que todos los campos estén llenos y que la clave del alumno no esté repetida en determinado grupo, además verifica que los campos del nombre y apellidos no admitan algún número, en caso contrario el sistema muestra mensajes de error. |
| | 5 | Rectifica si los datos que ingresó son correctos. | |
| | 6 | | Almacena la información del aluno en la ontología y muestra una lista actualizada con los datos llenados. |
| 7 | Visualiza los datos almacenados en una lista actualizada. | | |

Tabla 3.20 Narrativa del caso de uso “Registrar Reactivo”

| | | | |
|-------------------------|--|---|---|
| CU005 | Registrar Reactivo | | |
| Versión | V 1.0 | | |
| Fecha | 30/08/16 | | |
| Autor | Israel Colohua Cruz | | |
| Descripción | El sistema almacena la información de un reactivo en la ontología. | | |
| Precondición | Profesor autenticado. | | |
| Post-condición | Almacenar los datos del reactivo en la ontología. | | |
| Flujo de Eventos | | Entrada del actor | Respuesta del Sistema |
| | 1 | El caso de uso inicia cuando el profesor presiona del menú la opción “Reactivos”. | |
| | 2 | | Muestra el formulario correspondiente para llenar el texto, la respuesta correcta, el nivel y el tema al que pertenece el reactivo así como subir una imagen. |
| | 3 | El profesor llena los campos que solicita el formulario. | |
| | 4 | | Valida que todos los campos estén llenos, permite subir una imagen que seleccione el profesor, en caso contrario el sistema muestra mensajes de error. |
| | 5 | Rectifica si los datos que ingresó son correctos. | |
| | 6 | | Almacena la información del reactivo en la ontología y muestra una lista actualizada con los datos llenados. |
| | 7 | Visualiza los datos almacenados en una lista actualizada. | |

Capítulo III. Aplicación de la metodología.

En la Tabla 3.21 describe la narrativa del caso de uso “Registrar Examen”, este caso toma importancia porque involucra otros casos de uso antes mencionados (registrar grupo, curso y profesor), para que un examen se necesita indicar quién lo está elaborando, este último datos se logran se sabe mediante la sesión del profesor.

Tabla 3.21 Narrativa del caso de uso “Registrar Examen”

| | | | |
|-------------------------|--|--|--|
| CU006 | Registrar Examen | | |
| Versión | V 1.0 | | |
| Fecha | 30/08/16 | | |
| Autor | Israel Colohua Cruz | | |
| Descripción | El sistema almacena la información de un examen en la ontología. | | |
| Precondición | Profesor autenticado. | | |
| Post-condición | Almacenar los datos del examen en la ontología. | | |
| Flujo de Eventos | | Entrada del actor | Respuesta del Sistema |
| | 1 | El caso de uso inicia cuando el profesor presiona del menú la opción “Exámenes”. | |
| | 2 | | Muestra el formulario correspondiente para llenarla duración, fecha y hora del examen y la lista de grupos para asociarlo con la evaluación. |
| | 3 | El profesor llena los campos que solicita el formulario. | |
| | 4 | | Valida que todos los campos estén lleno y verifica que no exista un examen registrado en determinada fecha y hora, en caso contrario el sistema muestra mensajes de error. |
| | 5 | Rectifica si los datos que ingresó son correctos. | |
| | 6 | | Almacena la información del examen en la ontología y muestra una lista actualizada con los datos llenados. |
| | 7 | Visualiza los datos almacenados en una lista actualizada. | |

Capítulo III. Aplicación de la metodología.

En la Tabla 3.22 muestra la narrativa del caso de uso “Agregar Reactivo al Examen”, en esta sección se permite mostrar las opciones de filtrado múltiple dónde el profesor elije si desea o no un top de reactivos o si desea algún reactivo de algún otro docente o utilizar alguno de los más o menos implementados en otras evaluaciones.

Tabla 3.21 Narrativa del caso de uso “Agregar Reactivo al Examen”

| | | | |
|-------------------------|---|--|---|
| CU007 | Agregar Reactivo al Examen | | |
| Versión | V 1.0 | | |
| Fecha | 30/08/16 | | |
| Autor | Israel Colohua Cruz | | |
| Descripción | El sistema almacena la asignación de un reactivo a un examen en la ontología. | | |
| Precondición | Profesor autenticado. | | |
| Post-condición | Almacenar la asignación del reactivo en la ontología. | | |
| Flujo de Eventos | | Entrada del actor | Respuesta del Sistema |
| | 1 | El caso de uso inicia cuando el profesor presiona del menú la opción “Exámenes”, seguido de la opción “Elegir Reactivo”. | |
| | 2 | | Muestra una tabla con la información de los reactivos almacenados por los profesores con los datos y las opciones para filtrar una búsqueda de preguntas. |
| | 3 | El profesor llena los campos de búsqueda que se le muestran. | |
| | 4 | | Permite que visualizar la búsqueda de reactivos mediante los datos que ingreso el profesor. |
| | 5 | Selecciona el reactivo que se desea agregar al examen. | |
| | 6 | | Almacena la asignación del reactivo al examen seleccionado en la ontología. |
| 7 | Visualiza los reactivos asignados en una lista. | | |

3.5 Generación de páginas

En esta fase de la metodología *IWeb* consiste en generar páginas que se le muestran al usuario final, la cual es una actividad de construcción donde se aplican herramientas que agilizan la creación de la aplicación. Para el presente trabajo de tesis, la tecnología que se ocupó fue *Prime-Faces* porque facilita y agiliza la parte de interfaz de usuario, además de implementarse con el *Framework* de JSF. En la actualidad las plataformas en línea que están en el mercado se adaptan para visualizarse en diferentes dispositivos, la plataforma desarrollada ocupó la técnica que prioriza la correcta visualización o diseño responsivo para diversos medios que muestran páginas *Web*, la tecnología que permitió esta técnica fue el marco de trabajo *Bootstrap*, esta herramienta permite que la maquetación se adapte a la resolución de los dispositivo donde se visualiza. La Fig. 3.21 señala la interfaz principal del administrador, el cual pertenece al prototipo de la plataforma *Web ITOTest* en la versión 1.2



Fig. 3.20 Prototipo de la interfaz para el administrador.

Para cada tipo de usuario (profesor, alumno y administrador) se crearon interfaces adaptadas a páginas responsivas, en esta versión de la plataforma se muestran formularios con la información que se necesita para administrar cada módulo y para complementar la maquetación de vista se implementó la tecnología de *facelets*, la cual permite crear plantillas y reutilizar los elementos de la misma.

La Fig. 3.21 presenta la interfaz principal del alumno en forma responsiva, para visualizar de esa manera se configuro una red de manera local, la cual permite comunicar a diferentes dispositivos y mediante un visualizador *Web* del dispositivo móvil se ingresó a la dirección IP http://192.168.1.69:8084/ITOTest_v.1.3/.



Fig. 3.21 Prototipo de interfaz responsiva para el alumno.

La Fig. 3.22 muestra el formulario necesario para capturar los datos de un profesor, la información requerida no se almacena en una base de datos relacional como en [10], sino en la ontología diseñada para el dominio de las materias técnico-científicas. Además los datos se visualizan en una tabla elaborada en *PrimeFaces* donde se observa el listado de todos los profesores. En el trabajo base sólo se guardaba la información de los docentes de la asignatura Cálculo Diferencial, por lo tanto la extensión consiste en agregar diversos profesores para que posteriormente se les asigne un grupo de los diferentes cursos y la Fig. 3.23 muestra la interfaz para el manejo responsivo de la misma.

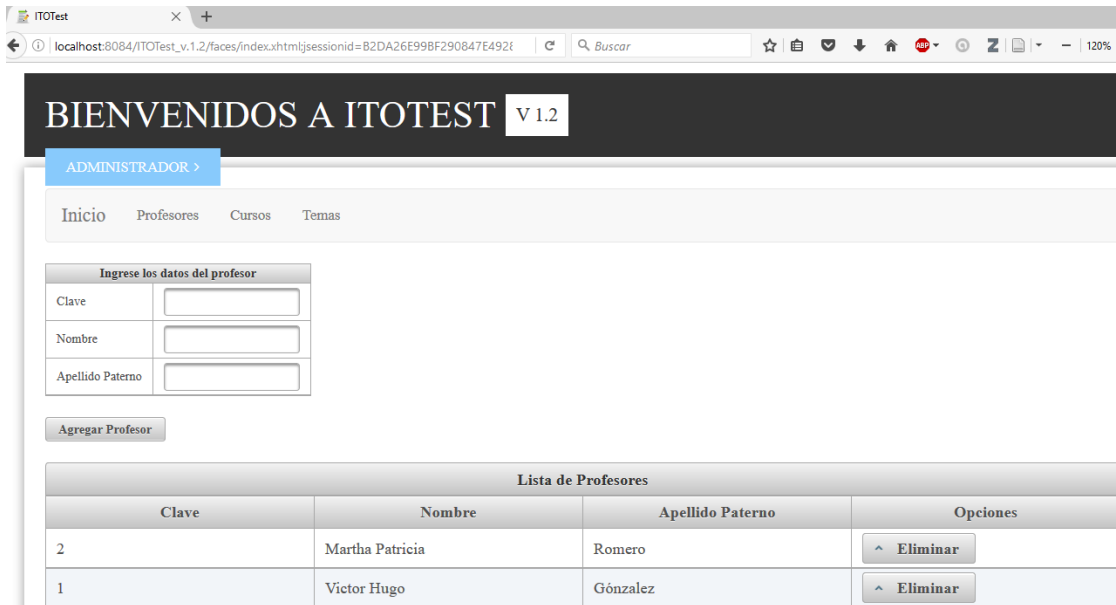


Fig. 3.22 Prototipo de interfaz para la gestión de profesores.



Fig. 3.23 Prototipo responsivo de interfaz para la gestión de profesores.

Para mostrar el listado de los profesores en una tabla se necesitó realizar de una consulta elaborada en *SPARQL*, con el operador *SELECT* permite escoger los atributos de instancia de un profesor y limitar los valores de retorno, al ejecutarse la consulta (línea 10-16) se almacenan los datos en un arreglo, el código necesario del método se encuentra en el archivo `Teacher.java` y se visualiza en el listado 3.5.

Listado 3.5 Método para buscar a todos los profesores.

```

1
2 public Teacher[] allTeachers() throws Exception{
3     Teacher[] arrTeacher = null;
4     ArrayList<ArrayList> rs=null;
5     ArrayList<Object> linea = null;
6     ArrayList<Teacher> arrTeachers = null;
7     Iterator it = null;
8     String sQuery = "";
9
10    sQuery="SELECT ?id ?name ?lastName ?secondLastName " +
11           "WHERE { ?x rdf:type itoTest:Teacher. " +
12           " ?x itoTest:hasTId ?id." +
13           " ?x itoTest:hasTName?name." +
14           " ?x itoTest:hasTLastName?lastName." +
15           " ?x itoTest:hasTSecondLastName ?secon LastName"+
16           "} ORDER BY ASC (?id)";
17
18    sQuery = prefix+query;
18    rs= oAO.ejecutarConsultaOnto(sQuery);
20
21    if (rs != null){
22        it = rs.iterator();
23        arrTeachers = new ArrayList<>();
24        while (it.hasNext()){
25            linea = (ArrayList<Object>)it.next();
26            Teacher oTeach = new Teacher();
27            oTeach.setIdTeacher((String) linea.get(0));
28            oTeach.setNameTeacher((String) linea.get(1));
28            oTeach.setLastNameTeacher((String) linea.get(2));
30            arrTeachers.add(oTeach);
31        }
32        arrTeacher =
33        arrTeachers.toArray(new Teach-er[arrTeachers.size()]);
34    }
35
36    return arrTeacher;
37
38 }

```

Una de las operaciones básicas para la gestión de la información es la de actualizar los datos de un profesor, el cual mediante el lenguaje de consultas de *SPARQL* se realiza una combinación de las operaciones *INSERT* y *DELETE*, dicho método se implementan en el archivo `Teacher.java` y permite llevar a cabo esta función. Para fines prácticos, el listado 3.6 muestra el código del método para eliminar un profesor de la ontología, la variable *sQuery* contiene la consulta para eliminar sólo al individuo de tipo *Teacher* (línea 14-16).

Listado 3.6 Método para eliminar a un profesor de la ontología.

```

1  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2
3  public void deleteTeacher() {
4      OntModel modOwl=null;
5      String sQuery="";
6      FileWriter out = null;
7      modOwl =
8          ModelFactory.createOntologyModel
9              (OntMoelSpec.OWL_MEM_MINI_RULE_INF);
10     modOwl.read("file:"+oAO.getNomArch(),"RDF/XML-ABBREV");
11
12     try{
13
14         sQuery=prefix+
15         "DELETE {itoTest:Teacher_ "+this.nId+" itoTest:hasTId ?x}"+
16         "WHERE {itoTest:Teacher_ "+this.nId+" itoTest:hasTId ?x}";
17
18         UpdateAction.parseExecute( sQuery, modOwl );
19
20         out = new FileWriter(oAO.getNomArch());
21         modOwl.write(new FileOutputStream(oAO.getNomArch()),
22             "RDF/XML-ABBREV", null);
23
24         if (out != null)
25             out.close();
26
27     }catch(Exception e){
28         e.printStackTrace();
29     }
30
31     modOwl.close();
32 }

```

La variable *sQuery* en el listado 3.7 visualiza la inserción de un individuo tipo *Teacher*, para ejecutar la consulta para insertar un nuevo profesor se necesita primero definir el tipo de individuo seguido de sus atributos.

Listado 3.7 Método para insertar un profesor a la ontología.

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2
3 public void insertTeacher() {
4     OntModel modOwl=null;
5     String sQuery="";
6     FileWriter out = null;
7
8     modOwl =
9         ModelFactory.createOntologyModel
10            (OntModelSpec.OWL_MEM_MINI_RULE_INF);
11
12     modOwl.read("file:"+oAO.getNomArch(),
13               "RDF/XML-ABBREV");
14     try{
15     sQuery=prefix+
16     "INSERT DATA {itoTest:Teacher_"+this.nId+
17     "rdf:type itoTest:Teacher}" ;
18
19         UpdateAction.parseExecute( sQuery, modOwl );
20
21     out = new FileWriter(oAO.getNomArch());
22     modOwl.write(new FileOutputStream(oAO.getNomArch()),
23                "RDF/XML-ABBREV", null);
24
25     if (out != null)
26         out.close();
27
28     }catch(Exception e){
29         e.printStackTrace();
30     }
31
32     modOwl.close();
33 }
```

Otra interfaz importante que se elaboró en esta fase fue para gestionar los cursos. La Fig. 3.24 muestra el formulario que solicita los campos de la clave, descripción y la selección del docente.

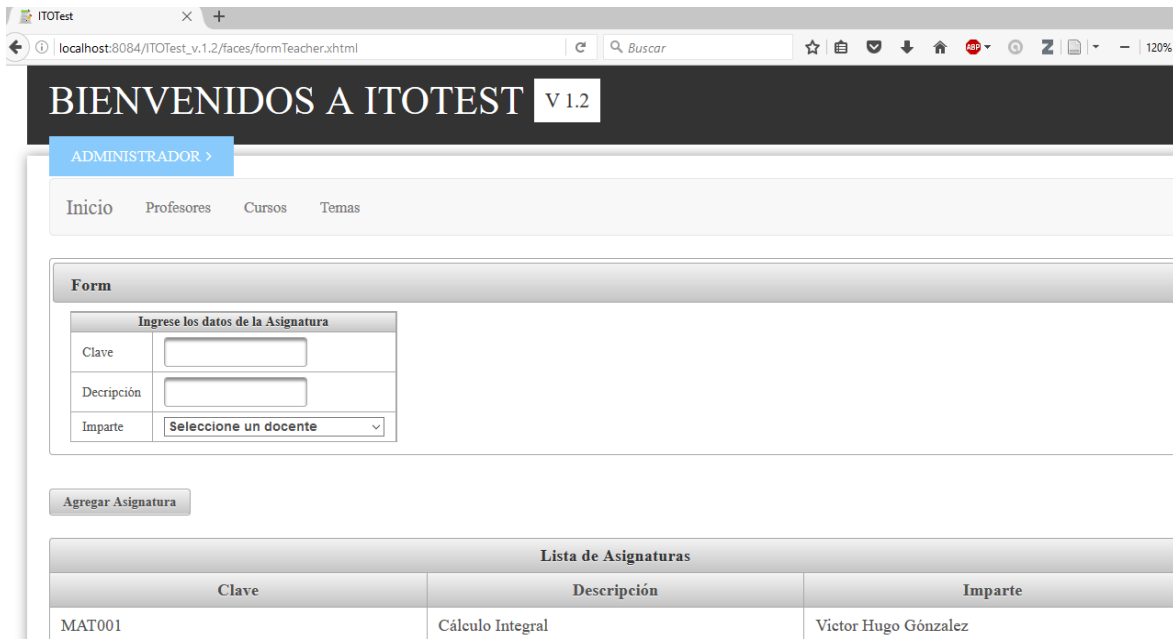


Fig. 3.24 Prototipo de interfaz para la gestión de curso.

La Fig. 3.25, muestra el prototipo de interfaz para la gestión de temas de cada curso.

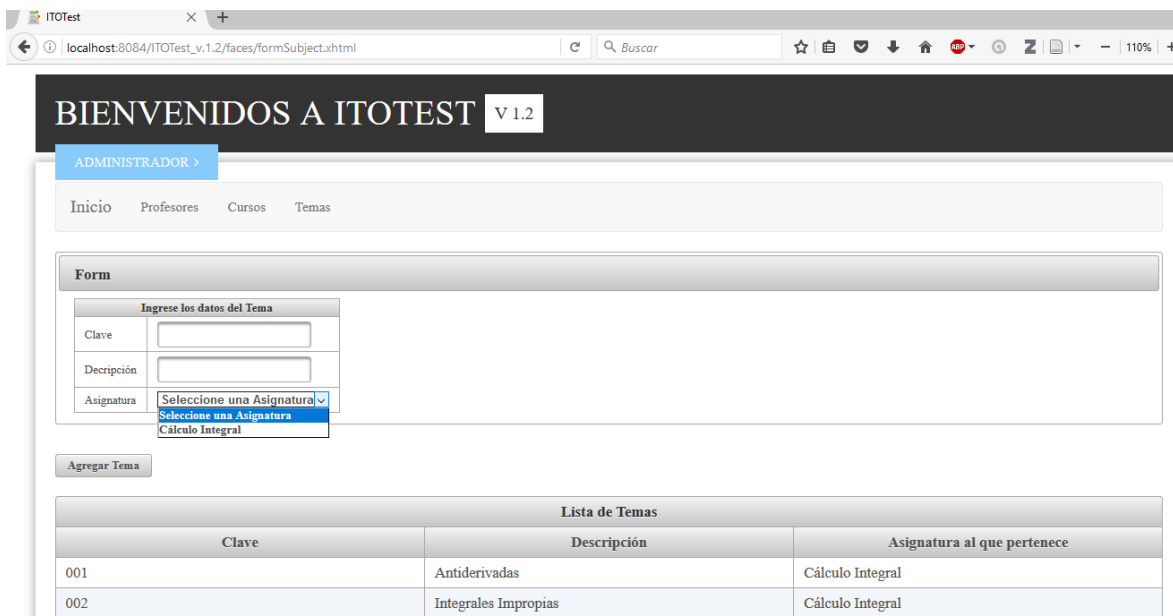


Fig. 3.25 Prototipo de interfaz para la gestión de temas.

.La Fig. 3.26 muestra el prototipo de interfaz que corresponde al formulario para agregar un reactivo, en donde se presentan los campos que se necesitan para almacenar una pregunta, como el nivel, quién lo elabora, a que tema corresponde y por lo consecuente a que curso. Esta interfaz le corresponde al rol de Profesor.

The screenshot shows a web browser window with the URL localhost:8084/ITOTest_v.1.2/faces/index.xhtml. The page title is 'BIENVENIDOS A ITOTEST V 1.2'. Below the title is a navigation menu with 'PROFESOR >' selected, and sub-menus for 'Inicio', 'Reactivos', 'Alumnos', and 'Evaluaciones'. The main content area is a form titled 'Ingrese los datos del Item'. The form has the following fields: 'Clave' (input with value '0'), 'Asignatura' (dropdown menu), 'Tema' (dropdown menu), 'Texto' (large text area), 'Respuesta Correcta' (input with value '0.0'), 'Nivel' (dropdown menu with value 'Fácil'), and 'Elabora' (dropdown menu with value 'Seleccione un docente'). A 'Agregar Reactivo' button is located at the bottom of the form.

Fig. 3.26 Formulario para almacenar un reactivo.

Este prototipo también contempla el listado de los reactivos, la Fig. 3.27 muestra el cómo se visualizará los datos que se almacenaron previamente de forma directa a la ontología.

| Lista de Reactivos | | | | | | | |
|--------------------|------------------|----------------------|---|--------------------|---------|-----------------|------------------------|
| Clave | Asignatura | Tema | Texto | Respuesta Correcta | Nivel | Veces Utilizado | Elaboró |
| 2 | Cálculo Integral | Integrales Impropias | ¿Cuál es el límite de la función $(x-2)/(x^2-4)$ cuando $x \rightarrow 2$? | 1.33 | Medio | 6 | Martha Patricia Romero |
| 1 | Cálculo Integral | Antiderivadas | ¿Cuál es el área debajo de la recta $y=x+4$? | 2.33 | Difícil | 4 | Victor Hugo González |

*Todos los derechos reservados I. Colohua.

Fig. 3.27 Interfaz del listado de reactivos.

Capítulo III. Aplicación de la metodología.

En la Fig. 3.28 se observa la interfaz que corresponde al formulario y listado para la gestión de alumnos en donde el profesor cuenta con la opción de guardar la información de sus estudiantes para que posteriormente se agreguen a determinado grupo.

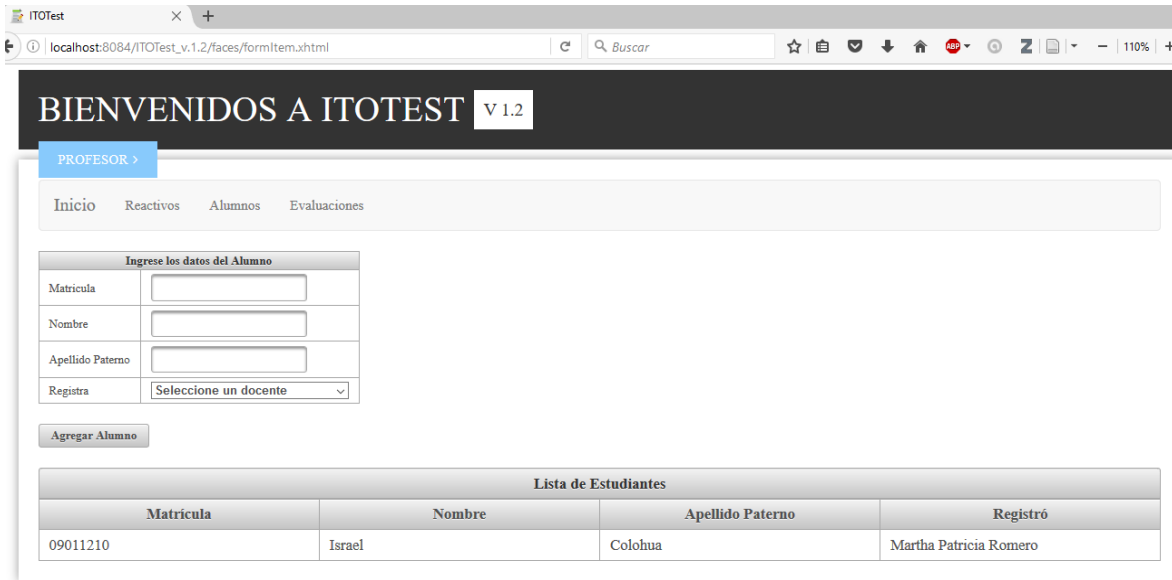


Fig. 3.28 Interfaz para la gestión de alumnos.

La Fig. 3.29 muestra el prototipo creado para la gestión de grupo

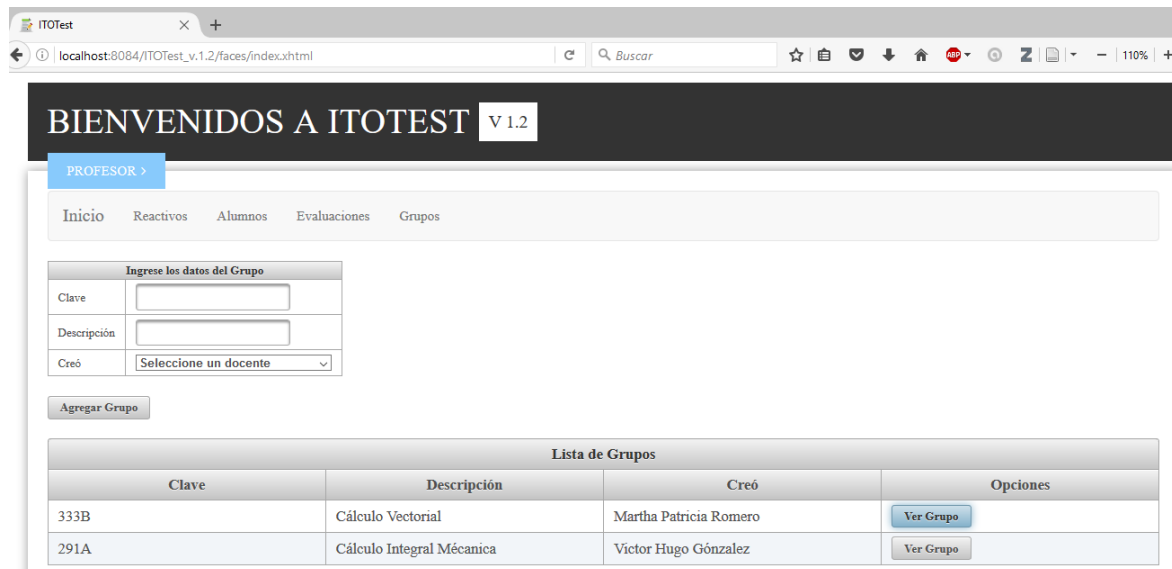


Fig. 3.29 Interfaz para la gestión de grupos.

En la Fig. 3.30 se visualiza la continuación de la interfaz del módulo para la gestión de grupos que permite agregar un alumno a un determinado grupo.

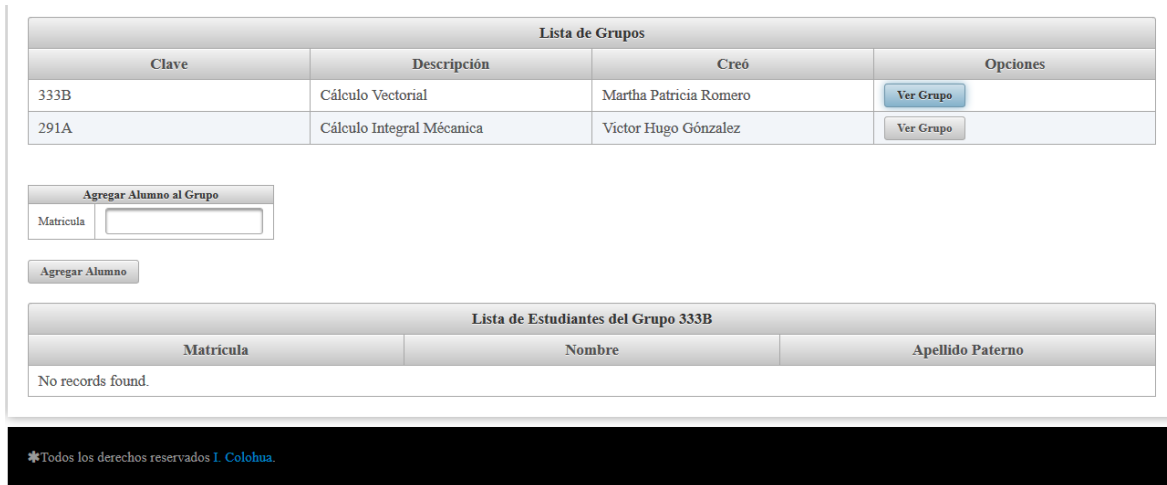


Fig. 3.30 Interfaz para la agregar un alumno a un grupo.

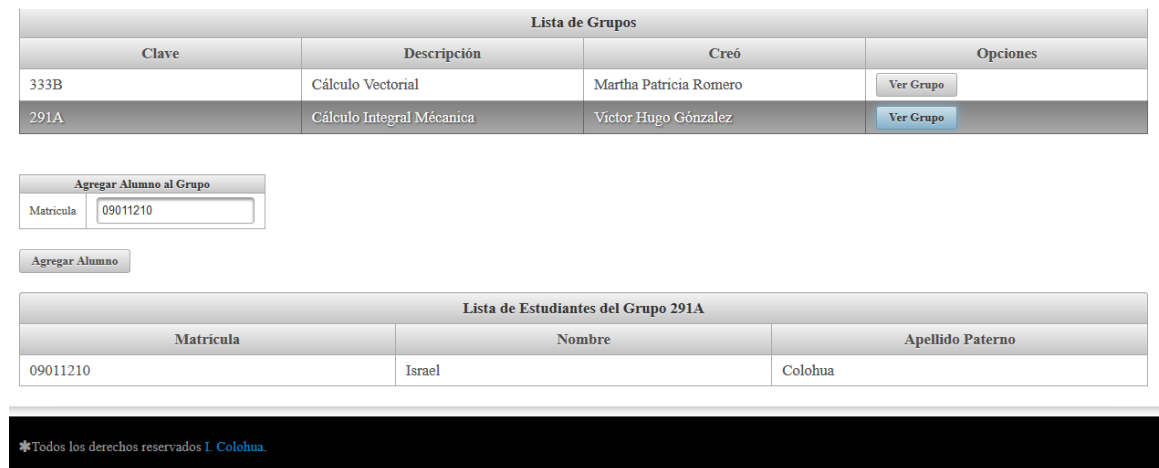


Fig. 3.31 Interfaz para agregar un alumno a un grupo.

Para finalizar las interfaces que le corresponden al rol de profesor, en la Fig. 3.32 se visualiza el formulario para crear un examen, en él, se solicita la fecha, hora de inicio y quien lo elabora.



Fig. 3.32 Interfaz para la gestión de evaluaciones.

En la Fig. 3.33 se muestra la continuación de la interfaz de la Fig. 3.32 que muestra el formulario para mostrar la agregación de un reactivo al examen.

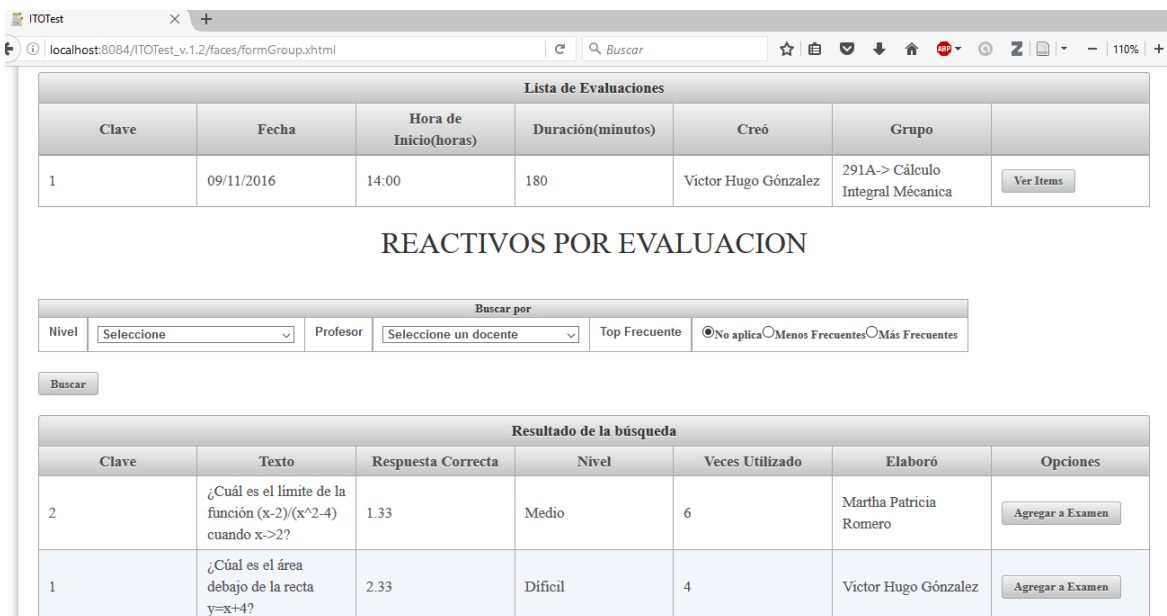


Fig. 3.33 Interfaz para agregar reactivos al examen.

Por último en la Fig. 3.34 se muestra la interfaz que se le presenta al alumno, la cual contiene las opciones de ver los exámenes disponibles y los resultados que ha obtenido.



Fig. 3.34 Interfaz del menú para el alumno.

3.6 Etapa de pruebas

En esta etapa el *Software* se pone a prueba para descubrir el máximo de errores posibles antes de su continuar mejorando prototipos y entregarle el producto final al cliente, como se trató de la generación de interfaces en la etapa anterior, las pruebas aplicadas para a estos elementos fueron:

1. Implementación de los conceptos de la ontología con las clases propuestas en *Java* (clases, *ManageBenas. etc.*).
2. Consistencia de datos de entrada.

Hasta el momento se aplicaron la pruebas que correspondían a las interfaces de usuarios del prototipo de la plataforma en su versión 1.2, la mejoras de la plataforma se presentaran mediante las Figuras de las interfaces en el siguiente capítulo.

3.7 Evaluación y aceptación del cliente

En esta fase de la metodología, el cliente aprueba la plataforma *Web*, en la cual posteriormente como parte del trabajo a futuro será entregado a los usuarios finales (administrador, alumnos y profesores)

Capítulo 4. Resultados

En este capítulo se describe el caso de estudio, en el cual abarca parte de la funcionalidad de los módulos y roles de usuarios. Los datos que se almacenarán serán del profesor Luis Ángel Reyes Hernández con clave 1050 cuya contraseña de acceso se le dará de “321”, el docente imparte el curso de Cálculo Diferencial al grupo “1g1a” cuyos temas se mencionan a continuación:

- Unidad I: Números reales.
- Unidad II: Funciones.
- Unidad III: Límites y Continuidad.
- Unidad IV: Derivadas.
- Unidad V: Aplicaciones de las Derivadas.

Para demostrar la parte responsiva de la plataforma *Web*, se accede de manera local mediante un dispositivo móvil para realizar la operación de eliminar los datos del docente “Lourdes Adbala Castillo”, después de esta acción, se procede a que el nuevo docente agregará al alumno Israel Colohua Cruz con la matrícula “09011210” al grupo que se le asigne y elaborará un reactivo para el tema de límites de la materia de Cálculo Diferencial.

4.1 Agregar Profesor

Para agregar al profesor se requiere que el administrador se firme dentro del sistema mediante la interfaz de la Fig. 4.1 al presionar sobre la imagen se muestra el formulario correspondiente al inicio de sesión, los campos que se necesitaban se remarcan de color rojo. Véase Fig. 4.2



Fig. 4.1 Interfaz para acceder al formulario de sesión.

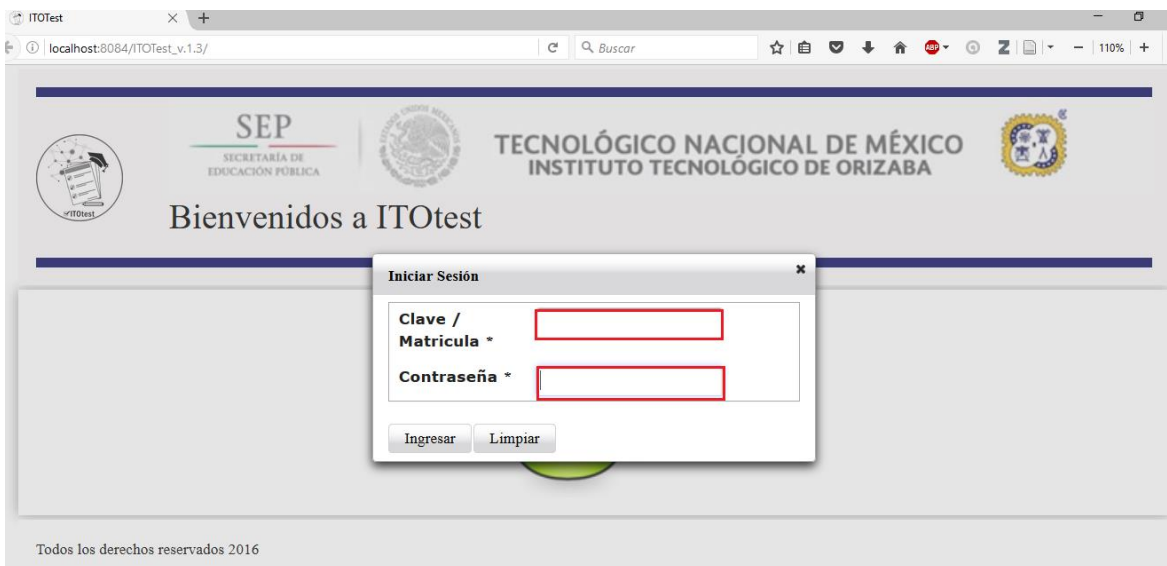


Fig. 4.2 Interfaz para iniciar sesión.

Si el administrador o cualquier usuario que ingrese su clave o matricula y/o contraseña incorrectamente, se muestra un mensaje de error como en la Fig. 4.3



Fig. 4.3 Interfaz para iniciar sesión.

Cuando los datos del administrador son correctos, se procede al redireccionamiento de la página principal, donde se localiza el menú, dentro del mismo se encuentra la opción “Lista de profesores” como se visualiza en la Fig. 4.4, cuando se presiona la imagen que representa el formulario para ingresar los datos del profesor como su nombre, apellidos, contraseña y la clave que se le asignó por parte de la institución, este valida que los campos que ingrese correspondan al tipo de elementos que se necesita, en caso contrario se muestran mensajes de error. Recordando que independientemente del tipo de información, todos los campos se consideran como obligatorios. Véase Fig. 4.4



Fig. 4.3 Menú para la gestión de profesores.

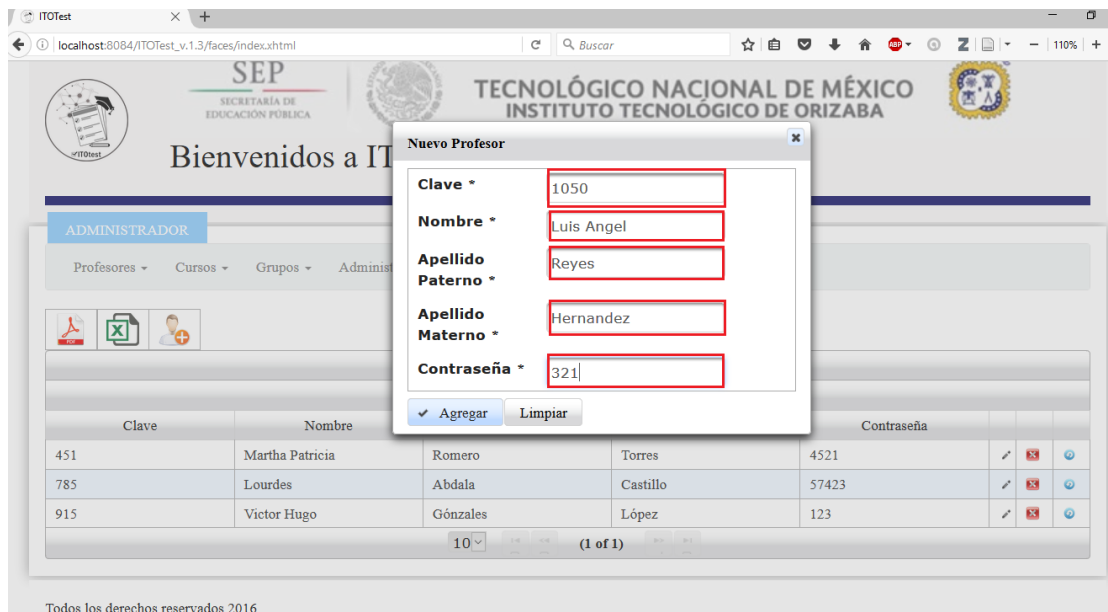


Fig. 4.4 Formulario para dar de alta un profesor.

Cuando se agrega exitosamente un nuevo profesor, el administrador visualiza el listado de todos los registros actuales, el cual mediante una tabla se organizan los datos de un docente, además para una mejor portabilidad de la información, se implementó el reporte de en PDF, La Fig. 4.5 muestra la interfaz que muestra las opciones de exportación (PDF y .csv) y la lista de profesores.



Fig. 4.5 Opciones de exportación de la lista de profesores.

Una vez que se seleccionó el formato del reporte, se permite visualizar la tabla de los registros actuales en un archivo *.pdf*. En ocasiones se facilita el uso de los dispositivos móviles para realiza alguna operación, como se mencionó con anterioridad la adaptación responsiva permite que se logró una vista de la plataforma *Web*. La Fig. 4.6 representa la interfaz para mostrar la lista de los profesores y como se trata de una pantalla de menor resolución, la información se recortó dando lugar a la visualización de los datos con mayor prioridad, para este caso se elige la clave y el (los) nombre(s) de los profesores y con la operación de eliminar, cabe señalar que se necesita acceder de manera local a la plataforma mediante una conexión a un red, por lo tanto al ingresar a la IP `192.168.1.65: 8084/ITOTest_v.1.3/` desde el dispositivo se obtiene la interfaz responsiva. La Fig. 4.7 muestra el diálogo que permite dar de baja al profesor “*Lourdes Adbala Castillo*” mediante la interfaz adaptada.



Fig. 4.6 Interfaz responsiva para la gestión de profesores.

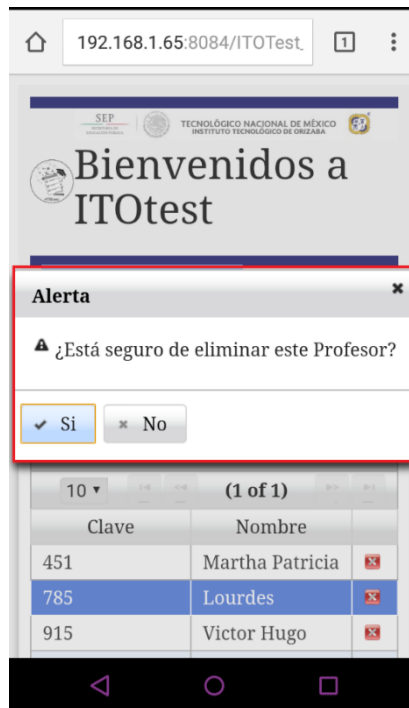


Fig. 4.7 Interfaz responsiva para eliminar un docente.

Conforme a la resolución del dispositivo, se van mostrando los demás elementos. La Fig. 4.8 muestra la interfaz responsiva de la lista de profesores donde se visualiza el apellido paterno de los profesores.



Fig. 4.8 Interfaz responsiva que muestra otro atributo de profesor.

4.2 Agregar curso y temas

El registro de un nuevo una nueva materia con sus respectivos temas conlleva a la gestión de cursos, las materias técnico-científicas abarcan asignaturas como física, química y matemáticas, por lo tanto este módulo permite la administración de dichas materias, el administrador necesitó de estar firmado en el sistema para acceder a esta sección, una vez dentro de su sesión al presionar “Lista de cursos” se muestra el formulario correspondiente como se visualiza en la Fig. 4 .9.



Fig. 4.9 Interfaz para gestionar cursos y temas.

El administrador cuenta con un formulario de ingreso de datos, el cual mediante un diálogo se le solicita el nombre del curso y la descripción de la temas, cabe mencionar que para que se identifique el número de cursos, se les asignó un clave consecutiva de manera implícita, esto se logra mediante el uso de una consulta que regresa el ultimo valor de la clave de la última asignatura registrada. La Fig. 4.10 muestra el formulario para un nuevo curso, el recuadro marcado con el número uno indica que se agregan las filas necesarias para escribir los temas y el recuadro marcado con el número dos elimina la fila seleccionada.

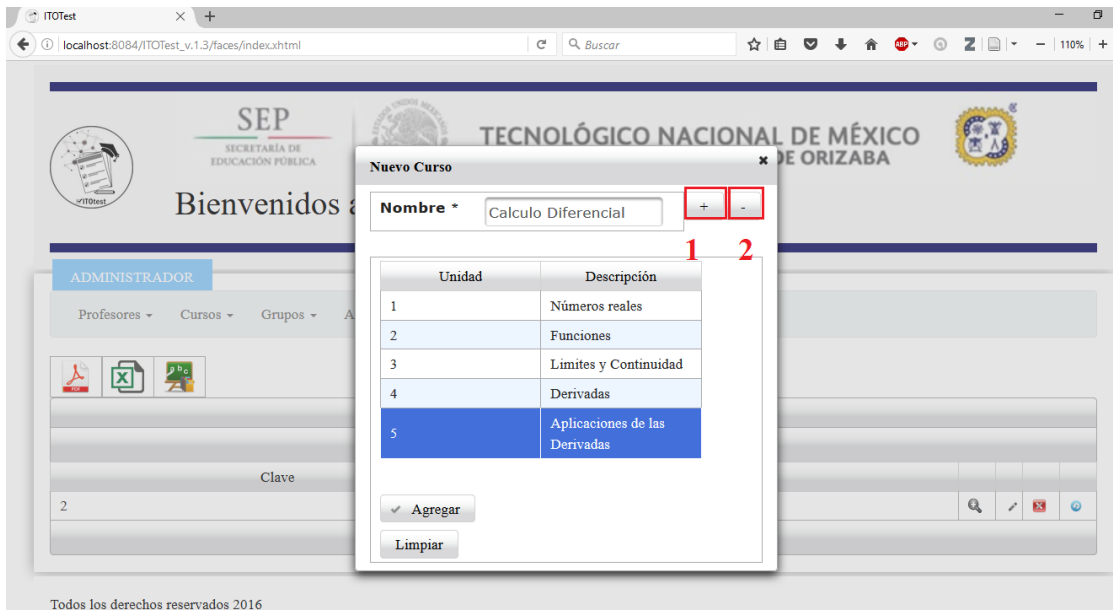


Fig. 4.10 Interfaz para dar de alta un curso.

Si en determinado momento se necesita saber los temas de un curso en particular, sólo al presionar el botón de “Ver Temas” se visualiza la descripción de los mismos mediante un diálogo y de igual manera se permite la exportación de todos los cursos que se imparten en el ITO. Fig. 4.11.



Fig. 4.11 Interfaz para visualizar los temas de un curso.

4.3 Asignar cursos

El administrador gestiona la asignación de grupos que se imparten en el tecnológico, los cuales están formados por un clave de cuatro dígitos y estos a la vez son impartidos en los periodos de ENE-JUN o AGO-DIC. En esta sección el usuario asigna los diferentes grupos de las materias técnico-científicas a los docentes registrados, para el presente caso se le asigna el grupo “1g1a” que corresponde al curso de Cálculo Diferencial de la carrera de “Ingeniera en Sistemas Computacionales” y lo impartirá el docente Luis Ángel Reyes Hernández en el periodo ENE-JUN de 2017, La Fig. 4.12 presenta la opción del menú para ingresar a la gestión de cursos.



Fig. 4.12 Interfaz para visualizar la gestión de cursos.

La Fig.4.13 muestra el formulario correspondiente donde se le solicita al administrador la información necesaria para una nueva asignación de curso, se recuerda que todos los campos son obligatorios y para una mejor selección del año se visualiza una lista con los mismos para evitar errores de escritura o rangos no permitidos para los cursos. La Fig. 4.14 representa el formulario con los datos ingresados por el administrador.



Fig. 4.13 Interfaz que visualiza que los campos son requeridos.

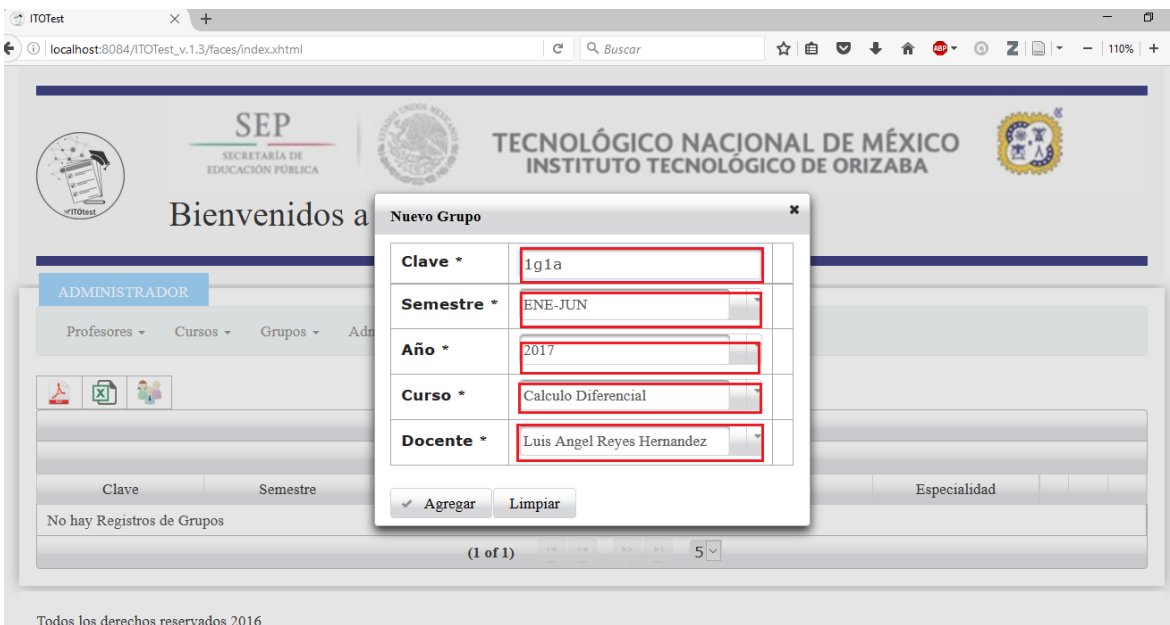


Fig. 4.14 Interfaz que visualiza el formulario con los datos requeridos.

En caso de que exista alguna modificación en el curso, la plataforma se limita a que sólo se permita actualizar el docente que imparte dicho curso, porque de ser modificada toda la información (clave, año y periodo) se tendría un descontrol, por lo tanto se consideró de esa forma. La Fig. 4.15 muestra la interfaz con la opción señalada mediante una flecha para editar al profesor asignado.



Fig. 4.15 Interfaz para editar el profesor asignado.

4.4 Consultar cursos asignados

Esta sección le pertenece al rol de profesor, una vez que se firma dentro del sistema, se le muestra una interfaz con un menú diferente al administrador, este usuario cuenta con la opción para consultar los grupos que se le asignaron, así mismo mediante la opción de "Ver alumnos" logra visualizar al número de estudiantes y su información personal como matricula, apellidos y la clave que se le asigna al momento de darlo de alta. La Fig. 4.17 muestra la interfaz de dicho caso de uso y la Figura 4.18 visualiza un diálogo con los datos de los estudiantes asignados al grupo "1g1a".



Fig. 4.16 Interfaz para consultar los grupos asignados.



Fig. 4.19 Lista de alumnos del grupo “1g1a”.

4.4 Registrar un alumno

Dentro de las funciones que se le permiten al profesor, la gestión de sus alumnos es una de ellas, mediante la opción que se muestra en la Fig. 4,20 entra a la pantalla donde se encuentra el listado y la opción para mostrar el formulario e ingresa los datos personales de un nuevo alumno. Como nota cada profesor llevará el control de sus estudiantes.



Fig. 4.20 Interfaz para la gestión de alumnos.

Una vez presionado la opción del menú anterior, al docente se le presenta la opción para llenar un nuevo formulario con los datos de un alumno. La Fig. 4.21 muestra la interfaz que corresponde al llenado de los datos del alumno “Israel Colohua Cruz” cuyo número de control y contraseña se guarda como “09011210”.



Fig. 4.21 Interfaz del formulario para un nuevo alumno.

Una vez que se ingresa la información solicitada, el profesor cuenta con las opciones de modificar y eliminar un alumno si lo desea, al presionar la opción de “Edición” sobre la fila de la tabla donde se encuentra el alumno, el sistema le permite modificar los datos del alumno desde

el nombre, apellidos, contraseña y grupo, después de finalizar la edición, el docente debe presionar “Terminar Edición” para que posteriormente se guarden los cambios en la ontología especializada. Fig. 4.22.

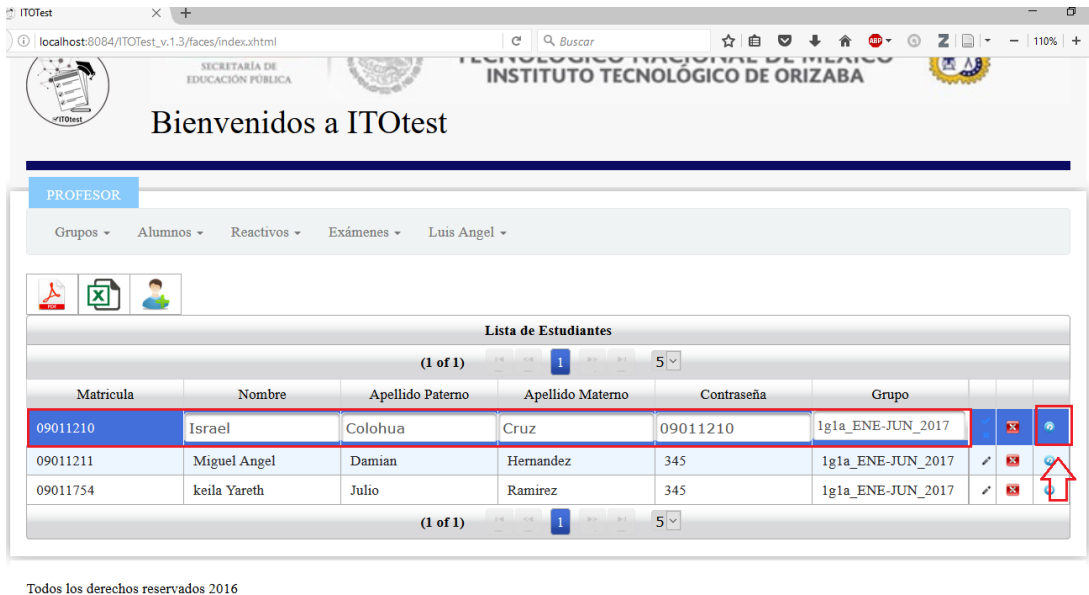


Fig. 4.22 Edición de la información de un alumno.

De igual manera, el profesor tiene la opción de eliminar la información de un alumno mediante el botón “eliminar”, el cual muestra un diálogo que pregunta la confirmación de dicha operación. Fig. 4.23.



Fig. 4.23 Confirmación eliminar la información de un alumno.

4.5 Registrar un reactivo

Cada profesor del área de ciencias básicas cuenta con la opción de llenar el banco de reactivos para sus evaluaciones, cabe mencionar que el reactivo registrado podrá reutilizarse posteriormente en otros exámenes que elabore o que otro docente necesite ocuparlo, La Fig. 4.24 muestra la opción para ingresar a la gestión de reactivos.



Fig. 4.24 Opción para ingresar a la gestión de reactivos.

La Fig. 4.25 visualiza el formulario correspondiente para almacenar un reactivo, los datos que se solicitan es: el texto de la pregunta, la respuesta, el nivel de dificultad, a que tema corresponde y una imagen por si se necesita de tal manera que implícitamente se almacena la clave del profesor que lo elaboró para que al momento de aplicar los filtros de búsqueda, el parámetro de “buscar por maestro” se permita. Otro aspecto que se visualiza en la Fig. 4.26 se trata de que no se satura el listado de las preguntas, se cuenta con la opción “Ver Imagen” que permite abrir un diálogo para mostrar la imagen que se almacenó al momento de crear el reactivo.

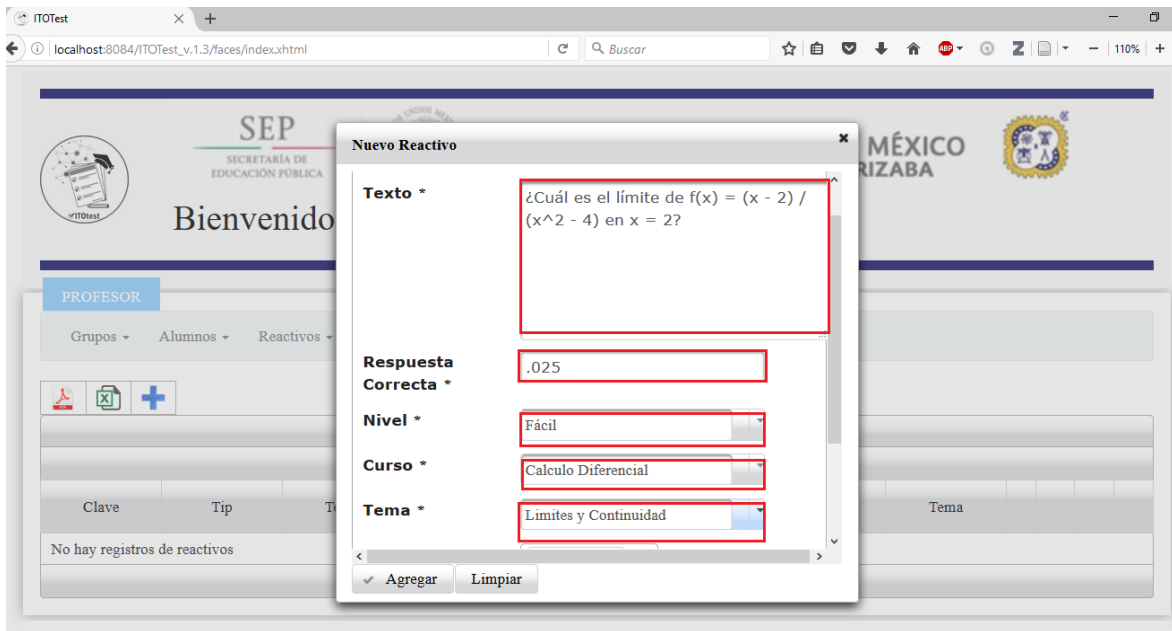


Fig. 4.25 Formulario para agregar un nuevo reactivo.

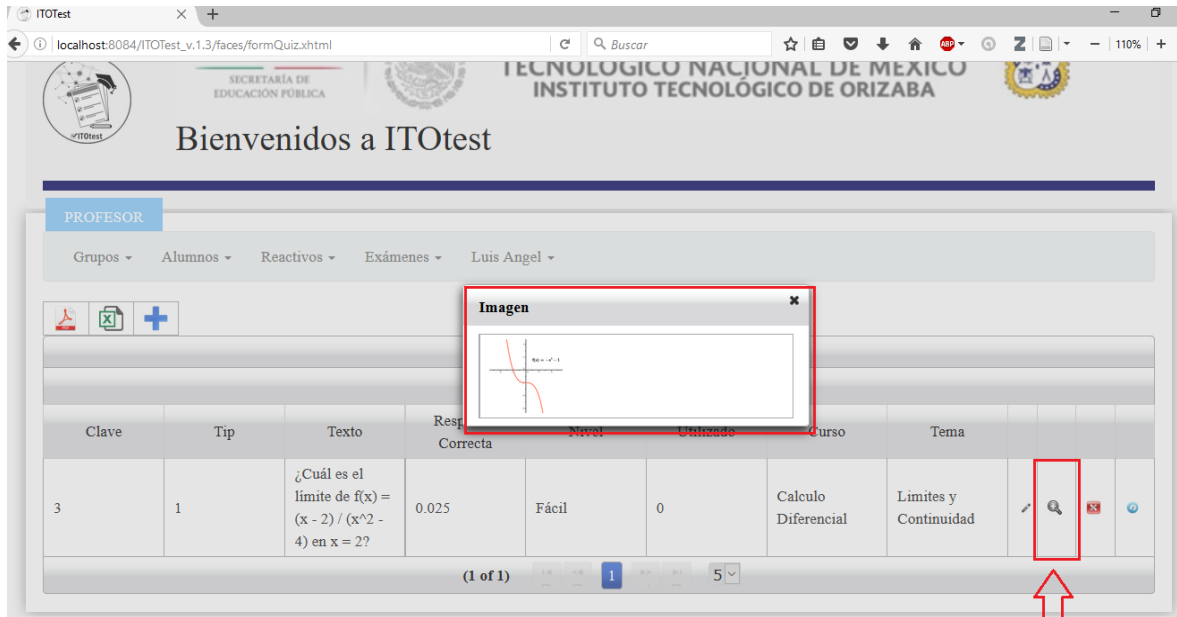


Fig. 4.26 Diálogo para mostrar la imagen del reactivo.

Capítulo 5. Conclusiones y trabajos a futuro

En este capítulo se presentan las conclusiones del trabajo de tesis, así como los posibles trabajos a futuro con respecto a la plataforma *Web*.

5.1 Conclusiones

En la actualidad, los contenidos, materiales y cursos que se imparten en línea van dirigidos a los estudiantes a través de plataformas de *E-learning*, comúnmente se utilizan las aplicaciones *Web* que aplican herramientas y servicios propios de Internet. A pesar de que existen diversas plataformas con las características antes mencionadas, una aplicación dotada de *Web* semántica se considera una de las mejores alternativas para implementar el aprendizaje en línea, ya que el uso de las ontologías tiene un gran potencial para este dominio porque una de sus principales características permite inferir conocimiento a partir de las definiciones dadas, es por eso que el desarrollo de la ontología presentada con base a los principios del *E-learning* facilita el proceso de evaluación de las asignaturas técnico-científicas en el ITO.

La reutilización de reactivos de determinado profesor permite enriquecer aún más una evaluación además de alimentar el banco de preguntas, el examen toma una importancia mayor porque cada elemento que lo forma cuenta con un nivel que el docente considera. Otro aspecto importante se vio reflejado en la implementación responsiva de páginas *Web*, en la actualidad existen diversos dispositivos que permiten la comunicación con *Internet* y se logra adaptar dichos elementos a diversas resoluciones, el manejo de esta técnica permitió que la plataforma desarrollada se adapte para mostrar su contenido a diversos dispositivos.

5.2 Trabajos a futuro

Esta plataforma gestiona profesores, cursos, alumnos y exámenes de las diferentes carreras ofertadas en el ITO, a pesar de que se consideran aspectos de las respuestas numéricas, se recomienda expandir el módulo de reactivos, por ejemplo, que una pregunta permita subir un archivo a la plataforma y se califique posteriormente por el docente.

Otro elemento a considerarse y necesario para la plataforma desarrollada consiste en el manejo de la concurrencia de los datos en la ontología, por lo cual se recomienda investigar sobre la API Jena para el control de la persistencia de los datos o en su defecto utilizar un servidor de tripletas como Virtuoso.

Glosario de Términos

API (*Application Programming Interface*, Interfaz de Programación de Aplicaciones), es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizada por otro *Software* como una capa de abstracción.

E-learning (Aprendizaje electrónico) hace referencia a la educación a distancia a través de sistemas de aprendizaje en línea utilizando para ello, herramientas o aplicaciones digitales para soporte a los procesos de enseñanza y aprendizaje.

LMS (*Learning Management System*, Sistema de Gestión de Aprendizaje), es un *Software* para administrar, distribuir y controlar las actividades de *E-learning*.

MCQ (*Multiple Choice Question*, Preguntas de Selección Múltiple), es una forma de evaluación que consta de una pregunta con cierto número de respuestas, de las cuales, una respuesta es correcta y las demás respuestas son incorrectas y se llaman distractores.

Moodle es una plataforma de aprendizaje diseñada para proporcionar a educadores, administradores y estudiantes un sistema integrado único, robusto y seguro para crear ambientes de aprendizaje personalizados.

Ontología hace referencia a la formulación de un exhaustivo y riguroso esquema conceptual dentro de uno o varios dominios dados.

OWL (*Web Ontology Language*), un lenguaje de marcado para publicar y compartir datos y tiene como objetivo facilitar un modelo de marcado construido sobre *RDF* y *XML*.

TIC (Tecnologías de la información y la comunicación) son el conjunto de tecnologías desarrolladas para gestionar información.

Productos académicos

- I. Colohua Cruz, L. A. Reyes Hernández, H. Muñoz Contreras, B. A. Olivares Zepahua, U. Juárez Martínez , “*Diseño de una Plataforma Web E-learning para la elaboración y evaluación de reactivos empleando distractores Shuffle Question/Option para asignaturas técnico-científicas*”, (9 de septiembre de 2016) , Cuernavaca, Morelos.

- II. I. Colohua Cruz, L. A. Reyes Hernández, G. Hernández Chan, J. L Sánchez-Cervantes. “*Diseño de una ontología para el proceso de evaluación de las asignaturas técnico-científicas del Instituto Tecnológico de Orizaba*”, (18 de noviembre de 2016), Apizaco, Tlaxcala.

Artefactos

ITOTest_v3.owl

ITOTest_v.1.3.war

Tesis.vpp

Bibliografía

- [1] Definición.de, «Definición.de,» 2017. [En línea]. Available: <http://definicion.de/lenguaje-de-programacion/>. [Último acceso: 06 01 2017].
- [2] J. J.Gutiérrez, «ISI,» [En línea]. Available: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf. [Último acceso: 28 08 2015].
- [3] N. Guarino, «What Is an Ontology?,» 2010.
- [4] S. E. S. López, «Ontologías y su representación jerárquica,» Cholula,Puebla, 2007.
- [5] J. J. J. B. J. A. Tabares García, «Ontología para el proceso evaluativo en la educación superior,» *Revista Virtual Universidad Católica del Norte*, nº 68-79, 2014.
- [6] w3c, «<http://www.w3.org>,» w3c, 2001. [En línea]. Available: http://www.w3.org/2001/sw/Europe/reports/dev_workshop_report_8/maria/slide19-0.html. [Último acceso: 28 08 2015].
- [7] O. M. F.-L. Corcho, «Construcción de ontologías legales con la metodología METHONTOLOGY y la herramienta WebODE,» *Intelligent Software Components*.
- [8] CCM, «CMS (sistema de gestión de contenido),» CCM, [En línea]. Available: <http://es.ccm.net/contents/825-cms-sistema-de-gestion-de-contenido>. [Último acceso: 28 08 2015].
- [9] Moodle, «Moodle.org,» Moodle.org, [En línea]. Available: <https://moodle.org/mod/forum/discuss.php?d=66743>. [Último acceso: 28 08 2015].
- [10] R. D. L. Antonio, «Entorno Web para la generación de reactivos en la evaluación de aplicación de derivadas,» Orizaba,Veracruz, 2014, p. 93.
- [11] A. Jurado-Núñez, «Distractores en preguntas de opción múltiple para estudiantes de medicina: ¿cuál es su comportamiento en un examen sumativo,» vol. 2, nº 2013.
- [12] kent, «www.kent.ac.uk,» kent, [En línea]. Available: <https://www.kent.ac.uk/elearning/files/moodle/moodle-quizzes.pdf>. [Último acceso: 28 08 2015].
- [13] R. B. Álvarez, «EL E-LEARNING, UNA RESPUESTA EDUCATIVA A LAS DEMANDAS DE LAS SOCIEDADES DEL SIGLO XXI,» *Pixel-Bit*, nº 35, pp. 87 - 96, 2009.

- [14] A. N. d. V. Rodríguez, «Postgrado.info.unlp.edu.ar,» 2009. [En línea]. Available: http://postgrado.info.unlp.edu.ar/Carreras/Especializaciones/Tecnologia_Informatica_Aplicada_en_Educacion/Trabajos_Finales/Rodriguez_Ana.pdf. [Último acceso: 10 02 2017].
- [15] W3C, «W3C,» 2014. [En línea]. Available: <http://www.w3.org/TR/rdf11-primer/> .
- [16] W3C, «W3C,» 2009. [En línea]. Available: <https://www.w3.org/2009/Talks/0615-qbe/>.
- [17] J. Cabero, «Bases pedagógicas del e-learning,» *Revista de Universidad y Sociedad del Conocimiento.*, 2006.
- [18] Nuria Vallejo Acebal, «Metodología De Elaboración De Materiales Didácticos Multimedia Accesibles,» *Fundación Andaluza Fondo de Formación y Empleo.*
- [19] M. y. A. J. AREA, «e-Learning: Enseñar y Aprender en Espacios Virtuales,» *Tecnología Educativa. La formación del profesorado en la era de Internet*, nº . (2009), Tecnología Educativa. . Aljibe, Málaga, pp. 391-424, 2009.
- [20] M. E. A. Obeso, «Metodología de Medición y Evaluación de la Usabilidad en Sitios Web Educativos».
- [21] S. Clara, «Propuesta de ejercicios interactivos para la autoevaluación del aprendizaje en la asignatura Medicina General Integral,» *EduMeCentro*, 2013.
- [22] A. L. Ruiz, «QUIZ LIBRE: Software de entretenimiento basado en un juego de preguntas y respuestas para plataforma Nintendo DS,» 2012.
- [23] L. C. Turienzo, «Wit Quiz Plataforma de aprendizaje sobre dispositivos móviles,» nº Universitat Oberta de Catalunya, Barcelona., 2010.
- [24] R. Eixarch, «WIRIS Quizzes, matemáticas para cuestionarios Moodle,» 2010.
- [25] C. U. d. C. E. e. I. U. D. M. Instrumental, «Manual Básico de WIRIS para Moodle 2,» 2012.
- [26] I. Akinsanmi, «Development of an E- Assessment Platform for Nigerian Universities,» vol. 2, nº 2, pp. 170-175, September 22, 2010.
- [27] L. D. M. ROBERTO BARCHINO, «Assessment Design: A Step Towards Interoperability,» April 2009.
- [28] Lucila Romero1, «EAOnto A Heavyweight Ontology for Supporting eAssessment Generation,» 2012.

- [29] B. Z. itko, «Dynamic test generation over ontology-based knowledge representation in authoring shell,» 2009.
- [30] John Dermo, «In search of Osiris random item selection fairness and defensibility in high stakes e-assessment,» 2010.
- [31] M. Foulonneau, «Generating educational assesment items from Linked Open Data the case of DBPEdia,» 2011.
- [32] F. A. V. Garzón, «Generation of Random Questions with Multiple Choice Single Answer for Moodle,» 2011.
- [33] J. L. Reguera1, « Una plataforma de evaluación automática con una metodología efectiva para la enseñanza aprendizaje en programación,» 2010.
- [34] F. Dalipi, «On Integration of Ontologies into E-Learning Systems,» 2014.
- [35] J. M. a. H. Jang, «Generating diagnostic multiple choice comprehension cloze questions,» pp. 136-146, 2012.
- [36] T. M. a. J. X. Wang, «A language learning support system using course-centered ontology and its evaluation,» pp. 278-293, 2014.
- [37] J. J. & J. B. J. Tabares García, «Ontología para el proceso evaluativo en la educación superior,» *vista Virtual Universidad Católica del Norte*, pp. 64-79, 2014.
- [38] Rodriguez-Gonzalez, «Towards an Ontology to Support Semantics Enabled Diagnostic Decision Support Sys-tems,» *Current Bioinformatics* 7, pp. 245-450, 2012.
- [39] B. Foster, «A completely client-side approach to e-assessment and e-learning of mathematics and statistics,» *International Journal of e-Assessment*, vol. 2, nº 2, 2012.
- [40] O. F. L. M. G.-P. A. L.-C. Corcho, «Construcción de ontologías legales con la metodología METHONTOLOGY y la herramienta WebODE,» 2005, .