

Benemérita Universidad Autónoma de Puebla.



Facultad de Ciencias de la Computación.

Diseño y desarrollo de un sistema de gestión educativo para la
formación a nivel superior

Tesis presentada como requisito para obtener el título de:

Licenciatura en Ingeniería en Ciencias de la Computación.

Presenta:

Javier Valderrábano Gutiérrez

Asesores :

Dra. Josefina Guerrero García.

Dr. Juan Manuel González Calleros.

Mayo de 2015

Agradecimientos

Después de tanto tiempo que hemos batallado, reído, sufrido, llorado y peleado, hemos llegado a este magnífico momento en el cual terminamos otro capítulo de esta historia juntos.

En primer lugar quiero agradecerle a un amigo, que nunca me ha abandonado, que ha estado siempre a mi lado hasta en momentos en los que he dudado de Él, quiero agradecerle a ti, Dios, porque solo nosotros dos sabemos absolutamente todas las cosas por las que hemos pasado juntos y que gracias a ti hemos podido alcanzar este logro más, nos hemos peleado, distanciado y demás pero al final comprendí el porqué de las cosas y aunque en ocasiones las cuestionó me has demostrado que todo lo haces por una razón, sin ti y sin tu ayuda, nada de esto hubiera sido posible. Gracias Dios.

A una estrella muy grande en el cielo, que se fue, horas después de haberse enterado mi ingreso a la universidad, sí, el padre de mi padre, mi abuelito Toño, que aunque no se encuentra aquí físicamente sé que me ha estado ayudando desde lo más alto del cielo y que nunca me ha desamparado. Gracias Abue, y recuerda que todavía tenemos una cita pendiente.

A dos personas con las que he convivido mis 24 casi 25 años de vida, que han estado allí como un gran soporte en estos años que he vivido, a mis papás, sé que el camino fue largo y difícil, pero les agradezco todo lo que han hecho por mí; sus sacrificios, lucha y entrega, tuvimos momentos en los que parecía que esto era interminable pero que hoy les puedo decir que hemos llegado a la meta. Papá, Mamá, gracias, en verdad, muchas gracias por todo lo que me han dado.

A dos personas que sin ellas, mis padres no hubieran nacido, a mi abuelita Celia y a mi abuelita lolita, pilar fundamental de valores y alegrías que realmente un nieto se merece, agradezco a Dios el que me haya dejado ser su nieto. Gracias por tanto cariño abues.

A mis tíos y tías, primos y primas que han estado siempre al pendiente de mí, apoyando a mis padres y dándome palabras de motivación para poder seguir con este camino llamado Ingeniería en Ciencias de la Computación, han sido de gran apoyo para mí y les agradezco todas las atenciones que han tenido conmigo.

A una pareja muy exitosa, que demuestran pasión dedicación y entrega por lo que hacen, que desde el momento que me comentaron de ellos, me lleno de mucha emoción querer trabajar con ambos, exactamente, hablo de mis profesores asesores de tesis, la Dra. Josefina Guerrero García y el Dr. Juan Manuel González Calleros, no sé cómo agradecerles el haberme tenido tanta paciencia en este proyecto y en haber confiado en mí para el desarrollo de este, honestamente, no dudo ni un poquito en la gran capacidad que tiene ambos y me ha dado muchísimo gusto trabajar con ellos porque siempre me ha gustado trabajar con los mejores. Gracias profes gracias por todo el apoyo y las palabras de aliento y motivación que me han dado en este tiempo.

A todos mis amigos ya que gracias a ellos, estos años en la carrera no han sido tan pesados como pensé que lo serían, por sus momentos de consuelo, apoyo y consejos, por las risas interminables que me sacaban en momentos de dolor, por sus juegos y absolutamente todas las cosas que han hecho por mí, a esos amigos que me han ayudado a no fallar en la tesis, que me han enseñado tantas cosas como por ejemplo a programar y demás, agradezco a Dios me los haya puesto en mi camino y no tengo otra manera de agradecerles más que apoyándolos en lo que pueda ayudarles, o aconsejarles, muchas gracias amigos, gracias por ser esa familia que uno escoge. Los estimo mucho.

A todas aquellas personas que no están aquí, por motivos de la vida, ya sea que se hayan distanciado u otras situaciones, porque gracias a ellas pude ver mis virtudes y fallas, en su momento estuvieron allí y me ayudaron a crecer como persona y también de manera intelectual, de unas ya no pude despedirme, de otras espero que Dios y la vida nos pueda volver a juntar para poder así agradecerles todo esto que han hecho por mí.

Este capítulo ha concluido y quiero agradecerles a todos su esfuerzo dedicación y tiempo hacia mi persona y antes de empezar a exponerles mi proyecto quiero decirles algo, no les puedo decir que lo logré, sino más bien:

¡LO LOGRAMOS!

Contenido

Agradecimientos	3
Capítulo 1: Introducción.....	7
Capítulo 2: Estado del Arte.....	9
2.1 Contexto de la educación en México	10
2.2 Educación a distancia	11
2.3 Una nueva Innovación dentro de las TIC: El e-Learning.....	13
2.3.1 ¿Qué es el e-Learning?.....	13
2.3.2 Plataformas dentro del E-Learning	14
2.4 Sistemas Gestores de Objetos de Aprendizaje.....	16
2.5 Objetos de Aprendizaje (OA's)	22
2.5.1 Propiedades y características de los OA's.....	23
2.5.2 Estándares utilizados dentro de los OA's.....	29
2.6 Objetos de aprendizaje como medio educativo	30
2.6.1 Diseño de Interfaces y OA atractivos mediante interacción seductiva.....	30
2.6.2 Sistema de puntuación "ELO"	34
2.7 Redes sociales como agentes de difusión masiva.....	37
2.8 En resumen.....	41
Capítulo 3: Modelado Conceptual	43
3.1 Metodología	43
3.2 Diagrama de Clases	46
3.3 Diagrama de Casos de Uso	50
3.4 Modelo Relacional.....	57
3.4.1 Diccionario de Datos	58
3.5 En Resumen.....	64
Capítulo 4: Diseño	65
4.1 Herramientas tecnológicas para la creación del gestor	65
4.1.1 Patrón de diseño MVC.....	65
4.1.2 Java Server Pages (JSP).....	68
4.1.3 Java Servlets	69
4.1.4 HTML	72
4.1.5 MYSQL	72
4.1.6 CSS.....	73

4.1.7 Lenguaje de Expresiones (LE)	73
4.1.8 JSTL	74
4.1.9 JDBC.....	74
4.1.10 Apache Tomcat.....	75
4.1.11 Java Script.....	76
4.1.12 Bootstrap.....	77
4.1.13 WebSockets.....	78
4.1.14 JQuery	80
4.1.15 JSR356 (Java)	81
4.2 Diagramas de secuencia	83
4.3 Desarrollo	85
4.3.1 Prototipos realizados.....	86
4.3.2 Diagramas de secuencia y sitio web desarrollado	93
4.4 En Resumen.....	118
Capítulo 5: Validación.....	119
En Resumen.....	121
Capítulo 6: Conclusiones y Trabajo a futuro	122
Evaluación de cara al trabajo a futuro	122
Conclusiones	125
Referencias.....	127
Apéndice.....	131

Capítulo 1: Introducción

La educación es la base fundamental de toda nación, cada una de ellas emplean diferentes sistemas educativos para poder cumplir las carencias y deficiencias que el sistema educativo tiene y México no es la excepción.

La base de una buena economía está en la formación que tienen los jóvenes, ya que ellos serán los futuros profesionistas que podrán generar nuevas tecnologías o formas de convivencia mediante los estudios que lleguen a obtener en las diferentes ramas de estudio como la medicina, la ciencia y tecnología, ciencias políticas entre otras.

En México, dos de las principales cuestiones por la que su juventud carece de una educación superior es por deserción, debido a la falta de apoyos económicos por un lado o la falta de infraestructura escolar. En el siguiente proyecto veremos de que manera se puede apoyar a disminuir ambos factores gracias a la propuesta que daremos más adelante.

Últimamente la tecnología ha crecido a pasos agigantados, en los cuales nos encontramos con redes sociales, las cuales sirven para informar o establecer comunicación desde diferentes puntos del planeta, también podemos encontrar información de manera más rápida y oportuna sin la necesidad de acudir a bibliotecas, que para la mayoría de las personas podría ser fácil de acceder a una de ellas, pero las personas de escasos recursos o en comunidades marginadas, esas oportunidades son casi nulas.

Internet empieza a ser una herramienta demasiado útil, debido a que podemos encontrar en ella libros, información y herramientas que nos ayudarían a poder formar nuestra educación pero toda esta información está dispersa, y eso dificulta el aprendizaje ya que en ocasiones no encontramos lo que queremos y desearíamos que todo eso estuviera en un solo lugar exclusivo para la educación.

Tenemos las herramientas necesarias, internet, información, redes sociales, solo falta algo que conjunte todo esto y es aquí donde la respuesta desemboca en nuestra propuesta en un **Sistema Gestor de Objetos de Aprendizaje.**

En este documento nos encontraremos con dos sustantivos muy importantes *Objeto de Aprendizaje* y *Sistema gestor de objetos de aprendizaje* que de manera breve podremos

decir que un *objeto de aprendizaje* es una entidad digital que apoya al progreso educativo de un usuario (videos, audios, imágenes, juegos interactivos etc.) de tal manera que puede ser reutilizado por cualquier usuario, y un *sistema gestor de objetos de aprendizaje* será el contenedor donde se encontrarán todos estos recursos digitales de manera procesada para una fácil búsqueda e interacción con el usuario, de tal manera que son creados por un generador de conocimiento (profesor) que ayudará en el proceso de aprendizaje.

¿Por qué mencionamos las redes sociales en este proyecto? La respuesta es sencilla de explicar, actualmente la sociedad debido al “boom” tecnológico, ha entrado en el ambiente de redes sociales para una fácil comunicación o medio de información (Twitter, Facebook, Google+, YouTube, WhatsApp etc.) de tal manera que esto a nosotros nos servirá para crear **una red social educativa**, teniendo como apoyo algunas de estas redes sociales para combatir con la tasa de deserción educativa.

Para el desarrollo de este proyecto, hemos creado una lista de objetivos que nos ayudaran a combatir la problemática planteada en un principio.

Como **objetivo principal** está el desarrollo de un sistema capaz de gestionar el uso y creación de objetos de aprendizaje con apoyo de las tecnologías de información de manera que dichos objetos pueda ser reutilizables para cualquier persona.

Como objetivos específicos tenemos los siguientes:

- I. Proponer una metodología MVC para el diseño de un sistema gestor de objetos de aprendizaje (repositorio web de contenidos educativos).
- II. Identificar las maneras de aprendizaje y estudio que son manejadas dentro de la pedagogía y adaptarlas a una plataforma web.
- III. Elaborar un sistema con enfoque de red social educativa
- IV. Hacer uso de las TIC como medio de información y enseñanza mediante los recursos que este ofrece (videos, audios, imágenes etc.) y acoplarlos a la plataforma que se creara
- V. Usar herramientas tecnológicas actuales e innovadoras para la implementación de la plataforma
- VI. Fomentar la participación educativa mediante un sistema de competencias evitando puntajes inalcanzables mediante la creación de rangos clasificatorios

Capítulo 2: Estado del Arte

Alrededor del concepto enseñanza-aprendizaje han surgido varias teorías como conductismo, cognitismo y constructivismo. Para los constructivistas es muy importante la interacción. Se hace énfasis en la identificación del contexto en el cual las habilidades serán aprendidas y aplicadas. De esta forma el estudiante es capaz de manejar la información. Ahora bien, el socio constructivismo es aquel modelo basado en el constructivismo, que dicta que el conocimiento además de formarse a partir de las relaciones ambiente-yo, es la suma del factor entorno social a la ecuación. Las Tecnologías de la Información y Comunicación (TIC) pueden ser utilizadas por el profesorado manteniendo un modelo pedagógico tradicional, o bien pueden ser usadas de forma innovadora para responder a los desafíos de la sociedad actual aplicando modelos socio-constructivistas y colaborativos [Diéz12].

Si bien las TIC han tenido gran aceptación en el ámbito académico, no necesariamente han impactado en el binomio enseñanza-aprendizaje ya que pueden ser utilizadas simplemente como un espacio de almacenamiento y difusión de los documentos, apuntes y materiales del profesorado; lo cual evidencia la aplicación de un modelo pedagógico que no añade nada significativo a los sistemas tecnológicos, ya que se siguen pensando los usos educativos virtuales con parámetros tradicionales [Sua14].

En la última década se ha dado la creación de herramientas precursoras de conocimiento, a lo que comúnmente conocemos como educación electrónica (e-Learning) basadas en el uso de dispositivos móviles (tablets, celulares, lap-tops, etc.) para poder dar alcance de material educativo a la sociedad, además de dar soluciones para lograr organizar tiempos, asistencias, viajes, entre otras dificultades que con la educación convencional suelen suceder.

Herramientas como Internet, multimedia (audio, video) son utilizadas por el e-Learning, dependerá del creador de contenidos (profesor) ver cuáles herramientas pedagógicas puede involucrar para poder cubrir los requerimientos educacionales que sean necesarios, esta combinación de medios visuales, interactivos y tecnológicos dan como resultado un entorno amigable para el aprendizaje.

Actualmente los e-Learning han ido evolucionando y con su evolución han ido naciendo nuevas metodologías y herramientas que ayudarán al proceso de aprendizaje, entre estas nuevas herramientas se encuentran los objetos de aprendizaje (OA's), estos, son

contenidos educativos los cuales han sido recopilados en contenedores, los cuales son llamados Repositorios de Objetos de Aprendizaje (ROA's) enfocados para poder abarcar los diferentes perfiles educativos de los usuarios.

En este capítulo abarcaremos algunos conceptos existentes en la literatura actual referente al uso de las TIC en educación.

A continuación se da a conocer el esquema de trabajo (*Figura 1*).

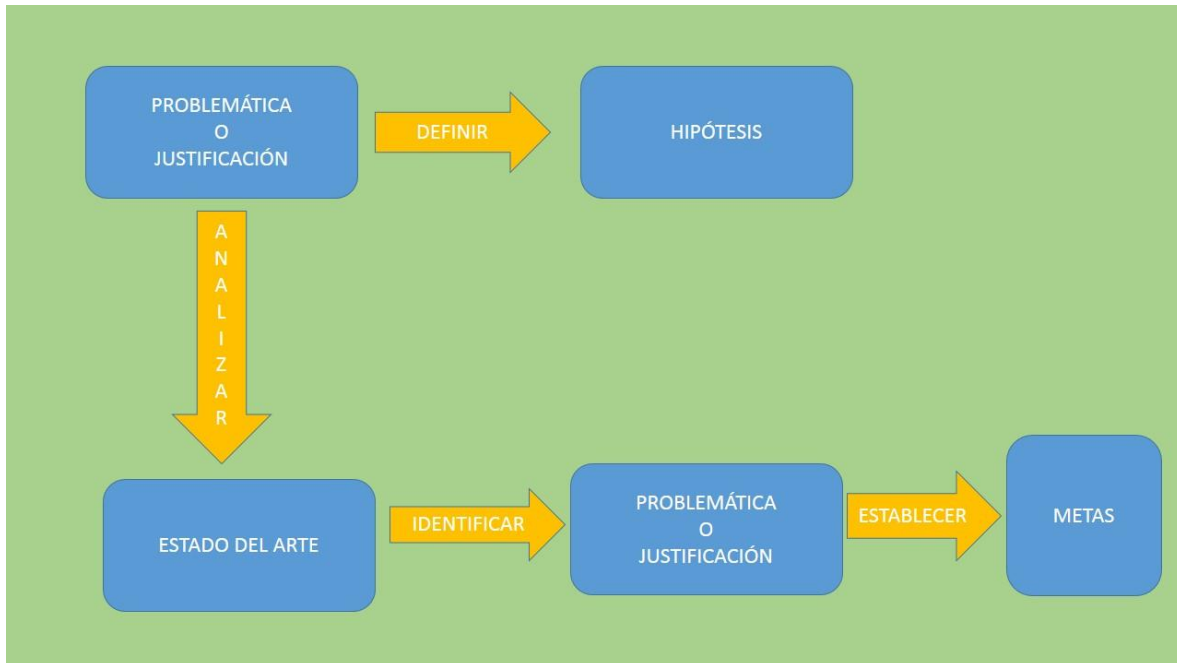


Figura 2.1: Desarrollo del estado del Arte

2.1 Contexto de la educación en México

Según datos del INEGI, de la población adulta de la nación aproximadamente 10,581,425 habitantes pudieron concluir una educación superior a nivel licenciatura, tomando en cuenta que un 26.8% de la población total está dentro de este rubro, siendo la población total de 112,336,538 habitantes en la nación, estamos hablando de alrededor de 30,000,000 de ciudadanos adultos [INEGI10] lo cual nos da como resultado un 34% aproximadamente del total de adultos que pudieron culminar sus estudios de nivel superior, esto, sin tomar en cuenta factores como pobreza, falta de infraestructura educativa o medios educativos (como libros, cátedras, etc.) ya que si fueran tomados en cuenta de una manera positiva (erradicando estos problemas), el número antes mencionado podría cambiar significativamente para el bien de la nación y por supuesto de la educación.

Citemos una frase de Platón que dice: “El Objetivo de la educación es la virtud y el deseo de convertirse en un buen ciudadano” [Frases14]; la educación es la base de cualquier individuo por tanto una nación con buenos niveles de educación tiende a ser una nación rica en cultura con gran poder social, económico y tecnológico, como lo dice el Banco Mundial. “Se trata de uno de los instrumentos más poderosos para reducir la pobreza y la desigualdad y sienta las bases para un crecimiento económico sostenido” [BM14], como talla educación es una herramienta la cual podrá crear individuos capaces de realizar tareas complejas las cuales impulsaran un desarrollo tecnológico y cultural a la humanidad.

En el último censo del INEGI se reportan aproximadamente 6233 universidades en el país, ofreciendo una gama muy amplia de licenciaturas e ingenierías, sin embargo el índice de deserción se acrecienta debido a diferentes factores como selección errónea de la licenciatura, bajos recursos económicos familiares, distancia de las universidades, embarazos no planeados, falta de interés en los programas educativos, personal académico no capacitado, gestión del tiempo, etc.

Debido a estos diversos factores, la educación ha evolucionado en desarrollo e implementación de nuevos modelos, técnicas y herramientas para dar soporte al aprendizaje en el camino al logro exitoso de concluir la educación básica, media superior y superior. Entre ello podemos mencionar el aprendizaje basado en problemas, aprendizaje basado en proyectos, aprendizaje colaborativo/cooperativo, aprendizaje por competencias.

La inserción de las tecnologías de información y comunicación en la educación representa la posibilidad de integrar otros medios de enseñanza bajo requisitos pedagógicos. Ante la necesidad de abarcar diversas regiones en las cuales la educación no llega, se han creado las famosas “Universidades a distancia”, las cuales no dependen de una enseñanza presencial sino que el alumno mediante videos, y material digital puede acceder a la enseñanza que requiere a lo cual llamamos como educación Virtual o E-Learning.

2.2 Educación a distancia

En [Loza10] podemos encontrar la siguiente definición: La educación a distancia ocurre cuando el profesor y el estudiante no se encuentran físicamente en el mismo lugar y no necesariamente al mismo tiempo.

La educación a distancia organizada puede remontarse al siglo XVIII, con un anuncio publicado en 1728 por la Gaceta de Boston, en donde se refería a un material auto-instructivo para ser enviado a los estudiantes con posibilidad de tutorías por

correspondencia. La educación por correo o correspondencia, también fue usada por las instituciones para ganar acceso a programas especializados o maestros que no estaban disponibles en el recinto.

El sistema de educación por correspondencia fue siendo reemplazado por el tipo de sistema de educación de una dirección, lo cual se concreta para los años 1970 con el uso de medios electrónicos tales como audio-cintas, videocintas, radio, televisión y las computadoras. Tenemos así que en 1950 la Ford Foundation comienza con programas educativos por televisión, en 1965 la University of Wisconsin da cursos basados en comunicación telefónica y en 1968 la Stanford University crea una red por televisión.

En México en 1947 el Instituto Federal de Capacitación de Magisterio inicia un programa para el perfeccionamiento del profesorado en ejercicio. En 1968 se crearon los Centros de Educación Básica de Adultos (CEBA). En 1971 aparece la Telesecundaria que utiliza la televisión para apoyar la acción de los centros. Otra institución pionera en la aplicación de los sistemas a distancia en México es el Centro para el Estudio de Medios y Procedimientos Avanzados de la Educación (CEMPAE) que, creado en 1971 y extinguido en 1983, inició en 1973 la aplicación de un modelo de Preparatoria Abierta. En 1974 el Instituto Politécnico Nacional inició su Sistema Abierto de enseñanza (SAE) en varias de sus escuelas. También el Colegio de Bachilleres inicia en 1976 su Sistema de Enseñanza Abierto. En el ámbito universitario, podemos destacar que en 1972 se iniciaron experiencias de educación a distancia a través del denominado Sistema Universidad Abierta (SUA) de la Universidad Nacional Autónoma de México (UNAM) que hoy ofrece algunos estudios, con validez académica similar a la que la misma Universidad ofrece a través del sistema presencial. En [Loza10] se mencionan 3 tipos de modelos educativos en los que ocurre la educación a distancia:

1. Estudio independiente guiado.
2. El aula remota
3. El modelo interactivo basado en TIC.

Es este último modelo, donde se utilizan tecnologías provistas por Internet para el acceso a los materiales y para mantener el contacto entre profesores y estudiantes, que favorece la implementación de modelos educativos basados en la construcción del conocimiento por parte del estudiante.

2.3 Una nueva Innovación dentro de las TIC: El e-Learning

Al hablar de educación a distancia tenemos que hablar de la tecnología de información y comunicación y del e-Learning ya que estas tres palabras van muy relacionadas conforme a los nuevos modelos educativos.

Como hemos comentado a lo largo de esta investigación la educación da las bases para formar individuos capaces de crear y transformar, de poder enriquecer a su nación de diferentes maneras además de darles una mayor formación social con otros individuos, nos hemos percatado que no en todas partes se cuenta con infraestructura tecnológica o arquitectónica para poder llevar a cabo la catedra y la interacción alumno-profesor, entonces se ha recurrido al internet, el cual apoya en esta difícil tarea de poder llevar la educación a zonas inimaginables e incluso permite la comunicación asíncrona, también comentamos las maneras de llevar a cabo esto y como hemos visto las redes sociales son un medio de comunicación del cual poder tomar ventaja para acrecentar la interacción de los alumnos.

2.3.1 ¿Qué es el e-Learning?

Se denomina aprendizaje electrónico (e-Learning en inglés) a la educación a distancia completamente virtualizada a través de los nuevos canales electrónicos (las nuevas redes de comunicación, en especial Internet), utilizando para ello herramientas o aplicaciones de hipertexto (correo electrónico, páginas web, foros de discusión, mensajería instantánea, plataformas de formación que aúnan varios de los anteriores ejemplos de aplicaciones, entre otras) como soporte de los procesos de enseñanza-aprendizaje.[Wiki14]

También puede definirse como un sistema basado en la comunicación masiva y bidireccional que sustituye la interacción personal en el aula del profesor y alumno, como medio preferente de enseñanza, por la acción sistemática y conjunta de diversos recursos didácticos y el apoyo de una organización tutorial, que proporcionan el aprendizaje autónomo de los estudiantes, además de reforzar la habilidad de la comunicación efectiva con los participantes a través de las plataformas usadas. [Bello13]

2.3.2 Plataformas dentro del E-Learning

Dentro de las herramientas e-Learning existentes, éstas se relacionan en un conjunto de hardware y software que soporten las funcionalidades y aplicaciones realizadas para el sistema.

Si buscamos los diferentes sistemas e-Learning nos encontraremos que este tendrá un contenedor el cual se divide en cinco bloques como lo demuestra la siguiente figura:

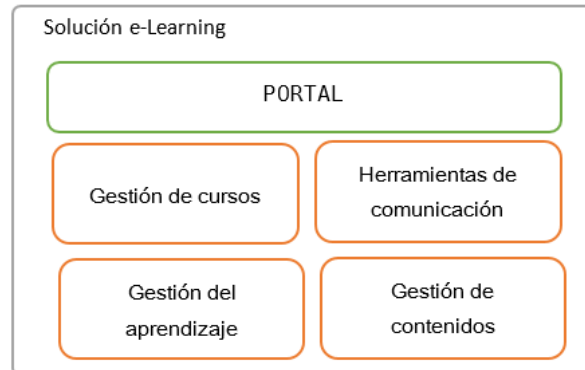


Figura 2.2: Bloques de una solución e-Learning, Fuente [Rold10]

- *PORTAL.*
- *SISTEMA GESTOR PARA LA ADMISION DE LOS CURSOS (CAMS “COURSE ADMINISTRATION MANAGEMENT SYSTEM”).*
- *SISTEMA GESTOR DE APRENDIZAJE (LMS).*
- *SISTEMA GESTOR DE CONTENIDOS (LCMS).*
- *HERRAMIENTAS DE COMUNICACIÓN.*

- 1) **Portal:** Encargado de mostrar los contenidos al usuario además de poder interactuar con ellos.
- 2) **Sistema gestor para la admisión de los cursos:** Encargado de la publicación de los cursos existentes además de englobar las características administrativas de los cursos (altas, bajas entre otras) y en dado caso puede generar informes acerca de ello.

- 3) **Sistema gestor de aprendizaje:** Encargado del control de actividades del usuario (resultados, evaluación de tareas, acceso a materiales) además que tiene como deber que los contenidos y herramientas de comunicación favorezcan el proceso de aprendizaje además de conseguir buenos resultados en el proceso formativo.
- 4) **Sistema gestor de contenidos:** Conocido también como *“courseware”* es el conjunto de herramientas para manejar los contenidos de aprendizaje y aunque no parezca es un sistema que no está al 100% relacionado con un sistema gestor de aprendizaje pero no logra su total independencia de él.
Resulta favorable que este rubro cumpla con los requisitos de usabilidad, estandarización y portabilidad los cuales pueden ser vistos en el tema de: *“Estándares utilizados en los OA’s”*.
- 5) **Herramientas para la comunicación:** Importantes para mantener una retroalimentación entre los participantes del e-Learning.
Dentro de estas podemos encontrar 2 maneras de establecer la comunicación:

- 1) *En tiempo real:* hechas por video-conferencias, chats y grupos de chats las cuales permiten dar solución al problema o duda planteados en ese instante.
- 2) *Mensajería:* estas pueden ser como mensajes o e-mails los cuales serán respondidos cuando uno de los actores pueda verlos, con ello quiere decir que la duda o cuestión no va a ser resuelta en el instante sino que pueden pasar horas o días para tener una respuesta de cualquier actor además de que estos pueden ser revisados cuantas veces quiera el participante ya que se quedan almacenados.

Dentro del E-Learning podemos encontrar diferentes plataformas las cuales necesitan ser administradas por un Sistema Gestor de Contenidos (como los sistemas gestores de objetos de aprendizaje, los cuales veremos más detalladamente en el siguiente capítulo), aunque se usan principalmente en páginas web (Web Content Management o WCM) Santillán [Sant08] indica que no solo se limitan a contenidos Web.

Un sistema gestor de contenidos debe permitir 3 funciones las cuales son principales para uno de ellos las cuales son:

- Gestión de contenido
- Publicación
- Presentación

Un SGC (Sistema Gestor de Contenidos) no puede abarcar por si solo todas las necesidades que se necesitan para un sistema que se basa en el aprendizaje es por ello que se recurre a nuevas especificaciones y necesidades que permiten la creación de un sistema gestor de aprendizaje tomando como base un SGC.

Según Bär et los requisitos indispensables para un sistema de interacción cuyo objetivo es el dominio del aprendizaje deben comprender lo siguiente [Bär07]:

- Facilidad de interacción durante y después de la conferencia
- Una arquitectura abierta en la cual debe haber la posibilidad de ampliaciones
- Ser escalable (el sistema debe de ser capaz de manejar un solo curso o una conjunción de varios)

Ahora desglosaremos las herramientas o elementos más comunes que necesita un entorno de aprendizaje los cuales son:

- Para la creación de materiales
- Para la evaluaciones
- De colaboración (formularios, mails, chats, etc.)

Estas herramientas antes mencionadas nos ayudaran en una interacción síncrona y asíncrona de los modelos.

2.4 Sistemas Gestores de Objetos de Aprendizaje

La gestión de contenidos en los entornos e-learning en conjunto con la demanda de herramientas para la educación a distancia ha creado como resultado plataformas e-Learning conocidas como Sistemas de Gestión de Aprendizaje o Entornos de aprendizaje Virtual (un subgrupo de los Gestores de Contenido) q cada uno cumpliendo diferentes

tópicos dependiendo de las demandas que les fueron solicitadas además del nivel de escolaridad para el cual iban dirigidos

Un sistema gestor de Objetos de aprendizaje es un contenedor de OA's que los organiza y mantiene disponibles para cualquier usuario y sus diferentes usos, a su vez estos sistemas gestores tienen el objetivo de crear redes para que el contenido de los OA's no sea un factor de incomodidad al no poder encontrar la fuente o ubicación de dicho recurso, logrando esto mediante la comunicación de las aplicaciones interoperables que existen entre ellos. [GueGo14]

A continuación presentamos una lista de los Sistemas que han tenido más auge en la web:



Moodle

Es uno de los gestores de aprendizaje más extendido debido a su capacidad de adaptación y modularidad. Su web está disponible en varios idiomas y en lo referente a la interoperabilidad trabaja con la especificación ADL Scorm, una de las especificaciones más extendida en el e-Learning, lo que significa que nos ofrece una cierta compatibilidad con otros gestores de aprendizaje que soporten la misma especificación. Al finalizar la creación del curso podemos exportarlo para así utilizarlo desde otro lugar.

Igualmente Moodle colabora en el desarrollo de estándares IMS, es decir, forma parte del proceso de estandarización y creación de especificaciones relacionadas con el aprendizaje. Cuenta con una serie de módulos de comunicación para que los alumnos puedan dirigirse al profesorado, comunicarse entre ellos y así crear su propia comunidad de aprendizaje.

Ventajas:

- Manejo de archivos como presentaciones, documentos de texto entre otros
- Creación de foros
- Amigable con el usuario
- Ritmo de estudio adaptable
- Fácil comunicación entre otros usuarios

Desventajas:

- Falta de aplicaciones
- Muy poca interacción autodidacta
- Muy sensible al manejo de peticiones por parte de los usuarios, esto se debe por estar hecho mediante un diseño CGI

**LAMS**

Es un sistema dirigido a la gestión de actividades de aprendizaje en el que me gustaría destacar que en ningún momento se pierde la secuencia de actividades. También permite exportar el curso realizado para importarlo más tarde a otro gestor.

Es uno de los mejores en el caso del aprendizaje adaptativo y colaborativo.

Además presenta beneficios como el hecho de permitir al profesorado el control sobre los materiales educativos, el fomento del crecimiento personal, el trabajo en grupo y la comunicación entre el profesorado y el alumnado.

Una de las mayores ventajas con las que cuenta es que al incluir líneas de transición de una actividad a otra podemos establecer la necesidad de superar la actividad anterior para seguir avanzando.

Ventajas:

- Ofrece tutoriales de aprendizaje
- En caso de ser principiante ofrece ayuda (mediante un ayudante que es una oveja) para poder familiarizarnos con el sistema
- Control sobre los contenidos.
- El aprendizaje se divide en etapas para poder crear un plan de estudio
- Posee la facilidad de grabar video
- Fácil manejo de archivos entre alumno-docente

Desventajas:

- Al ser un sistema secuencial de aprendizaje, un alumno al cual se le dificulta el tema, puede quedarse “estancado” en una lección y no poder avanzar a otra hasta cumplir satisfactoriamente en la que esta
- La creación de foros no está soportada, se crea como si fuera una actividad mas no como un lugar para compartir retroalimentación o conocimiento

**Sakai**

Si lo comparamos con LAMS una de sus mayores desventajas es que llega a perder el manejo de flujo de aprendizaje. Se puede introducir de forma textual la mejor secuencia para realizar las actividades pero finalmente es el estudiante el que elige. Se puede decir que su uso se ajusta más al de un gestor de recursos educativos. Además tampoco es posible exportar el curso creado.

Ventajas:

- Fácil disposición de herramientas para el tipo de actividad creada
- Publicaciones en línea
- Cuenta con herramientas de comunicación
- Herramientas para la gestión de exámenes, tareas y actividades para la fácil evaluación de los alumnos.

Desventajas:

- Difícil de usar para un usuario principiante
- Poco amigable al usuario
- No intuitiva
- Necesita ser instalado

Claroline

Posibilita la creación de actividades de forma ordenada. También se pueden definir diferentes secuencias de aprendizaje para el mismo curso, pudiendo así fraccionar el curso en diferentes niveles y permitiendo al estudiante comenzar en el nivel que él cree que posee.

Se puede emplear para reforzar materias dejando que sean los estudiantes quienes escojan el camino que más se adapta a sus necesidades. Sin embargo, tiene el inconveniente de que no permite exportar el curso creado.

Ventajas:

- Fácil interacción debido a que presenta una respuesta amigable con el usuario, similar a Moodle pero no es tan exigente en la secuencia de tareas como LAMS
- Herramientas suficientes para la interacción entre profesor-alumno
- Contiene un sistema seguimiento de tareas
- Permite cualquier formato de archivos
- Permite guardar los foros de discusión ya sean públicos o privados

Desventajas:

- Difícil de personalizar
- Numero de módulos y plugins limitado
- Difícil comprensión en lenguajes con caracteres especiales
- No existen las categorías
- Tedioso a la hora de abrir archivos
- Velocidad de carga de archivos lenta



Dokeos

Permite crear una lección SCORM, lo que posibilita su compatibilidad con otros gestores de aprendizaje.

Al dar el nombre a la actividad se puede modificar su posición en la secuencia de aprendizaje y asignarla a una determinada sección del curso. También permite seleccionar qué actividad es necesaria haber hecho para pasar a la siguiente.

Ventajas:

- Fácil acceso por parte de cualquier Institución educativa
- Buen medio de comunicación dentro de la plataforma
- Adaptable a las necesidades del docente
- Puntos de discusión como foros y blogs
- Permite controlar y revisar las actividades que el alumno ha hecho en la plataforma
- Video-conferencias

Desventajas:

- Debido a contar con muchas herramientas, requiere de un aprendizaje constante de la plataforma
- Es instalable
- Interfaz compleja



OpenStax

Creado por el Dr. Richard Baraniuk en 1999 de la Universidad de Rice provee material de los autores y espacio para los alumnos donde pueden crear material acerca de libros, cursos y reportes (<http://cnx.org>)

Ventajas:

- Creación de libros electrónicos

- Intercambio de libros electrónicos

Desventajas:

- Solo es un sistema de intercambio de archivos

Además de los anteriormente señalados podemos encontrar los siguientes repositorios de América Latina:

- ✓ UNAM Galería de Objetos Educativos (<http://ccobagaleria.cuaed.unam.mx>).
- ✓ Contenidos Educativos Digitales para la educación Superior, Universidad de Córdoba. Colombia. (<http://www.aves.edu.co/ovaunicor/>).
- ✓ EducarChile el portal de la educación de Chile. (www.educarchile.cl).

En una breve conclusión y como ya lo habíamos mencionado anteriormente un Sistema Gestor de Aprendizaje nos ayudara a facilitar la interacción alumno-profesor por ser una medio en el cual se aportan contenidos para el aprendizaje autodidacta y además en caso de requerirlo una valoración y evaluación del profesor al alumno además de darle una retroalimentación del tema que abarquen en ese instante, haciendo esto más práctico para comunidades o situaciones en las cuales el aprendizaje común o como lo conocemos como presencial es muy difícil de alcanzar, entonces, al unir el E-learning con los SGC y los SGA obtenemos el resultado que estamos esperando un SISTEMA GESTOR DE CONTENIDOS PARA EL APRENDIZAJE.

Nos encontraremos con que llamaran a los Sistemas de aprendizaje y a los Sistemas de contenidos de aprendizaje como uno solo “Sistemas gestores de contenidos de aprendizaje” y es que estos dos trabajan en conjunto ya que el primero de ellos maneja los cursos, el seguimiento de las actividades y el avance del alumno y el segundo maneja todo lo que se refiere a el desarrollo de contenidos, accesos y almacenamiento de la información.

2.5 Objetos de Aprendizaje (OA's)

Al hablar de Objetos de aprendizaje podremos encontrar variedad de definiciones, una propuesta por el Comité de Estándares de Tecnologías de Aprendizaje (LTSC *Learning Technology Standards Committee*) es la siguiente:

“Un objeto de aprendizaje es cualquier entidad digital o no digital, la cual puede ser usada, re-usada o referenciada durante el aprendizaje apoyado por la tecnología, estos pueden ser sistemas de entrenamiento basados en computador, ambientes interactivos de aprendizaje, sistemas inteligentes de instrucción asistida por computador, educación a distancia y ambientes colaborativos de aprendizaje; dentro de ellos podemos encontrar imágenes, la lección a aprender, contenido multimedia como videos, sonidos, simulaciones, animaciones, entre otras...” [Ossa05].

Mientras que otros definen a un objeto de aprendizaje como “la conexión de <objetos> con <tecnología>” [Loza10].

El término data de la década de los setenta, debido al constante uso del lenguaje “Simula 67” pero tomo más fuerza cuando tuvo aparición los lenguajes de programación orientados a objetos como es el caso de C++ y JAVA ya que los orígenes de los OA’s datan que son entidades asociadas a la programación, diseño, análisis y teoría orientada a objetos.

2.5.1 Propiedades y características de los OA’s

Todo OA está conformado por dos partes, la primera es el contenido del objeto y la segunda los meta-datos, los cuales describen la encapsulación del OA, un ejemplo dentro de esta encapsulación puede ser las palabras claves del contenido, qué objetivos se tienen como meta, el nivel al que se quiere llegar, los prerrequisitos indispensables para progresar dentro de la interacción, la evaluación, autor, fecha de creación, versión, lenguaje, etc.

Estos metadatos son necesarios debido a que en ellos se almacenan los objetos de aprendizaje mediante repositorios donde pueden ser tomados para su integración de algún curso, los repositorios no son más que una base de datos en la cual se logra la localización de objeto requerido.

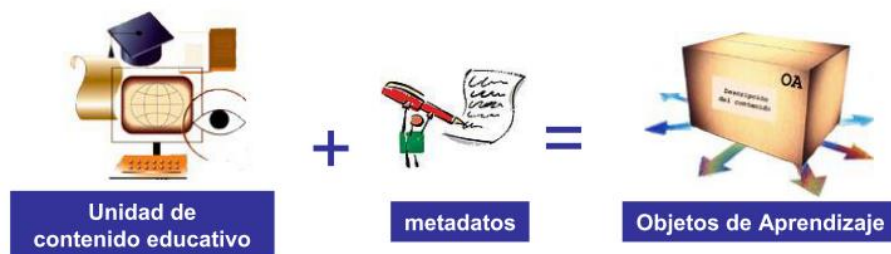


Figura 2.3 Conceptuación de un OA. Fuente: [Lope05]

Dentro de las características de los OA's podemos encontrar las siguientes:

- Nombre que lo identifique
- Objetivo: especificación de las metas que se pueden lograr al concluir el tema
- Descripción: muestra las características del objeto
- Actividades y evaluaciones
- Palabras clave para la descripción del contenido

NOTA: Los OA' no tienen la exigencia de estar formados solo por un objeto de aprendizaje, tienen la prestación de contener uno o más objetos de aprendizaje.

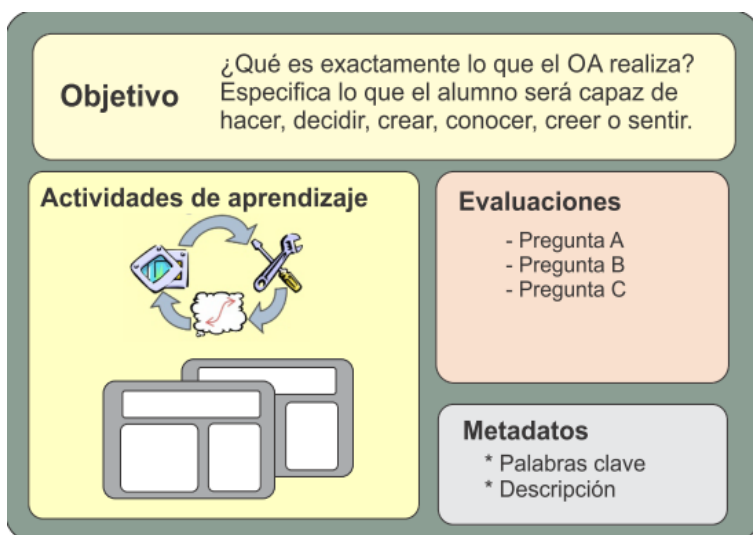


Figura. 2.4 Estructura básica de un OA [Hort06]

Para que un OA pueda tener un alto grado de reutilización debe de cumplir con las siguientes características:

- **Granularidad o Nivel de Agregación:** Hace referencia al tamaño del recurso y su descomposición visto de otra manera es el grado en el que el recurso puede formar parte de un recurso más grande; este concepto se liga con la idea de reutilización, en otras palabras, el nivel de granularidad va a dar como resultado la dependencia del contexto y reducirá su potencial de reutilización, a más alto nivel más dependiente y viceversa.
- **Interoperabilidad:** Es la capacidad que tiene para poder relacionarse e interactuar con otras aplicaciones que pueden ser desarrolladas por terceros.

- **Metadatos:** Según [Rold10] los metadatos se pueden definir como “datos estructurados sobre otros datos, creados y gestionados para describirlos de algún modo, que puedan utilizarse para una función diferente de las funciones de los datos originales del objeto”; los metadatos describirán las características especiales de los recursos como lo son el autor, información técnica y educativa, con el fin de en un futuro poder dar un soporte y ayuda dentro de una indexación y búsqueda de los repositorios.

Aunque todas las características antes mencionadas son de gran importancia, también contienen una jerarquía la cual nos ayudara a crear un buen OA , un ejemplo es la cuestión de los metadatos como lo mencionan en [Muño07], según Currier, los metadatos de baja calidad pueden implicar que un recurso sea básicamente invisible en el repositorio y permanece sin utilizarse; para evitar caer en este problema, la estandarización mediante el IEEE LOM clasificó nueve categorías para la manipulación de los metadatos las cuales son descritas en la siguiente tabla:

Categoría	Descripción
General	Describe el recurso con información sobre si título e idioma
Ciclo de vida	Contiene características sobre el estado actual del recurso, como autores y fecha de creación
Meta-metadatos	Describe el registro de metadatos
Técnica	Agrupar las características técnicas y requisitos de ejecución
Uso educativo	Contiene información sobre aspectos educativos y pedagógicos, tales como la interactividades y tipo de recurso
Derechos	Describe condiciones de uso y derechos de propiedad intelectual del material
Relación	Relaciones que el objeto de aprendizaje tenga con otros objetos de aprendizaje
Anotación	Contiene comentarios sobre el uso educativo del objeto de aprendizaje
Clasificación	Describe el objeto de aprendizaje según un sistema de clasificación específico

Tabla 2.1: Categoría de los metadatos

En [Garc05] se considera un buen OA a todo aquel que cumpla con las características antes mencionadas, además de reutilización, medios educativos, interoperabilidad, accesibilidad, durabilidad, independencia y autonomía, granularidad, generatividad escalabilidad, portabilidad, flexibilidad, versatilidad y durabilidad.

Cuando hablamos de reutilización damos referencia que el OA puede ser utilizado para diferentes contextos y poder adaptarse a nuevas formas educativas para poder abarcar con ello el medio educativo del que hablábamos y con ello poder generar el aprendizaje al cual queremos llegar en el alumno que utilice el medio.

El OA también debe de poder integrarse a diferentes plataformas sin importar las diferentes tecnologías usadas en cuanto a software y hardware se refiere en este punto es donde logramos que nuestro OA contenga la característica de interoperabilidad.

En párrafos anteriores hacíamos mención a los metadatos, ya que en ellos recae otra característica: la accesibilidad; la cual esta afianzada a los metadatos para poder lograr la facilidad de que un OA puede ser encontrado mediante las descripciones contenidas en el metadatos del OA, estos metadatos a su vez permiten que el OA pueda ser almacenado en un repositorio.

La definición de Wiley [Muño11] define la reutilización a la idea de ensamblado y descomposición de los recursos existentes, con el fin de crear nuevos recursos que solo serán combinados más no serán modificados ni adaptados

Wiley [Muño11] además menciona cuatro modos de hacer uso de los contenidos:

- 1. Reutilizar:** Usar y copiar de la misma manera en la que el objeto de aprendizaje está disponible.
- 2. Revisar:** Modificar el recurso de manera que satisfaga nuestras necesidades.
- 3. Combinar:** Mezclar el recurso con otros recursos.
- 4. Redistribuir:** Compartir con otros el nuevo recurso que es derivado de las tres formas mencionadas anteriormente.

Continuando con las características que debe de comprender un buen OA nos encontramos con la durabilidad donde esta es definida como la capacidad de flexibilidad y actualización que debe de tener un Objeto de Aprendizaje [Med13], en cuanto a autonomía e independencia se refiere a que deben de ser independientes con respecto a los sistemas en los que fueron creados, y por último pero no menos importante (si no es que es la característica principal de un OA) la granularidad la cual desemboca en la el nivel de agregación que tiene un OA en el cual se ve el tamaño del recurso y a la manera en la que el recurso puede ser utilizado para poder construir o formar recursos más grandes.

A pesar de las características antes mencionadas para la creación de un buen objeto de aprendizaje encontramos otras bases fundamentales las cuales son esenciales en la creación de un objeto de aprendizaje de calidad [Delo94].

- **Saber Hacer:** Integrar el conocimiento transmitido de manera que pueda ser utilizado en el ámbito profesional además de requerir una evaluación del contexto en el cual en el que el alumno se desenvuelve.
- **Saber Conocer:** En esta parte es necesario que se adquieran no solo los conocimientos teóricos sino que aparte de ello se pueda reconocer el entorno y comprenderlo.
- **Saber Ser:** Con el objetivo de desarrollar las habilidades del estudiante de manera que puedan puestas ser en práctica dentro una vida cotidiana, desarrollando actitudes, competencias y valores que son la base para cualquier individuo productivo.
- **Saber Convivir:** Además de los conceptos antes mencionados se debe de tomar en cuenta un desarrollo social, en pocas palabras, hacer que el estudiante conviva con diferentes individuos para que así pueda distinguir las diferentes diferencias y semejanzas que puede encontrar en sus compañeros de entorno con esto se crea un conciencia de independencia entre todos los individuos, en cambio, cuando el trabajo se realiza en conjunto, las diferencias entre los individuos se ven reducidas con el objetivo de tener metas comunes que serán de beneficio para la comunidad en general.

En resumen, un objeto de aprendizaje es una estructura reutilizable básica que contiene información detallada de un tema en específico el cual permite relacionarse con estructuras similares, teniendo como atributos un *id* el cual servirá como identificador del objeto, un *nombre* haciendo referencia al tema del objeto, *fecha* y una *descripción* que contiene información acerca del objeto y por ultimo un *objetivo* en el cual se detalla cual es la meta que tiene el OA

Subiendo un escalón en la jerarquía de los objetos de aprendizaje nos encontraremos que este requiere de un *área* el cual se relaciona con el conjunto de conocimientos que brinda el OA y además de contener un *categoría* haciendo mención al subtema del área que fue comprendida en primer rubro, además es necesario un *contenido* que es la información que presentara el objeto de aprendizaje al usuario que lo solicite.

Para una mejor indexación y búsqueda del OA además de manejar un *id* y un *nombre* se requiere de un *autor* el cual es el nombre del creador del OA, y para un mejor manejo del acceso a los OA's se requiere de un *password*.

Dentro del método de aprendizaje se encuentra el rubro de *saber* que son los conocimientos adquiridos durante a lo largo de la educación, dentro de él, se desglosa en *saber ser*; en el cual se desarrollan todas las actividades además de que se enseñan tanto valores como actitudes que puede poner en practica dentro de su vida cotidiana, *saber hacer*; busca la integración del conocimiento adquirido de manera que pueda ser usado a nivel profesional y el *saber conocer* su meta es adquirir no solo los conocimientos teóricos, sino en comprender, conocer y descubrir el entorno en el cual se labora.

Título		
Descripción		
Objetivo		
Autor		
Saber		
Conocer	Ser	Hacer
Evaluación	Evaluación	Evaluación
Tipo de aprendizaje	Tipo de aprendizaje	Tipo de aprendizaje

Figura 2.5: Estructura de un OA

Para poder ver la estructura antes mencionada de un OA se presenta a continuación la siguiente imagen la cual remarca la estructura descrita

2.5.2 Estándares utilizados dentro de los OA's

Para saber de lo que vamos a hablar en este tema primeramente debemos de saber que es un estándar, el cual es un conjunto de reglas y normas que sirven para describir o seguir paso a paso las características que debe de tener un producto, bien o servicio para poder decir que es de calidad.

Hay estándares propios que pueden ser creados por la empresa donde se labora, pero también hay estándares que son elaborados por comités de talla internacional que regulan la calidad de los bienes y servicios, los cuales deben de ser seguidos por todas las corporaciones para poder acreditar la calidad de sus productos, dentro de los comités más reconocidos podemos encontrar el IEEE (*Institute of Electrical & Electronics Engineers*) el ISO (*International Organization for Standardization*) o el ANSI (*American National Standards Institute*) entre otros más.

A continuación vamos a nombrar diferentes estándares los cuales son evaluadores de los OA's:

- **IEEE-LTSC**

Formado en 1996, el Comité para la Normatividad de la Tecnología de Aprendizaje ha sido el encargado de desarrollar y promover normas de tecnología institucional. El comité describe a los Objetos de Aprendizaje como entidades digitales o no digitales las cuales pueden ser utilizadas, reutilizadas o referenciadas durante un aprendizaje asistido mediante una interacción humano-computadora. La especificación más reconocida dentro de este rubro es la de los metadatos asociados a los OA's o mejor llamado como LOM (Learning Object Metadata) cuya meta es describir los recursos del aprendizaje.

- **IMS**

El IMS por sus siglas en inglés (Instruction Management System) es un consorcio creado por vendedores, productores, implementadores y consumidores de e-Learning cuya tarea es promover especificaciones abiertas que faciliten la interoperabilidad en todos los rubros de la formación on-line.

- **ADL-SCORM**

Advance Distributed Learning por sus siglas en inglés ADL es una organización promovida por el gobierno de los Estados Unidos cuya meta es favorecer la

adopción de e-Learning de manera eficiente y eficaz; se ha convertido en el impulsor principal de la normalización de este rubro tecnológico, sus especificaciones se han convertido potencialmente en estándares.

SCORM (Shareable Content Object Reference Model) ha sido la rama de ADL mas fuerte ya que incorpora elementos de LTSC, AICC e IMS en una sola especificación, además de ofrecer una visión global del entorno e-Learning otorgando los mecanismos indispensables para la distribución y gestión de OA's así como dar un seguimiento de la interacción de los estudiantes con los OA's.

2.6 Objetos de aprendizaje como medio educativo

Para tomar ventaja de los objetos de aprendizaje, debemos captar y retener el interés de los alumnos en el momento de su consulta, en el sistema educativo convencional (presencial) el alumno llega a tal grado que pierde el interés en lo que se le está enseñando ¿Qué garantiza que dentro de la educación a distancia los OA's no perderían el control total? Es allí donde las herramientas pedagógicas en conjunto con las herramientas tecnológicas crearon una unión y con esto pudieron lograr obtener un resultado el cual mediante las emociones, sentimientos y psicologías de los alumnos en conjunto con las maneras de visualización de la información lograron el resultado deseado.

A continuación se explicara cuáles de las herramientas más eficientes logran obtener ese control que se desea: la interacción seductiva y los sistemas de puntuación (ELO el más benéfico para este rubro)

2.6.1 Diseño de Interfaces y OA atractivos mediante interacción seductiva

Desde tiempos memorables la atracción y la seducción ha jugado un papel importante en la vida del ser humano, ya sea la atracción o seducción hacia otra persona u objeto. La seducción, normalmente asociada a la sexualidad, se refiere a:

“El proceso de atraer deliberadamente a una persona con el fin de modificar su opinión o hacerle adoptar un determinado comportamiento según la voluntad del que seduce” [GueGo11], lo primero es fácil de conseguir debido a las hormonas de cada individuo, pero lo segundo es más difícil de tener, debido a que un objeto no genera esa sensación; diferentes empresas se dieron a la tarea de llevar a cabo investigaciones para que los productos elaborados puedan “seducir” al mercado objetivo e incrementar sus ventas.

Lo primero a lo que se enfocaron fue hacer atractivo su producto, y aunque nosotros lo podemos asociar con sentimientos, los medios lo ven como una manera de poder obtener

la atención del usuario, pero ¿cómo obtener esa atención? Sencillamente, mediante la elaboración de una estética del producto.

La estética cumple uno de los principales factores de este tema, debido a que “mientras más bonito, es mejor” ya que en la estética están involucrados todos nuestros sentidos y reaccionamos de diferentes maneras mediante estímulos que esta estética crea en nosotros, por ejemplo: ¿a un niño le gustara más un oso de peluche suave y está bien elaborado de tal manera que parece un osito de verdad o uno cuya forma no se asemeja a un oso? Es allí donde se crea una parte afectiva por medio de la estética y esta a su vez nos da un resultado.

Ahora, llevemos esto a la programación de aplicaciones o mejor aún, a los OA's, ¿cómo lograr que el alumno se interese por utilizar los OA's? Una opción es a través de las recompensas a cambio de la tarea que ha realizado, a esto le llamamos incentivos. Otra forma es el uso de colores en el diseño de interfaces de usuario que transmitan seguridad al usuario, por ejemplo no diseñar una interfaz de “Bienvenido” con letras rojas o con una combinación de rojo y amarillo, ya que estos colores el ser humano los asocia como una señal de peligro o precaución.

Para poder atraer a nuestro usuario y sin que pierda el interés, es necesario considerar la interfaz de tal manera que el usuario tenga la confianza de seguir en ella, que además no sienta que es tardada y si en dado caso de que sea tardada, hacer que el usuario se distraiga dentro de ella mediante diferentes maneras para que el tiempo de espera del usuario no se vea o no sea sentido por él ya que, un usuario descansado y relajado trabaja mejor que uno que no lo está, además de tener una interfaz intuitiva, se necesita que sea amigable con el usuario, abarcando colores dentro de la interfaz que lo hagan sentirse cómodo de manera física y psicológica, ya que no podemos tener una interfaz con colores muy llamativos, porque, aunque sea algo extravagante para el usuario, al término de unos 10 minutos aproximadamente o menos, esos colores (naranja, verde limón, rosa mexicano, etc.) pueden afectar la vista del usuario, además de llegar a estresarlo.

El hablar de un buen diseño de la interfaz, en ocasiones resulta algo difícil, ya que en ocasiones no abarcamos todas las características idóneas para una interfaz eficiente, intuitiva y amigable con el usuario, es por eso que en el artículo “10 heurísticas de usabilidad para el diseño de interfaces” de Jakob Nielsen [JAK95] nos habla acerca de 10 puntos

importantes que nos ayudan en el diseño de una buena interfaz, teniendo como prioridad, los principales problemas que el usuario presenta en ella:

1. Visibilidad del estado del sistema:

El sistema debe de permitir siempre al usuario estar informado que es lo que está pasando a través de una retroalimentación apropiada en un tiempo razonable, esto podría verse con mensajes de alerta o informativos acerca de lo que sucede en el sistema.

2. Relación entre el sistema y el mundo real:

Va de la mano con la primera característica antes mencionada solo que ahora estos mensajes deben contener palabras, frases o contenido que el usuario pueda entender y con el cual se pueda familiarizar.

3. Control de usuario y libertad de elección

Por lo general el usuario al querer familiarizarse con el sistema hace elecciones erróneas dentro de él, esta característica debe de permitir al usuario una salida fácil para salir de la elección no deseada sin tener que pasar por toda la secuencia que esta opción mal seleccionada pueda implicar.

4. Consistencias y estándares

No se debe poner al usuario en duda de que diferentes acciones pueden ser sinónimas de otras, debe de ser claro y conciso con lo que se expresa en la plataforma.

5. Prevención de errores

Aun cuando exista una retroalimentación al usuario acerca de los errores ocurridos dentro del sistema, lo mejor es disminuir el porcentaje de errores, además de comprobar la acción que desea realizar con un mensaje de información al usuario antes de ejecutarla.

6. Preferencia en reconocimiento en lugar de recordar el objetivo

Hacer que el usuario no tenga una carga de “recordar la información vista” para así reducir su carga de memorización, en vez de eso manejar la información que necesita el usuario en ese momento mediante objetos visibles y de fácil interacción para su recuperación cuando la información sea solicitada.

7. Flexibilidad y eficiencia de uso

Uso de aceleradores: esto ayuda mucho a un usuario experto ya que permite acelerar la interacción con el sistema, pero también en cierto grado, puede ayudar a los usuarios sin experiencia mediante el uso de acciones frecuentes y poder acceder a ellas.

8. Diseño estético y minimalista

Los diálogos manejados dentro de la interfaz no deben de contener información irrelevante o innecesaria ya que este tipo de información puede hacer que el usuario pierda el objetivo que tiene la interfaz.

9. Reconocimiento, diagnóstico y recuperación de errores

Al aparecer un mensaje de error este no debe de manejarse mediante códigos, sino que debe tener un lenguaje de fácil uso para el usuario además de proponer una posible solución.

10. Ayuda y documentación

Debe de contener un área de ayuda y documentación la cual debe de ser centrada en los posibles problemas que pueden ocurrir y esta documentación no debe de ser tediosa y extensa

Además de los puntos antes vistos en cuento se refiere al diseño de la interfaz, el uso de colores entre otras cosas, el diseño de buenos objetos de aprendizaje es indispensable y aún más importante, porque estos servirán para el apoyo al aprendizaje, ¿qué queremos decir con esto? Los objetos de aprendizaje realizados deben de ser fáciles de usar comprender, “divertidos” y lo más importante que deje en el usuario una nueva enseñanza porque ese es el principal objetivo

Una de las maneras en las que podemos evitar que el sistema se vuelva aburrido o perdamos el interés en el usuario es mediante la creación de competencias y obtención de premios para estimular aún más el aprendizaje, esto se puede realizar mediante diferentes dinámicas pero ¿cómo evitar caer en un sistema de puntuación inalcanzable el cual haría que el usuario perdiera interés en la competencia y premio de la actividad?

Para eso se ha propuesto un sistema de puntuación llamado ELO el cual explicaremos a continuación.

2.6.2 Sistema de puntuación “ELO”

El sistema de puntuación ELO es un método matemático, basado en cálculo estadístico, para calcular la habilidad relativa de los jugadores de juegos como el ajedrez o el go [Go14].

Debe su nombre a su inventor, el profesor Árpád Élő (1903-1992), un físico estadounidense de origen húngaro, contrariamente a la suposición generalizadamente extendida de que se trata de un acrónimo. Se recurre al uso de mayúsculas ELO para distinguir al método respecto de la persona que lo inventó.

Pero, ¿cómo nace ELO? ELO nace debido al Dr. Élő, quien trabajó para la United States Chess Federation o USCF (Federación de Ajedrez de Estados Unidos), fue designado en 1959 como presidente del Comité de Puntuación.

ELO consiste en generar rangos de puntuación y meterlos en categorías, para poder dar un sistema de puntaje más competitivo de tal manera que así el usuario de ser un simple “novato” con dedicación y esfuerzo puede alcanzar y vencer al “campeón del mundo” por lo tanto la competición no se vuelve algo inalcanzable y aburrido para los usuarios; de esta manera tenemos la atención del usuario, divirtiéndolo y enseñándole de una manera no convencional. [ELO14]

Elo	Número de jugadores	Rango equivalente
0-1399	40158	Principiante
1400-1599	10402	Aficionado
1600-1799	20398	Jugador de club medio
1800-1999	26285	Jugador de club fuerte
2000-2199	22478	Experto nacional
2200-2299	5386	Candidato a Maestro (CM)
2300-2399	2697	Maestro FIDE (MF)
2400-2499	1299	Maestro Internacional (MI)
2500-2599	554	Gran Maestro (GM)
2600-2699	173	Super Gran Maestro
2700-2799	46	Candidato a Campeón Mundial
2800 o más	3	Campeón del Mundo

Figura 2.6. Tabla comparativa del sistema de puntuación ELO para la FIDE,
Fuente [ELO14]

Además de manejar una interacción seductiva para que el usuario no se “aburra” con la aplicación, debemos de incentivarlo de alguna manera para que la participación sea un reto y no un deber, que sea de manera divertida y no una obligación para obtener solo una calificación, además, se ha observado que en ambientes amigables y de menos estrés se obtienen mejores resultados.

El ser humano además de ser curioso es competitivo por naturaleza, nunca le gusta quedarse atrás de los demás, un ejemplo y aunque sea demasiado burdo podemos verlo en las aplicaciones de las redes sociales como Facebook, esas aplicaciones que hacen que el usuario las ganas de jugar se conviertan en una necesidad ya que si no entra a la aplicación “mi amigo puede ganarme en puntuación y tengo que demostrarle que soy mejor que él” y es ese efecto viral que hace que las aplicaciones sean tan demandadas en las redes sociales.

Un ejemplo de una aplicación usada mediante los principios de ELO es:



“Tetris Battle”, el cual consiste en un juego de tetris donde compites contra un amigo tuyo en Facebook, al iniciar por primera vez esta aplicación el usuario se encuentra en el primer rango al que llaman “**nuevo**” como lo muestra la siguiente figura:



Figura 2.7. Competencia de la aplicación “Tetris Battle” en Facebook, Fuente [Facebook]

Como podemos observar en la figura, esta aplicación aplica los principios de ELO, ya que primeramente hemos sido catalogados como un “**nuevo**” dentro del juego y además no nos lanzó a una competencia “imposible” contra un contrincante más fuerte que nosotros, sino que nos lanzó a un oponente el cual tiene la misma experiencia que nosotros, de esta manera la competencia se vuelve entretenida debido a que si se gana la partida se obtiene una recompensa en puntos.

Al lograr superar el puntaje dentro del rango “**nuevo**”, ya no nos enfrentaremos a oponentes inferiores que nosotros, ahora serán oponentes que estarán dentro de nuestro nuevo rango el cual ahora es “**principiante**” y nos enfrentaremos contra oponentes que estén dentro de nuestro mismo rango y así sucesivamente, haciendo la eterna competencia por alcanzar al usuario con el puntaje insuperable de algo aburrido e imposible a algo entretenido y que se puede lograr.



Figura 2.8. Aumento de rango dentro de la aplicación “Tetris Battle” de Facebook
Fuente [Facebook]

Entonces si aterrizamos esto a algo educativo podemos crear una aplicación la cual la enseñanza se vuelva aún más divertida y para ello necesitamos tener un sistema de puntuación, pero no cualquier sistema de puntuación porque podríamos caer en el problema de los “usuarios inalcanzables” y esto haría que nuestro “usuario novato” pierda el interés en la aplicación ya que “jamás podrá alcanzar al primer lugar”, es por ello que recurrimos a el sistema de puntuación ELO.

2.7 Redes sociales como agentes de difusión masiva

Una red social es una estructura social que se puede representar en forma de uno o varios grafos en el cual los nodos representan individuos (a veces denominados actores) y las aristas representan las relaciones entre ellos. Las redes sociales de internet permiten a los individuos construir un perfil público o semipúblico dentro de los límites de la plataforma que ofrece el servicio; tienen como propósito facilitar la comunicación y otros temas

sociales, compartir fotografías, noticias, archivos de música y videos personales, entre otras cosas. Los orígenes históricos de las redes sociales se remontan a 1994-1995 cuando algunos sitios de Internet fueron añadiendo, con las capacidades técnicas de entonces y en ámbitos más bien restringidos, la oportunidad de agregar comentarios en foros, mensajería instantánea y, eventualmente, listas de amigos. El fenómeno Facebook en el 2004 dio pie al desarrollo, consolidación, masificación y proliferación de numerosas redes sociales en los meses y años subsiguientes

Dentro de las redes sociales que podemos hacer uso se encuentran las siguientes:

- **Facebook**

The Facebook logo, consisting of the word "facebook" in white lowercase letters on a blue rectangular background.

Facebook (<https://www.facebook.com>) es una de las redes sociales con más popularidad en esta década, su auge ha venido gracias a que en ella puedes poner status, compartir fotos, videos, notas, además de estar vinculada con otras redes sociales (como YouTube, Twitter, Spotify , etc.) para poder compartir el contenido de estas otras.

Creado por Mark Zuckerberg en el año de 2004, ha sido hasta ahora la red social con mayor poder en la web.

- **Twitter**



Twitter (<https://www.twitter.com>) catalogada como la red social del pajarito, está basada en el manejo de tweets (la traducción al español significa “llamada del pajarito”) en los cuales puede en un espacio menor a 160 caracteres expresar ideas, links de diferentes páginas web e incluso videos además de subir imágenes, es por ello que es caracterizada como un “microblogging”.

Su mayor fuerza en el mundo de las redes sociales es el uso de #HashTags, los cuales hacen referencia a hechos o frases muy mencionadas dentro de esa red

social (además de ser esta red social creadora de esta interacción) las cuales si tienen mucho empuje en esta red, pueden llegar a ser “Trend Topic”, los cuales son los 5 temas más hablados en ese momento.

Creado en el 2006 por Jack Dorsey, fue la red social que en algún momento se pensó que desbancaría a Facebook debido a su practicidad y fácil manejo.

- **YouTube**



YouTube (<http://www.youtube.com>) es un servidor web de videos en el cual los usuarios pueden compartir vídeos sin necesidad de descargarlos a sus dispositivos. Es una página gratuita creada en el año 2005, y adquirida por la compañía **Google** en el año 2006.

- **Flickr**



Flickr (<http://www.flickr.com>) es un sitio web gratuito que permite almacenar, ordenar, buscar, vender y compartir fotografías y videos en línea.

Actualmente Flickr cuenta con una importante comunidad de usuarios que comparte las fotografías y videos creados por ellos mismos. La popularidad de Flickr se debe fundamentalmente a su capacidad para administrar imágenes mediante herramientas que permiten al autor etiquetar sus fotografías y explorar y comentar las imágenes de otros usuarios.

- **SlideShare**



SlideShare (<http://www.es.slideshare.net>) es un sitio web que ofrece a los usuarios la posibilidad de subir y compartir en público o en privado presentaciones de diapositivas, documentos de Word, OpenOffice, PDF, entre otros.

SlideShare también proporciona a los usuarios la capacidad para evaluar, comentar, y compartir el contenido subido. Ha sido elegido en 2010 entre de los 10 mejores sitios del mundo para la enseñanza y el aprendizaje en línea.

El servicio de alojamiento es gratuito y los usuarios pueden acceder registrándose o usando el registro de su cuenta de Facebook.

- **Scribd**

usuarios  para publicar diversos formatos y embeberlos en una página web utilizando su formato iPaper.

Scribd fue fundada por Trip Adler en 2006.

Scribd (<http://www.scribd.com>) es un sitio web para compartir documentos que permite a los usuarios publicar diversos formatos y embeberlos en una

- **Blogger**



Blogger (<http://www.blogger.com>) como su nombre lo dice, es un sitio creado para el desarrollo de blogs, pero la idea fundamental de blogger es crear una bitácora de contenidos que son publicados en diferente rubros que este tiene mediante el manejo de etiquetas las cuales controlan los temas y dan acceso a los contenidos de diferentes autores siempre y cuando tengan la referencia de la etiqueta del tema creado.

Desarrollado por Pyra Labs en el 2000 y un poco más tarde adquirido por Google (2003), se ha convertido en uno de los medios masivos de divulgacion de contenidos ya sea tutoriales o simplemente como un diario de anécdotas del usuario, además que si se tenían conocimientos básicos de html estos pueden ser añadidos mediante el scriptin de esta red social

- **Wordpress**



WordPress (<http://es.wordpress.com>) desarrollado en el 2003 por Matt Mullenweg es un sitio CMS o dicho en español, un Sistema gestor de contenidos creado bajo Php y Mysql, fue la evolución de b2/cafelog, es de código abierto y es un sistema el cual ha tenido mucho auge debido a su código

abierto el cual permite a desarrolladores y programadores poder crear plugins o diversas herramientas dentro de él.

Es un sistema el cual abarca gran parte de lo hablado hasta ahora en este documento debido a ser un CMS pero destinado a la creación de Blogs es una herramienta de difusión muy poderosa en estos tiempos debido a que es reestructurable y adaptable, en pocas palabras, comprende todas las características que un sistema gestor necesita.

Las redes sociales satisfacen, en cierta forma, el deseo fundamental de las personas de entrar en relación unas con otras. Es un anhelo de comunicación y amistad que tiene su raíz en nuestra propia naturaleza humana y no puede comprenderse adecuadamente sólo como una respuesta a las innovaciones tecnológicas [Muji14].

En este contexto, las redes sociales pueden ser aprovechadas para transmitir conocimiento, Las redes sociales dan a los enseñantes una oportunidad para continuar el contacto con los alumnos, para dialogar con ellos incluso fuera del encuentro académico propio de las instituciones escolares [Bene09]

2.8 En resumen...

Después de haber abordado el tema, viendo factores tanto sociales, educativos y tecnológicos podemos ver que los objetos de aprendizaje vienen a reforzar la educación en el mundo, además de poder romper barreras para lograr los diferentes fines que se proponga la sociedad.

A continuación mostramos una comparativa de las plataformas educativas que hay actualmente, para poder así proponer un diseño completo del sistema a desarrollar

Plataforma	Amigable	Manejo de Archivos	Foros	Carga de Archivos	Mensajería Instantánea	Manejo Multimedia	Reestructurable	ON-LINE
Moodle	Normal	Sí	Sí	Sí	Sí	Depende del Alumno	No	Sí
LAMS	Mucho	Sí	Sí	Sí	Sí	Secuencial	Sí	Sí
Sakai	No	Sí	Sí	Sí	No	Depende del Alumno	Sí	No
Claroline	Normal	Sí	Sí	Sí	Sí	Depende del Alumno	Poco	Sí
OpenStax	Poco	Sí	No	Sí	No	Ninguno	No	Sí

Tabla 2.2: Comparativa de los diversos gestores de aprendizaje

Tomando ventaja de las tecnologías a nuestro alcance y las necesidades de estudiantes de tener un espacio virtual que facilite la enseñanza-aprendizaje, nace la inquietud de crear un espacio para la gestión de objetos de aprendizaje y el intercambio de ideas y conocimiento para apoyar en la educación en nuestro país.

El futuro de cada nación está en la juventud, en los niños y en la educación, entonces, si tenemos todos los medios tecnológicos como lo es el internet, redes sociales, dispositivos móviles inteligentes, ¿Por qué no apostar en la combinación de educación y tecnología para así poder tener gente competitiva para un futuro?

Capítulo 3: Modelado Conceptual

3.1 Metodología

Dentro del desarrollo de sistemas E-learning nos podemos encontrar con varias dificultades, las cuales son afrontadas mediante un paradigma basado en modelos (Model Driven Architecture "MDA") el cual ofrece a los desarrolladores Web la posibilidad de expresar una estructura, comportamiento y funcionalidad independientemente de los aspectos tecnológicos y lograr con ello la fácil integración con otros sistemas.

Este paradigma contempla cuatro componentes básicos para el desarrollo de sistemas interactivos como lo son el modelo, lenguaje, software enfoque

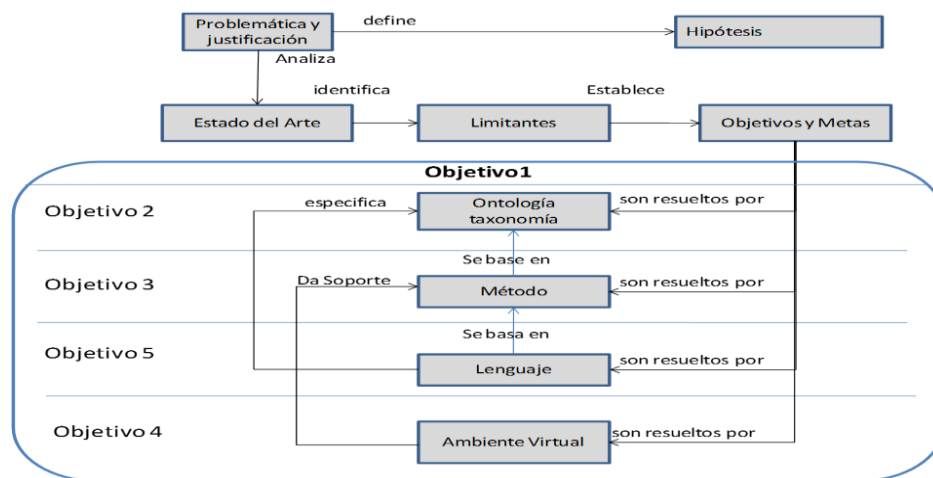


Figura 3.1: Metodología de desarrollo

Dentro de la metodología a usarse, encontramos los siguientes sujetos involucrados dentro del sistema como lo son:

- **Lenguaje:** Este es el encargado de dar una especificación de la interfaz de usuario representando diferentes modelados para que de alguna manera el computador pueda procesar además de que permite al desarrollador compartir, comunicar e intercambiar pequeños pedazos de las especificaciones los cuales permitirán a las diferentes herramientas operar en conjunto con las especificaciones del sistema. *UsiXML* es una arquitectura que permite el manejo de modelos y en conjunto con el patrón de diseño *MVC* podrá ser más escalable y rediseñable el sistema a crear.

- **Software:** Indispensable para el diseño de la arquitectura y modelado del sistema para asegurar las características del gestor mencionadas con anterioridad.
- **Modelo:** Al hablar de este tipo de modelo, no nos estamos refiriendo al que pertenece en el patrón de diseño MVC, a lo que nos enfocamos en esta parte es a todas las características que nos permitirán la creación de modelos los cuales describen los diferentes cambios o facetas que presenta la GUI entre las que podemos encontrar:
 - **Tareas**
 - **Modelos de Datos**
 - **Perfil de Usuario**
 - **Interfaz**
 - **Comportamiento**

Estos procesos se representan mediante diferentes diagramas los cuales tienen soporte en UML (Lenguaje Unificado de Modelado) para poder ayudar a la abstracción de las necesidades que requiere el sistema en un mundo real y poder plasmarlo en la programación.

- **Enfoque:** Al hablar de un enfoque nos estamos refiriendo a un paradigma ordenado el cual nos dará los pasos a seguir una metodología diseñada para el desarrollo de GUI's.

El proceso de diseño comienza con un modelo de tarea que se desarrolla a través de un enfoque gradual para al final derivar la interfaz de usuario [Cupp05]. UsiXML tiene como base el Marco de Referencia Camaleón [Calv03].

Esta metodología consta de 4 pasos dentro de su desarrollo:

1. **Tareas & Conceptos:** En esta actividad son descritas las tareas del usuario y los conceptos referentes al modelo de datos indispensables para el desarrollo pleno de las tareas.

2. **Interfaz de Usuario Abstracta (AUI):** En esta parte hacemos la abstracción de la GUI completa con respecto a la interacción; la interfaz se construye de definiciones de entradas del usuario y respuestas por parte del sistema aunque no define nada con respecto a los módulos de interacción de ésta (táctil, gráfica, sonora, etc.), además de que en este momento son definidos los contenedores de una manera abstracta en conjunto con los componentes de interacción individual, estas tareas van de la mano con los contenedores para su ejecución o por otra parte con los objetos individuales para poder tener cierta manipulación en ellos.

3. **Interfaz de Usuario Concreta (CUI):** Una CUI es definida como una modalidad de interacción, compuesta de elementos descriptivos de ella, entre estos encontramos CIO's (Objetos de Interacción concretos) con la meta de dar un significado a los componentes de diseño y navegación dentro de la interfaz; es independiente de cualquier plataforma pero puede hacer explícita la manera en la que se comporta de la GUI final, se podría decir en pocas palabras, que es el esqueleto funcional dentro de un sistema particular.
Un sinónimo a las CUI's puede ser definido como el nivel superior y abstracción de la interface final de una AUI.

4. **Interfaz de Usuario Final (FUI):** Es el resultado de todos los pasos anteriores en el cual es encargada de la funcionalidad y la operatividad de la interfaz, en resumen, la interfaz final que puede ser desplegada en cualquier plataforma.

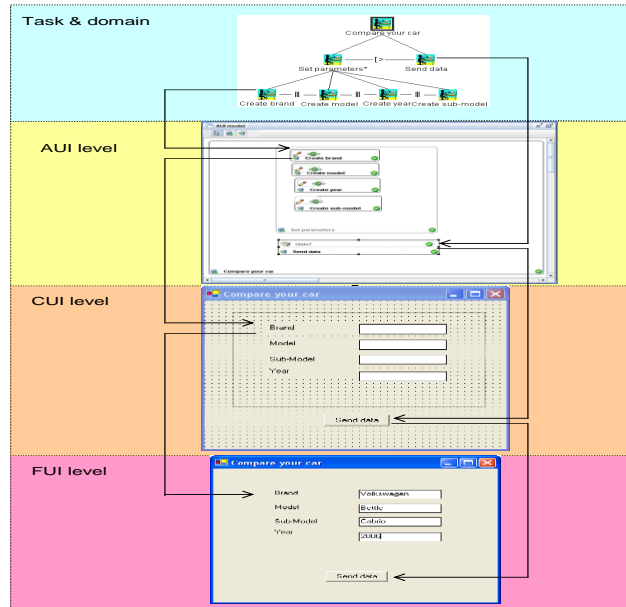


Figura 3.2: Marco de referencia
CAMELEON

3.2 Diagrama de Clases

Los diagramas de clase son una herramienta básica y fundamental para el desarrollo de un sistema en el cual, se representan los objetos fundamentales de éste, en otras palabras, son con los que interactúa el usuario para poder completar la tarea que tiene que realizar dentro del sistema.

De manera más concreta, un diagrama de clases sirve para visualizar las relaciones que hay entre las clases en las cuales se ve involucrada el sistema, dentro de ellas encontramos que pueden clasificarse en ser asociativas, de herencia, de uso y de contención.

Un diagrama de clases está compuesto por:

- **Clases:** atributos, métodos y visibilidad (alcance, scope)
- **Relaciones:** herencias, composiciones, agregaciones, asociaciones y usos.

Para este sistema se elaboró el siguiente diagrama mostrado y explicado a continuación.

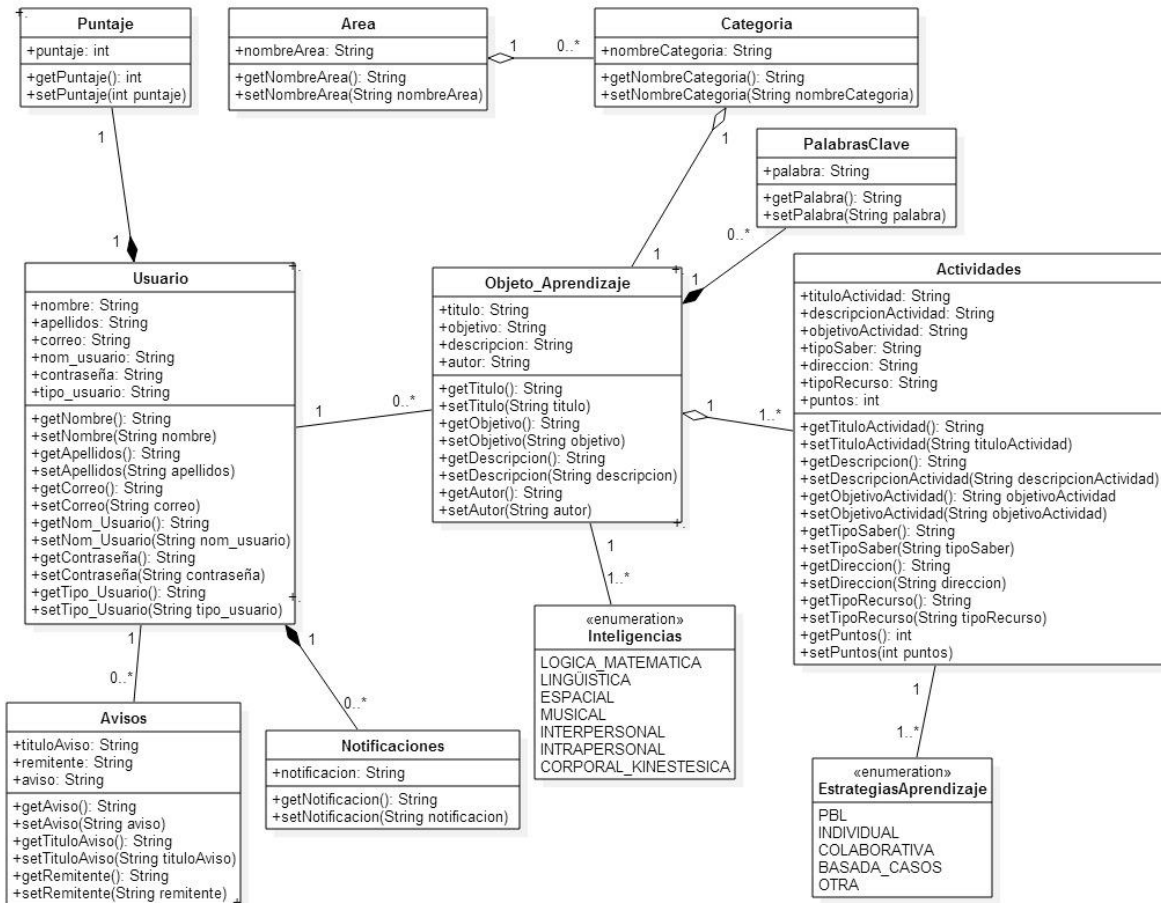


Figura 3.3 Diagrama de clases del SGOA

Fuente: Propia

Como podemos observar, tenemos tres clases que podríamos catalogar como principales ya que de ellas se basan la mayoría de las acciones realizadas dentro del sistema.

En la clase *Objeto_Aprendizaje* podemos observar que está conformada por información básica como título, objetivo, descripción y autor, dicha información es elemental para distintos propósitos, algunos para ver con mayor detalle y generar agrado al usuario consumidor del recurso (título, objetivo, descripción) y otros para poder colaborar con la indexación del recurso (título, autor).

La clase *Actividades* es de gran importancia para el sistema, ya que sin ella, la clase *Objetos_Aprendizaje* no contendría nada para la exposición y enseñanza del recurso, dicha clase se conforma de un título, descripción, objetivo para describir la actividad, además de un tipo de saber (que contendrá el tipo de saber que se quiere desarrollar en la actividad),

dirección (almacenara el nombre o url del recurso a compartir), tipo de recurso (describe que recurso se visualizará <PDF, VIDEO, IMAGEN, URL... etc.>) y puntos, estos últimos que serán obtenidos por el usuario una vez que haya finalizado su actividad

Usuarios es la clase que nos permitirá tener un registro del usuario que se haya registrado dentro del sistema, para ello es necesario solicitarle nombre, apellidos y correo y contraseña, los cuales serán almacenados en las variables con sus respectivos nombres, el campo *nom_usuario* es creado por el sistema siendo este su correo electrónico, este campo el usuario puede modificarlo en el momento que desee para que deje de ser su cuenta de correo el nombre de usuario registrado en el sistema. *Tipo_usuario* es recolectado en el registro del usuario, dicho campo, contendrá el tipo de usuario que se manejara, el cual maneja dos tipos, *Profesor* y *Alumno*.

Ahora describiremos las siguientes 6 clases de una manera más concreta, ya que, la mayoría de ellas, realizan acciones similares pero con distintos nombres de campos.

- *Avisos*: conformada por 3 variables que conformaran el cuerpo del aviso, estas tres variables son: *tituloAviso*, *remitente* y *aviso*, cada una desempeñando la labor que su nombre de variable demuestra.
- *Notificaciones*: esta clase funge como una sección de noticias ya que permite a cualquier usuario ver lo que otros usuarios han realizado, está conformada simplemente por una variable llamada *notificación*, la cual contiene el mensaje de lo que algún otro usuario ha realizado
- *Área*: formada por una sola variable, desempeña la función de guardar el área en la cual se desenvuelve el objeto de aprendizaje creado
- *Categoría*: de igual manera que la clase antes explicada, la clase *Categoría* está conformada por una variable, la cual almacenará el nombre de la categoría a la que pertenece un Objeto de aprendizaje, pero está tiene el agregado que necesita de la compañía de la clase *Área* para su correcto funcionamiento, ya que un área puede contener categorías para una mejor indexación.

- *Puntaje*: clase que contiene una variable de tipo INT la cual está encargada de almacenar el puntaje del usuario que obtiene de realizar diferentes actividades de los objetos de aprendizaje creados en el sistema.
- *PalabrasClave*: clase que contiene una variable de tipo string la cual está encargada de almacenar la palabra clave con la que se puede localizar de manera más sencilla un objeto de aprendizaje

Para casi concluir la explicación del diagrama, encontramos 2 clases más, distintas a las anteriores, ya que estas son “*Enumeraciones*” las cuales contienen constantes que solamente se escogerán en diferentes acciones del sistema.

Las enumeraciones que localizamos en el diagrama son las siguientes:

- *Inteligencias*: esta enumeración contiene el tipo de inteligencias que se quieren desarrollar en la elaboración del OA, el usuario dentro de un formulario puede visualizarlas todas y escoger la que sea necesaria para describir su objeto de aprendizaje
- *EstrategiasAprendizaje*: contiene las estrategias de aprendizaje que pueden ser desarrolladas en la actividad creada dentro de un OA, para esto, el usuario de tipo profesor puede visualizar mediante un formulario, todas las estrategias de aprendizaje disponibles y así él pueda seleccionar la estrategia que desarrolle su actividad.

Los métodos de las clases antes mencionadas realizan las funciones de modificar (set) y visualizar la información (get), cada una cumpliendo con la convención de clases que la plataforma *JAVA EE 6* propone para el manejo, control y organización de la *Programación Orientada a Objetos* dentro de un sistema WEB con patrón *MVC (Modelo-Vista-Controlador)*, dentro de dicha sintaxis se propone que los métodos que modificarán la información tendrán el prefijo “set” seguido del nombre de la variable que se modificará en cuestión (*setNombre(String nombre)*) y para los métodos que devolverán información para ser trabajada por el sistema o mostrada al usuario final llevarán el prefijo “get” de igual manera, seguido de la variable de la cual se requiere su valor. Esto se ha propuesto, ya que si en algún futuro se requiere actualizar el sistema con “*Beans*” (*otra herramienta de JSP*) facilitará dicha actualización.

3.3 Diagrama de Casos de Uso

En un sistema, podemos encontrarnos con situaciones respecto al uso del sistema y a los tipos de usuarios con los que nos podemos topar en él; para ello se presentan los *Diagramas de casos de Uso* los cuales están encargados de describir las diferentes situaciones que se pueden presentar en el sistema.

En este apartado dejaremos a un lado la palabra “*usuario*” para referirnos a la persona que maneja el sistema, cambiaremos este sustantivo por la palabra “*ACTOR*” los cuales son los participantes en las diferentes interacciones que se tienen con el sistema.

En resumen los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema, al mostrar cómo reacciona a eventos que se producen en su ámbito o en el mismo. A continuación detallaremos más los diferentes componentes que integran a un diagrama de casos de uso.

- **Actor**

Se le denomina autor a cualquier entidad que interactúa con el sistema y que no pertenece a éste, estos a su vez son catalogados mediante roles y pueden ser desde personas hasta dispositivos periféricos u otros ordenadores que participen en la interacción del sistema del que se recrea su diagrama de casos de uso.

La representación de un actor es mediante un “muñequito” hecho con bolitas y palitos el cual es llamado también “*stick-man*”



Figura 3.4: Representación gráfica de un Actor en un Diagrama de casos de Uso. Fuente: Propia

- **Caso de Uso**

Es la secuencia de interacciones entre un sistema y alguien o algo que utiliza los servicios de éste. La sintaxis que reconoce a un caso de uso es que el nombre que se le da a este, comienza por un verbo en infinitivo (terminaciones ar, er, ir) seguido de la entidad a la que se quiere hacer mención en el diagrama



Figura 3.5 Representación gráfica de *un Caso de uso*. Fuente: Propia

▪ **Relaciones**

Para poder unir los diferentes casos de uso con otros casos de uso o con un actor utilizamos líneas con diferente simbología a las que llamamos *relaciones* que, como su nombre lo dice, indica la relación que llevan los diferentes elementos de un diagrama de casos de uso. En la siguiente tabla se explican los tipos de relaciones existentes en este tipo de diagrama.

Representación	Tipo de relación
	Comunica (<<communicates>>): Relación (asociación) entre un actor y un caso de uso que denota la participación del actor en dicho caso de uso.
<<include>>	Usa (<<uses>>) (o <<include>> en la nueva versión de UML): Relación de dependencia entre dos casos de uso que denota la inclusión del comportamiento de un escenario en otro.
<<extend>>	Extiende (<<extends>>): Relación de dependencia entre dos casos de uso que denota que un caso de uso es una especialización de otro.
	Generalización es la actividad de identificar elementos en común entre conceptos y definir las relaciones de una superclase (concepto general) y subclase (concepto especializado).

Tabla 3.1 diferentes tipos de relaciones en un diagrama de casos de uso Fuente: Propia

Dentro de un diagrama de casos de uso es importante saber la descripción o especificación de este, ya que este nos dirá de qué manera interactúan los diferentes componentes del diagrama con el sistema, para ello, debemos realizarnos ciertas preguntas para de esa manera, dar respuesta al diagrama que realizaremos, entre esas preguntas se encuentran las siguientes:

- ¿Cuáles son las principales funciones o tareas realizadas por el actor?
- ¿Qué información del sistema adquiere, produce o transforma el actor?
- ¿Deberá el actor informar al sistema de los cambios producidos en el entorno?
- ¿Qué información del sistema desea el actor?
- ¿Debe informarse al actor de algún cambio inesperado?

Además, para una mejor descripción de los actores y casos de uso que se encuentren, debemos manejar una tabla, la cual retenga la información básica de los elementos mencionados con anterioridad.

Las siguientes tablas describen la manera en la cual debe de ser detallado los casos de uso o actores

Nombre:	<< Nombre del actor >>	<<identificador>>
Descripción:	<< Una breve descripción del actor>>	

Tabla 3.2 Tabla de detalles para un actor. Fuente: Propia

Nombre:	<< Nombre del CU >>	<< Identificador >>
Propósito:	<< Propósito general del CU >>	
Descripción:	<< Resumen o descripción de alto nivel del flujo normal (básico) del caso de uso >>	
Actores:	<< Listado de los actores participantes en el CU >>	
Referencias:	<<Indicamos que requisitos se pueden incluir dentro de este CU>>	<<CU que tienen relación con este>>
Precondición:	<<Condiciones sobre el estado del sistema que tienen que ser ciertas para que se pueda realizar este CU>>	
Flujo normal (Básico)		
<<Secuencia de acciones realizadas por el actor>>	<<Secuencia de acciones realizadas por el sistema>>	
Flujo alternativo: << Descripción de la secuencia de acciones alternas a las acciones del flujo normal>>		
Post-condición:	<<Efectos que provoca la realización del CU sobre el estado del sistemas>>	

Tabla 3.3 Tabla de detalles para un caso de uso. Fuente: Propia

Dentro de este sistema se han encontrado 4 actores los cuales se describirán en breve

- Visitante

Nombre:	Visitante	AC1
Descripción:	Usuario que no requiere registrarse en el sistema y que solo puede acceder a la página de inicio	

Tabla 3.4 Descripción del actor "Visitante". Fuente: Propia

- Profesor

Nombre:	Profesor	AC2
Descripción:	Es el usuario que puede generar avisos, actividades y objetos de aprendizaje dentro del sistema.	

Tabla 3.5 Descripción del actor “Profesor”. Fuente: Propia

- Alumno

Nombre:	Alumno	AC3
Descripción:	Usuario que realiza las actividades de un objeto de aprendizaje y es clasificado mediante su actividad dentro del sistema	

Tabla 3.6 Descripción del actor “Alumno”. Fuente: Propia

- SGOA

Nombre:	SGOA	AC4
Descripción:	Sistema encargado de realizar las tareas de creación, edición y eliminación de datos dentro de la plataforma	

Tabla 3.7 Descripción del actor “SGOA”. Fuente: Propia

A continuación se muestran dos de los diagramas de casos de uso realizados para este sistema, si se solicita observar los demás diagramas de casos de uso, se pueden encontrar en el apartado **A. Diagramas de casos de Uso** del Apéndice localizado al final de este documento

- Crear Objeto de Aprendizaje

Nombre:	Crear Objeto de Aprendizaje	CU3
Propósito:	Crear un nuevo OA dentro del sistema	
Descripción:	Un usuario de tipo profesor, dentro de las opciones que el sistema brinda para este tipo de usuarios, puede realizar la creación de un Objeto de aprendizaje como apoyo o herramienta a la educación que brinda mediante asesorías o el aula	
Actores:	Profesor, SGOA	

Referencias:	CU2
Precondición:	El usuario de tipo profesor debió de autenticarse dentro del sistema, seleccionar el menú “Actividades” y seleccionar el submenú “Crear OA”
Flujo normal (Básico)	
<p>1.- El usuario debe iniciar sesión como tipo de usuario “Profesor” (ver caso de uso 2 en la sección “Apéndice” al final de este documento)</p> <p>3.-El usuario deberá escoger el menú “Actividades-Crear OA”</p> <p>5.-Ingresar el título del OA</p> <p>6.-Ingresar la descripción del OA</p> <p>7.- Ingresar el objetivo del OA</p> <p>8.- Seleccionar un área para el OA</p> <p>10.-Seleccionar categoría para el OA</p> <p>11.- Presionar el botón “Siguiente”</p>	<p>2.-Si el inicio de sesión es correcto, el sistema llevará al usuario a su muro, si no lo es, enviará un mensaje de error marcando la falla del usuario</p> <p>4.- El sistema presenta un formulario solicitando los datos básicos del OA (título, descripción, objetivo)</p> <p>9.- El sistema carga todas las categorías dependiendo el área seleccionada por el usuario</p> <p>12.- Se agrega la información básica del OA y se muestra la ventana “Agregar Actividad”</p>
Flujo alternativo:	
<p>8.1.- Agregar área</p> <p>10.1.- Agregar categoría</p> <p>12.1.-El usuario deberá a partir de este momento agregar las actividades que contendrá el OA</p>	
Post-condición:	Se guarda la información básica del OA y se presenta la ventana para agregar actividades

Tabla 3.8 Descripción del caso de uso *Crear Objeto Aprendizaje*. Fuente: Propia

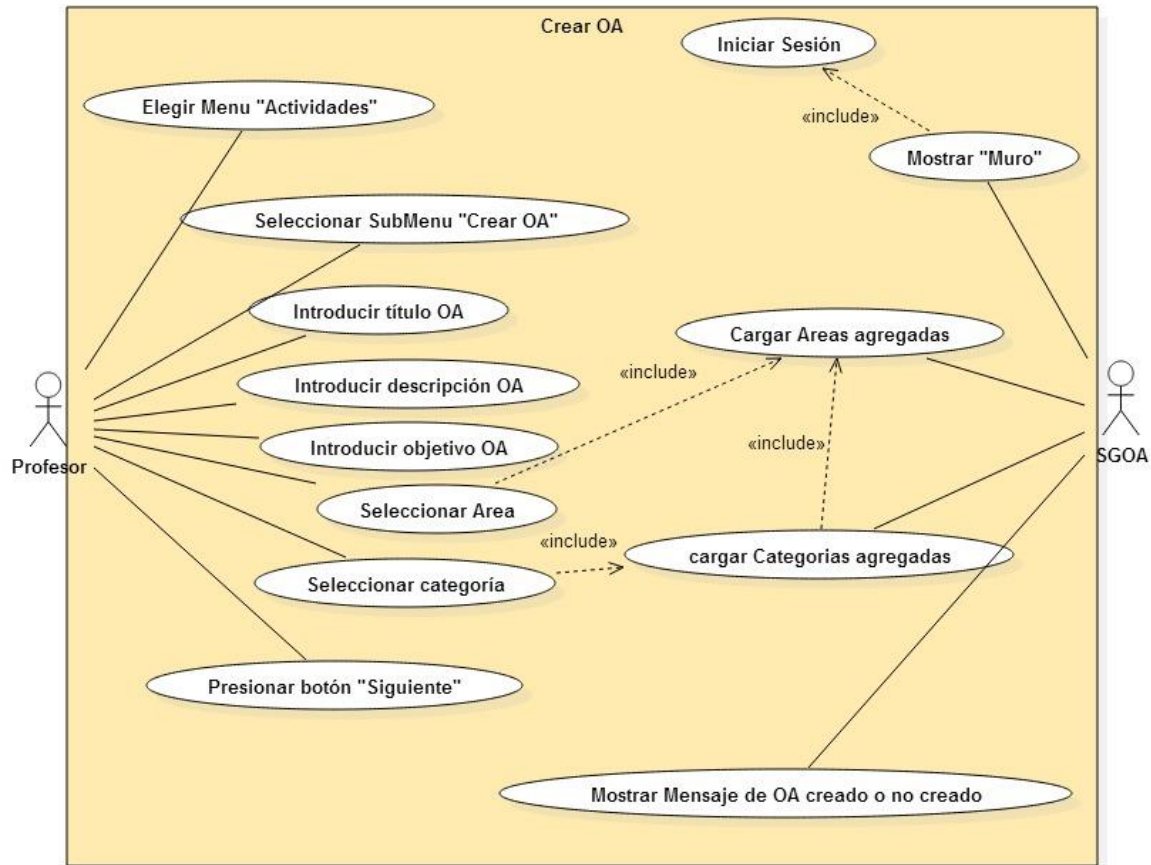


Figura 3.6 Diagrama de casos de uso "Crear OA". Fuente: Propia

- Crear Actividad

Nombre:	Crear Actividad	CU6
Propósito:	Crear una actividad dentro de un OA	
Descripción:	El usuario de tipo profesor crea un tipo de actividad dentro del objeto de aprendizaje que este creando.	
Actores:	Profesor, SGOA	
Referencias:	CU3	
Precondición:	Llenar el formato para crear un objeto de aprendizaje	
Flujo normal (Básico)		
1.- Presionar el botón del recurso que desea agregar	2.-El sistema muestra el formulario para agregar una actividad dependiendo del recurso solicitado	
3.- Ingresar título de la actividad		

<p>4.-Ingresar descripción de la actividad</p> <p>5.- Ingresar objetivo de la actividad</p> <p>6.- Ingresar puntos de la actividad</p> <p>7.- Seleccionar el tipo de saber de la actividad</p> <p>9.-Seleccionar tipo de origen del recurso digital</p> <p>11.- Subir archivo o pegar enlace del recurso que proporcionara la actividad</p> <p>12.- Seleccionar tipo de inteligencias que contendrá la actividad</p> <p>13.- Presionar botón “Agregar actividad” para finalizar la creación de ésta</p>	<p>8.-El sistema carga los tipos de saberes existentes para colocar en una actividad</p> <p>10.-Mostrar campo según el recurso digital seleccionado</p> <p>14.- Mostrar mensaje de actividad creada</p> <p>15.- Mostrar la actividad creada dentro de las actividades añadidas al OA</p>
<p>Flujo alternativo:</p> <p>15.1 Mostrar la tabla de las actividades que contiene el OA</p>	
<p>Post-condición:</p>	<p>Insertar la actividad al OA creado con anterioridad</p>

Tabla 3.9 Descripción del caso de uso *Crear Actividad*. Fuente: Propia

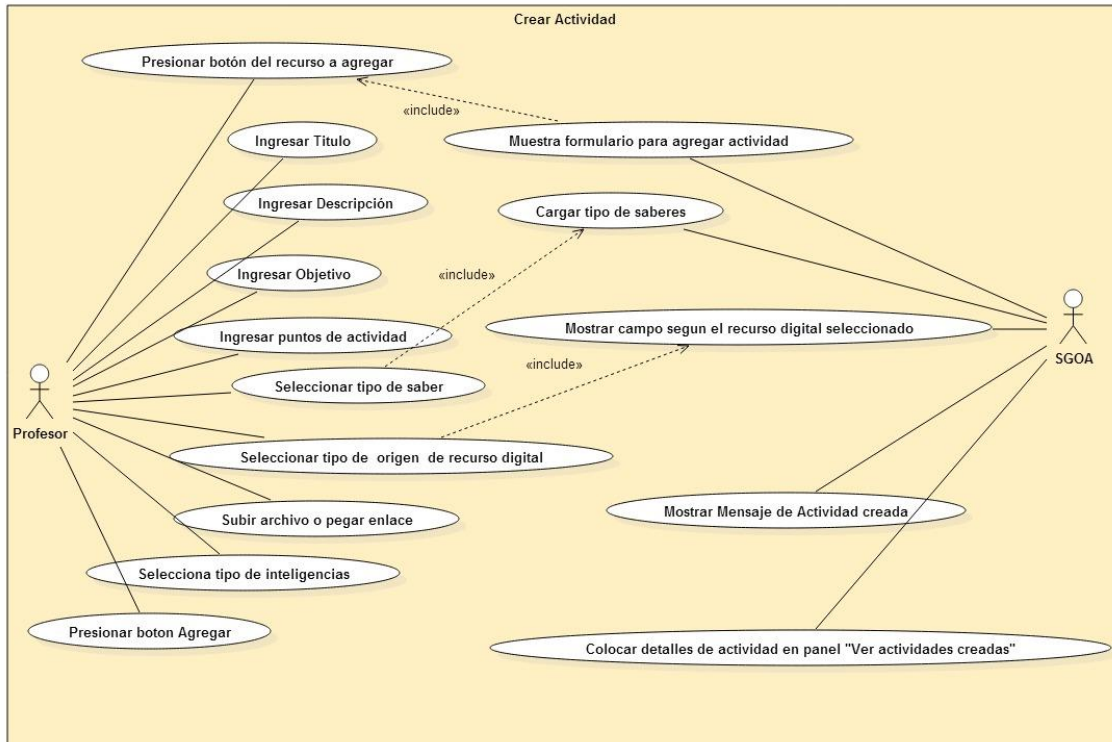


Figura 3.7 Diagrama de casos de uso "Crear Actividad". Fuente: Propia

3.4 Modelo Relacional

Todo sistema que requiere el uso de datos necesita de un mecanismo que controle la inserción eliminación y consulta (entre otras) de los datos a manejar en dicho componente, para ello es necesario una *Base de Datos* en la cual se almacenará toda la información necesaria para que el sistema pueda funcionar con normalidad.

Dentro de un sistema que solicita el apoyo de una base datos para su funcionamiento, es necesario antes de realizar ésta, elaborar un Modelo Relacional el cual se enfoca en mostrar las entidades, relaciones y propiedades que se poseen además de ser una ayuda en el modelado de los datos que se recopilarán dentro del sistema.

A continuación se presenta en la siguiente figura el Modelo Relacional perteneciente al *Sistema Gestor de Objetos de Aprendizaje*:

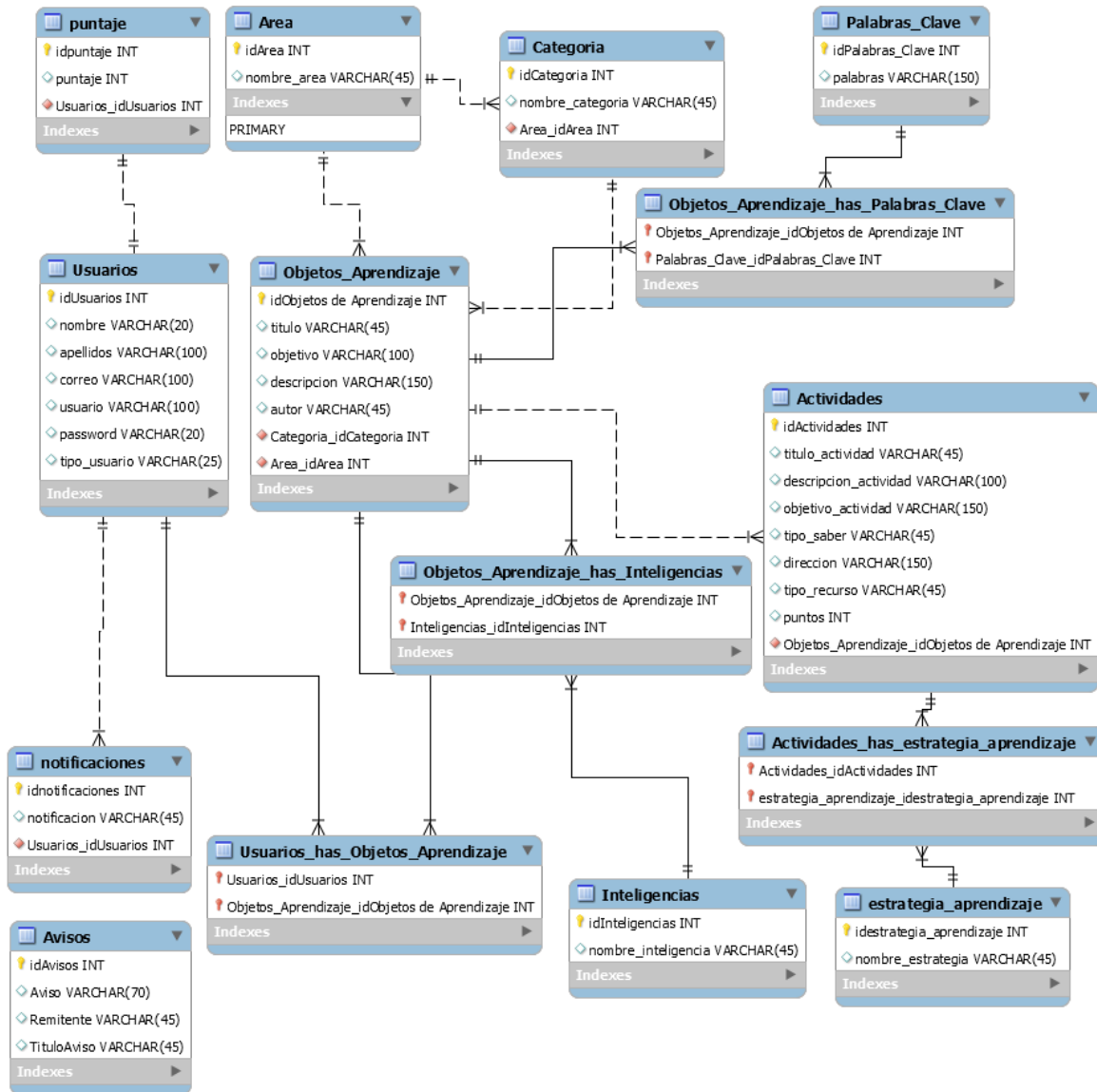


Figura 3.8 Modelo Relacional

Fuente: Propia

3.4.1 Diccionario de Datos

Una vez comprendido el modelo relacional del sistema, pasamos a la explicación de éste mediante un Diccionario de Datos el cual se encarga de detallar las entidades y propiedades de un esquema o diagrama.

Las siguientes tablas describen las diferentes entidades que componen el sistema:

- **Tabla “Objetos_Aprendizaje”**

La tabla “Objetos_Aprendizaje” almacena los datos principales del objeto de aprendizaje a crear, estos pueden ser creados solo por un profesor y tanto el como el alumno pueden observar la descripción básica de dicho objeto una vez que éste es creado.

Campo	Tipo	Tamaño	Descripción
Id_Objetos_Aprendizaje	INT	-----	Identificador de OA
Título	VARCHAR	45	Título del OA
Objetivo	VARCHAR	100	Objetivo a lograr en el desarrollo del OA
Descripción	VARCHAR	150	Descripción del OA
Autor	VARCHAR	45	Creador del OA (solo puede ser un usuario de tipo “Profesor”)

Tabla 3.10 Tabla Objetos_Aprendizaje

Fuente: Propia

- **Tabla “Usuarios”**

La tabla “Usuarios” almacena la información referente al usuario, haciendo mayor énfasis en su correo, contraseña y tipo de usuario con el cual se registra en el sistema.

Campo	Tipo	Tamaño	Descripción
idUsuarios	INT	-----	Identificador de Usuario
Nombre	VARCHAR	20	Nombre del usuario registrado
Apellidos	VARCHAR	100	Apellidos del usuario registrado
Correo	VARCHAR	100	Correo electrónico del usuario registrado
Usuario	VARCHAR	100	“Alias” con el cual puede ingresar el usuario al sistema
Password	VARCHAR	20	Contraseña con la que puede ingresar el usuario al sistema
Tipo_usuario	VARCHAR	25	Tipo de usuario con el que se puede acceder al sistema (Profesor/Alumno)

Tabla 3.11 Tabla Usuario

Fuente: Propia

- **Tabla “Puntaje”**

La tabla “*Puntaje*” almacena los puntos que el usuario obtiene siempre que realiza una actividad para así poder clasificarlo con otros usuarios que estén dentro del sistema además de poder evaluar la actividad que tiene dentro del sistema mediante la realización de actividades.

Campo	Tipo	Tamaño	Descripción
idPuntaje	INT	-----	Identificador de Puntaje
Puntaje	INT	2	Puntaje del usuario dentro del sistema

Tabla 3.12 Tabla Puntaje

Fuente: Propia

- **Tabla “Área”**

La tabla “*Área*” resguarda las áreas de los objetos de aprendizaje creados además de que pueden ser seleccionadas “n” cantidad de veces por otros objetos de aprendizaje para poder dar una mejor clasificación y búsqueda de cada uno de ellos.

Campo	Tipo	Tamaño	Descripción
idArea	INT	-----	Identificador de Área
Nombre_area	VARCHAR	45	Nombre del área a la que corresponde el Objeto de aprendizaje

Tabla 3.13 Tabla Área

Fuente: Propia

- **Tabla “Categoría”**

La tabla “*Categoría*” concentra las categorías de un objeto de aprendizaje, esta, a su vez, se relaciona con la tabla “*Área*” por lo tanto, una categoría no puede ser seleccionada sin antes escoger el área a la que pertenece.

Campo	Tipo	Tamaño	Descripción
idCategoría	INT	-----	Identificador de Categoría
Nombre_categoria	VARCHAR	45	Nombre de la categoría a la que corresponde el Objeto de Aprendizaje

Tabla 3.14 Tabla Categoría

Fuente: Propia

- **Tabla “ Palabras Clave”**

La tabla “*Palabras Clave*” retiene las palabras que son indispensables para agilizar la búsqueda de un Objeto de aprendizaje

Campo	Tipo	Tamaño	Descripción
idPalabras_Clave	INT	-----	Identificador de Palabra Clave
palabra	VARCHAR	45	Palabra clave del Objeto de Aprendizaje

Tabla 3.25 Tabla Palabras Clave

Fuente: Propia

- **Tabla “Notificaciones”**

La tabla “*Notificaciones*” concentra los mensajes de las acciones que ha realizado un usuario dentro del sistema, desde cuándo se ha creado un OA, hasta cuándo se ha finalizado una actividad para después, poder desplegarlas a los usuarios que se encuentren dentro del sistema.

Campo	Tipo	Tamaño	Descripción
idNotificacion	INT	-----	Identificador de la Notificación
notificacion	VARCHAR	45	Mensaje que se mostrará a todos los usuarios que se encuentren dentro del sistema

Tabla 3.16 Tabla Notificaciones

Fuente: Propia

- **Tabla “Avisos”**

La tabla avisos contiene pequeños mensajes que se publicarán al inicio del sistema, estos son controlados por usuarios de tipo profesor, los cuales son los únicos que pueden presentar este tipo de mensajes

Campo	Tipo	Tamaño	Descripción
idAvisos	INT	-----	Identificador de Aviso
Aviso	VARCHAR	70	Aviso publicado por un usuario de tipo profesor
Remitente	VARCHAR	45	Usuario que emite el aviso
TítuloAviso	VARCHAR	45	Encabezado del aviso a publicar

Tabla 3.16 Tabla Avisos

Fuente: Propia

- **Tabla “Actividades”**

La tabla “*Actividades*” almacena toda la información requerida para crear una actividad, un capo a destacar en esta tabla es el campo “*puntos*” en el cual se concentra los puntos que se asignaran al usuario cuando termine su actividad, de esta manera se podrá evaluar la realización de la actividad en conjunto con las notificaciones que el sistema aporte.

Campo	Tipo	Tamaño	Descripción
idActividades	INT	-----	Identificador de Actividad
Título_actividad	VARCHAR	45	nombre que se muestra de la actividad
Descripción_actividad	VARCHAR	100	Mensaje de lo que se requiere de la actividad
Objetivo_actividad	VARCHAR	150	Meta que se requiere al finalizar la actividad
Tipo_saber	VARCHAR	45	Desempeño de saber que se desea lograr con la realización de la actividad
Dirección	VARCHAR	150	Campo en el cual se coloca la dirección URL o el nombre del archivo o recurso que se incrustó en la actividad
Tipo_recurso	VARCHAR	45	Clasificación del recurso que se crea en la actividad (Video, Imagen Audio, URL)
Puntos	INT	2	Puntos que se otorgan al usuario al terminar la actividad

Tabla 3.17 Tabla Actividades

Fuente: Propia

- **Tabla “Inteligencias”**

La tabla “*Inteligencias*” contiene el nombre de las inteligencias que se requieren desarrollar en un objeto de aprendizaje

Campo	Tipo	Tamaño	Descripción
idInteligencias	INT	-----	Identificador de Inteligencias
Nombre_inteligencias	VARCHAR	45	Nombre de la inteligencia que se quiere abarcar en el OA

Tabla 3.18 Tabla Inteligencias

Fuente: Propia

- **Tabla “Estrategia_aprendizaje”**

La tabla “Estrategia_aprendizaje” recolecta el nombre de las estrategias de aprendizaje que se quieren cumplir dentro de la actividad realizada.

Campo	Tipo	Tamaño	Descripción
idEstrategia_aprendizaje	INT	-----	Identificador de Estrategia de Aprendizaje
Nombre_estrategia	VARCHAR	45	Nombre de la estrategia de aprendizaje que contendrá la actividad a realizar

Tabla 3.19 Tabla Estrategia_Aprendizaje

Fuente: Propia

3.5 En Resumen

Dentro el trayecto de este capítulo, hemos visto que herramientas serán necesarias para la creación de un SGOA, la arquitectura de software que este necesita de manera general y de manera particular abordando los objetos de aprendizaje y las actividades.

Vimos que el sistema puede ser utilizado por dos partes importantes dentro de la educación, un profesor, que será el encargado de crear objetos de aprendizaje, exponiendo las factores que quiere realizar, como las inteligencias que quiere desarrollar dentro de él, actividades mediante recursos digitales que a lo largo de los últimos años se ha vuelto una herramienta muy poderosa para el aprendizaje, y el alumno, que es el público que más exigirá en este medio, es por ello que dentro de esta arquitectura, se ha propuesto un mecanismo de competencia mediante la asignación de puntos al terminar de realizar la actividad y de esta manera poder ser clasificado dentro de una competencia sana.

Una vez concluidos los diagramas antes presentados, podemos empezar a abordar el diseño y las herramientas de desarrollo que serán necesarias para la creación de la plataforma para además, cumplir con las necesidades que las TIC y el e-learning requieren, además de sacar el máximo beneficio que las herramientas multimedia (videos, imágenes, audios etc.) para el enriquecimiento educativo y de la plataforma.

Para concluir el capítulo, dentro de la arquitectura, como hemos visto, se ha propuesto un sistema de reutilización de recursos, ya que cualquier alumno puede acceder a cualquier OA creado, sea o no realizado por un profesor con el cual tenga un contacto o retroalimentación, además de que el profesor, también puede seleccionar el OA creado sin que haya sido realizado por el, para que sus alumnos puedan realizarlo, ya dependerá del profesor, y del sistema educativo en el que se encuentre, tomarlo como un apoyo como complemento a su sistema de enseñanza o como un medio absoluto en donde él solo tenga que dar las conclusiones y retroalimentaciones del tema

Capítulo 4: Diseño

4.1 Herramientas tecnológicas para la creación del gestor

Por lo mismo de que en este documento hemos estado hablando de un sistema Gestor, este mismo será desarrollado como una aplicación, para ello, usaremos tecnologías amigables tanto para un programador como para un diseñador debido a los lenguajes que utilizaremos y mencionaremos a continuación con más detalle

4.1.1 Patrón de diseño MVC

Se ha decidido incluir el modelo MVC (Modelo-Vista-Controlador) el cual es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, los cuales suelen manejarse últimamente como un paradigma de programación por capas en el cual encontramos 3 capas, las cuales son la capa de presentación, la capa de negocio y la capa de datos.

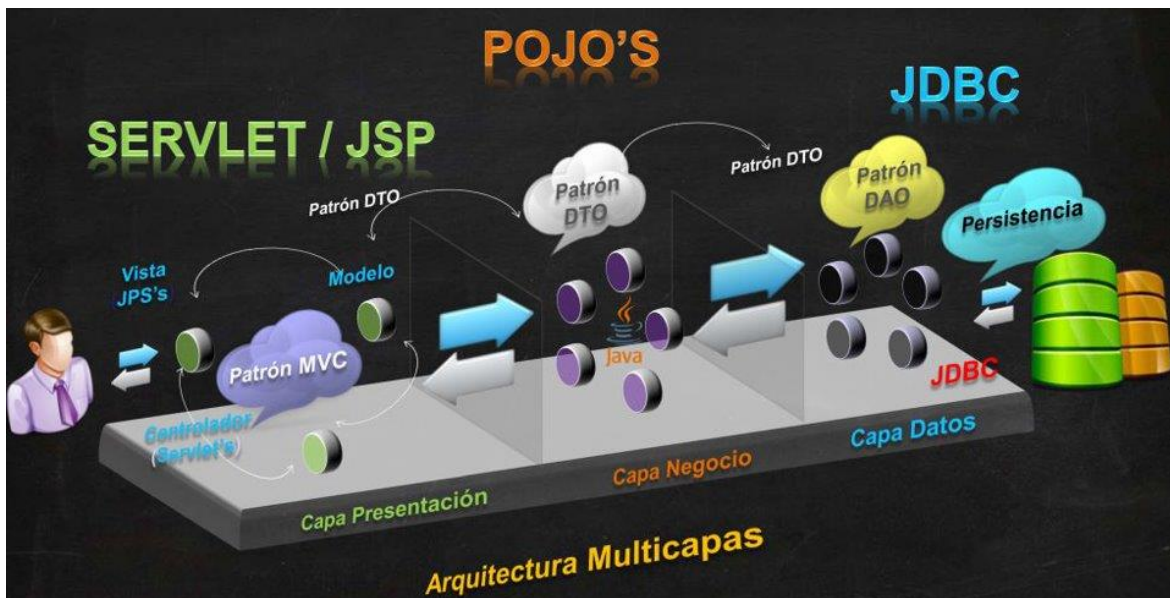


Figura 4.1. Arquitectura Multicapas en Java. Fuente: Propia

MVC no pretende discriminar entre capa de negocio y capa de presentación pero si pretende separar la capa visual gráfica de su correspondiente programación y acceso a datos, algo que mejora el desarrollo y mantenimiento de la Vista y el Controlador en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí.

Dentro de este patrón podemos ver las diferentes subcapas y sus respectivas tareas, los cuales son:

1. Modelo:

- Es el responsable de acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: “Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor”.
- Lleva un registro de las vistas y controladores del sistema.
- Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero bath que actualiza los datos, un temporizador que desencadena una inserción, etc.).
- A veces puede necesitar de una BD u otro sistema.

2. Controlador:

- Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos, del tipo “SI Evento Z, entonces Acción W”. Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método “Actualizar ()”. Una petición al modelo puede ser “Obtener_tiempo_de_entrega (nueva_orden_de_venta)”.

3. Vista:

- Recibir datos del modelo y los muestra al usuario en formato HTML.
- Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- Pueden dar el servicio de “Actualización()”, para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).[MVC14]

En resumen:

- La vista es la página HTML y el código que provee de datos dinámicos a la página
- Es responsable de tomar los resultados de la computación desde el modelo y presentarlo al usuario, en algunos sistemas el “usuario” tal vez no es humano pero tal vez puede ser otro sistema de cómputo, eso no altera lo esencial del diseño.
- El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio
- Es un modelo computacional del problema a ser resuelto esto no tiene en cuenta como se proporciona la entrada ni la presentación de la salida
- El controlador es el responsable de recibir los eventos de entrada desde la vista, además de ser el responsable de tomar la salida y pre-procesarla, puede ser presentada al modelo en una forma genérica que es apropiada para el modelo. En un sistema basado en web, el controlador extrae la esencia de la petición y los datos de los argumentos de la petición HTTP.

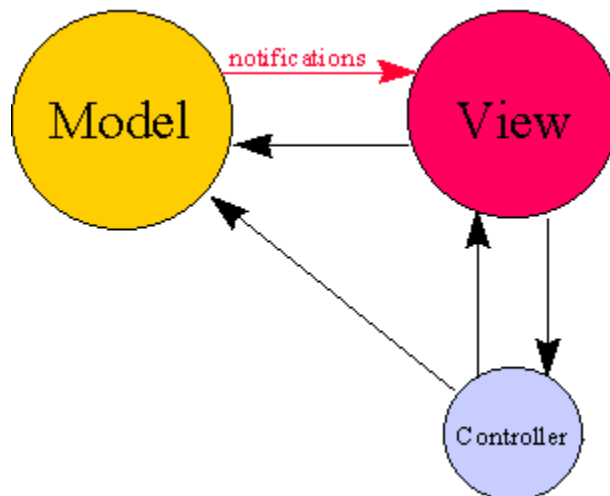


Figura 4.2 Funcionalidad de: Modelo-Vista-Controlador. Fuente: Propia

El realizar esta combinación de tecnologías puede dar como resultado un sistema más moldeable y robusto, ya que por un lado se desarrolla una ingeniería de software bajo un marco teórico adaptable a tecnologías y un desarrollo bajo un paradigma que en los últimos años se ha manejado en los sistemas web.

4.1.2 Java Server Pages (JSP)

JavaServer Pages (JSP) es una tecnología que ayuda a los desarrolladores de software a crear páginas web dinámicas basadas en HTML, XML, entre otros tipos de documentos. JSP es similar a PHP, pero usa el lenguaje de programación Java.

Para desplegar y correr JavaServer Pages, se requiere un servidor web compatible con contenedores servlet como Apache Tomcat o Jetty.

El rendimiento de una página JSP es el mismo que tendría el servlet equivalente, ya que el código es compilado como cualquier otra clase Java. A su vez, la máquina virtual compilará dinámicamente a código de máquina las partes de la aplicación que lo requieran. Esto hace que JSP tenga un buen desempeño y sea más eficiente que otras tecnologías web que ejecutan el código de una manera puramente interpretada.

La principal ventaja de JSP frente a otros lenguajes es que el lenguaje Java es un lenguaje de propósito general que excede el mundo web y que es apto para crear clases que manejen lógica de negocio y acceso a datos de una manera prolija. Esto permite separar en niveles las aplicaciones web, dejando la parte encargada de generar el documento HTML en el archivo JSP.

Otra ventaja es que JSP hereda la portabilidad de Java, y es posible ejecutar las aplicaciones en múltiples plataformas sin cambios. Es común incluso que los desarrolladores trabajen en una plataforma y que la aplicación termine siendo ejecutada en otra.

Los servlets y Java Server Pages (JSPs) son dos métodos de creación de páginas web dinámicas en servidor usando el lenguaje Java. En ese sentido son similares a otros métodos o lenguajes tales como el PHP, ASP o los CGIs, programas que generan páginas web en el servidor. Sin embargo, se diferencian de ellos en otras cosas.

Para empezar, los JSPs y servlets se ejecutan en una máquina virtual Java, lo cual permite que, en principio, se puedan usar en cualquier tipo de ordenador, siempre que exista una máquina virtual Java para él. Cada servlet (o JSP, a partir de ahora lo usaremos de forma indistinta) se ejecuta en su propio hilo, es decir, en su propio contexto; pero no se comienza

a ejecutar cada vez que recibe una petición, sino que persiste de una petición a la siguiente, de forma que no se pierde tiempo en invocarlo (cargar programa + intérprete). Su persistencia le permite también hacer una serie de cosas de forma más eficiente: conexión a bases de datos y manejo de sesiones, por ejemplo.

Las JSPs son en realidad una forma alternativa de crear servlets ya que el código JSP se traduce a código de servlet Java la primera vez que se le invoca y en adelante es el código del nuevo servlet el que se ejecuta produciendo como salida el código HTML que compone la página web de respuesta.

Los JSP antes de ser ejecutados por primera vez, son transformados a un equivalente de un Servlet, esto es similar a la fase de compilación, esta transformación es creada automáticamente por el contenedor web, mientras que los servlets son más difíciles de dar manteniendo los JSP tiene la ventaja de que su mantenimiento es mucho más sencillo[OCEJWCDEE614]

4.1.3 Java Servlets

El servlet es una clase en el lenguaje de programación Java, utilizada para ampliar las capacidades de un servidor. Aunque los servlets pueden responder a cualquier tipo de solicitudes, éstos son utilizados comúnmente para extender las aplicaciones alojadas por servidores web, de tal manera que pueden ser vistos como applets de Java que se ejecutan en servidores en vez de navegadores web. Este tipo de servlets son la contraparte Java de otras tecnologías de contenido dinámico Web, como PHP y ASP.NET.

La palabra servlet deriva de otra anterior, applet, que se refiere a pequeños programas que se ejecutan en el contexto de un navegador web.

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

Un Servlet es similar a un programa CGI, responde a las peticiones del protocolo HTTP y se ejecuta dentro de un servidor. Los tipos de tareas que se pueden realizar con servlets son los mismos que se pueden hacer con un CGI pero la arquitectura será diferente.

Los procesos básicos de un Java servlet son similares a los de CGI, el Servlet corre en un hilo dentro de un contenedor web separado de los procesos del sistema operativo; el contenedor web es por sí mismo un proceso del sistema operativo pero corre como un

servicio y es disponible continuamente, cosa contraria a un script CGI en el cual es creado un proceso para cada petición.

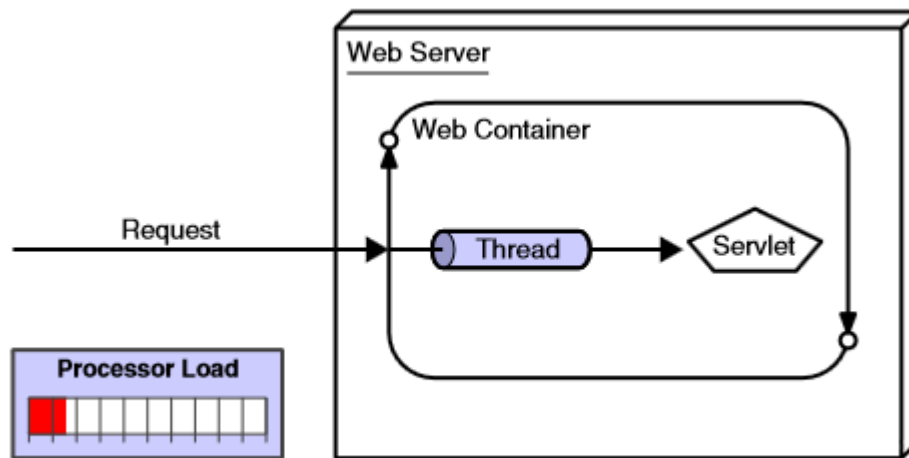


Figure 1-4 Running a Single Instance of a Servlet

Figura 4.3: Ejecución de un Servlet como un hilo
Dentro del contenedor web de procesos. Fuente: OCEJWCD

Cuando el número de peticiones para un servlet se eleva, no se crean nuevos procesos o se instancian nuevos servlets, cada petición es procesada concurrentemente usando un hilo por petición. A continuación veremos como ocurre este proceso antes mencionado

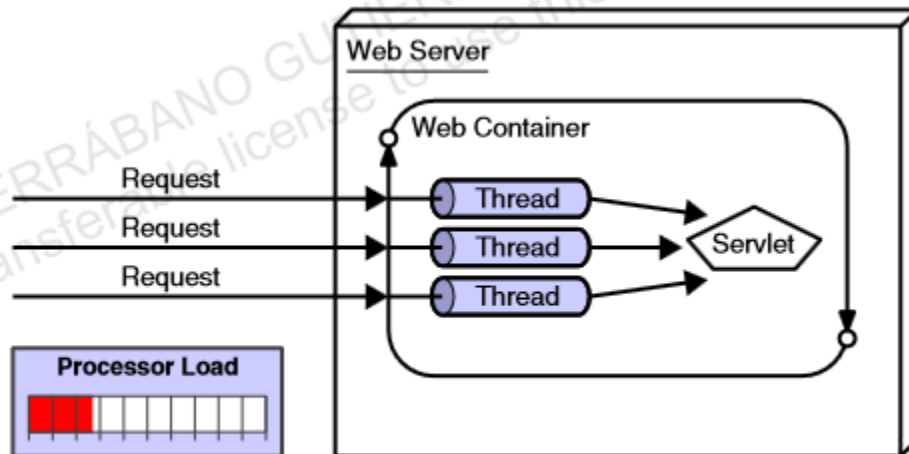


Figure 1-5 Running Multiple Instances of a Servlet

Figura 4.4: Ejecución múltiple de instancias en un Servlet. Fuente: OCEJWCD

Las ventajas de utilizar servlets en este proyecto son las siguientes:

- Cada ejecución corre en un hilo diferente dentro de un proceso, además de que las peticiones dentro del ámbito de los servlets son mucho más rápidos que a comparación con los CGI.
- Los servlets son más escalables que los programas diseñados dentro de CGI, muchísimas más peticiones pueden ser ejecutadas porque el contenedor web usa un hilo preferentemente a usar un proceso del S.O. además de que los hilos son ligeros y el S.O. del host puede soportar más de uno.
- Son simples, robustos, de plataforma independiente y Orientado a objetos lo cual es la naturaleza del lenguaje Java
- Los servlets tienen acceso a ser estandarizados y fáciles de usar para el “Log in” de cualquier servicio Web
- El contenedor web de los servlets da servicios a estos mismos de seguridad y manejo de errores.

Los servlets pueden ejecutarse dentro de la plataforma JAVA EE dentro de una arquitectura de contenedor de componentes, este contenedor de componentes es llamado contenedor web (o también conocido como mecanismo de servlets).

El contenedor web es una tecnología de Java que implementa una interface programable de aplicaciones para servlets. Los servlets son componentes que responden a las peticiones HTTP. El contenedor web se ejecuta y después selecciona el Servlet manejando la petición dada, el contenedor web aparte controla el ciclo de vida del Servlet

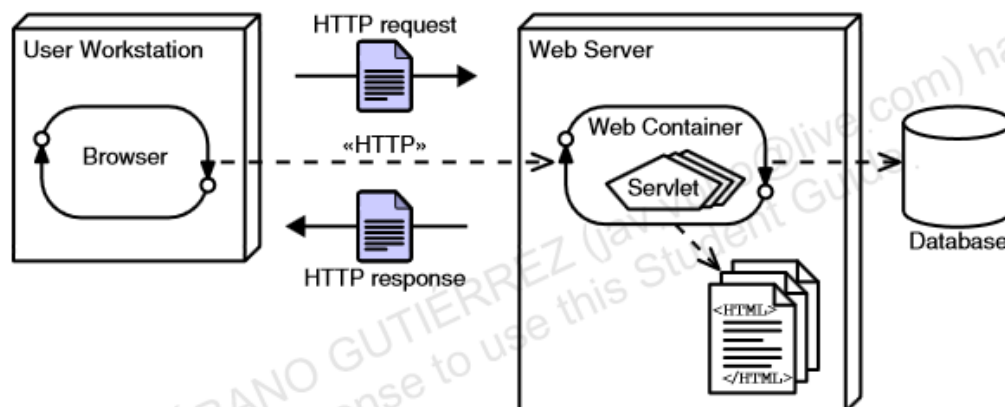


Figura 4.5: Arquitectura Web Server con Servlets. Fuente: OCEJWCD

Algunas arquitecturas el contenedor web trabaja como un contenedor de servicios http "StandAlone" y en otras arquitecturas el servicio HTTP las peticiones son a corto plazo y el que tiene que ejecutarlas es el contenedor web

Un Servlet es invocado por el contenedor el cuándo una petición apropiada es recibida por el contenedor web, la URL o URL PATTERN especificara que es usado para responder a 'x'.

Las tareas de los servlets son 2:

1. Debe de realizar lo requerido por el cliente
2. Debe de mostrar el resultado en una página HTML con un buen formato [OCEJWCDEE614]

4.1.4 HTML

HTML, siglas de HyperText Markup Language (Lenguaje de Marcado de Hipertexto), es el lenguaje de marcado para la elaboración de páginas web. Es usado para describir la estructura y el contenido de una página web en forma de texto, también permite complementar su contenido con objetos tales como imágenes, videos, audio, animaciones etc.

HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (< >). El lenguaje HTML es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado World Wide Web Consortium, más conocido como W3C. El propio W3C define el lenguaje HTML como "un lenguaje reconocido universalmente y que permite publicar información de forma global".

HTML sirve únicamente para transmitir la información que contiene un documento, dejando para hojas de estilo (CSS) y JavaScript su aspecto y diseño en distintos medios.

4.1.5 MYSQL

Es un sistema administrativo relacional de bases de datos (RDBMS por sus siglas en inglés Relational Database Management System). Este tipo de bases de datos puede ejecutar desde acciones tan básicas, como insertar y borrar registros, actualizar información o hacer consultas simples, hasta realizar tareas tan complejas como la aplicación lo requiera.

MySQL es un servidor multi-usuarios muy rápido y robusto de ejecución de instrucciones en paralelo, es decir, que múltiples usuarios distribuidos a lo largo de una red local o Internet podrá ejecutar distintas tareas sobre las bases de datos localizadas en un mismo servidor.

Utiliza el lenguaje SQL (Structured Query Language) que es el estándar de consulta a bases de datos a nivel mundial.

4.1.6 CSS

Las hojas de estilo en cascada (Cascading Style Sheets, CSS), es un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C es el encargado de formular la especificación de las hojas de estilo que servirá de estándar para los agentes de usuario (navegadores web).

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. La información de estilo puede ser adjuntada tanto como un documento separado o en el mismo documento HTML.

Los estilos definen la forma de mostrar los elementos. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a ella en las que aparezca ese elemento. De esta forma, CSS permite controlar el estilo y formato de múltiples páginas Web al mismo tiempo. CSS funciona a base de reglas, esto es, declaraciones sobre el estilo de uno o más elementos.

4.1.7 Lenguaje de Expresiones (LE)

El lenguaje de expresiones es un lenguaje simple diseñado para conocer las necesidades de una aplicación web, aparte de ser visto como un evaluador de expresiones o como un expresión de literales, además de aceptar expresiones compuestas donde múltiples expresiones de LE's son agrupadas en un solo conjunto. [ExpLe14]

El Lenguaje de Expresiones (LE) provee acceso a los atributos de un mapa mediante una simple sintaxis que es similar a las usadas por UNIX en sus variables de desarrollo además de que va de la mano con las Java Server Pages (JSP)

Estas Expresiones son utilizadas para el manejo de los datos entre las capas del patrón MVC, es muy eficiente y de un mecanismo flexible que permite acceso a los datos almacenados en los atributos de la petición además de otros no introducidos aún. [OCEJWCDEE614]

4.1.8 JSTL

Java Standard Tag Library, (JSTL) es un conjunto de etiquetas que se relacionan con JSP y el patrón de diseño MVC, encapsulan en una simple etiqueta todas las funcionalidades comunes para varias aplicaciones Web, tiene soporte para tareas comunes y estructurales como interacciones y condicionales, etiquetas para la manipulación de XML, etiquetas internacionalizadas y además de etiquetas para el manejo de SQL.

Las JSTL tienen mucha relación con el Lenguaje de Etiquetas (LE) que fue desarrollado para ayudar a las JSP 2.1; gracias a la unificación del LE, las etiquetas JSTL pueden interactuar con ellas. [OCEJWCDEE614]

Por ser un componente web, pertenece a una aplicación web como se muestra en la siguiente figura:

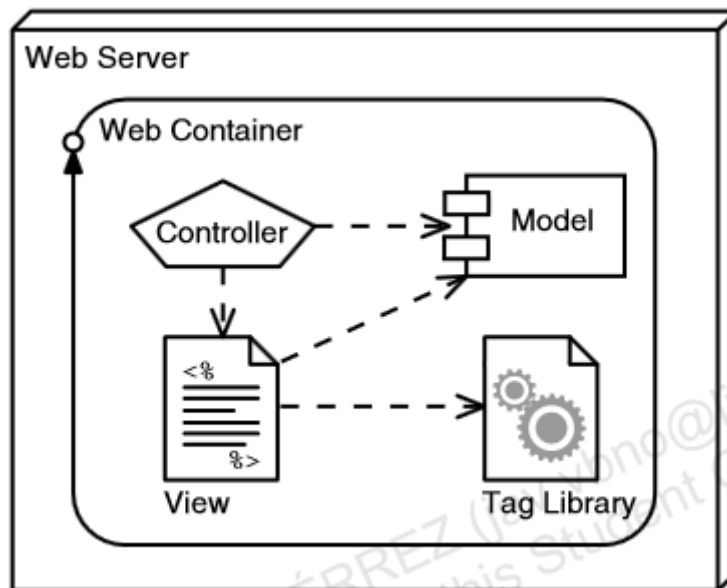


Figura 4.6: Mecanismo y funcionamiento de una JSTL dentro del patrón de diseño MVC de una aplicación Web. Fuente: OCEJWCD

4.1.9 JDBC

Java Database Connectivity, más conocida por sus siglas **JDBC**, es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

El API JDBC se presenta como una colección de interfaces Java y métodos de gestión de manejadores de conexión hacia cada modelo específico de base de datos. Un manejador de conexiones hacia un modelo de base de datos en particular es un conjunto de clases que implementan las interfaces Java y que utilizan los métodos de registro para declarar los tipos de localizadores a base de datos (URL) que pueden manejar. Para utilizar una base de datos particular, el usuario ejecuta su programa junto con la biblioteca de conexión apropiada al modelo de su base de datos, y accede a ella estableciendo una conexión; para ello provee el localizador a la base de datos y los parámetros de conexión específicos. A partir de allí puede realizar cualquier tipo de tarea con la base de datos a la que tenga permiso: consulta, actualización, creación, modificación y borrado de tablas, ejecución de procedimientos almacenados en la base de datos, etc.

4.1.10 Apache Tomcat

Apache Tomcat (también llamado Jakarta Tomcat o simplemente Tomcat) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems.

Tomcat es mantenido y desarrollado por miembros de la Apache Software Foundation y voluntarios independientes. Los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la Apache Software License. Las primeras distribuciones de Tomcat fueron las versiones 3.0.x. Las versiones más recientes son las 7.x, que implementan las especificaciones de Servlet 3.0 y de JSP 2.2. A partir de la versión 4.0, Jakarta Tomcat utiliza el contenedor de servlets Catalina.

Tomcat es un servidor web con soporte de servlets y JSPs. Tomcat no es un servidor de aplicaciones, como JBoss o JOnAS. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de

transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

Las características de Tomcat son las siguientes:

- Recarga de servlets
- Funciones básicas HTTP
- Contenedor de servlets rediseñado como Catalina.
- Motor JSP rediseñado con Jasper.
- Conector Coyote
- Java Management Extensions (JMX), JSP Y administración basada en Struts.
- Recolección de basura reducida.
- Capa envolvente nativa para Windows y Unix para la integración de las plataformas.
- Análisis rápido JSP
- Soporte para Unified Expression Language 2.1
- Diseñado para funcionar en Java SE 5.0 y posteriores.
- Soporte para Comet a través de la interfaz CometProcessor.
- Mejoras para detectar y prevenir "fugas de memoria" en las aplicaciones web.
- Limpieza interna de código.
- Soporte para la inclusión de contenidos externos directamente en una aplicación web.

4.1.11 Java Script

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.

Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas

escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java. Legalmente, JavaScript es una marca registrada de la empresa Sun Microsystems, como se puede ver en <http://www.sun.com/suntrademarks/>.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Una cuarta edición está en desarrollo e incluirá nuevas características tales como paquetes, espacio de nombres y definición explícita de clases.

4.1.12 Bootstrap

Twitter Bootstrap es una colección de herramientas de software libre para la creación de sitios y aplicaciones web. Es el proyecto más popular en GitHub y es usado por la NASA y la MSNBC junto además organizaciones.

Fue desarrollado por Mark Otto y Jacob Thornton de Twitter, como un framework para fomentar la consistencia a través de herramientas internas. Antes de Bootstrap, se usaban varias librerías para el desarrollo de interfaces de usuario, las cuales guiaban a inconsistencias y a una carga de trabajo alta en su mantenimiento.

Bootstrap tiene un soporte relativamente incompleto para HTML5 y CSS 3, pero es compatible con la mayoría de los navegadores web. La información básica de compatibilidad de sitios web o aplicaciones está disponible para todos los dispositivos y navegadores. Existe un concepto de compatibilidad parcial que hace disponible la información básica de un sitio web para todos los dispositivos y navegadores. Por ejemplo, las propiedades introducidas en CSS3 para las esquinas redondeadas, gradientes y sombras son usadas por Bootstrap a pesar de la falta de soporte de navegadores antiguos. Esto extiende la funcionalidad de la herramienta, pero no es requerida para su uso.

Desde la versión 2.0 también soporta diseños sensibles. Esto significa que el diseño gráfico de la página se ajusta dinámicamente, tomando en cuenta las características del dispositivo usado (Computadoras, tabletas, teléfonos móviles).

Bootstrap es de código abierto y está disponible en GitHub. Los desarrolladores están motivados a participar en el proyecto y a hacer sus propias contribuciones a la plataforma.

4.1.13 WebSockets

La especificación WebSocket (llevada a cabo por la IETF) provee un canal de comunicación bidireccional full-duplex sobre un único socket TCP entre el navegador (cliente) y el servidor, enviando y recibiendo mensajes de manera simultánea. El cliente puede enviar datos al servidor por este canal y el servidor puede enviar datos al cliente sin necesidad de que éste realice una petición para solicitar datos. Se establece una única conexión entre cliente y servidor que permite que cualquiera de los dos actores pueda enviar mensajes al otro en cualquier momento, sobre todo para aplicaciones que requieren constantes actualizaciones en el front-end debido a la interacción de terceras personas (o sistemas). Son las denominadas "aplicaciones en tiempo real", que son aquellas en las que el cliente puede estar informado de todo aquello que sucede en el sistema desde el mismo momento en que se produce un cambio. Ejemplos de este tipo de aplicaciones pueden ser: juegos multijugador, aplicaciones de monitorización, chats, herramientas de trabajo colaborativo, etc...

La API de WebSocket está siendo normalizada por el W3C, mientras que el protocolo WebSocket ya fue normalizado por la IETF como el RFC 6455. Debido a que las conexiones TCP comunes sobre puertos diferentes al 80 son habitualmente bloqueadas por los administradores de redes, el uso de esta tecnología proporcionaría una solución a este tipo de limitaciones proveyendo una funcionalidad similar a la apertura de varias conexiones en distintos puertos, pero multiplexando diferentes servicios WebSocket sobre un único puerto TCP (a costa de una pequeña sobrecarga del protocolo).

Antes de la aparición de WebSockets se utilizaban dos técnicas para implementar este comportamiento:

- **AJAX polling:** Consiste en realizar constantemente peticiones al servidor preguntándole si se ha producido algún evento que requiera una actualización en la vista. Por ejemplo: en un chat, desde la pantalla donde llegan los posibles mensajes

que recibe el usuario por parte de otros usuarios, se estarían enviando peticiones HTTP al servidor (cada 2 segundos, 5, 10, o lo que sea) preguntándole preguntando constantemente por alguna actualización. Esta es una solución bastante ineficiente debido a que se genera una gran cantidad de tráfico con el servidor, sobre todo en aplicaciones con un elevado número de usuarios e intervalo entre peticiones pequeño. Además, pensemos que la mayoría de las veces el servidor responderá lo mismo: "no, no hay cambios".

- **Comet o "long polling"**: Es una técnica muy parecida a la que utilizan los WebSockets. Consiste en realizar una única petición al servidor de forma que éste responde diciendo que va a devolver la respuesta en "trozos" (streaming). La petición queda abierta hasta que el servidor responda con todas las porciones de respuesta que solicita el cliente, que no son otra cosa que eventos que se producen en el servidor notificando un cambio de estado en el cliente. Como digo, es parecido a lo que hacen los WebSockets, sin embargo su implementación es bastante más compleja.

En el lado del cliente, WebSocket está ya implementado en Mozilla Firefox 8, Google Chrome 4 y Safari 5, así como la versión móvil de Safari en el iOS 4.2.1 y en Internet Explorer 10.2

WebSockets es una especificación relativamente nueva pero que cada vez va sonando con más fuerza de la mano de eso a lo que conocemos como HTML5.

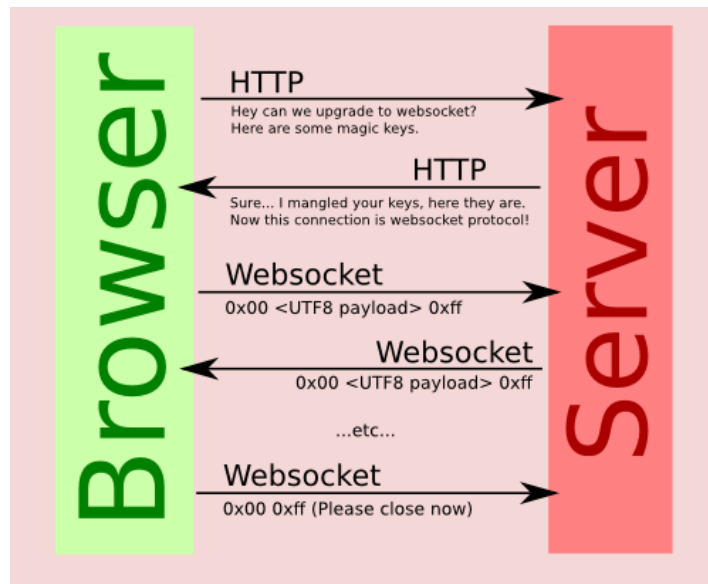


Figura 4.7 Ciclo de Vida de un WebSocket

Fuente (ApuntesJava.com)

4.1.14 JQuery

jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. JQuery es la biblioteca de JavaScript más utilizada.

JQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos.² jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

Las empresas Microsoft y Nokia anunciaron que incluirán la biblioteca en sus plataformas.³ Microsoft la añadirá en su IDE Visual Studio⁴ y la usará junto con los frameworks ASP.NET AJAX y ASP.NET MVC, mientras que Nokia los integrará con su plataforma Web Run-Time. La característica principal de la biblioteca es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX.

4.1.15 JSR356 (Java)

JSR 356, API de Java para WebSocket, especifica la API que los desarrolladores de Java pueden utilizar cuando quieren integrar WebSockets en sus aplicaciones -tanto a nivel de servidor, así como en el lado del cliente Java. Cada implementación del protocolo WebSocket que pretende ser compatible con JSR 356 debe implementar esta API. Como consecuencia de ello, los desarrolladores pueden escribir sus aplicaciones basadas en WebSocket independientes de la aplicación WebSocket subyacente. Este es un gran beneficio, ya que evita que un vendedor de bloqueo y permite más opciones y la libertad de las bibliotecas y los servidores de aplicaciones.

JSR 356 es una parte de la próxima estándar de Java EE 7; por lo tanto, todos los servidores de aplicaciones 7 compatible con Java EE tendrán una implementación del protocolo WebSocket que se adhiere al estándar JSR 356. Una vez que se hayan establecido, el cliente WebSocket y compañeros de servidor son simétricos. La diferencia entre un API cliente y un servidor de API es, por lo tanto, mínima. JSR 356 define una API de cliente de Java, así, que es un subconjunto de la API completa requerida en Java EE 7.

Una aplicación cliente-servidor aprovechando WebSockets contiene típicamente un componente de servidor y uno o más componentes de cliente, como se muestra en la siguiente figura:

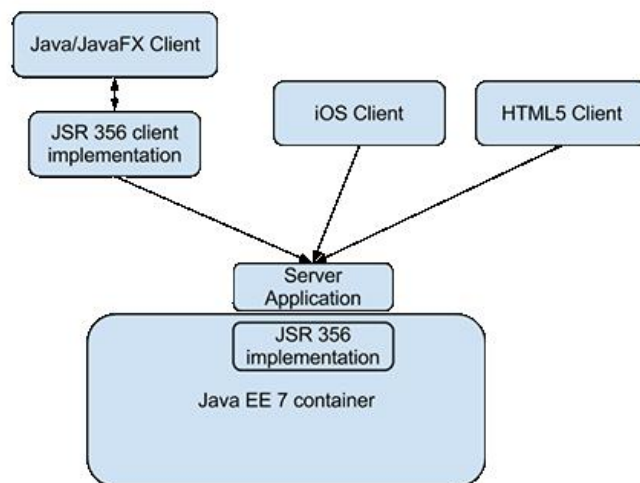


Figura 4.8 Procedimiento de un WebSocket JSR 356 de Java

Fuente (Oracle)

En este ejemplo, la aplicación de servidor está escrito en Java, y los detalles del protocolo WebSocket son manejados por el 356 implementación JSR contenido en el contenedor Java EE 7.

Un cliente JavaFX puede confiar en cualquier aplicación cliente-356 compatible con JSR para el manejo de las cuestiones de protocolo-WebSocket específico. Otros clientes (por ejemplo, un cliente de iOS y un cliente HTML5) pueden utilizar otros (no-Java) implementaciones que cumplan con RFC 6455 con el fin de comunicarse con la aplicación de servidor.

El Grupo de Expertos que define JSR 356 quería apoyar a los patrones y técnicas que son comunes a los desarrolladores de Java EE. Como consecuencia, JSR 356 aprovecha anotaciones

En general, se admiten dos modelos de programación diferentes:

- Anotación impulsada. Utilizando POJOs anotados, los desarrolladores pueden interactuar con los eventos del ciclo de vida WebSocket.
- Interfaz de motor. Los desarrolladores pueden implementar la interfaz Endpoint y los métodos que interactúan con los eventos del ciclo de vida.

Ciclo de Vida de Eventos

El evento del ciclo de vida típico de una interacción WebSocket es como sigue:

- Uno de ambos (un cliente) inicia la conexión mediante el envío de una petición HTTP apretón de manos.
- El otro punto (el servidor) responde con una respuesta apretón de manos.
- Se establece la conexión. A partir de ahora, la conexión es completamente simétrica.
- Ambos compañeros envían y reciben mensajes.
- Uno de los compañeros cierra la conexión.

La mayor parte de los eventos del ciclo de vida WebSocket se puede asignar a los métodos de Java, tanto en los enfoques de anotación impulsada y de interfaz impulsada. [ORA13]
Dentro de los métodos esenciales que maneja JSR356 encontramos:

- ***onOpen()***
- ***onMessage(String message)***
- ***onClose()***
- ***onError()***

Estos métodos son los encargados de realizar la comunicación cliente-servidor y avisar si en algún momento la conexión se interrumpe, no puede ser establecida o soportada por el explorador.

- *onOpen*: Este método es el encargado de recibir las solicitudes de conexión y establecerla con la petición de conexión que el cliente envíe.
- *OnMessage*: Se encarga de recibir los mensajes que el cliente envíe y realizar la comunicación cada determinado tiempo establecido por el servidor, por lo general el servidor envía una respuesta cada vez que registra un cambio en el.
- *onClose*: Maneja el cierre de la conexión entre el servidor y el cliente.
- *onError*: Método que es lanzado cada vez que no se pudo establecer la conexión o fue interrumpida, además de manejar y verificar si el explorador soporta la tecnología WebSocket instanciada en el servicio.

4.2 Diagramas de secuencia

Para poder tener una referencia en cuanto a cómo será el diseño de la plataforma debemos realizar prototipos, los cuales nos ayudaran a un mejor desarrollo del sistema, agregando diagramas de secuencia que nos explicaran el flujo que llevan dichas acciones realizadas por el usuario

Cada sistema realizado, sea en un entorno web o local, realiza una secuencia de operaciones para poder realizar la tarea que el usuario ha solicitado, para esto, los diagramas de secuencia nos ayudarán, describiendo el flujo de trabajo que se procesa al solicitar y realizar una tarea o cierta parte de ella, en resumen, son un esquema conceptual, que permite representar el comportamiento de un sistema, dentro de ellos, encontraremos diversos componentes que ayudan en la descripción del diagrama, como lo son los objetos,

un escenario donde se encuentran y la manera de comunicación que tienen estos objetos entre sí mediante el uso de mensajes.

A demás de todo lo descrito con anterioridad, ¿Para qué otras situaciones nos sirven los *diagramas de secuencia*?

Los diagramas de secuencia nos ayudan a:

- **El modelado de escenarios de uso.** Un escenario de uso es una descripción de una posible forma en que un sistema se utiliza. La lógica de un escenario de uso puede ser parte de un caso de uso, por ejemplo, una secuencia alternativa o un paso completo a través de un caso de uso, tal como la lógica que describe la secuencia normal de la acción o una parte de ella. Un escenario de uso también puede ser un paso a través de la lógica contenida en varios casos de uso.
- **El modelado de la lógica de los métodos.** Los diagramas de secuencia se pueden utilizar para explorar la lógica de una operación, función o procedimiento complejo, ya que ofrece una forma de observar las invocaciones a las operaciones definidas en las clases.
- **La detección de cuellos de botella en un diseño orientado a objetos.** Al observar los mensajes enviados a un objeto y cuánto se tardan en ejecutar el método invocado, es posible concluir que es necesario cambiar el diseño con el fin de distribuirla carga dentro del sistema.

Dentro de los diagramas que mostraremos más adelante, podemos observar que contienen los siguientes elementos:

- **Un actor:** Este realiza la actividad de empezar las interacciones con el sistema
- **Entidades/Objetos:** Estas interactuaran con el autor y/o entre ellas mismas. En la parte inferior de la entidad podremos observar una línea punteada la cual representa el tiempo o vida que tiene la entidad.
- **Mensajes:** Es la manera en la que se generará la interacción entre el autor y las entidades que estén dentro del diagrama, los mensajes no solo pueden ser entre

un actor y las entidades, sino también entre las misma entidades puede darse la comunicación.

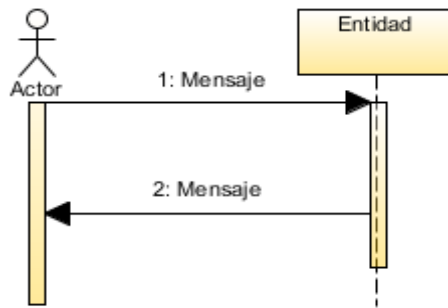


Figura 4.9 Componentes de un diagrama de secuencia. Fuente: Propia

En el siguiente bloque se presentan los diseños de la plataforma hechos en Balsamik, en conjunto de los diagramas de secuencia pertinentes, además de presentar vistas de la plataforma ya realizada y una breve explicación de lo que se realiza y como se realiza en cada una.

4.3 Desarrollo

Una vez completada la arquitectura básica del sistema, procedemos a la etapa de desarrollo del sistema, que como lo hemos mencionado anteriormente, se utilizara tecnología JAVA con el uso de JSP (Java Server Pages) para crear el dinamismo con el servidor de aplicaciones (Tomcat 7), para la conexión con la base de datos que será en MYSQL utilizaremos JDBC (Java Data Base Connectivity) y una clase JAVA creada por uno mismo que actuará con JDBC y las funciones básicas para la gestión del sistema; eso por parte del back-end (lado del servidor), para la parte del front-end se utiliza Bootstrap que hará una interfaz más agradable para el usuario, HTML5 por ser un sistema Web y JavaScript para realizar ciertas acciones de las JSP o HTML que necesiten, ya sea para realizar acciones asíncronas (AJAX) o verificar formularios.

Los prototipos que veremos a continuación han sido hechos en Balsamick, la cual nos proporciona las herramientas necesarias para realizar diseños de interfaces, de una manera intuitiva y sin necesidad de programar.

En el apartado *4.3.2 Diagramas de secuencia y vistas finales del sistema* veremos las secuencias que realizan ciertas acciones que son realizadas debido a peticiones requeridas

mediante los diagramas de secuencia realizados, además de observar y describir las vistas del sistema que ha sido desarrollado, dando una breve descripción de lo que es y qué acción realiza cada apartado.

4.3.1 Prototipos realizados

Una interfaz gráfica (GUI) es la manera en la que un sistema puede comunicarse o tener interacción con un usuario, y, a mejor diseño y atractivo de la interfaz será mayor el gusto de que nuestro usuario quiera volver a visitar la plataforma o se olvide de ella.

Una interfaz gráfica debe de ser lo más intuitiva posible para el usuario, además de ofrecer colores que no lastimen a la vista o demasiado contenido que haga que nuestro usuario termine aburriéndose y perdamos la atención de éste, es por ello que se requieren crear diseños para obtener una respuesta positiva por parte del usuario y así proceder a el desarrollo en diferentes plataformas.

A continuación presentamos algunos diseños de la plataforma con una breve descripción de lo que haría cada una de ellas.

- **Crear un OA**

El prototipo para crear un OA es realizado mediante pasos, donde podemos ver 3 tipos de pestañas, cada una con una instrucción al principio de la ventana, indicándole al usuario que es lo que debe de realizar en esa sección:

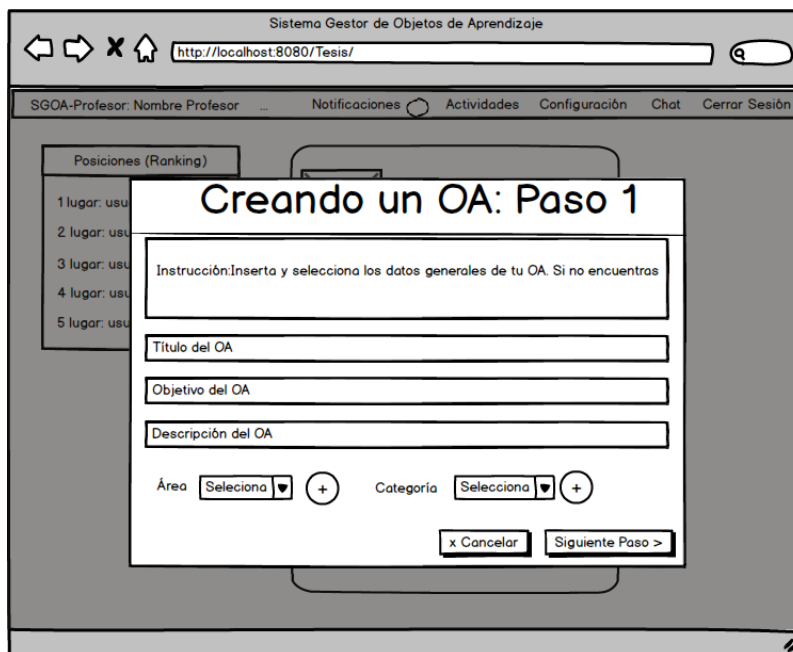


Figura 4.10 Prototipo *Creando un OA Paso 1*. Fuente: Propia

En el paso uno encontramos una ventana en la cual solo requiere llenar el formulario, y seleccionar las áreas y categorías a las cuales pertenecerá el OA, si la categoría o área no están enlistadas, se pueden agregar presionando el botón que tiene un signo de más (+) el cual desplegara una nueva ventana en la cual pedirá al usuario anote el nuevo sustantivo a ingresar.

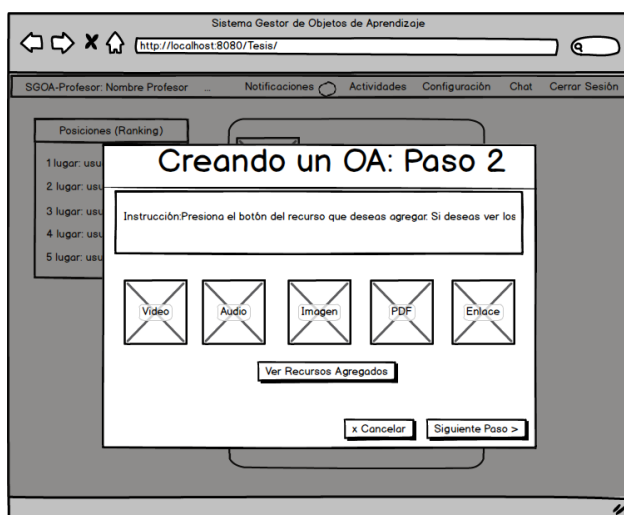


Figura 4.11 Prototipo *Creando un OA Paso 2*. Fuente: Propia

En el paso dos encontramos otra ventana tipo modal en la cual, además de colocar las instrucciones a realizar, distribuye una serie de imágenes tipo botón que al presionarlas desplegaran un nuevo formulario para agregar una actividad, una vez agregada una actividad, se pueden observar las actividades creadas en el OA presionando el botón “Ver recursos Agregados” esta opción desplegara una tabla que mostrara las actividades que han sido agregadas en el objeto de aprendizaje creado.

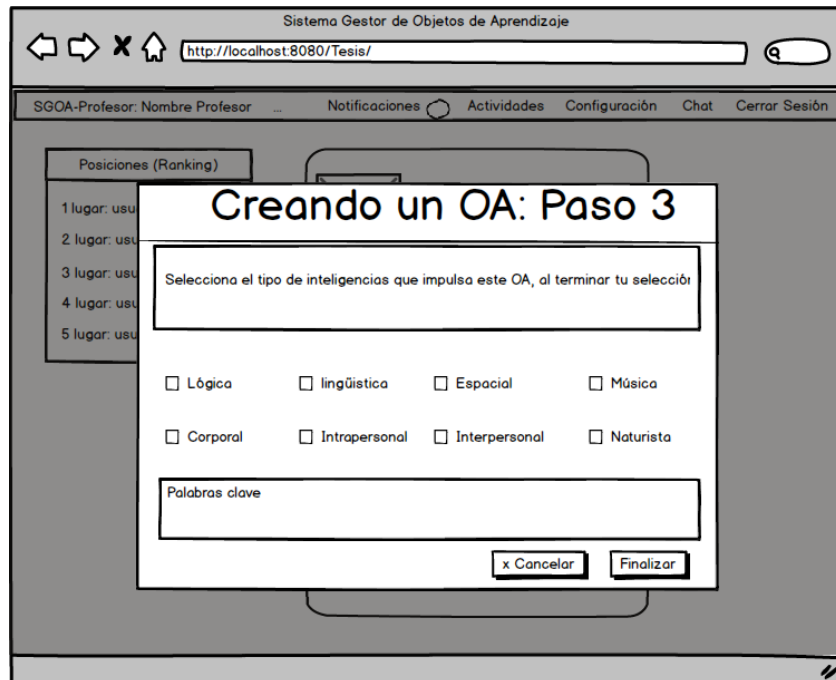


Figura 4.12 Prototipo *Creando un OA Paso 3*. Fuente: Propia

Para finalizar, el paso tres coloca las instrucciones a seguir en esa ventana, las cuales promueven la elección de las inteligencias que quiere desarrollar el OA una vez haya terminado su realización, además de colocar un área en donde se pide ingresar palabras clave para poder identificar al OA en la sección de búsqueda, una vez finalizado el proceso el usuario presionara el botón finalizar y el OA habrá sido creado y publicado en su totalidad.

- **Crear Actividad**

La creación de una actividad, tiene como pre-requisito que el formulario con la información básica del OA (Paso 1) haya sido completada de manera satisfactoria, una vez que se haya completado, el usuario seleccionará el recurso a agregar (Paso 2) presionando el botón que desee, una vez hecho esto se desplegará una ventana como la siguiente:

The screenshot shows a web browser window with the URL `http://localhost:8080/Tesis/`. The page title is "Sistema Gestor de Objetos de Aprendizaje". The navigation bar includes "SGOA-Profesor: Nombre Profesor", "Notificaciones", "Actividades", "Configuración", "Chat", and "Cerrar Sesión". A "Posiciones (Ranking)" sidebar is partially visible on the left. The main content area is a modal form titled "Agregar Actividad" with the following fields and controls:

- Text input: "Escribe el título de la Actividad"
- Text input: "Escribe el Objetivo de la actividad"
- Text input: "Escribe la descripción de la actividad"
- Text input: "Puntos a Asignar"
- Dropdown menu: "Tipo de Saber" (Selecciona)
- Dropdown menu: "Fuente del recurso" (Selecciona)
- Section: "Estrategia de Aprendizaje" with checkboxes for:
 - Individual
 - Colaborativa
 - PBL
 - Basada en casos
 - Otras
- Buttons: "x Cancelar" and "Agregar >"

Figura 4.13 *Agregando Actividad* (incluida en el paso 2 de crear un OA). Fuente: Propia

Dentro de esta ventana se muestra el formulario para la creación de una actividad contenida en el OA que está siendo creado, además de pedir la información básica, se pide asignar puntos a la actividad (estos se suman al puntaje del alumno, una vez que haya finalizado al actividad) el tipo de saber que contendrá la actividad y la fuente del recurso, en el cual acepta archivos y códigos embebidos dependiendo la selección del recurso, si se desea añadir un enlace, el recurso a escoger deberá ser el último y en el campo donde pide el enlace este tendrá que empezar con el protocolo que maneje la página a insertar (`http://` `ftp://`, etc.), una vez que se ha añadido el recurso, se dispone el usuario a seleccionar el tipo de estrategias de aprendizaje que impulsaran dicha actividad, la actividad se creará dando clic en "Agregar".

- **Mostrar OA's creados**

Cada vez que un OA es creado, este pasa a ser registrado dentro de la sección de "Mis OA's" en donde se puede observar mediante una breve descripción el OA que ha sido creado.

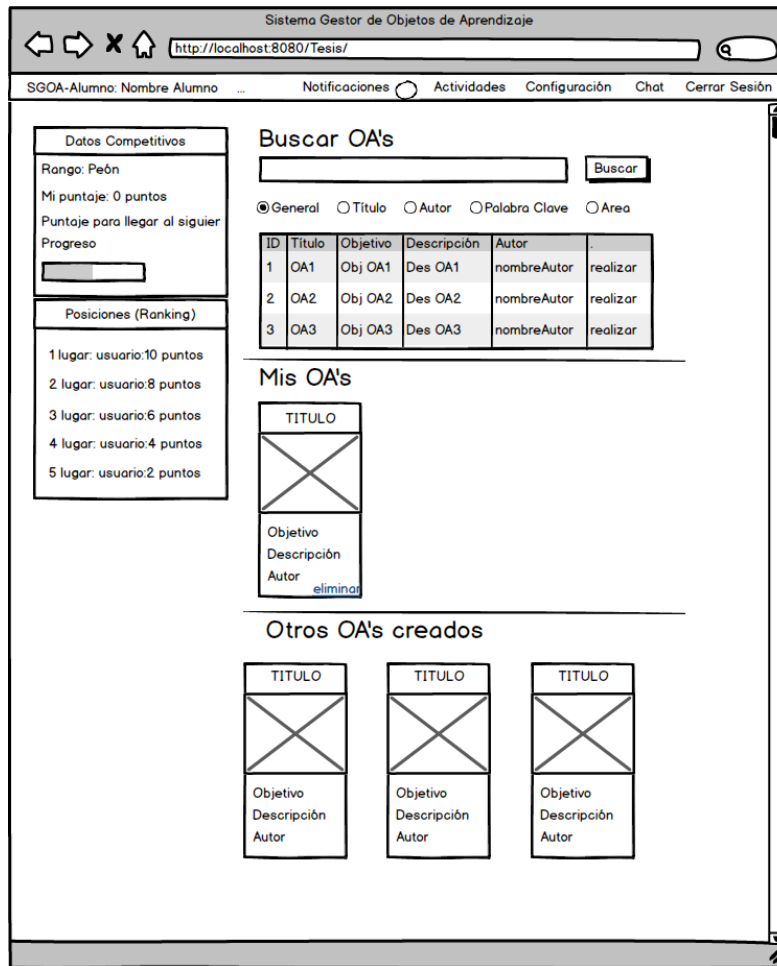


Figura 4.14 Sección "Mis OA's". Fuente: Propia

Esta ventana ofrece 3 tipos de opciones que el usuario necesita, la primera, una sección de búsqueda de los OA's creados, ya sea una búsqueda general, por autor, nombre de OA, palabra clave o área; esta búsqueda se desplegará en una tabla y se podrá seleccionar el OA que se necesite.

El segundo apartado, es que hemos descrito con anterioridad, en el cual se muestran los OA's creados por el usuario, aparece con una pequeña imagen para darle más vistosidad, una breve descripción de su información y un enlace por si se desea eliminar ese OA del sistema.

Por último, el apartado "Otros OA's creados" contiene OA's creados por otros usuarios, de igual manera se pueden seleccionar para realizarlos.

- **Realizar OA**

Una vez agregado el OA este puede ser seleccionado y a su vez realizado, al presionar el botón realizar se mostrara una interfaz la cual presentara todos los detalles de OA y además, mostrará las actividades que contiene el OA en el para poder acceder a ellas, solo se necesita presionar el enlace “realizar”

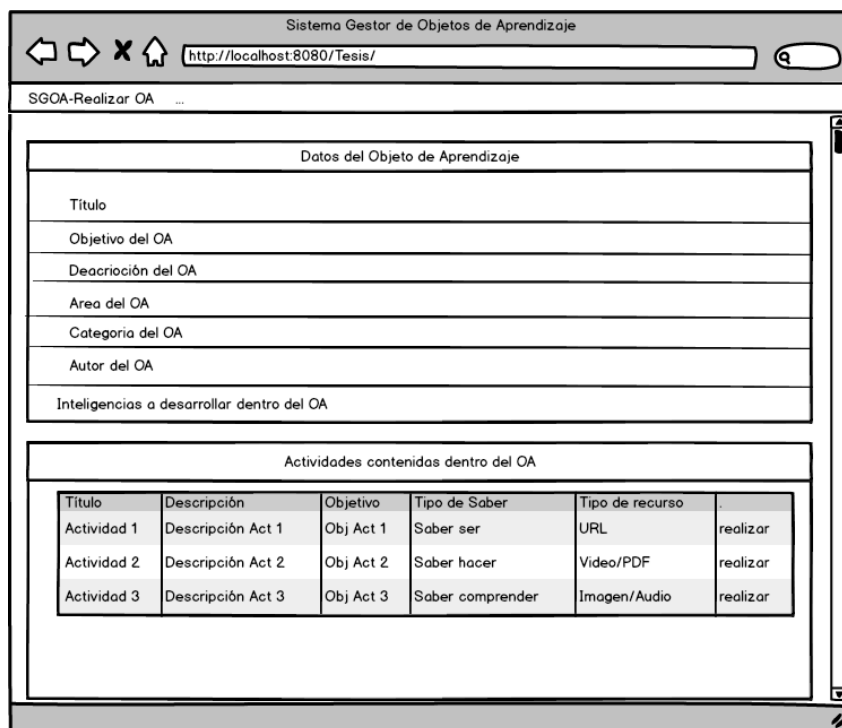


Figura 4.15 *Interfaz para realizar un OA*. Fuente: Propia

- **Otros prototipos del sistema**

A demás de los prototipos mostrados anteriormente, hay otros prototipos que requieren de su visualización antes de ser desarrollados, uno de ellos es la página principal o “index” que este contendrá los últimos cinco OA’s creados por distintos usuarios, además de contener una serie de avisos publicados por cualquier usuario de tipo profesor; la sección de últimos OA’s solo rinde información acerca del contenido del OA creado, no se puede acceder a él a menos que el usuario inicie sesión.

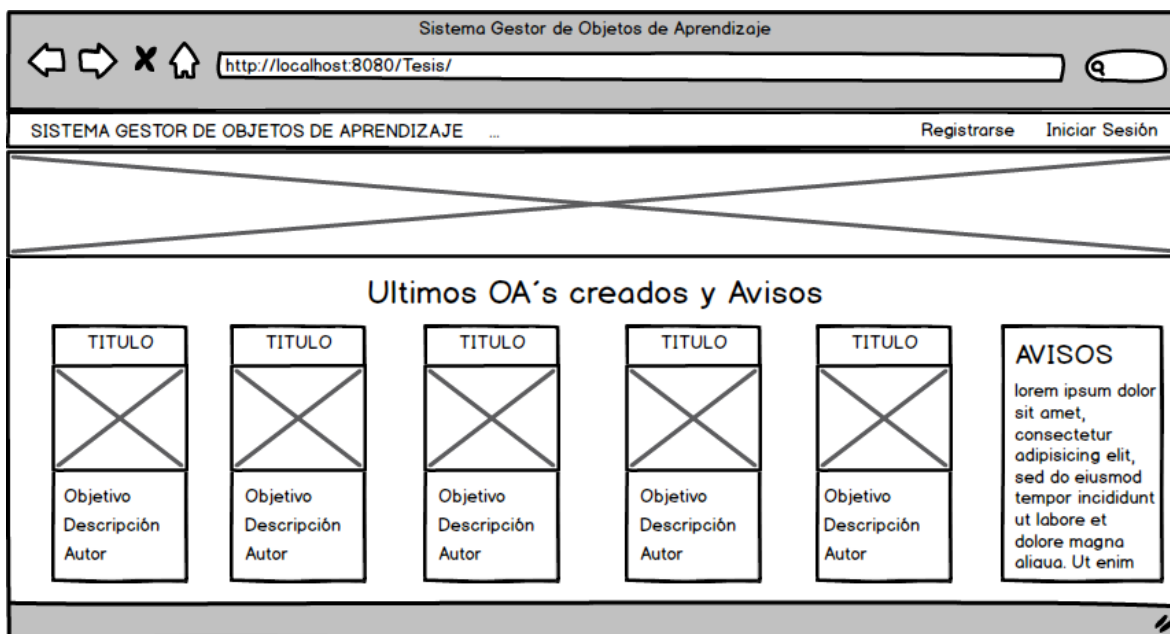


Figura 4.16 *Página principal del sistema.* Fuente: Propia

Al iniciar sesión los usuarios se encuentran con una sección a la que denominaremos “Muro” en ella se encuentran publicadas las actividades creadas por otros usuarios y dependiendo del tipo de usuario ingresado mostrará detalles adicionales; una sección de “Datos competitivos” si el inicio de sesión fue hecho por un usuario de tipo alumno, y submenús adicionales (Crear OA, Generar Aviso) si el logueo se realizó con un usuario de tipo profesor; además de dichas cualidades antes mencionadas por cada tipo de usuario, el “muro” contiene menús para:

- ✓ Ver notificaciones
- ✓ Actividades (contiene los menús adicionales para el profesor y el menú general “Mis OA's”)
- ✓ Configuración (con submenús para modificar información personal y de acceso)
- ✓ Chat (Sirve para establecer comunicación con cualquier tipo de usuario que haya iniciado sesión)
- ✓ Cerrar Sesión (Encargado de terminar la sesión del usuario)

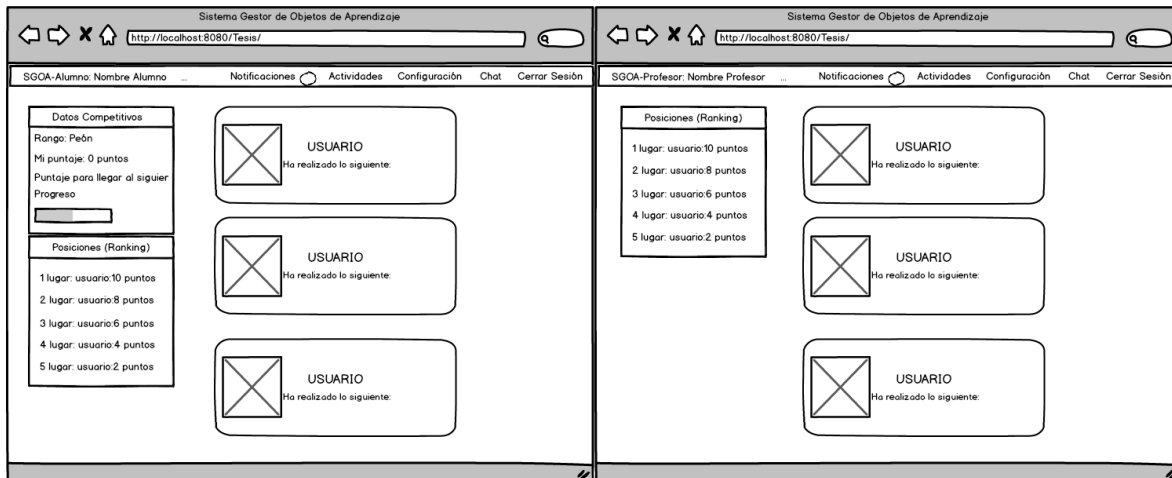


Figura 4.17 A la izquierda “Muro Alumno”, a la derecha “Muro Profesor”. Fuente: Propia

4.3.2 Diagramas de secuencia y sitio web desarrollado

Una vez creados los prototipos procedemos al desarrollo del sistema tal y cual los prototipos fueron señalados, además de ello, presentar la secuencia o como se realizan las acciones que el usuario no ve dentro del sistema mediante los diagramas de secuencia.

- **Diagramas de secuencia**

Una vez identificados los diagramas del sistema y sus casos de uso, procedemos a mostrar los diagramas de secuencia para distintas actividades del sistema, las cuales son las más comunes a realizar por los diferentes tipos de usuarios.

- **Crear un OA**

Como lo habíamos mencionado con anterioridad, el crear un OA se divide en tres pasos, los cuales cada paso lanzara un mensaje de información, indicando si dicho paso se realizó con éxito o tuvo un fallo durante su proceso.

En la siguiente figura podemos observar cual es el proceso que realiza el paso uno para crear un OA.

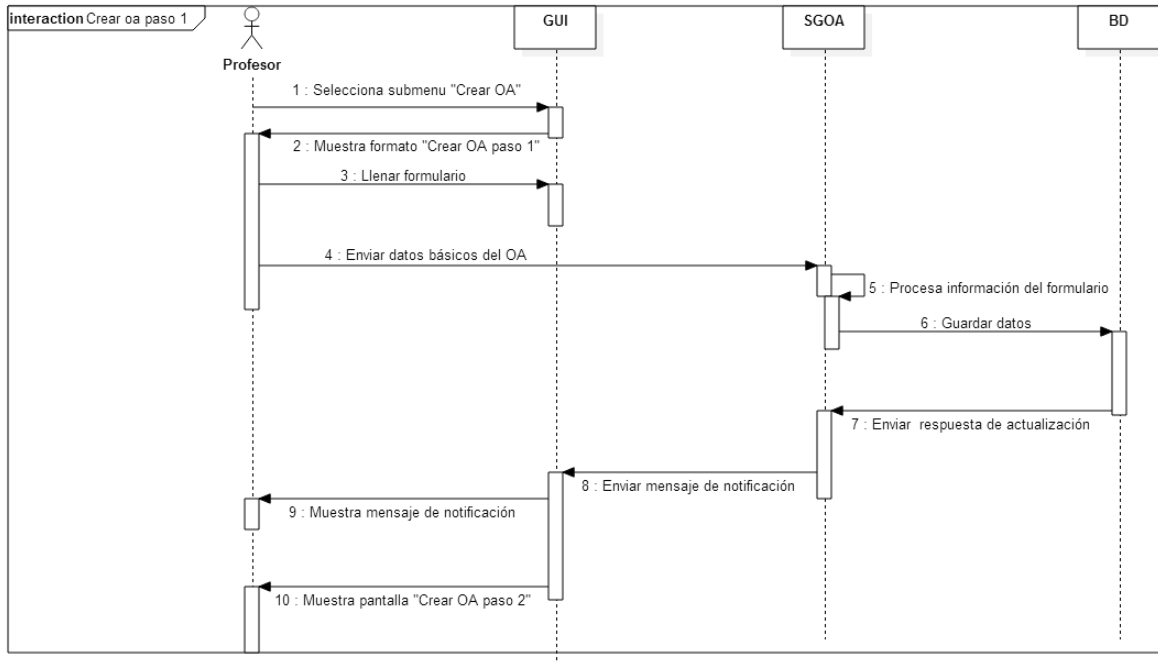


Figura 4.18 Diagrama de secuencia del paso 1 para crear un OA. Fuente: Propia

- **Agregar una actividad**

Para poder crear una actividad, dependemos que el sistema nos arroje un resultado positivo al realizar con anterioridad el paso uno, siendo esto correcto, se mostrará una segunda ventana donde seleccionaremos el recurso a agregar dentro del objeto de aprendizaje.

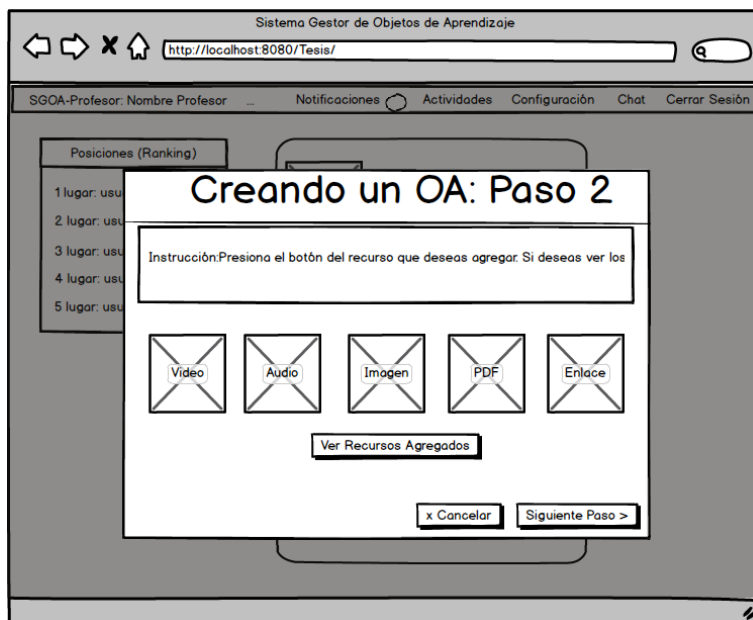


Figura 4.19 Ventana del paso 2 para crear un OA. Fuente: Propia

La secuencia que se maneja dentro de este proceso esta descrita en la siguiente imagen.

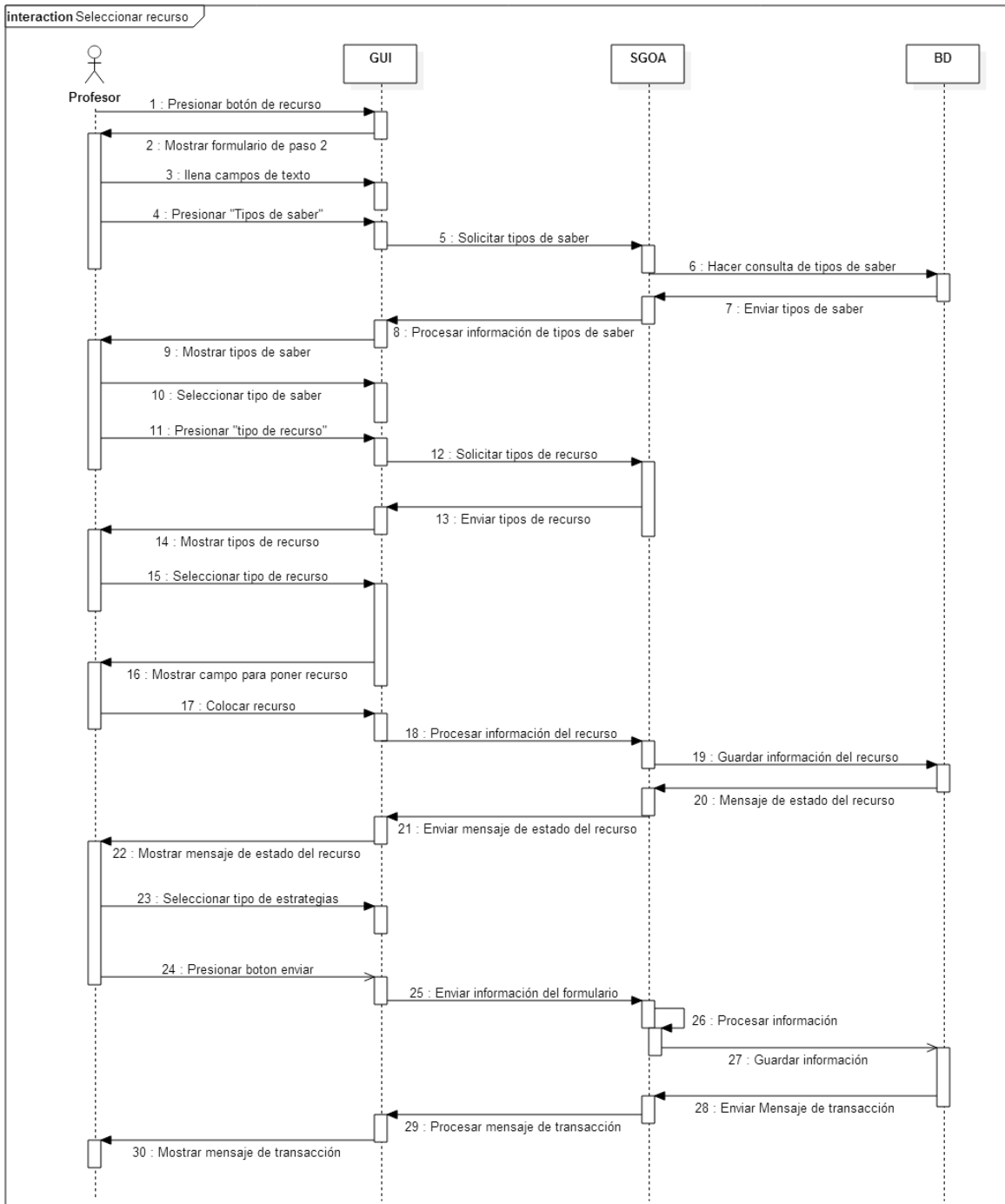


Figura 4.20 Diagrama de secuencia del paso 2 para crear un OA. Fuente: Propia

- Finalizar creación del OA

Al concluir de proporcionar la información requerida en la ventana de “Agregar actividad” y oprimir el botón “siguiente paso” nos llevará a la última ventana de la creación de un OA y con ello automáticamente el OA será publicado en el momento que es guardado a la base de datos.

En la ventana tres, se requiere que el usuario inserte los tipos de inteligencias y palabras claves que necesita el sistema para una mejor indexación al momento de su búsqueda, a continuación en la siguiente imagen se puede observar la secuencia del último paso para crear un OA.

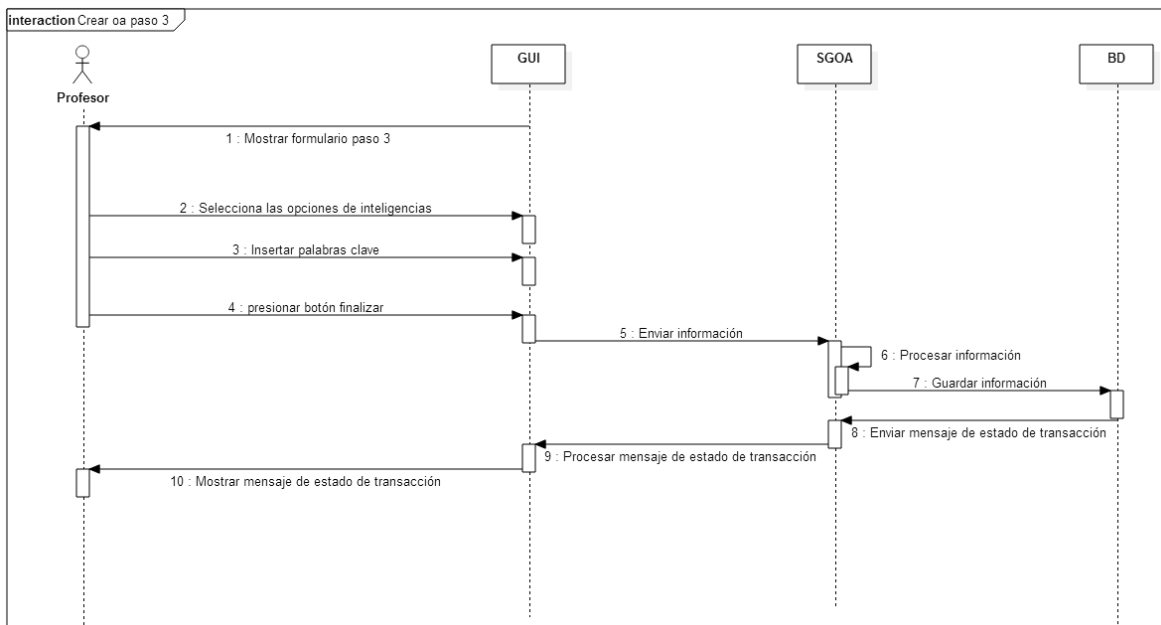


Figura 4.21 *Diagrama de secuencia del paso 3 para crear un OA.* Fuente: Propia

- Realizar actividad

Para que se pueda realizar una actividad, es necesario entrar al apartado “Actividades” seguido de “Mis OA’s” en esa sección podremos buscar OA’s que han sido creados y poder seleccionarlos, una vez seleccionado el OA y pulsado el botón o enlace “realizar” nos direccionaran a una página en donde se encuentra toda la información del OA además, en la parte inferior de la página, encontraremos una tabla que contiene las actividades pertenecientes a dicho OA, podemos seleccionar cualquiera de ella presionando el enlace “realizar”, una vez hecho esto, se mostrara otra parte de la información de la actividad y el recurso multimedia agregado para su realización, al finalizar de realizar la actividad,

presionando el botón “Finalizar actividad” obtendremos los puntos que se han dado en la actividad que hemos realizado.

Para una mayor descripción, en la siguiente imagen podemos observar la secuencia que se tienen al realizar una actividad.

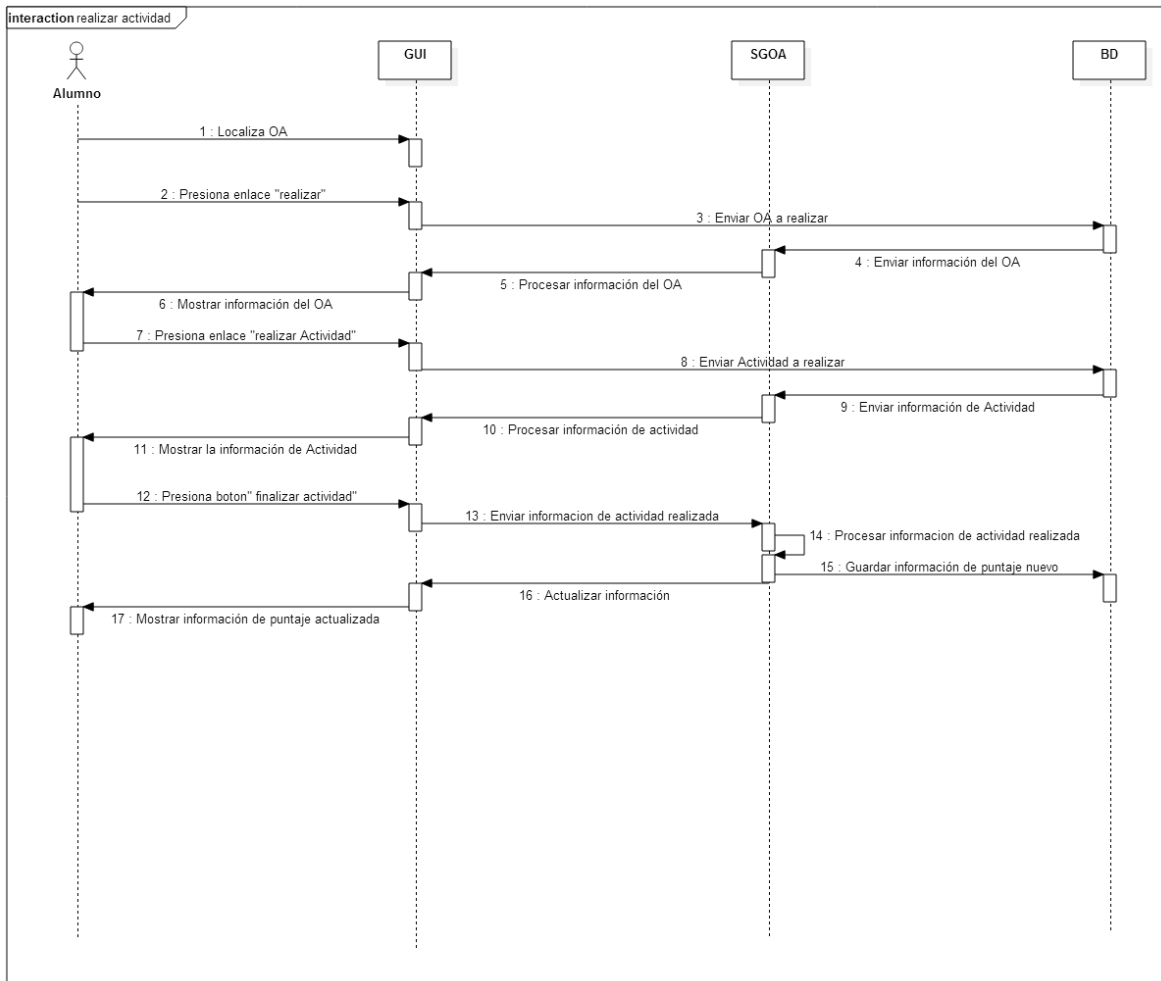


Figura 4.22 Diagrama de secuencia para realizar una actividad. Fuente: Propia

- **Sitio web desarrollado**

Ahora que ya hemos realizado todos los diagramas correspondientes al sistema, es momento de presentar como ha finalizado el desarrollo de la plataforma y una breve descripción de cada una de las interfaces que se mostraran a continuación.

- Página Principal (Index)

En la siguiente interfaz se muestra la página con la que cualquier persona es recibida a la interfaz



Figura 4.23 *Página principal del Sistema.* Fuente: Propia

En el index podremos encontrar 2 opciones las cuales desembocan en un registro al sistema y un ingreso, cada uno con sus respectivos campos de texto como veremos más adelante, además esta página contiene una sección de avisos y los últimos OA's que han sido creados.

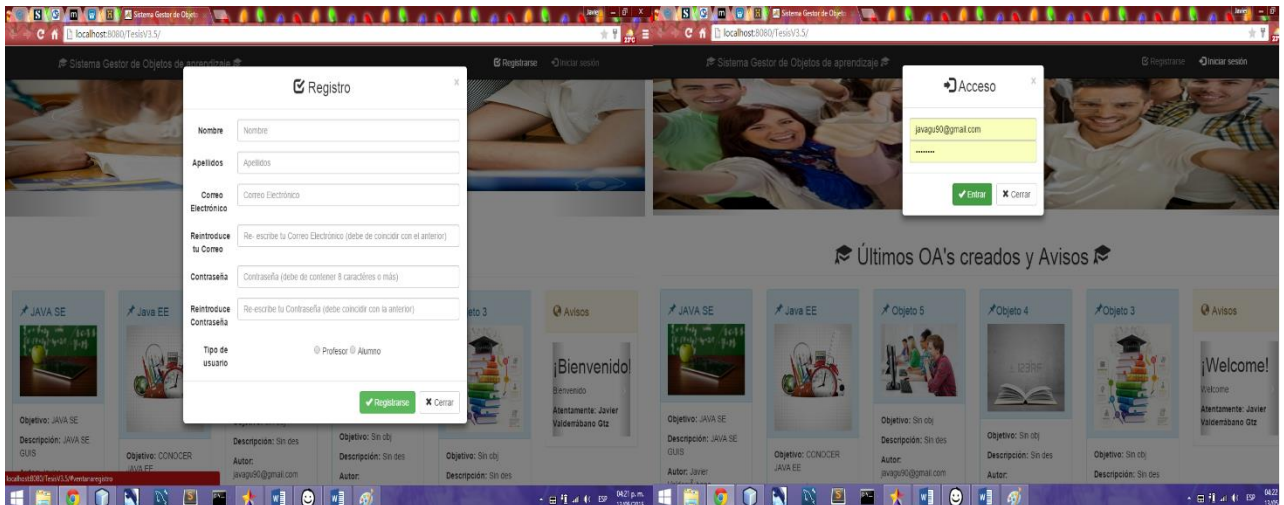


Figura 4.24 *a la izquierda "Ventana de registro" a la derecha "Ventana de Acceso".*

Fuente: Propia

Ambas ventanas confirman que haya información en sus respectivos campos, si no lo hay, envía una advertencia, si la información añadida es inexistente o no concuerda con la registrada, el usuario es redirigido a una página HTML donde se ve el error que ha tenido.

➤ "Muro"

Al ingresar al sistema, nos toparemos con lo que llamamos "muro" que dependiendo el tipo de usuario que haya ingresado (Alumno o Profesor) tendrá vistas o menús adicionales



Figura 4.25 “Muro de un usuario tipo Profesor”. Fuente: Propia

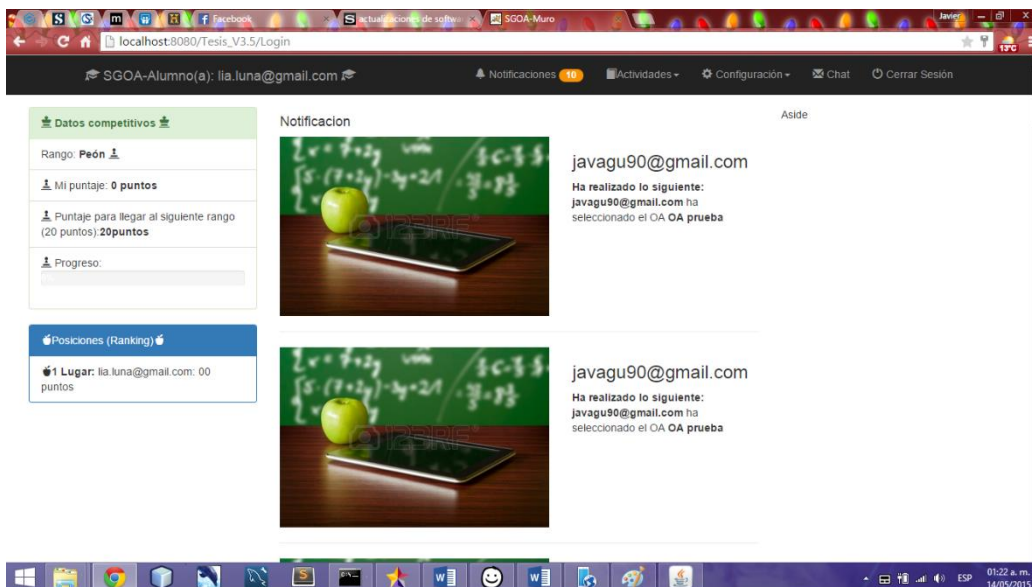


Figura 4.26 “Muro de un usuario tipo Alumno”. Fuente: Propia

Como podemos apreciar, la estructura es casi idéntica, solo con el plus, de que se puede observar una ventana verde que corresponde a los datos competitivos del alumno, en los cuales se determina el rango clasificados en cuatro,

- **Peón:** de 0 a 19 puntos
- **Caballero (Caballo):** de 20 a 39 puntos
- **Espadachín (Alfil):** de 40 a 59 puntos
- **Guardia (Torre):** de 60 a 79 puntos
- **Monarca (Rey):** de 80 a 100 puntos

Este criterio se ha tomado con base en el juego de ajedrez y sus valores relativos de cada pieza como se ha descrito con anterioridad

Para lograr que el llegar al siguiente puntaje no se vuelva difícil se colocó una barra de progreso el cual indicará cuanto ha progresado en el nivel.



Figura 4.27 “Datos competitivos de un usuario tipo Alumno”. Fuente: Propia

Los menús que podemos encontrar en el muro de cualquier usuario son los siguientes:

- *Notificaciones:*

Este menú solo contiene un desplegable de las acciones que han hecho diferentes usuarios dentro del sistema

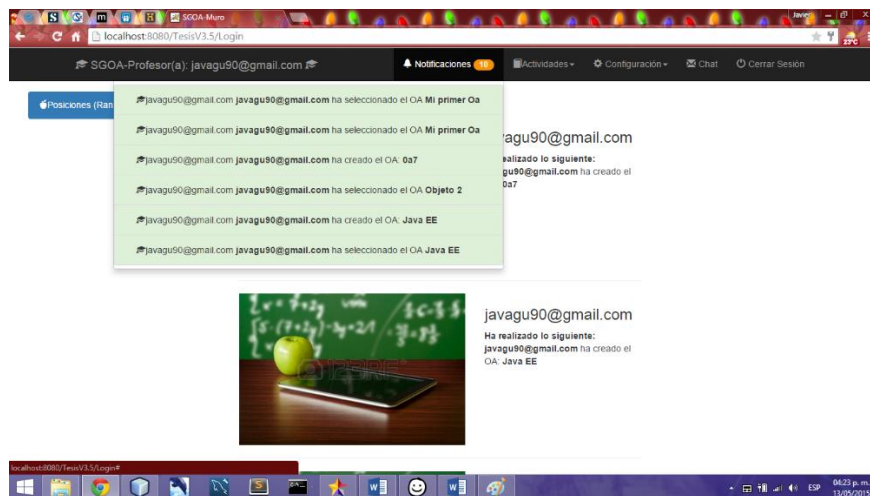


Figura 4.28 “Menú notificaciones desplegado”. Fuente: Propia

- **Actividades:**

Menú que contiene las actividades a realizar por el usuario, con la diferencia que el usuario de tipo profesor contiene dos submenús más (Crear un OA y Generar Aviso), el submenú “Mis OA's” está contenido en ambos tipos de usuario.

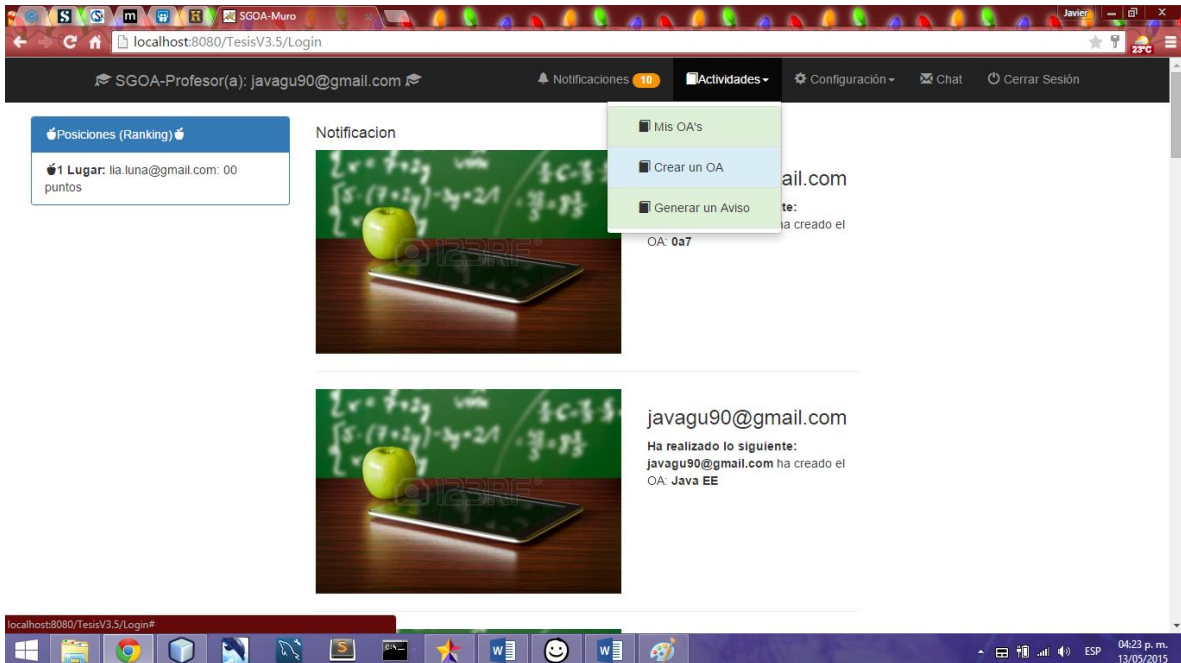


Figura 4.29 “Menú Actividades de un usuario tipo profesor”. Fuente: Propia

- **Configuración**

El menú configuración contiene los submenús esenciales para realizar el cambio de datos del usuario, como nombre de usuario, contraseña y datos personales

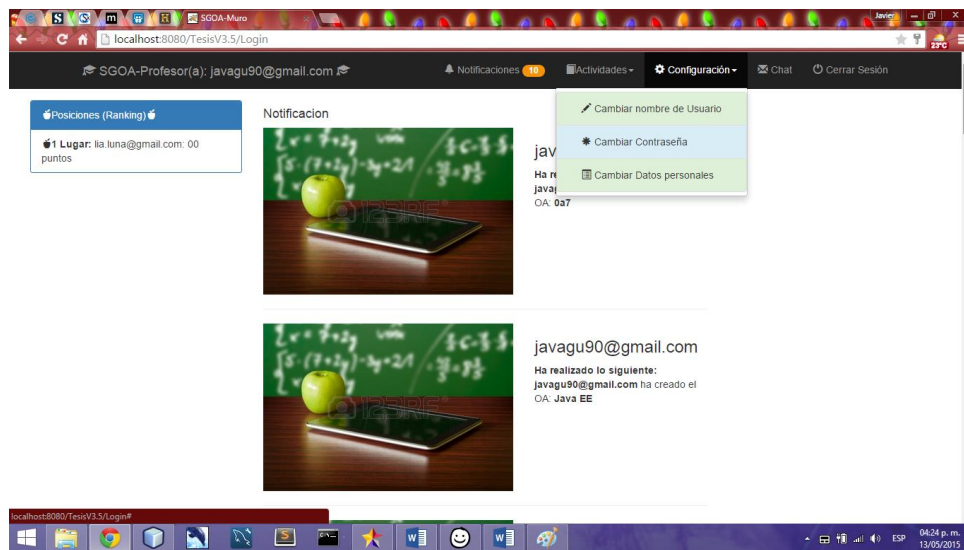


Figura 4.30 “Menú Configuración”. Fuente: Propia

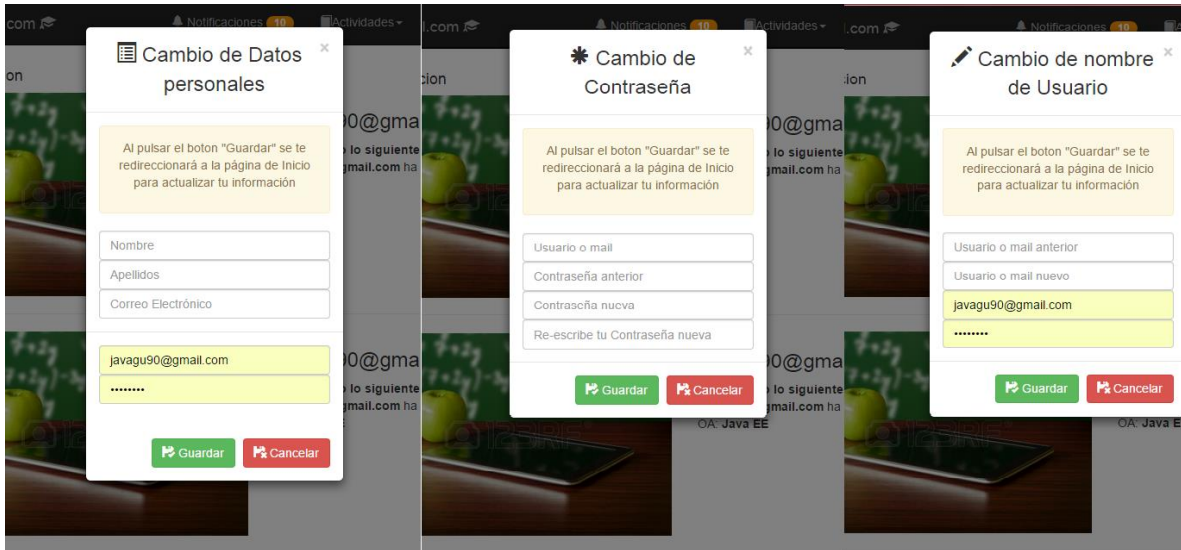


Figura 4.31 Registros “Cambio de datos personales”, “Cambio de contraseña” y “Cambio de nombre de usuario” respectivamente. Fuente: Propia

- Chat

El chat se ha creado para dar retroalimentación entre un profesor y un alumno o entre alumnos para que haya un ambiente cooperativo; este mecanismo se abrirá en una ventana diferente esperando el nombre del usuario y la conexión para empezar la comunicación

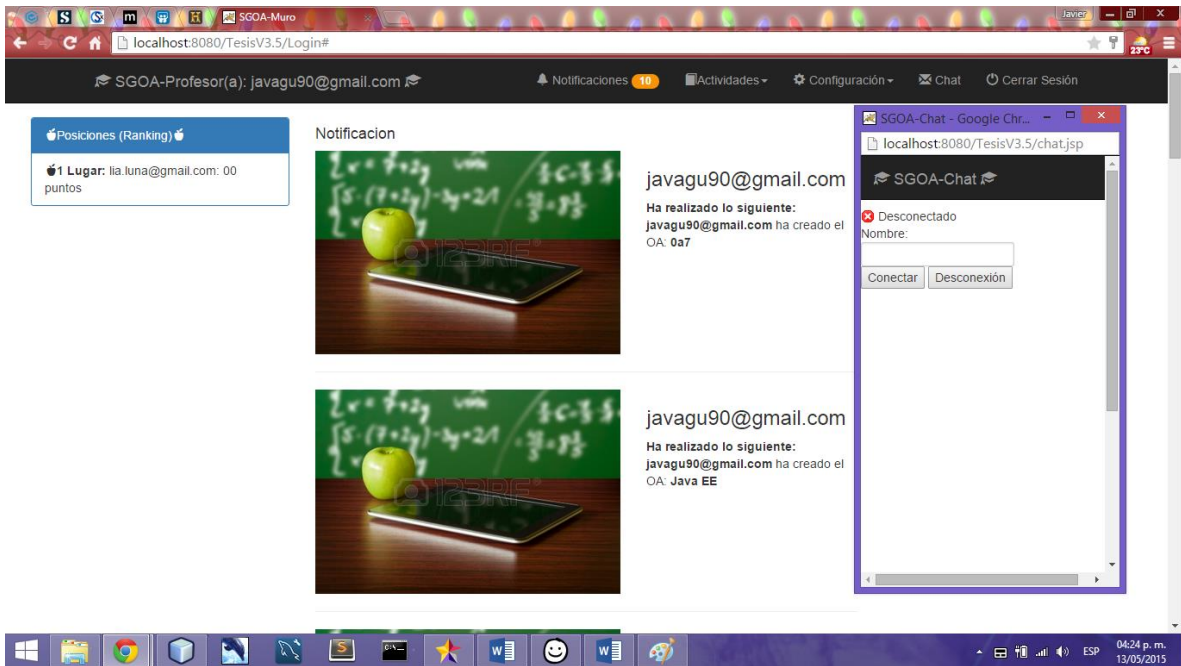


Figura 4.32 ventana de chat desplegada al seleccionar el menú “Chat”. Fuente: Propia

A continuación mostramos una imagen de la interacción que tienen 2 usuarios dentro del chat en diferentes computadores.

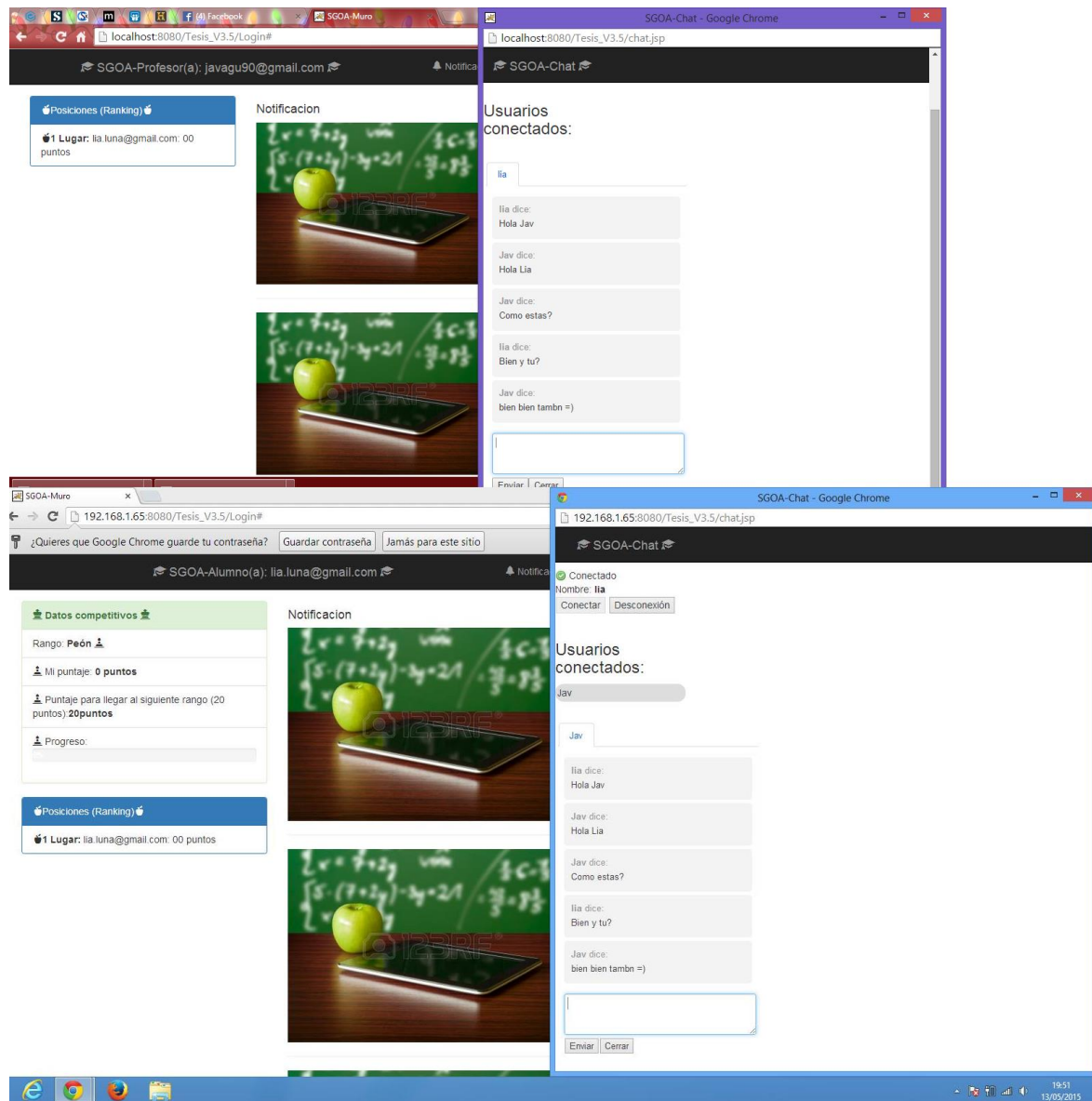


Figura 4.33 Interacción en el chat de dos usuarios en computadoras diferentes. Fuente: Propia

- Cerrar sesión

Este menú como su nombre lo dice, concluye la participación temporal de un usuario que haya ingresado.

➤ Crear un OA

Para crear un OA es necesario ingresar al menú “Actividades” y seleccionar el submenú “Crear un OA” una vez que se seleccione.

La creación de un OA se realiza mediante una serie de pasos consecutivos, además de ir ofreciendo instrucciones de ayuda para facilitar al usuario su creación. Para la realización de un OA completo, incluyendo actividades debemos de seguir tres pasos, los cuales explicaremos con sus debidas interfaces a continuación

○ *Paso 1: Crear un OA*

En el paso uno observaremos un formulario que pedirá la información básica del OA para su creación

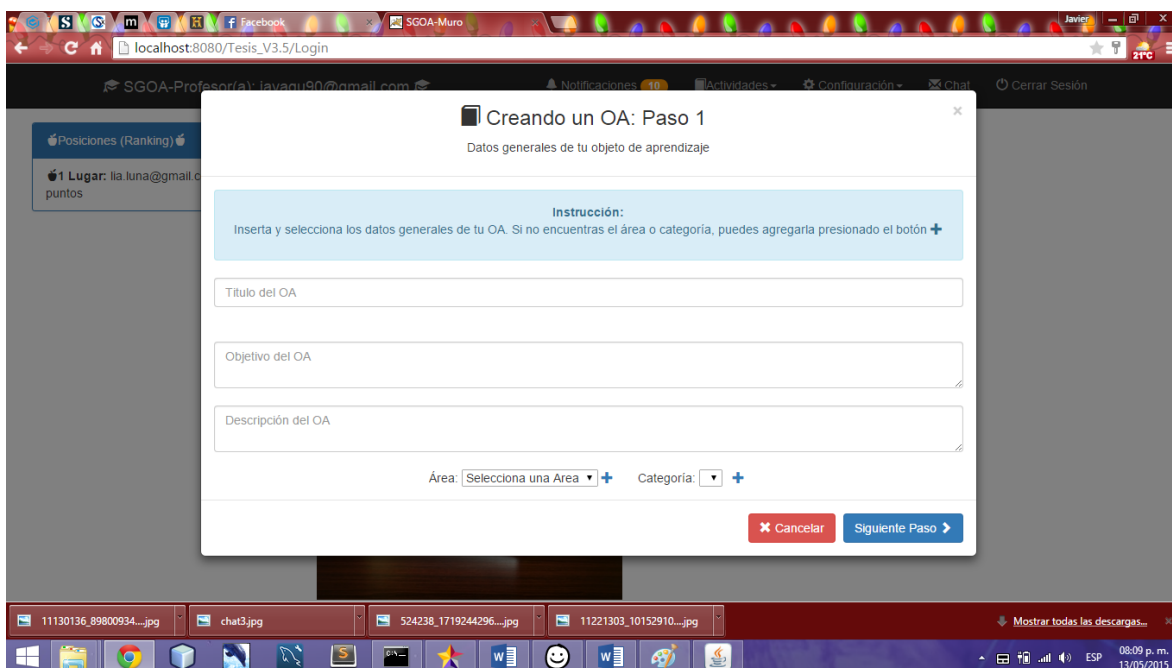
The image shows a screenshot of a web browser displaying a form titled "Creando un OA: Paso 1". The form is a modal window with a white background and a dark border. At the top, it says "Datos generales de tu objeto de aprendizaje". Below this, there is a light blue box with the instruction: "Instrucción: Inserta y selecciona los datos generales de tu OA. Si no encuentras el área o categoría, puedes agregarla presionando el botón +". The form contains three text input fields: "Título del OA", "Objetivo del OA", and "Descripción del OA". At the bottom, there are two dropdown menus: "Área: Selecciona una Área +" and "Categoría: +". There are two buttons at the bottom right: a red "Cancelar" button and a blue "Siguiente Paso" button. The background shows a blurred view of the web application's interface, including a navigation menu and a sidebar.

Figura 4.34 *Formulario para la creación de un OA.* Fuente: Propia

Si el usuario de tipo profesor requiere de un área o categoría no establecida, éste puede presionar el botón que tiene el signo de más y saldrá un formulario similar al de inicio de sesión solo que éste pedirá la información del área o categoría a agregar, y se agregará al dar clic en el botón agregar, cabe aclarar que para insertar una nueva categoría, debe de seleccionarse primero el área a la que pertenecerá.

Al completar el formulario y presionar el botón azul (Siguiente paso) aparecerá una ventana gris en la parte superior dando un mensaje si fue exitoso el proceso o no, si lo fue, el

mensaje dirá la palabra “hecho” del caso contrario mandara un mensaje con la palabra “falló”

- *Paso 2: Añadir Actividades*

En el paso dos podremos añadir las actividades que nosotros queramos dando clic en el botón de la actividad que deseamos crear; para observar los recursos añadidos al OA presionamos el botón “Ver recursos agregados” el cual nos desplegará una tabla con las actividades creadas dentro del OA

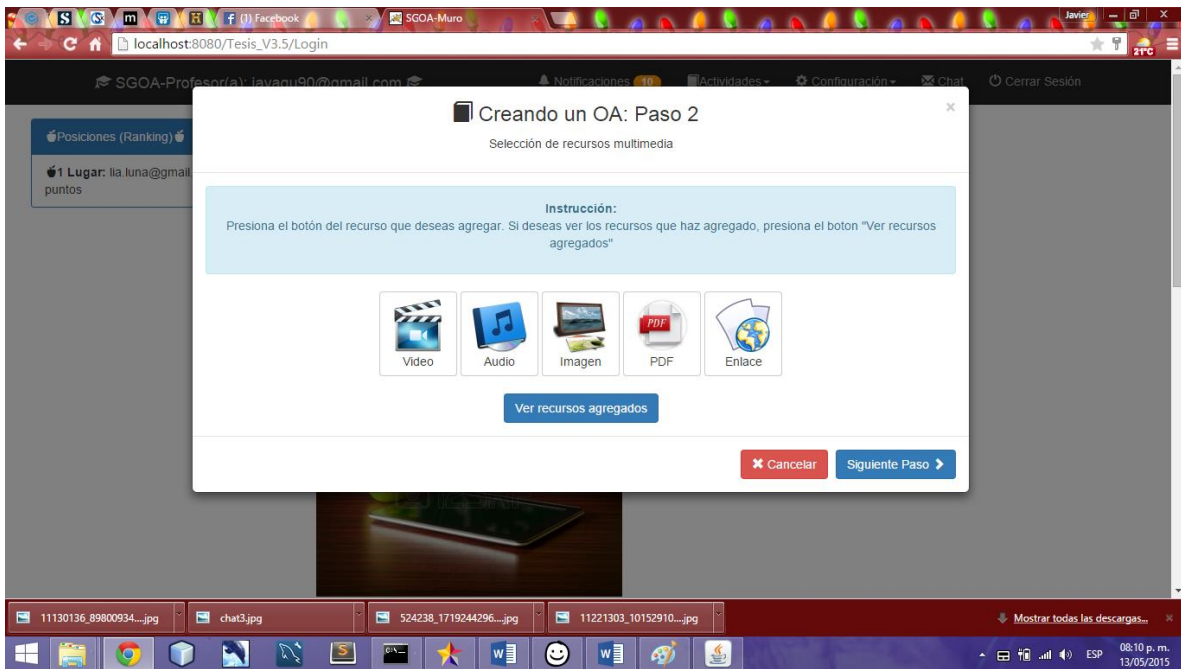


Figura 4.35 *Ventana de selección de actividad.* Fuente: Propia

Una vez presionado el botón de la actividad que queremos agregar al OA se desplegará un formulario pidiendo la información básica del recurso y el tipo de recurso a agregar, si escogemos audio o imagen podemos subir un archivo, si seleccionamos video o pdf, además de poder subir un archivo, podemos insertar código embebido, para finalizar, si queremos insertar un enlace de cualquier tipo, el botón de enlace es nuestra opción, en el campo de recurso, debe de ser insertado el protocolo que la página sigue (http:// o ftp://) de caso contrario nos saldrá un error en el sistema, al finalizar presionamos el botón azul con la leyenda “Agregar” y si fue agregada nuestra actividad, de igual manera que al crear un OA, aparecerá una pantalla gris en la parte superior diciendo “hecho” si el proceso fue exitoso y “fallo” si ocurrió un error dentro de la transacción

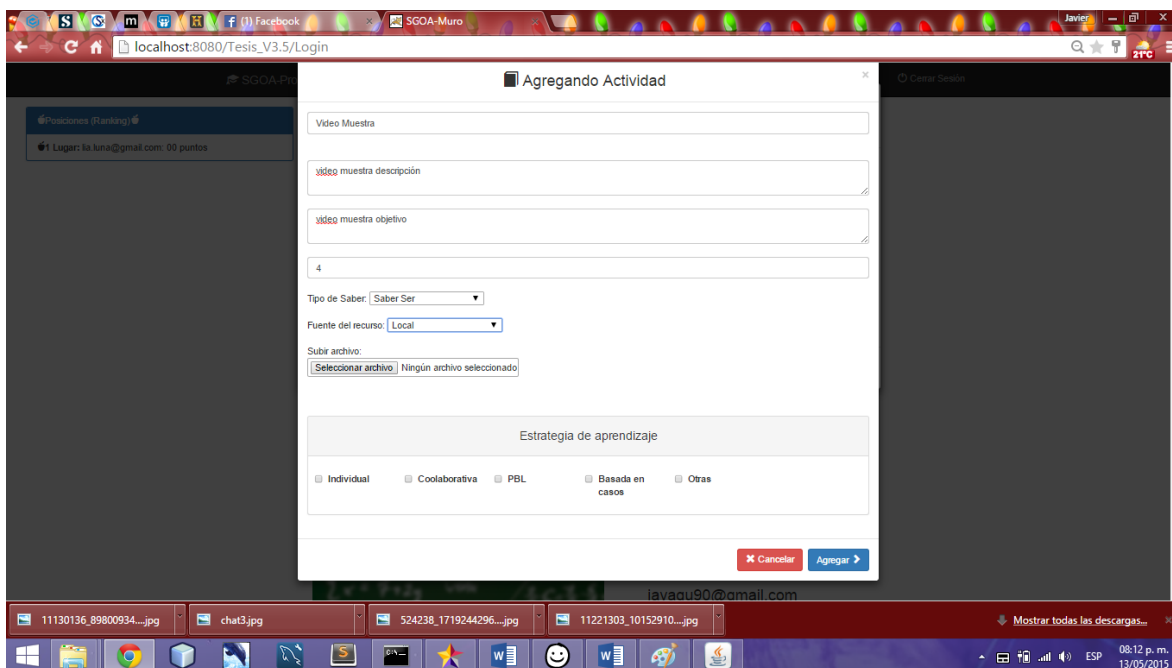


Figura 4.36 *Formulario para la creación de actividades.* Fuente: Propia

Ya creadas nuestras actividades podemos ver cuantas hemos realizado y se han añadido al OA dando clic en “Ver recursos Agregados” donde se desplegará una tabla y mostrara detalles de la actividad creada y añadida.

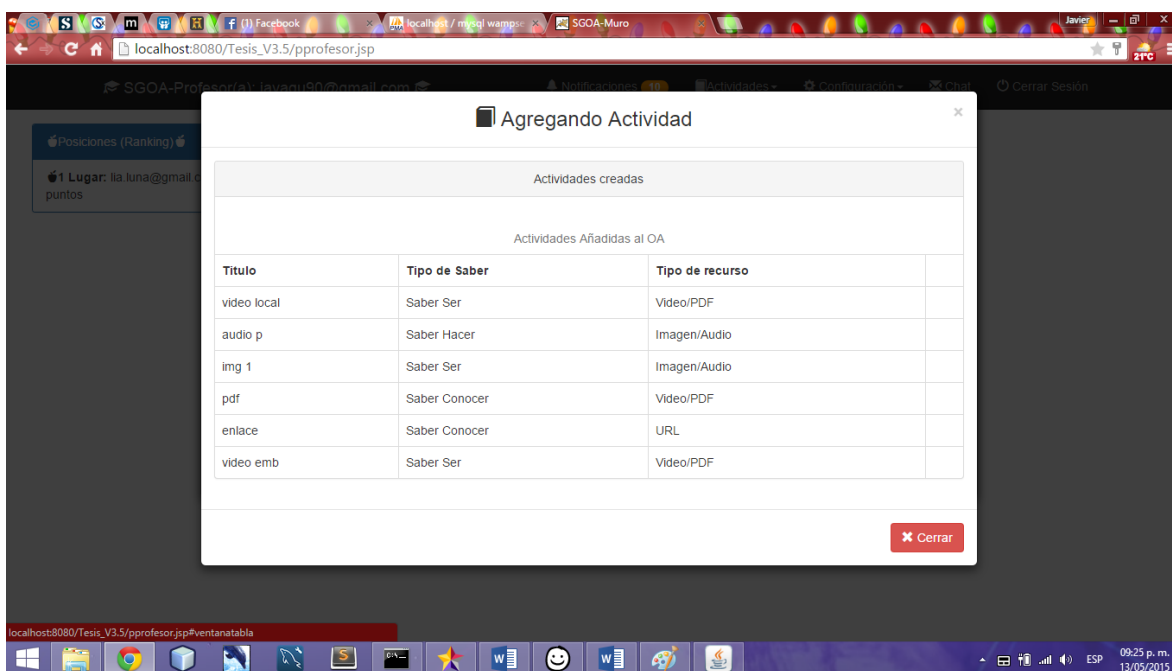


Figura 4.37 *Tabla de actividades añadidas al OA.* Fuente: Propia

- *Paso 3: Añadir inteligencias y palabras clave*

En el paso tres se despliega una ventana con varias opciones de inteligencias, las cuales serán parte de nuestro OA, es por eso que podemos seleccionar las que nuestro OA mediante su realización desarrolle, en la parte inferior encontramos un campo vacío, en el cual podremos incluir las palabras clave que nos servirán para buscar un OA en el apartado de "Buscar OA's" al finalizar todo el proceso, saldrá una ventana gris como en los anteriores pasos, describiendo si la operación fue exitosa o falló.

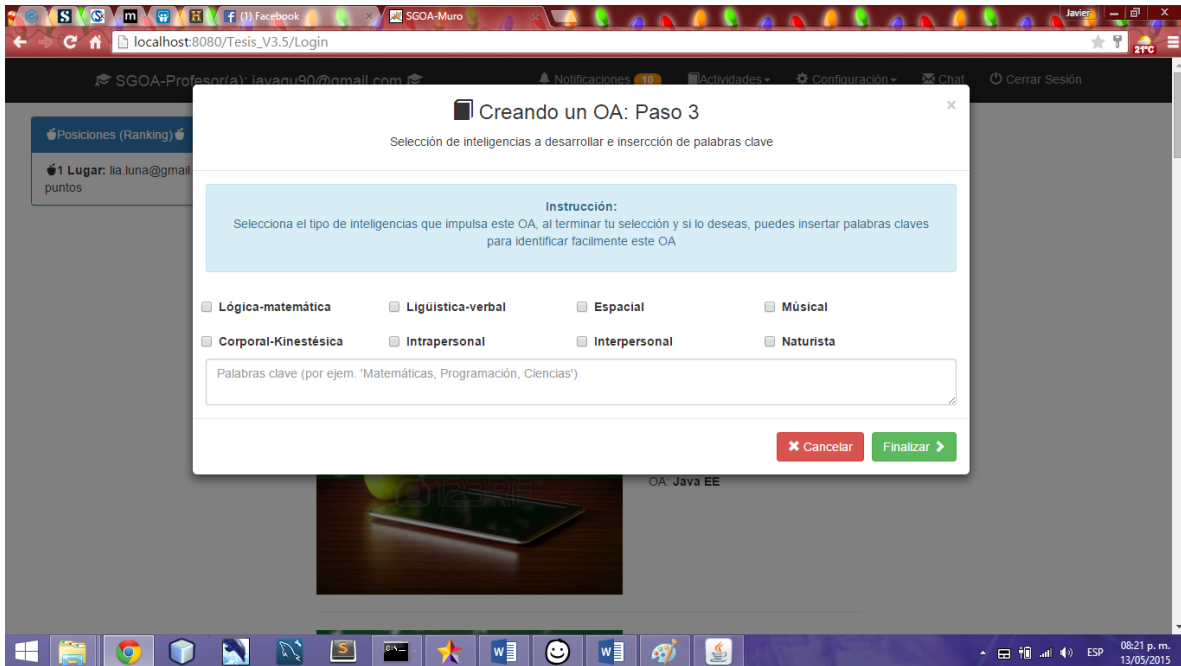


Figura 4.38 *Ventana final para la creación de un OA.* Fuente: Propia

- **Generar avisos**

Un usuario de tipo profesor puede generar avisos, los cuales se mantendrán en la primera página del sistema.

Los avisos requieren de tres campos, un título, el remitente de la persona que lo publica y el contenido del aviso, al presionar el botón "Agregar" el aviso se publicará en el instante, guardándose dentro de la base de datos.

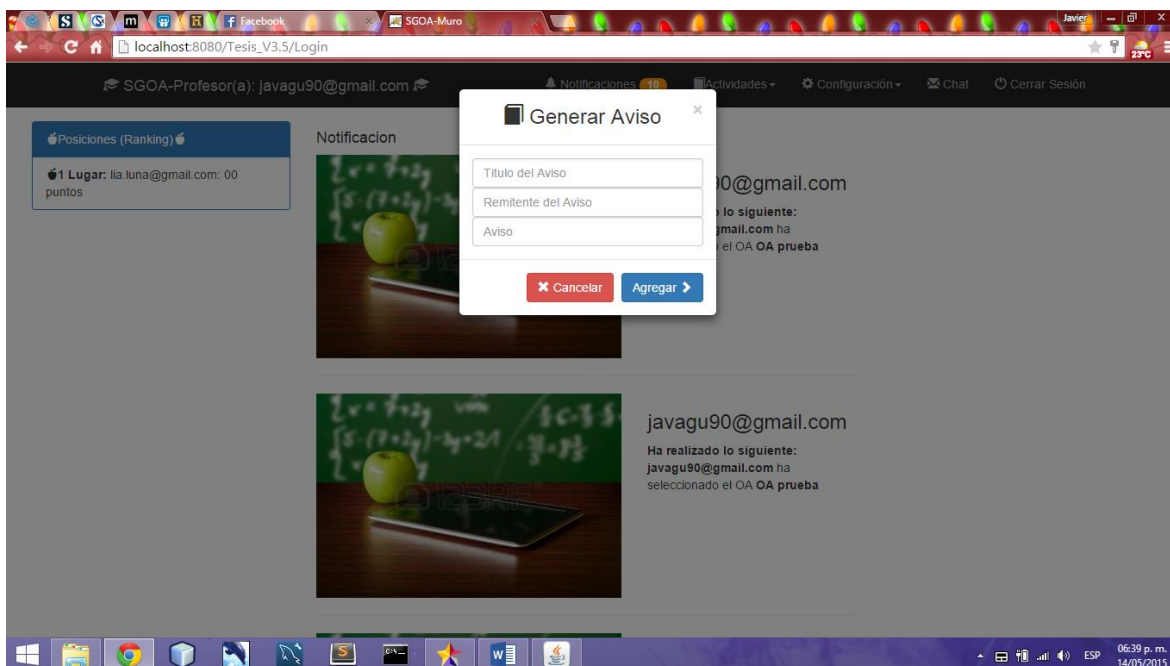


Figura 4.39 *Formulario para generar un aviso.* Fuente: Propia

➤ Búsqueda de un OA

Para poder localizar un OA es necesario que entremos al menú “Actividades” seguido del submenú “Mis OA’s” una vez realizado esto, el sistema nos enviara a la página de actividades en la cual podemos realizar la búsqueda de un OA.

La búsqueda de un OA puede realizarse de cinco maneras:

- *General:* Sin necesidad de insertar algo en el campo de texto, simplemente con oprimir el botón “Buscar” nos arrojará todos los OA’s que han sido creados.
- *Por título:* Debemos insertar el título del OA que queremos localizar, de esta manera, si existe, nos arrojará un resultado
- *Por autor:* Insertando cualquier nombre o apellido del autor nos lanzará los OA’s creados por dicho personaje
- *Por palabra clave:* insertando una de las palabras clave que contenga el OA, mostrará todos los OA’s que contengan esa palabra.
- *Por área:* colocando el área a la que pertenece el OA, la búsqueda mostrara todos los OA’s que pertenezcan a área buscada.

Todos los resultados de búsqueda son mostrados en una tabla con su información básica, además de un enlace llamado “Realizar” en el cual, dando clic, nos envía a la realización del objeto de aprendizaje.

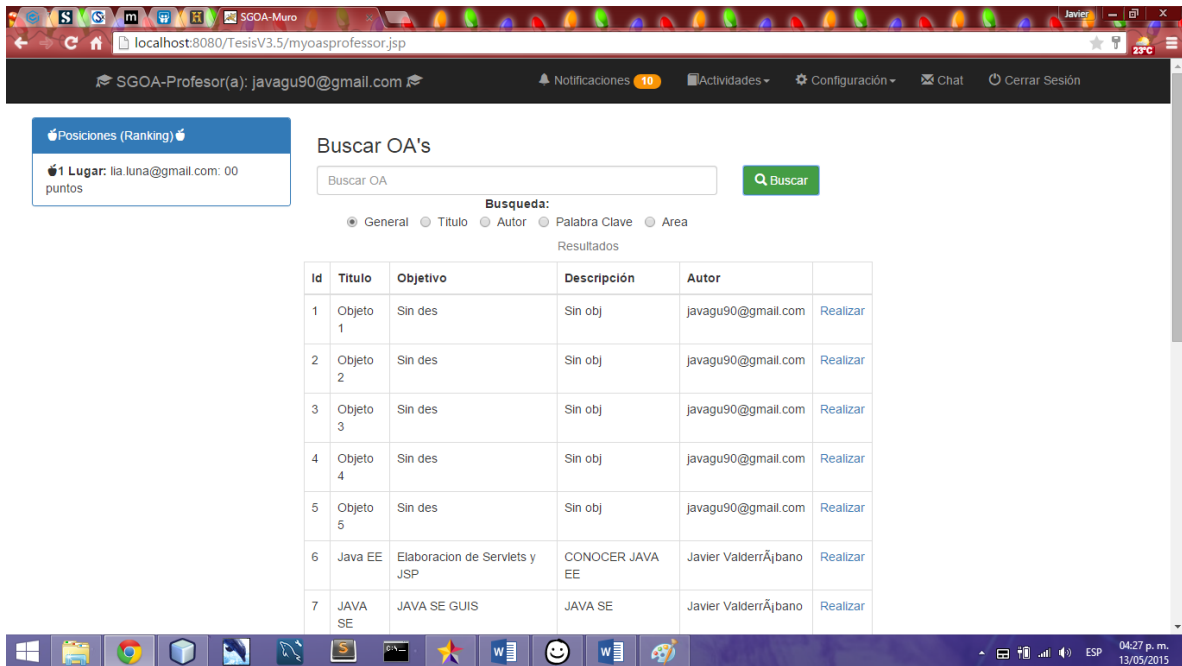


Figura 4.40 Búsqueda de un OA. Fuente: Propia

➤ Otros OA's creados

En la sección "Otros OA's creados" se muestran OA's de diferentes autores, estos OA's no son resientes, son OA que han sido creados con mayor antigüedad, para que diferentes usuarios puedan hacer uso de ellos, simplemente, presionando en la imagen o en el título del OA, se les enviará a la página de realización del OA

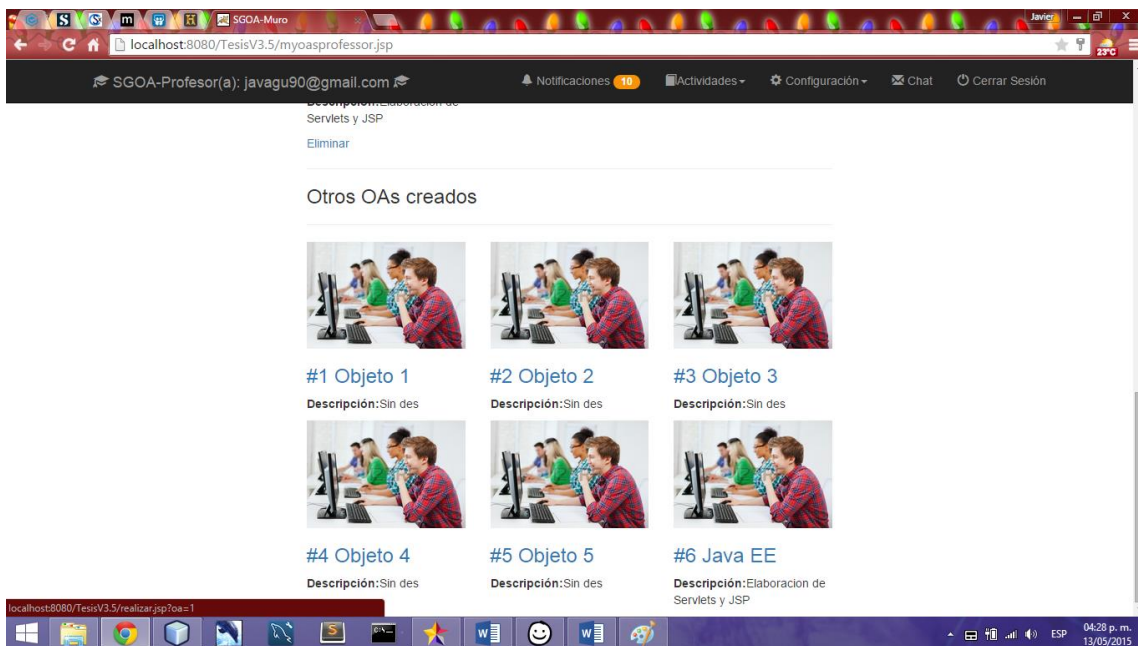


Figura 4.41 Sección "Otros OA's creados". Fuente: Propia

➤ OA's creados (Mis OA's)

En esta sección mostramos los OA's que han sido creados por el usuario, dando la descripción de éste además de colocar un enlace llamado "Eliminar" si es que el autor de este OA requiere eliminarlo, si es eliminado, la página se actualiza y muestra los demás OA's que ha creado

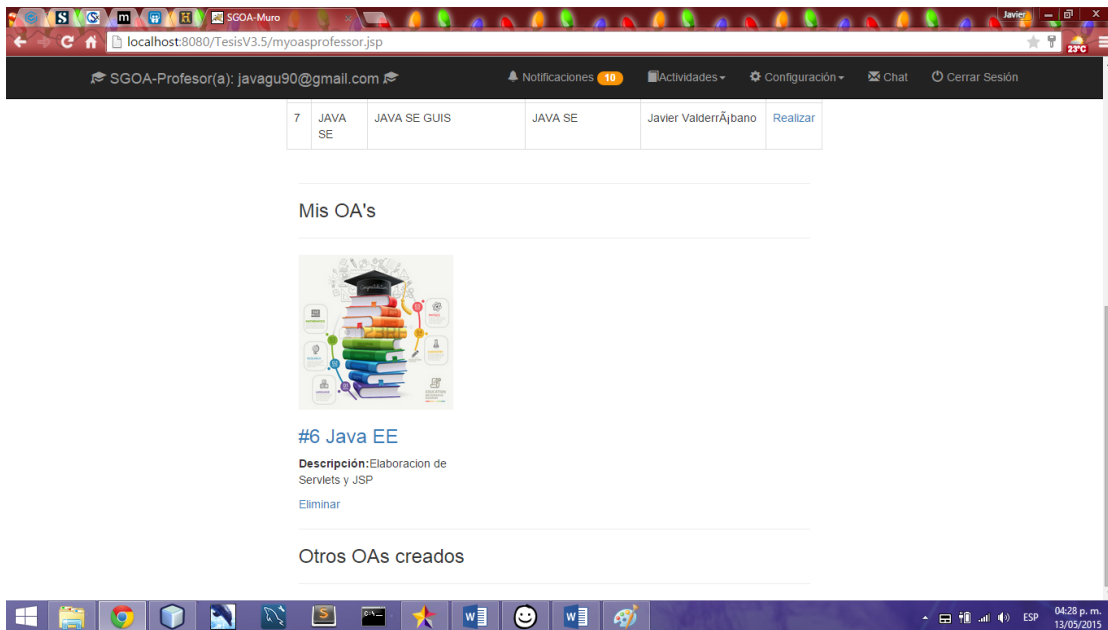


Figura 4.42 Sección "Mis OA's". Fuente: Propia

➤ Realizar un OA

Una vez que haya sido localizado el OA que queremos realizar y haber presionado el acceso a él (ya sea por su imagen, título o enlace "realizar") se redirigirá a una página en la cual encontraremos toda la información del OA en una tabla y en una segunda, encontraremos todas las actividades que pertenecen a él, con la información básica de dicha actividad (título, descripción, objetivo, tipo de saber y tipo de recurso). La manera en la que se pueden acceder a las actividades es presionando el enlace llamado "realizar".

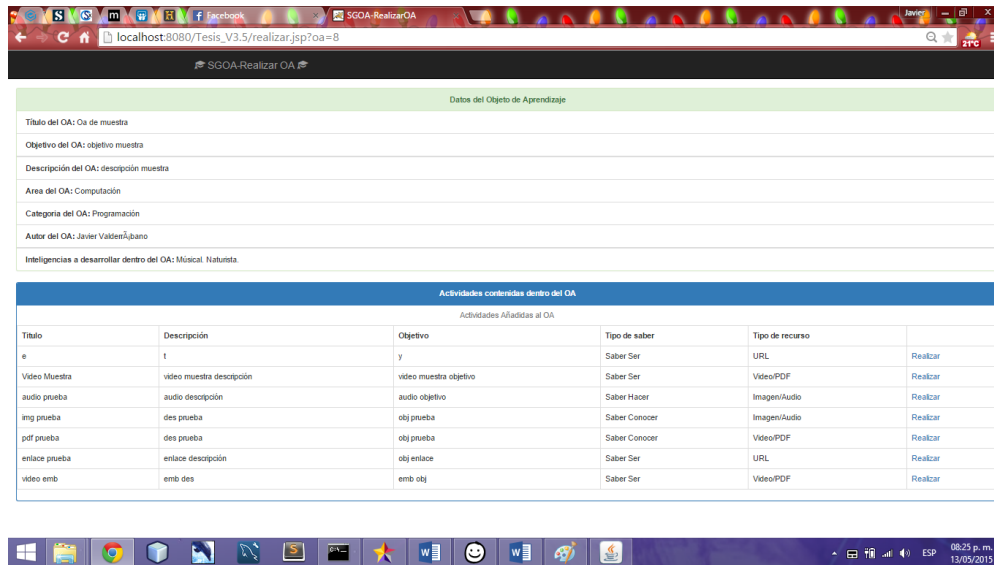


Figura 4.43 *Página “Realizar OA”*. Fuente: Propia

➤ Realizar actividad (recurso de código embebido)

Una vez que accedemos a la actividad seleccionada, si esta es de un recurso en el cual se agregó un código embebido (disponible para videos y archivos PDF), se redireccionará a una página en la cual contiene una tabla con la información restante de la actividad y el recurso añadido a ésta

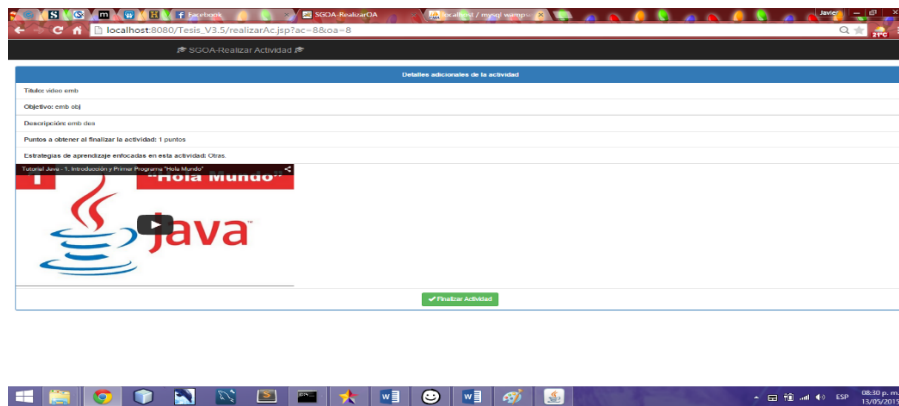


Figura 4.44 *Página “Realizar Actividad” correspondiente a un recurso embebido*. Fuente: Propia

➤ Realizar actividad (recurso de tipo enlace “URL”)

Una actividad de tipo URL se basa en colocar un enlace nombrado “Ir al sitio web” dentro de este se encuentra la página a la cual el sistema abrirá una nueva pestaña con el contenido de la página web almacenada para su actividad

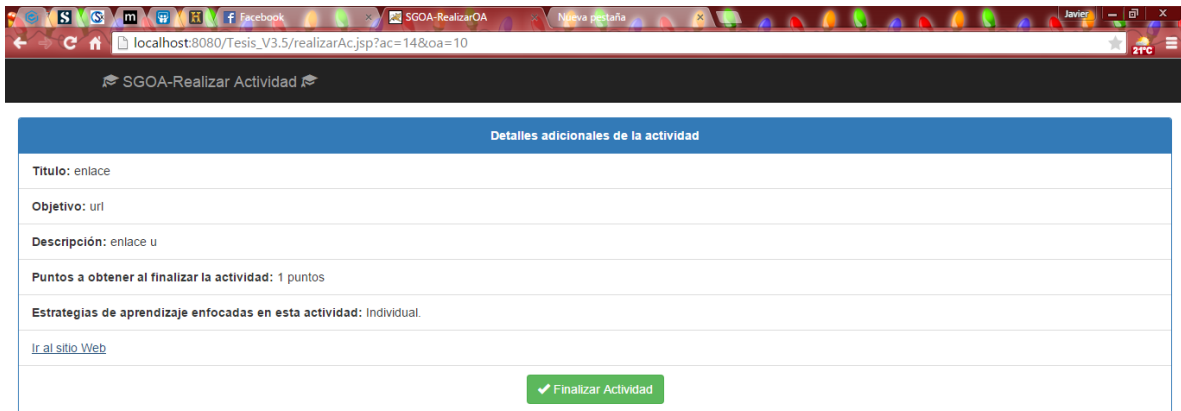


Figura 4.45 *Página “Realizar Actividad” correspondiente a un enlace.* Fuente: Propia

➤ Realizar actividad (recurso de imagen)

La actividad de imagen coloca y muestra la imagen que ha sido almacenada dentro de la creación de la actividad, mostrándola al usuario que la realizará.

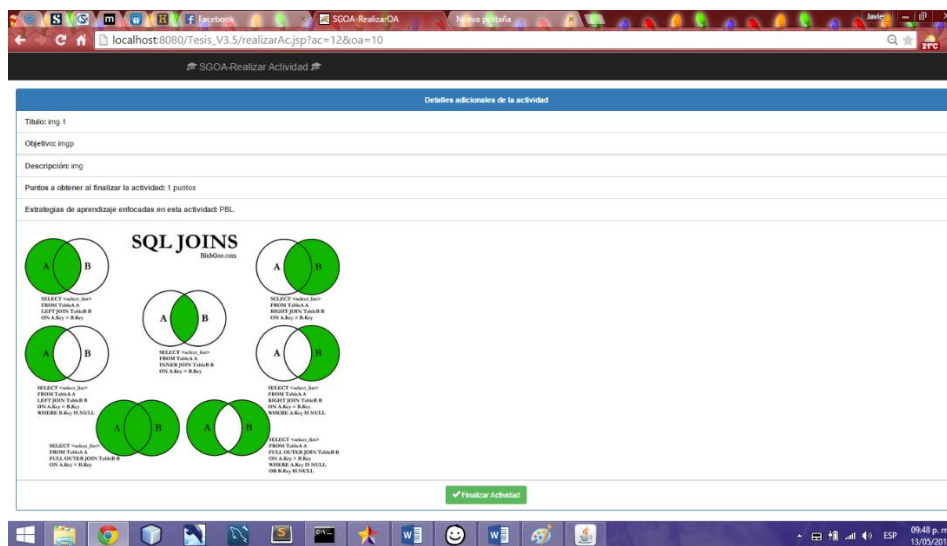


Figura 4.46 *Página “Realizar Actividad” correspondiente a un recurso de imagen.* Fuente: Propia

➤ Realizar actividad (recurso de archivo PDF)

Cuando es seleccionada una actividad que tiene como recurso un archivo PDF se muestra una página en la cual, además de la información, se muestra el archivo PDF para que el

usuario pueda consultarlo dentro del sistema, si él decide guardarlo dentro del archivo se encontrará el botón de guardar documento mismo que inserta HTML para su visualización

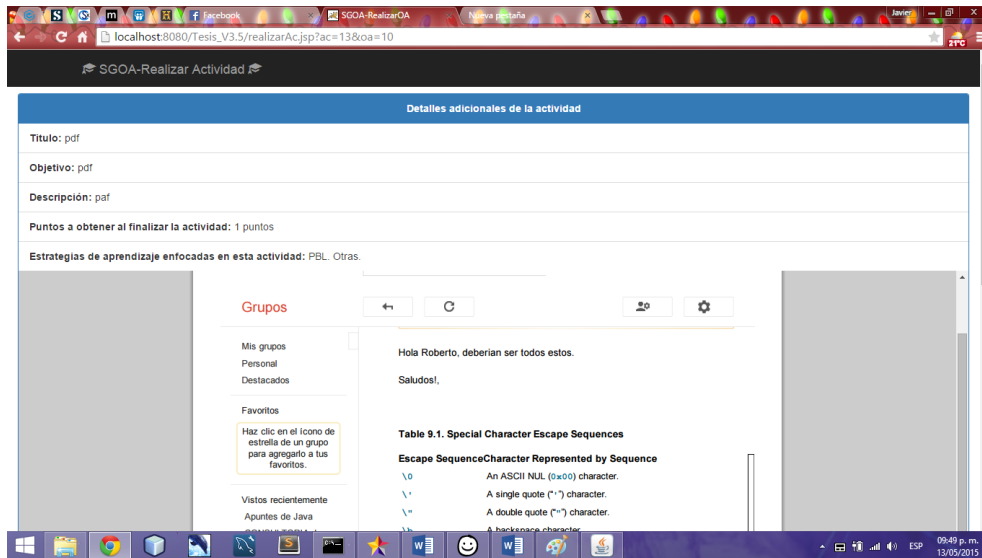


Figura 4.47 *Página “Realizar Actividad” correspondiente a un archivo PDF.* Fuente: Propia

➤ Realizar actividad (recurso de audio)

Una actividad de audio agrega un mini-reproductor en el cual se auto reproduce el sonido que se haya insertado dentro de la actividad, el usuario puede volver a escucharlo y controlar el volumen con los respectivos controles del mini-reproductor

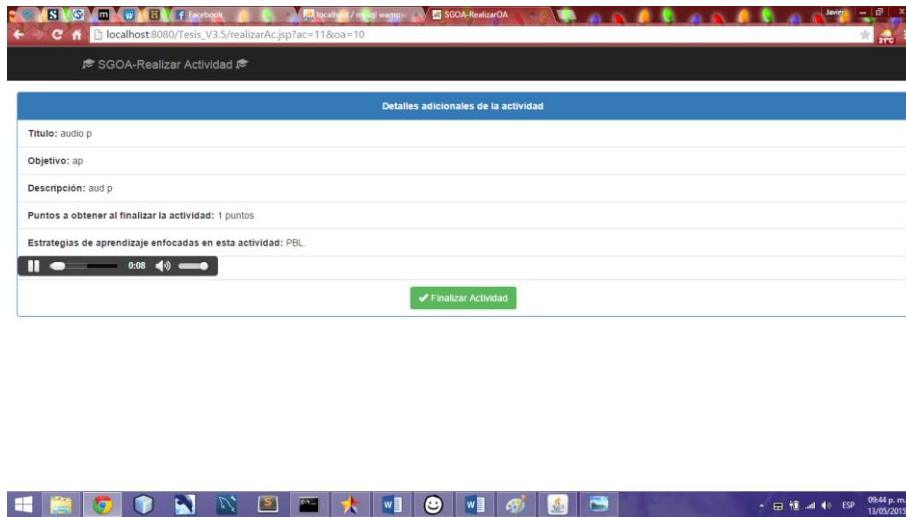


Figura 4.48 *Página “Realizar Actividad” correspondiente a un recurso de audio.* Fuente: Propia

➤ Realizar actividad (recurso de video)

Una actividad con recurso de video muestra en la página un reproductor de video, el cual requiere que el usuario presione “play” sobre el control del reproductor para que este video pueda ser visualizado dentro del sistema, además de poder controlar su reproducción sonido y si lo quiere ver a pantalla completa o como se visualiza al principio de la página

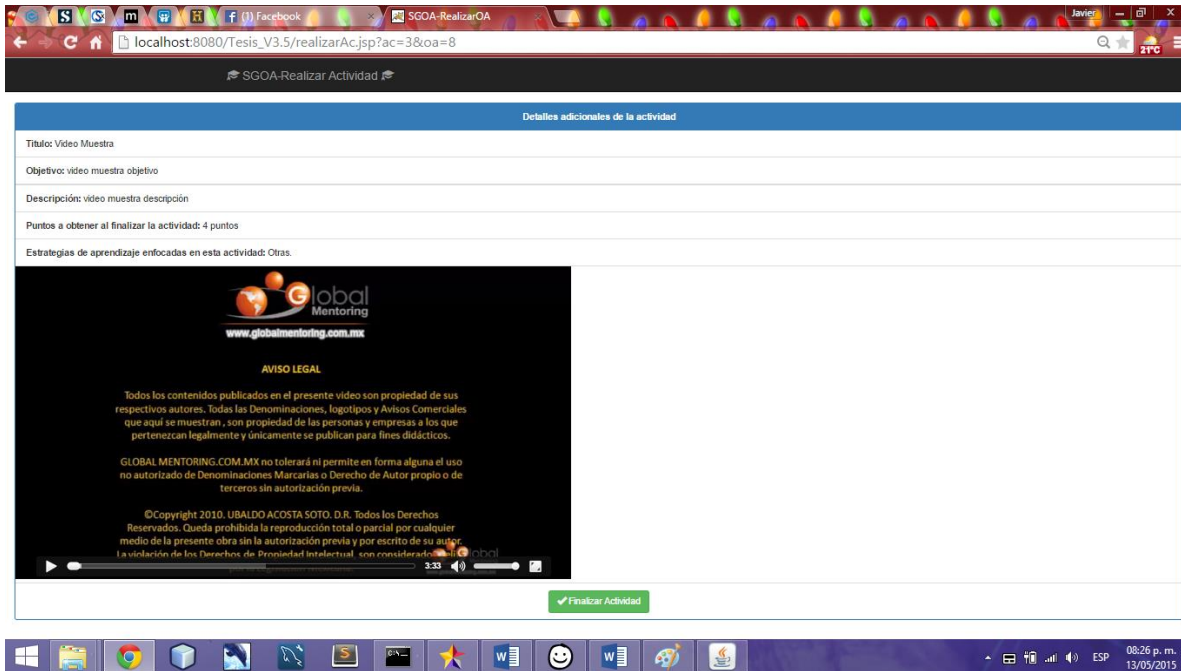


Figura 4.49 *Página “Realizar Actividad” correspondiente a un recurso de video.* Fuente: Propia

Para finalizar, dentro de esta página podemos encontrar un botón en la parte inferior central, el cual tiene la leyenda “Finalizar actividad” una vez presionado este botón, se le dice al sistema que el usuario completo la actividad seleccionada y es entonces cuando asigna los puntos que se ofrecen en la actividad al puntaje del usuario para poder de este modo seguir con la competencia que se generó desde que el usuario se registra

- **Codificación y MVC**

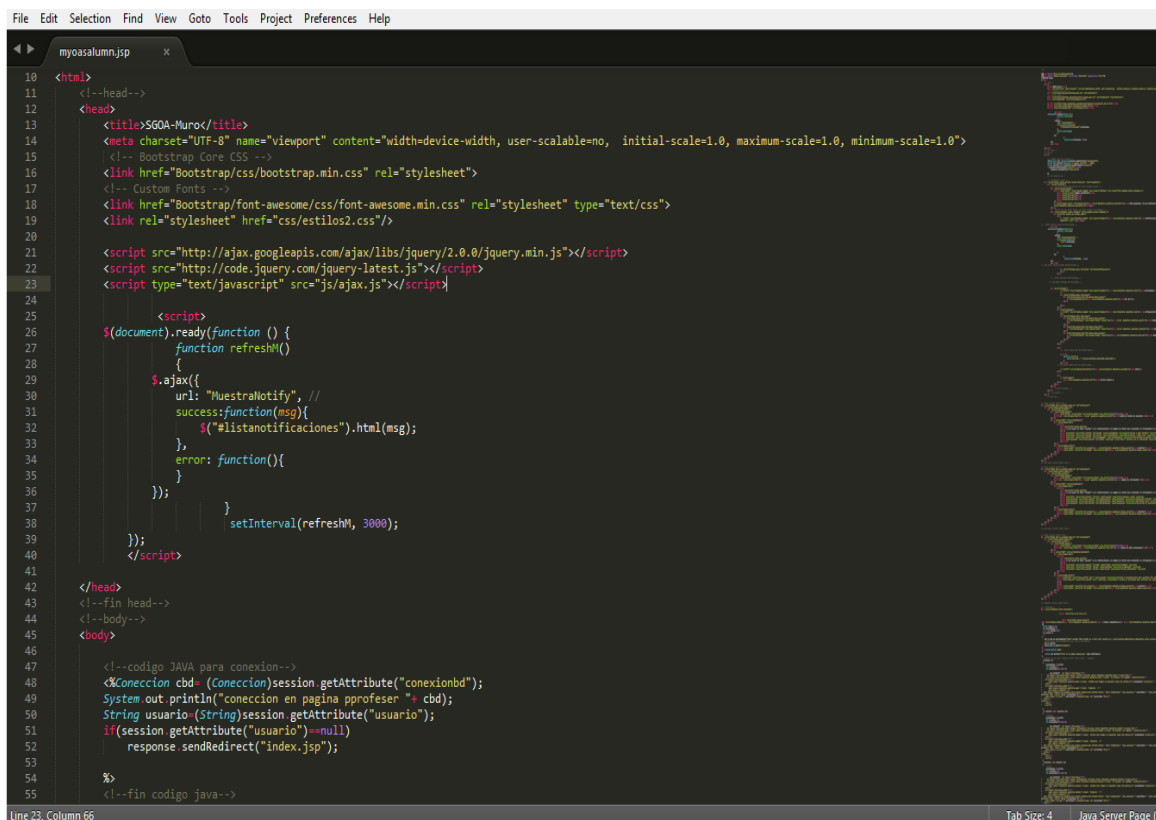
Debido a la elaboración de este sistema de manejar un patrón de codificación MVC (Modelo-Vista Controlador) mostraremos fragmentos de código con una breve explicación conforme al modelo de codificación planteado y explicado con anterioridad.

- Java Server Pages (JSP) & HTML5: Vista

Para manejar el lado del cliente/usuario o lo que llamamos en computación como front-end (vista) se han utilizado tecnologías recientes para el desarrollo web, entre ellas

encontramos las JSP que son páginas dinámicas que pueden contener código java dentro de ellas y que son compiladas por un contenedor de aplicaciones como Tomcat o Glassfish, estas páginas son las que muestran al usuario la información que necesita ver, en conjunción con scripts de javascript y hojas de estilos (CSS).

En este sistema podremos observar que la mayoría de las páginas que son mostradas a los usuarios tienen la extensión “.jsp” ya que estas nos servirán de motor para tener una interacción en primera instancia con el usuario, seguido después de un servlet o un POJO (siglas que serán descritas más adelante) y poder así crear la conjunción MVC.



```
10 <html>
11 <!--head-->
12 <head>
13 <title>SGOA-Muro</title>
14 <meta charset="UTF-8" name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
15 <!-- Bootstrap Core CSS -->
16 <link href="Bootstrap/css/bootstrap.min.css" rel="stylesheet">
17 <!-- Custom Fonts -->
18 <link href="Bootstrap/font-awesome/css/font-awesome.min.css" rel="stylesheet" type="text/css">
19 <link rel="stylesheet" href="css/estilos2.css">
20
21 <script src="http://ajax.googleapis.com/ajax/libs/jquery/2.0.0/jquery.min.js"></script>
22 <script src="http://code.jquery.com/jquery-latest.js"></script>
23 <script type="text/javascript" src="js/ajax.js"></script>
24
25 <script>
26 $(document).ready(function () {
27     function refreshM()
28     {
29         $.ajax({
30             url: "MuestraNotifiy", //
31             success: function(msg){
32                 $("#listanotificaciones").html(msg);
33             },
34             error: function(){
35             }
36         });
37     }
38     setInterval(refreshM, 3000);
39 });
40 </script>
41
42 </head>
43 <!--fin head-->
44 <!--body-->
45 <body>
46
47 <!--codigo JAVA para conexion-->
48 <%Conexion cbd= (conexion)session.getAttribute("conexionbd");
49 System.out.println("conexion en pagina pprofeser "+ cbd);
50 String usuario=(String)session.getAttribute("usuario");
51 if (session.getAttribute("usuario")==null)
52     response.sendRedirect("index.jsp");
53
54 %>
55 <!--fin codigo java-->
```

Figura 4.50 Fragmento de código de la página “myoasalumn.jsp” en él se puede observar código java, etiquetas HTML y scripts de javascript. Fuente: Propia

- Servlets Java: Controlador

Para enlazar las acciones que tiene el usuario con la base de datos es necesario un controlador, tarea que es realizada por los servlets, los cuales mediante el envío de parámetros por medio de los métodos POST o GET de HTML controla las acciones que realiza el usuario, conecta la vista con el modelo, en pocas palabras, es un puente entre ambas partes del patrón MVC.

```
File Edit Selection Find View Goto Tools Project Preferences Help
UpdatePass.java x
13 import javax.servlet.http.HttpServlet;
14 import javax.servlet.http.HttpServletRequest;
15 import javax.servlet.http.HttpServletResponse;
16 import modelo.Coneccion;
17
18 /**
19  *
20  * @author Javy
21  */
22 @WebServlet(name = "UpdatePass", urlPatterns = {"/UpdatePass"})
23 public class UpdatePass extends HttpServlet {
24
25     /**
26      * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
27      * methods.
28      *
29      * @param request servlet request
30      * @param response servlet response
31      * @throws ServletException if a servlet-specific error occurs
32      * @throws IOException if an I/O error occurs
33      */
34     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
35         throws ServletException, IOException {
36         response.setContentType("text/html;charset=UTF-8");
37         RequestDispatcher rd=null;
38         String passViejo, passNuevo, passNuevoR, usuario;
39         Coneccion cbd= (Coneccion)request.getSession().getAttribute("conexionbd");
40         passViejo=(String)request.getParameter("passViejo");
41         passNuevo=(String)request.getParameter("passNuevo");
42         passNuevoR=(String)request.getParameter("passNuevoR");
43         usuario=(String)request.getParameter("usuario");
44
45         boolean ok=cbd.cambiarPass(usuario, passViejo, passNuevo, passNuevoR);
46         request.getSession().invalidate();
47         if(ok==false)
48         {
49             if(cbd.error==5)
50             {
51                 .....
52             }
53         }
54     }
55 }
```

Figura 4.51 Servlet “UpdatePass.java” encargado de realizar la actualización de la contraseña del usuario, puede observarse la obtención de parámetros mediante “request.getParameter()”.Fuente: Propia

- Plain Old Java Object (POJO) :Modelo

Por ultimo para guardar la información y realizar consultas a la base de datos, nos encontramos con un POJO que es código java puro, (como una simple clase) en la cual realizamos la conexión a la base de datos y creamos funciones para interconectar el modelo con el controlador y este le dé una respuesta al usuario mediante la vista.

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Coneccion.java x
26 public Coneccion ()
27 {
28     try
29     {
30         Class.forName("com.mysql.jdbc.Driver");
31         conexion = DriverManager.getConnection("jdbc:mysql://localhost/mydb", "javagu90", "heil");
32         s = conexion.createStatement();
33         //OptionPane.showMessageDialog(null, "Exito!!!");
34     }
35     catch(Exception e)
36     {
37         System.out.println("mal "+ e.getMessage());
38     }
39 }
40
41 public void insertarUsuario(String nombre, String apellidos, String correo, String usuario, String pass, String tipoUsuario)
42 {...15 lines }
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57 public boolean insertarPuntaje(String usuario, int puntaje)
58 {...16 lines }
59
60
61
62
63
64
65
66
67 //updatepuntaje UPDATE puntaje SET puntaje = puntajeNuevo WHERE Usuarios_idUsuarios = id;
68 public void updatePuntaje(int puntajeNuevo, int idUser)
69 {...11 lines }
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89 public boolean acceso(String user, String pass)
90 {...23 lines }
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114 public boolean repetido(String user)
115 {...19 lines }
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136 public String tipoUser(String user)
137 {...22 lines }
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160 public boolean cambiarUser(String userViejo, String userNuevo, String userNuevoR, String pass)
161 {...48 lines }
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
```

4.4 En Resumen

Una buena organización en los diagramas correspondientes a una ingeniería de software es fundamental para un buen desarrollo e implementación de cualquier sistema.

En este capítulo vimos los diferentes diagramas que fueron realizados para tener una base de cómo crear y desarrollar el sistema, observamos prototipos de diseño que nos ayudaron a tener una idea más clara de cómo podríamos formar la interfaz de la plataforma sin tener que realizar programación para estos, una vez creados y aprobados los diseños se empezó la implementación del conjunto de diagramas y prototipos para dar como resultado el sistema gestor de objetos de aprendizaje que vimos a detalle en este capítulo.

En el siguiente capítulo veremos la validación del sistema mediante pruebas y aclarar las fallas que se lleguen a presentar a través de pruebas formales de ingeniería de software

Capítulo 5: Validación

Ningún sistema está exento a fallos, debido a que los sistemas están hechos por humanos que en cualquier momento pueden equivocarse, es por eso que cada proyecto tiene su monitoreo de calidad y búsqueda de fallos y en este sistema, se trató de que el número de fallos sea mínimo para poder crear en el usuario que lo utilice una mayor confiabilidad en la plataforma.

A continuación se presentan pruebas de caja negra realizadas a ciertos módulos de la aplicación, se denominan así, porque solo verifican si el proceso de salida de un módulo corresponden con lo que se espera que haga el sistema a partir de datos de entrada.

Modulo	Entrada	Salida	Status
Crear OA	Título: Curso Hibernate Descripción: Aprender el framework Hibernate parte del patrón MVC Objetivo: descubrir las funcionalidades de Hibernate Área: Computación Categoría: Programación	Se mostró el mensaje de que el OA ha sido creado, la información se guardó en la BD y se mostró la ventana para añadir actividades al OA	Realizado Satisfactoriamente
Eliminar OA	Id_OA: 6	Actualiza la página Mis OA's	Realizado Satisfactoriamente
Añadir Actividad	Título: img1 Descripción: joins sql Objetivo: aprender y comprender los joins de sql Tipo de saber: Saber Ser Puntos: 1 Archivo: sql_joins.jpg Estrategias de aprendizaje: PBL	Muestra mensaje de que se ha agregado la actividad al OA, se guarda en la base de datos, se inserta en la tabla actividades añadidas y muestra la ventana del paso 3	Realizado Satisfactoriamente
Añadir área	Área: Matemáticas	Muestra mensaje que se ha realizado	Realizado Satisfactoriamente

		la captura del área se guarda en base de datos y se actualiza la lista de áreas	
Añadir categoría	Categoría: Calculo	Muestra mensaje que se ha realizado la captura del área se guarda en base de datos y se actualiza la lista de categorías	Realizado Satisfactoriamente
Buscar OA	Palabra a buscar: Java	Muestra los OA que contienen la palabra clave "java"	Realizado Satisfactoriamente
Actualizar puntaje	Id_usuario= 1 Puntos= 2	Redirecciona a la pantalla de actividades del OA y actualiza el puntaje del usuario que realizo la actividad	Realizado Satisfactoriamente

Tabla 5.1 Descripción de pruebas de caja negra del sistema Fuente: Propia

Después de haber realizado las pruebas de caja negra se encontraron otras fallas, una de las fallas que se había presentado era al momento de crear una actividad, ya que la actividad no era agregada o era insertada en un tipo de recurso diferente al que se había especificado, esto se daba debido a que una sentencia if, involucraba a otras variables que no necesitaban participar en la condicional, la solución, eliminando de la condicional dichas variables de las cuales no se necesitaba su participación.

Dentro de esa misma sentencia se propagaba el error de que no se subía el archivo o no se guardaba la ruta del recurso agregado, esto se daba debido a que los campos de los recursos jamás limpiaban la información y con ello se confundía el tipo de recurso entre la

carga y el muestreo de éste, la solución fue que cada vez que se cargue el formulario, el campo del recurso se vacié para que de esa manera la condicional se pueda cumplir para los diferentes casos propuestos.

Otro fallo que presentaba la plataforma era un fallo de compatibilidad, ya que distintas librerías de jQuery participaban dentro de la plataforma, una librería para la interfaz del muro y otra para el chat, lo que hacía que no sirviera una de las dos secciones en cualquiera de los casos, la solución que se hizo fue rediseñar la interfaz de la plataforma y agregar el menú “Chat” el cual maneja otra librería independientemente de la librería de la interfaz del “muro” y con ello evitamos que se pierda la conexión del chat o las ventanas emergentes del muro no funcionen adecuadamente.

Otro fallo encontrado fue al momento de eliminar un OA, ya que al presionar el enlace “Eliminar” este eliminaba TODOS los registros creados dentro de la tabla OA’s de la base de datos, se verifico el comando MYSQL para ver su falló, y aunque manejaba el identificador del OA a eliminar, este no lo identificaba ya que faltaban algunos caracteres especiales, al no identificar el número del OA a eliminar, se eliminaban todos aunque pareciera que la consulta estaba bien, se reestructuró esta con los caracteres especiales faltantes se corrigió el problema.

En Resumen

El proceso de validación, calidad y búsqueda de fallos es un factor primordial en la elaboración de un sistema, debido a que en esto se basa la confianza del usuario en querer utilizar nuestra plataforma o no, es por ello que se deben de hacer las pruebas necesarias para hallar los posibles errores que pueda tener un sistema, porque como lo mencionamos desde un principio, ningún sistema está vetado de tener fallos, debido a que esta realizado por personas las cuales en cualquier momento pueden equivocarse, el punto principal aquí, es identificar el fallo y solucionarlo para poder dar una mayor calidad a nuestro sistema y por consiguiente una confiabilidad al usuario de que nada malo puede ocurrir con su información.

Capítulo 6: Conclusiones y Trabajo a futuro

Evaluación de cara al trabajo a futuro

Una vez que fue el sistema montado en un servidor, se realizaron diferentes tipos de pruebas además, sugerencias que se pueden desarrollar en la aplicación como un trabajo a futuro, dentro de las que más fueron mencionadas y más interés hubo fueron las siguientes:

➤ **Mejora del chat**

Los usuarios comentaron que el chat debería de ser más funcional, permitir el envío de mensajes pulsando la tecla “Enter” y no tener que dar clic forzosamente en el botón enviar, además de que el chat se ingrese automáticamente al seleccionarlo y no tener que dar un nombre de usuario y oprimir el botón “conectar” para poder establecer la comunicación, además de poder rescatar los mensajes antiguos y para finalizar, evitar la apertura de varias ventanas de chat.

➤ **Recuperar contraseña en caso de olvido**

Los usuarios propusieron que se pueda recuperar la contraseña mediante un registro en el cual se completen datos que solo el dueño de la cuenta pueda saber y al ser validados, el sistema envíe un correo, enviando la contraseña que el usuario requiere para el acceso.

➤ **Creación de evaluaciones**

Una vez creado el OA y la actividad, los usuarios propusieron la creación de evaluaciones para poder así, dar retroalimentación y calificación al usuario mediante la realización del OA en su totalidad (actividades y evaluaciones)

➤ **Crear una sección/menú de ayuda**

Algunos de los usuarios registrados, no se adaptaron fácilmente a la plataforma por lo cual, sugirieron se les explicara que realiza cada sección de la plataforma, con lo cual se propone crear un menú o sección de ayuda, en la cual se pueda, mediante videos u otras TIC asesorar al usuario en su estancia dentro de la plataforma.

➤ **Permitir modificaciones a los OA's**

Por el momento, los OA's no pueden ser modificados en ninguna parte de ellos, los usuarios al ver la necesidad de replantear alguna actividad, propusieron que los OA's puedan ser modificables en cualquier aspecto para poder así brindar un mejor recurso.

➤ **Restricción para la realización repetitiva de una actividad**

Los OA's y actividades al manejarse dentro de un sistema de puntajes, deben de poder solo realizarse una vez, ya que como los usuarios lo comentaron ellos pueden alcanzar un puntaje alto con la realización de la misma actividad varias veces, es por ello que se maneja la restricción de solo poder realizar la actividad una única vez.

➤ **Manejar el registro de un profesor de manera diferente a la de un alumno**

Con ello se propone crear un registro diferente, ya sea mediante una llave o dar de alta al usuario mediante un módulo exclusivo para las directivas académicas, las cuales permitan el registro de un profesor, de esta manera se evita que un profesor pueda ser registrado como profesor.

Dentro de otras propuestas de trabajo a futuro que se puede realizar en este proyecto se destacan las siguientes:

➤ **Inserción de juegos educativos**

Debido a que este sistema se vio dirigido a la educación superior y se propuso un modelo de red social, la inserción de juegos educativos, de esta manera vinculado con el sistema de puntajes, la educación se volvería más didáctica y dinámica, de esta manera evitaríamos la deserción al sistema y podríamos obtener la atención de más público.

➤ **Reusabilidad de actividades**

La usabilidad es un tema que está tomando mucho interés en diferentes medios web, en este sistema se utilizó reusabilidad de los objetos de aprendizaje, siendo que cualquier persona que este registrada en el sistema puede acceder a ella, ahora el reto está en hacer lo mismo pero para las actividades, de esta manera, al crear un OA podríamos poner la opción de, crear una actividad o *añadir una actividad creada con anterioridad* de esta manera ahorraríamos tiempo al profesor y la usabilidad podría llegar a niveles inimaginables.

➤ **Mejora de la interfaz**

La interfaz que se ha propuesto en este sistema ha sido una interfaz agradable pero de cierta manera, básica, ya que no contiene demasiados elementos y no cansa al usuario por la sobrecarga de componentes, pero se podría mejorar esta interfaz con diseños creados por gente especializada en manejo de colores, mercadotecnia entre otras, haciendo el sistema más agradable a la vista del usuario.

➤ **Publicación manual por parte del usuario**

Por el momento, el sistema registra las actividades que ha hecho un usuario y las publica en el muro para dar una información de lo que el alumno ha realizado o en su caso también el profesor, pero todo esto se hace de manera automática por el sistema, la propuesta a futuro es crear el sistema, de tal manera que las publicaciones se realicen de manera autónoma por el usuario, para que de esa manera se pueda compartir más información.

➤ **Mejora del sistema de puntajes**

El sistema de puntajes que se ha utilizado en este sistema es llamado “ELO” utilizado en el ajedrez para poder clasificar a los participantes y así evitar que un oponente muy fuerte se enfrente contra un principiante, además de esta manera evitar los “puntajes inalcanzables” que en vez de motivar al usuario a realizar la actividad, lo desalienta ya que “jamás podrá vencer al líder”. Este sistema de puntajes podría ser mejorado en conjunción con la creación de juegos educativos (como lo propusimos con anterioridad) para que de esta manera la competencia se vuelva más interactiva y agradable para el competidor

➤ **Asignación de OA al alumno**

Un alumno puede acceder a un objeto de aprendizaje, pero no se queda registrado en como parte de su cuenta, la propuesta que se hace, es que una vez que el alumno seleccione el objeto de aprendizaje, este, pueda ser pausado, reanudado y sea parte de la cuenta del alumno, sin necesidad de volver a buscar el objeto de aprendizaje y realizarlo de nuevo.

➤ **Interacción con otras redes sociales**

Debido a que nuestro sistema tienen como objetivo ser una red social educativa, no puede quedarse sin expandir, es por ello que se propone crear vínculos en los cuales se empiece

a compartir la información hecha dentro del sistema hacia otras redes sociales para que así el sistema pueda ir expandiéndose poco a poco y ser conocido por más personas.

➤ **Blog de preguntas y respuestas**

Un sistema de blog como preguntas y respuestas en donde todo el mundo que está dentro de plataforma pueda opinar acerca de diferentes temas posteados y dar un puntaje por realizar una posible respuesta y dar un mayor puntaje a la respuesta correcta para así incentivar la cooperación e interactividad entre usuarios de diferentes niveles.

Conclusiones

Uno de los principales objetivos en cualquier país es la educación, ya que los niños y jóvenes de hoy, serán los profesionistas del mañana, esa base en la que cualquier país puede salir adelante por sí mismo y es por ello que cada nación propone diferentes planes educativos y poder así eliminar cada día más la deserción.

En este documento hemos abarcado una propuesta de solución, que si no acabara de manera total el abandono en las escuelas, si ayudará significativamente a evitar este hecho desafortunado. Se planteó la propuesta de una educación a distancia, la cual, en México es necesario, debido a los diferentes problemas (como paros o huelgas de profesores o la falta de infraestructura para poder tomar clases) que pasan en esta nación.

Vimos que la propuesta además de cumplir con un esquema que está tomando fuerza, promueve la gestión de objetos de aprendizaje, fundamentales para el e-learning, pero no solo eso, objetos de aprendizaje que pueden ser reutilizables y seleccionados por cualquier persona la cual tenga el interés de aprender y tener nuevos conocimientos.

Pudimos observar lo que es un objeto de aprendizaje y con qué especificaciones debe de cumplir para lograr la eficiencia de dicho objeto, debido a las nuevas tecnologías, los objetos de aprendizaje transforman la educación de una manera convencional a una más interactiva y participativa mediante recursos multimedia como videos, audios, imágenes y enlaces a páginas, los cuales llegan enseñar al alumno de distintas maneras, porque, recordemos que no todos los alumnos tienen las mismas capacidades de retención por un solo medio, hay alumnos que aprenden realizando ejercicios, otros estudiando los temas o simplemente

escuchando o viendo lo que el profesor enseña en clase, en este caso, lo que la plataforma publica al público.

Las redes sociales están tomando un gran impacto en la comunidad cibernética, es por ello que en este proyecto se propuso manejar una interfaz y concepto de red social, para abarcar y conquistar al estudiante que piensa que el estudio no le ayudará de nada en la vida, y que “Facebook” o “Twitter” son lo de hoy, es por ello que tomamos la debilidad del estudiante, que son las redes sociales y las adecuamos a la educación para poder fortalecer sus distintas habilidades.

Promovimos un nuevo esquema dentro de la educación a distancia, que es el sistema de puntajes, dicho sistema, nos ayudará a evaluar al alumno y además, crear cierto interés en la realización de las actividades para poder así “ser siempre el primer lugar” pero, para evitar esto, se ha creado un sistema de rangos clasificatorios, de tal manera que evitamos los puntajes inalcanzables y cualquier persona puede ser el número uno, con esto, creamos una competencia sana y divertida que ayuda a la formación e intelecto del usuario.

Un sistema gestor de objetos de aprendizaje ayudará como suplemento o complemento a la educación y formación de cualquier persona en vías de desarrollo y ganas de superación.

Para concluir este trabajo quiero compartir una frase que me agradó y que me gustaría que las tuvieran en mente en cada momento de su estudio y de su vida:

“Estudia como si fueras a vivir por siempre, y vive como si fueras a morir mañana, porque recuerda que lo que con mucho trabajo se adquiere, más se ama.”

Referencias

[Bello13] R. E. Bello Díaz. *E-Learning Uso de Internet y sus herramientas en la educación a distancia modalidad virtual* <http://www.educar.org/articulos/educacionvirtual.asp>

(Consulta Agosto 2014)

[Bene09] Benedicto XVI, *Nuevas tecnologías, nuevas relaciones. Promover una cultura de respeto, de diálogo y de amistad*. Mensaje para la XLIII Jornada Mundial de las Comunicaciones Sociales, 24 de mayo de 2009.

(Consulta Agosto 2014)

[BM14] Banco Mundial- Tema: Educación <http://datos.bancomundial.org/tema/educacion>

(Consulta Mayo 2014)

[Delo94] Delors, J., *Los cuatro pilares de la educación, La educación encierra un tesoro*. Informe a la UNESCO de la Comisión internacional sobre la educación para el siglo XXI, Madrid, España: Santillana/UNESCO

(Consulta Agosto 2014)

[Díez12] E. J. DÍEZ GUTIÉRREZ. Modelos socio-constructivistas y colaborativos en el uso de las TIC en la formación inicial del profesorado. *Revista de educación*, 358, 175-196. (ISSN: 0034-8082), 2012.

(Consulta Agosto 2014)

[ELO14] Sistema de Puntuación ELO, Wikipedia

http://es.wikipedia.org/wiki/Sistema_de_puntuaci%C3%B3n_Elo

(Consulta Agosto 2014)

[ExpLe14] Expression Language 2.2, Preparation to OCEJWCD EE 6 Exam®, Oracle®, California, USA. (2014)

(Consulta Febrero 2014)

[Frases14] Frases sobre la educación: <http://www.taringa.net/post/apuntes-y-monografias/2018526/Frases-celebres-sobre-educacion.html> (Consulta Enero 2014)

[Garc05] García, L., *Objetos de Aprendizaje. Características y repositorios*, BENED, Abril 2005
(Consulta Agosto 2014)

[Go14] Go. Wikipedia.
<http://es.wikipedia.org/wiki/Go>
(Consulta Agosto 2014)

[GueGo11] Guerrero García Josefina, González Calleros Juan Manuel. Interacción Humano-Computadora, Interacción Seductiva, Diapositiva 5, (2013)
(Consulta Agosto 2014)

[GueGo14] Josefina Guerrero García | Juan Manuel González Calleros, Hacia un Sistema Gestor de Objetos de Aprendizaje, Conciencia Tecnológica 2014 (47)
(Consulta Octubre 2014)

[Hort06] Horton W., 2010, e-Learning by Design, Ed. Pfeiffer.
(Consulta Agosto 2014)

[INEGI10] Indicadores principales del Banco de Información INEGI
<http://www3.inegi.org.mx/sistemas/mexicocifras/#P> (Consulta Mayo 2014)

[JAK95] Nielsen Jakob, 10 Usability Heuristics for User Interface Design
<http://www.nngroup.com/articles/ten-usability-heuristics/>
(Consulta Octubre 2014)

[Lope05] López C., 2005, Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-Learning, Tesis doctoral, Salamanca, España.
(Consulta Abril 2014)

[Loza10] Lozano Rodríguez A., Burgos J. (Compiladores). (2010). Tecnología educativa en un modelo de educación a distancia centrado en la persona, Limusa, México DF.
(Consulta Abril 2014)

[Med13] Medina, C., *Objetos de Aprendizaje*, UAM. 2013
(Consulta Agosto 2014)

[Muji14] J. E. Mújica. Redes sociales: historia, oportunidades y retos.
http://www.forumlibertas.com/frontend/forumlibertas/noticia.php?id_noticia=16428.
(Consulta Agosto 2014)

[Muño07] Muñoz, J., Álvarez, F., *Tecnología de Objetos de Aprendizaje*, Universidad Autónoma de Aguascalientes, 2007.
(Consulta Agosto 2014)

[Muño11] Muñoz, J., Álvarez, F., *Avances en objetos de aprendizaje*, Universidad Autónoma de Aguascalientes, México 2011.
(Consulta Agosto 2014)

[MVC14] *Modelo MVC y sus características:*
<http://www.comusoft.com/modelo-vista-controlador-definicion-y-caracteristicas>
(Consulta Febrero 2014)

[OCEJWCDEE614] Oracle Certified Expert Java Web Component Developer Enterprise Edition 6, Preparation to OCEJWCD EE 6 Exam®, Oracle®, California, USA. (2014)
(Consulta Febrero 2014)

[ORA13] JSR 356, Java API for WebSocket, Johan Vos, California, USA. (2013)
(Consulta Febrero 2015)

[Ossa05] Ossandón Núñez Y. (2005). *Objetos de aprendizaje: Un recurso para el e-Learning*, Universidad de Tapanca, Arica Chile.
(Consulta Agosto 2014)

[Rold10] Roldan D., Buendía F., Ejarne E. García P, Hervás A., Martín J., C. Santos Olga, Oltra J. (2010). *Gestión de proyectos e-Learning*, Alfaomega Grupo Editor, México DF., Pág. 17.
(Consulta Abril 2014)

[Sant08] Santillán J. (2007). Sistema de Gestión de Contenidos, Diapositiva 5, <http://www.slideshare.net/santillan/introduccion-a-los-sistemas-de-gestion-de-contenidos-cms>,(2014)

(Consulta Mayo 2014)

[Sua14] C. SUÁREZ GUERRERO. Estructura didáctica virtual para Moodle. Revista DIM: Didáctica, Innovación y Multimedia, 13. Recuperado el 28 de agosto de 2014 de <http://www.pangea.org/dim/revistaDIM13/Articulos/cristobalsuarez.doc>, 2014.

(Consulta Agosto 2014)

[Wiki14] Aprendizaje Electrónico. Wikipedia, Mayo 2014

http://es.wikipedia.org/wiki/Aprendizaje_electr%C3%B3nico#cite_ref-1

(Consulta Mayo 2014)

	15.- El sistema verifica que el usuario ingresado no este registrado 16.- El sistema crea un nuevo usuario en la plataforma
Flujo alternativo:	12.1.- Si falta algún campo, el sistema lanza un mensaje de error. 13.1.- Si los correos no coinciden, el sistema lanza un mensaje de error 14.1.- Si las contraseñas no coinciden de ninguna forma (incluyendo especificaciones) el sistema lanza un mensaje de error 15.1.- Si el usuario a registrado ya existe, el sistema lanza un mensaje de error
Post-condición:	Se genera un nuevo usuario en la plataforma

Tabla A.1 Descripción del caso de uso *Registrar Usuario*. Fuente: Propia

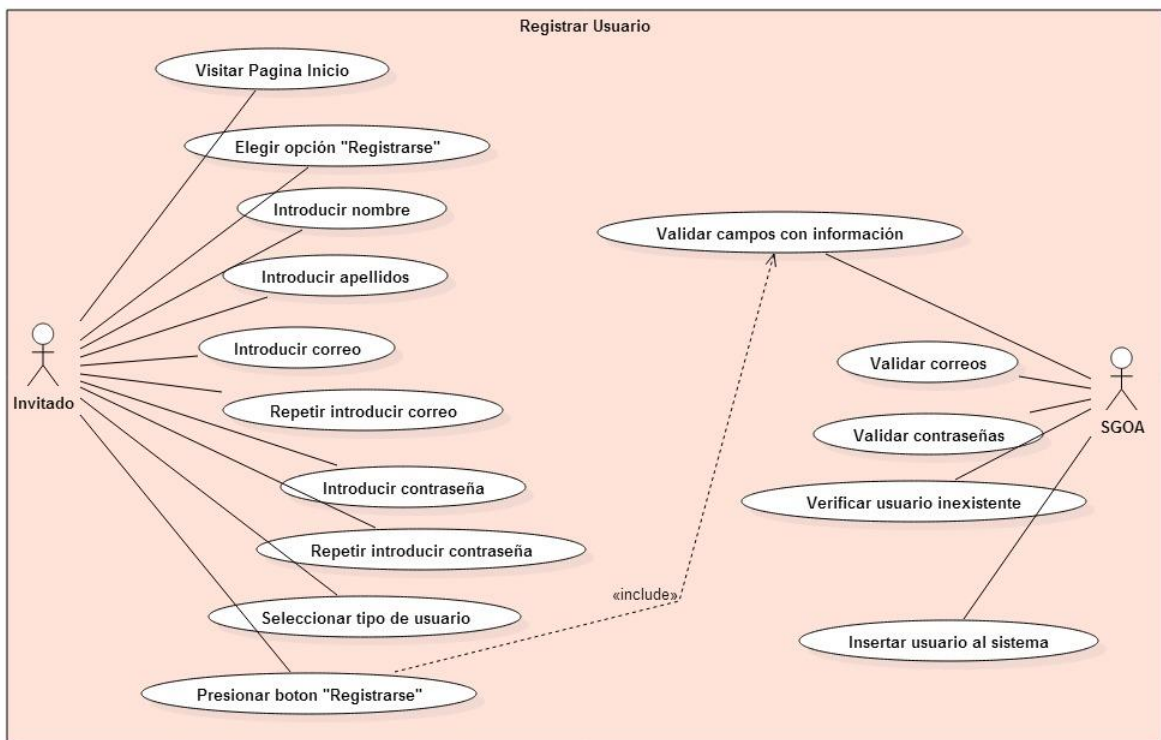


Figura A.1 Diagrama de casos de uso "Registrar Usuario". Fuente: Propia

- Iniciar Sesión

Nombre:	Iniciar Sesión	CU2
Propósito:	Permitir el acceso a un usuario registrado	
Descripción:	Permite el acceso a un usuario registrado en el sistema con los diferentes tipos de usuarios existentes, ya sea profesor o alumno y enviarlos a su "muro" dependiendo el usuario logueado	

Actores:	Usuario registrado (Puede ser profesor o alumno)	
Referencias:	CU1	
Precondición:	Haberse registrado en la plataforma (ver el diagrama y tabla correspondientes al caso de usos “Registrar usuario ”)	
Flujo normal (Básico)		
1.- Entrar al portal del sistema 2.- Presionar la opción “Iniciar Sesión” 4.-Ingresar correo/usuario 5.- Ingresar contraseña 6.- Presiona botón “Entrar”		3.- Mostrar el registro para iniciar sesión 7.-Validar usuario y contraseña 8.-Mostrar “Muro” del usuario que ha ingresado
Flujo alternativo:		
7.1- Mostrar mensaje de error si el usuario no está registrado o la contraseña es invalida		
Post-condición:	Ingresa al “muro” del usuario registrado dentro de la plataforma	

Tabla A.2 Descripción del caso de uso *Iniciar Sesión*. Fuente: Propia

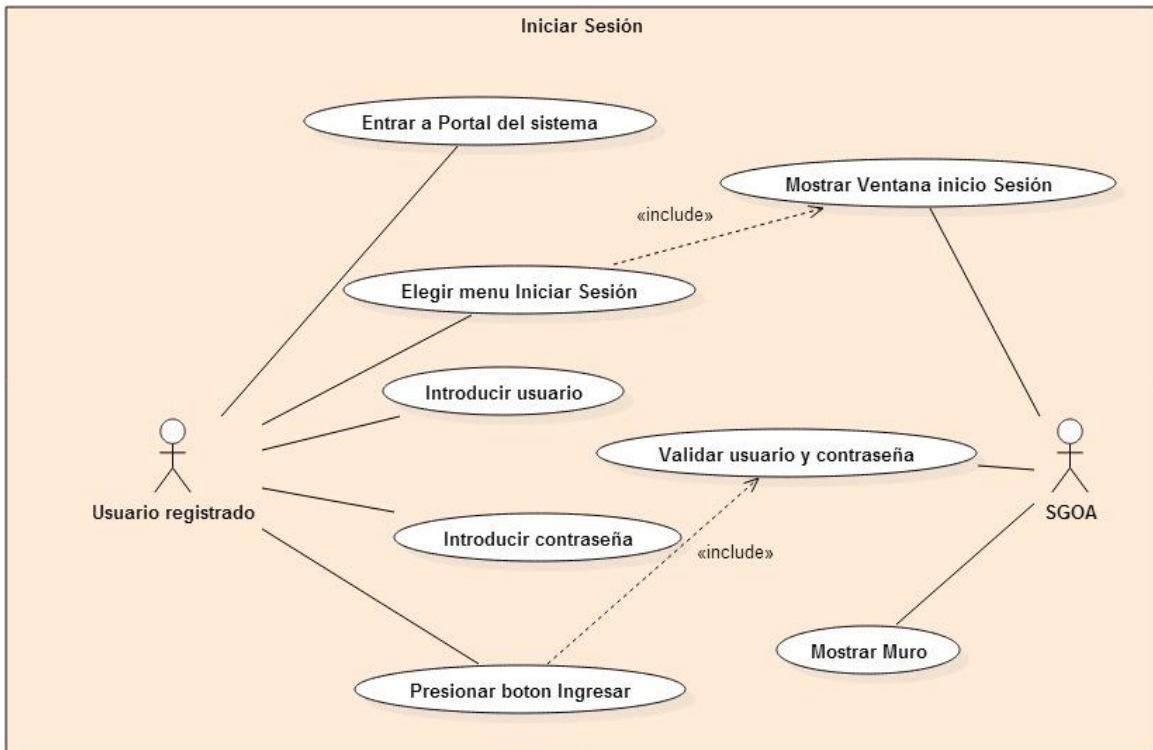


Figura A.2 Diagrama de casos de uso “Iniciar Sesión”. Fuente: Propia

- Agregar Área

Nombre:	Agregar Área	CU4
Propósito:	Agregar un área inexistente en el sistema	
Descripción:	El usuario agregara un área de no encontrarse en la selección siempre y cuando oprima el botón “agregar área” y llene los requisitos de dicho formulario	
Actores:	Profesor, SGOA	
Referencias:	CU3	
Precondición:	Haber comenzado la creación de un OA dentro del formulario “Crear OA”	
Flujo normal (Básico)		
1.-El profesor presiona el botón “Agregar Área”	2.-El sistema muestra el formulario con los campos básicos para agregar un área nueva 5.- El sistema agrega el área aportada si es que no existe una con el mismo nombre	
3.-El usuario llena el formulario con los datos requeridos		
4.- El usuario presiona el botón Agregar		
Flujo alternativo:		
Ninguno		
Post-condición:	Se agrega un área nueva al sistema	

Tabla A.3 Descripción del caso de uso *Agregar Área*. Fuente: Propia

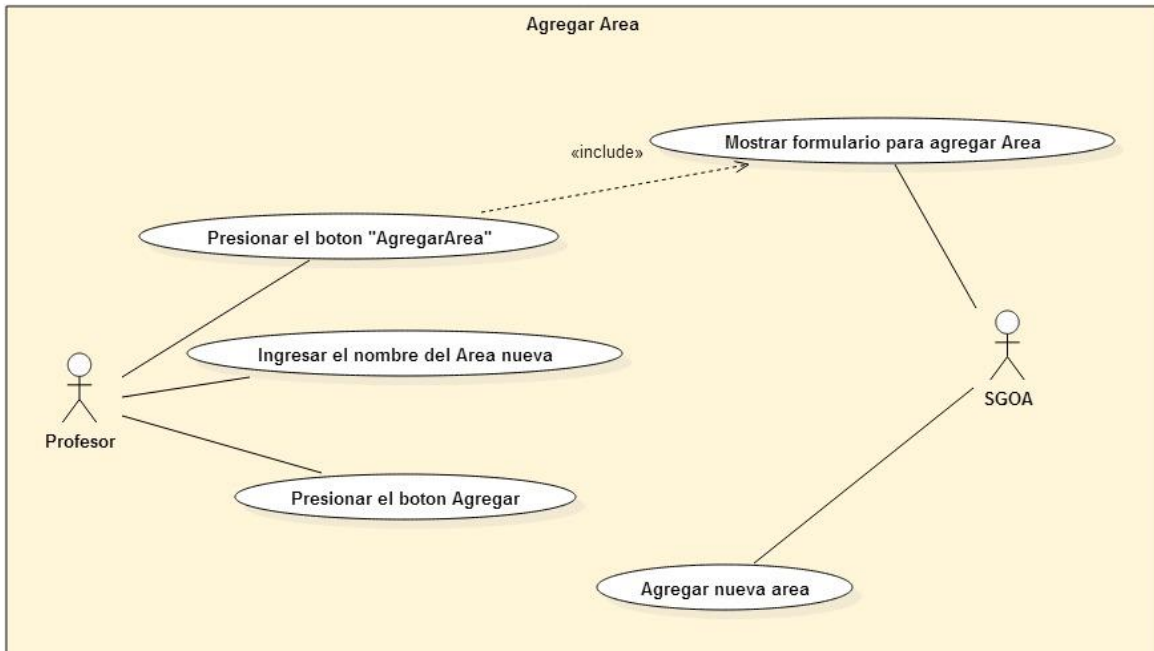


Figura A.3 Diagrama de casos de uso “Agregar área”. Fuente: Propia

- Agregar Categoría

Nombre:	Agregar Categoría	CU5
Propósito:	Agregar una categoría inexistente en el sistema	
Descripción:	Agrega una categoría que no exista en un área dentro del sistema para poder así catalogar a los diferentes objetos de aprendizaje creados	
Actores:	Profesor, SGOA	
Referencias:	CU4	
Precondición:	Seleccionar una categoría dentro del formulario de creación de un OA	
Flujo normal (Básico)		
1.-El profesor debe de seleccionar un área donde se agregara la categoría	3.-El sistema despliega un formulario para la creación de una nueva categoría 5.-El sistema agrega una nueva categoría dentro del área que se seleccionó en el sistema	
2.- El profesor presiona el botón “Agregar categoría”		
4.-El profesor presiona el botón agregar		
Flujo alternativo:		
Ninguno		

Post-condición:	El sistema agrega una nueva categoría para una mejor indexación del OA
------------------------	--

Tabla A.4 Descripción del caso de uso *Agregar Categoría*. Fuente: Propia

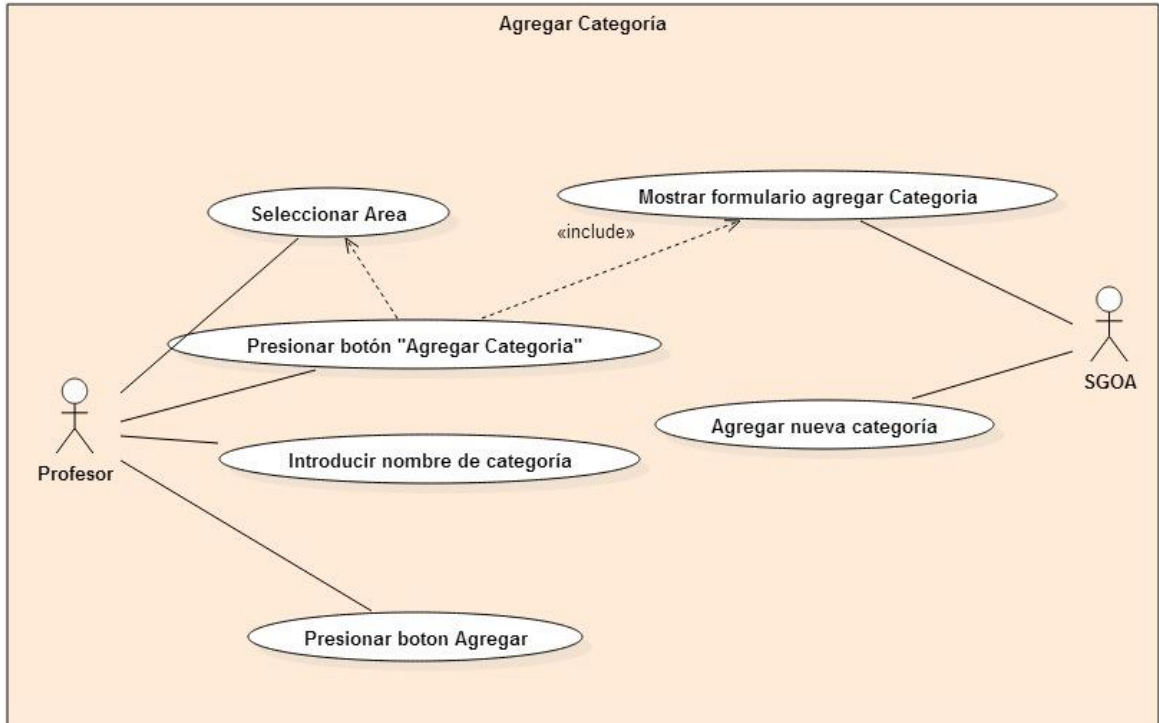


Figura A.4 Diagrama de casos de uso "Agregar Categoría". Fuente: Propia

- Eliminar OA

Nombre:	Eliminar OA	CU7
Propósito:	Eliminar el registro de un objeto de aprendizaje dentro del sistema	
Descripción:	Elimina la información básica de un objeto de aprendizaje dentro del sistema para que no se pueda hacer uso de el	
Actores:	Profesor, SGOA	
Referencias:	CU6	
Precondición:	Haber creado un objeto de aprendizaje	
Flujo normal (Básico)		
1.- Iniciar sesión 2.-Seleccionar el menú "Actividades" 3.-Seleccionar el menú "Mis OA's" 4.- Desplazarse hasta la sección "Mis OA's "		

5.- Localizar el OA a eliminar 6.- Presionar enlace “Eliminar”	7.-El sistema elimina el registro del OA seleccionado 8.- Se actualiza la sección “Mis OA’s”
Flujo alternativo:	
Post-condición:	Elimina el registro de un OA en el sistema

Tabla A.5 Descripción del caso de uso *Eliminar OA*. Fuente: Propia

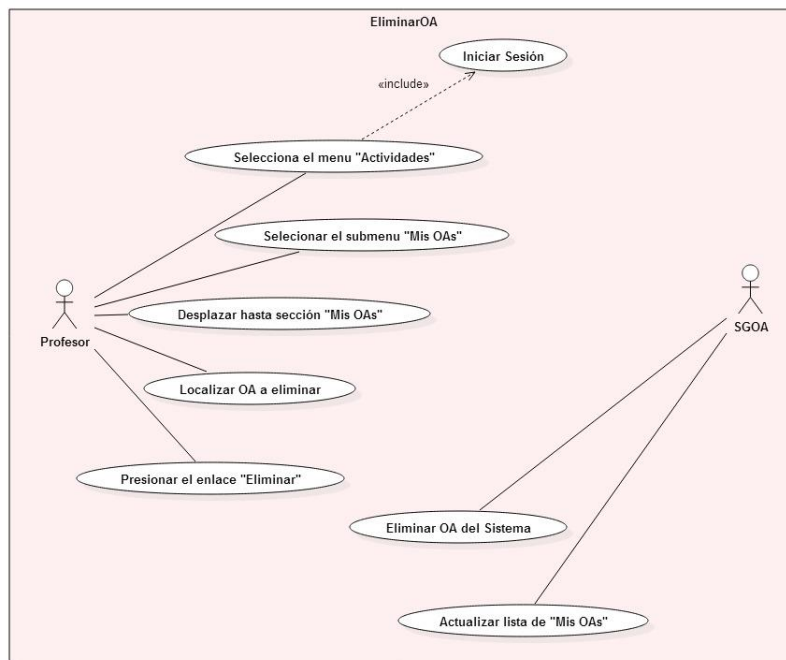


Figura A.5 Diagrama de casos de uso “Eliminar OA”. Fuente: Propia

- Realizar Actividad

Nombre:	Realizar Actividad	CU8
Propósito:	Mostrar el proceso de realización y evaluación al realizar una actividad	
Descripción:	Un usuario realiza la actividad siguiendo una serie de pasos para que al finalizarla, éste obtenga los puntos que se han otorgado a dicha actividad y se sumen a su puntaje	
Actores:	Alumno, SGOA	
Referencias:	CU6	
Precondición:	Crear una actividad	
Flujo normal (Básico)		
1.-Iniciar sesión en modo de usuario Alumno 2.- Seleccionar un OA a realizar	3.-Desplegar las actividades dentro del OA	

4.- Seleccionar la actividad a realizar	5.- Mostrar los detalles de la actividad
6.- Presionar botón finalizar actividad	7.- Sumar los puntos de la actividad a el puntaje del usuario que la realizo
	8.- Regresar a pantalla de actividades
Flujo alternativo:	
Post-condición:	Sumar puntos que otorga la actividad al puntaje del usuario

Tabla A.6 Descripción del caso de uso *Realizar Actividad*. Fuente: Propia

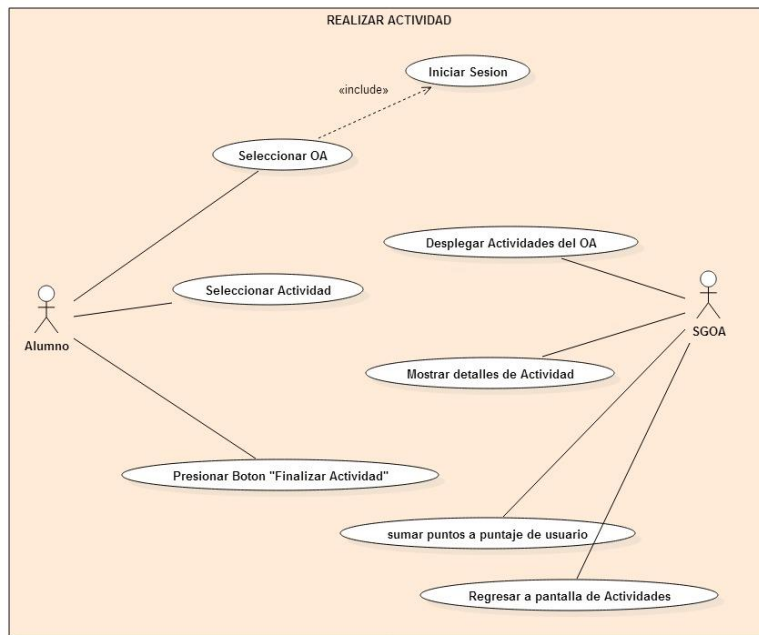


Figura A.6 Diagrama de casos de uso “Realizar Actividad”. Fuente: Propia