

ANÁLISIS, DESARROLLO, IMPLEMENTACIÓN Y MANTENIMIENTO DE
MÓDULOS RELACIONADOS CON LOS NUEVOS SERVICIOS DEL
ADMINISTRADOR WEB Y RENOVACIÓN DE SERVICIOS EN EL AULA
VIRTUAL DE LA PLATAFORMA COMA

OMAR SÁNCHEZ ARCINIEGAS

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
AÑO 2021

ANÁLISIS, DESARROLLO, IMPLEMENTACIÓN Y MANTENIMIENTO DE
MÓDULOS RELACIONADOS CON LOS NUEVOS SERVICIOS DEL
ADMINISTRADOR WEB Y RENOVACIÓN DE SERVICIOS EN EL AULA
VIRTUAL DE LA PLATAFORMA COMA

OMAR SÁNCHEZ ARCINIEGAS

Trabajo de grado para optar título de Ingeniero de Sistemas

Director

LUIS IGNACIO GONZÁLEZ RAMÍREZ

Magíster en informática

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
AÑO 2021

AGRADECIMIENTOS

A mi madre Ana Melba Arciniegas, por el apoyo y la confianza que siempre me ha tenido, por sacrificar todo para que todos sus hijos podamos cumplir nuestras metas. A mi familia en general por la confianza que me tienen, por sentirse orgullosos a medida que he cumplido mis sueños.

Al profesor Luis Ignacio por la confianza y el apoyo que me ofreció en todo el proceso de desarrollo de este proyecto de grado.

A mis amigos David Vanegas y Karenlyn Díaz por siempre estar ahí disponible ofreciéndome su ayuda en todo el transcurso de mi proceso formativo en la universidad. Y a Yahira Porras, que llegó en un momento crucial para mi vida y para mi desarrollo profesional, dándome la confianza y el apoyo para no desistir de mis sueños y para inspirarme a siempre ir por más, a buscar nuevas metas que perseguir.

CONTENIDO

	pág.
INTRODUCCIÓN	14
1. PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA	15
1.1 DEFINICIÓN DE LA SITUACIÓN ACTUAL	15
1.2 JUSTIFICACIÓN	16
1.3 OBJETIVOS	18
1.3.1 Objetivo general..	18
1.3.2 Objetivos específicos.....	18
1.4 IMPACTO Y VIABILIDAD	19
1.4.1 Impacto.....	19
1.4.2 Viabilidad.....	20
2. MARCO TEÓRICO	21
2.1 ARQUITECTURA CLIENTE/SERVIDOR.....	21
2.1.1 Características de la arquitectura Cliente/Servidor.....	22
2.1.2 Clasificación de las arquitecturas Cliente/servidor.	23
2.1.2.1 Arquitectura Cliente/Servidor de dos capas.....	23
2.1.2.2 Arquitectura Cliente/Servidor de tres capas.	24
2.1.3 Arquitectura Cliente/Servidor aplicada.	24
2.1.4 Ventajas del esquema Cliente/Servidor.....	25
2.1.5 Desventajas del esquema Cliente/Servidor.	25
2.2 TECNOLOGÍAS DE DESARROLLO DE PÁGINAS WEB DINÁMICAS	26
2.2.1 Código del Lado del Cliente (Client Side Scripts).	26
2.2.2 Código del Lado del Servidor (Server Side Scripts).	27
2.2.3 Tecnología aplicada.....	28
2.2.3.1 Modelo de acceso a JSP.....	28
2.3 BASES DE DATOS	29

2.3.1	Modelos de Bases de Datos.....	30
2.3.1.1	Base de Datos Jerárquica..	30
2.3.1.2	Bases de Datos de Red.....	30
2.3.1.3	Bases de Datos Relacional.....	31
2.3.2	Gestores de Bases de Datos.....	31
2.3.3	MySQL.....	32
2.3.4	Ventajas de MySQL.....	32
2.4	NETBEANS	33
2.5	SISTEMAS DE CONTROL DE VERSIONES	34
2.5.1	Subversión.....	35
2.6	PROGRAMACIÓN UTILIZADA	35
2.6.1	Clases.....	36
2.6.2	Objetos.	36
2.6.3	Atributos.	36
2.6.4	Métodos.....	36
2.6.5	Herencia.	36
2.6.6	Beneficios de la Programación Orientada a Objetos.	36
2.6.7	Java y JDK (Java Development Kit).	37
2.7	SERVIDORES WEB.....	38
2.7.1	Servidor Jakarta Tomcat.....	38
3.	MARCO METODOLÓGICO.....	39
3.1	PROTOTIPO EVOLUTIVO	39
3.2	LENUAJE DE MODELADO UNIFICADO	41
3.2.1	Diagramas de UML.....	42
3.2.2	Diagramas de casos de uso.	42
3.2.3	Diagramas de secuencias.	44
3.3	ESTÁNDARES DE PROGRAMACIÓN	45
3.3.1	Modelo de datos.	45
3.3.2	Nombres de las tablas.....	46

3.3.3	Clases.....	47
3.3.4	Paginas JSP.....	47
3.3.5	Organización de directorios.....	47
4.	DESARROLLO DE LA HERRAMIENTA, ADMINISTRACIÓN Y MANTENIMIENTO.....	48
4.1	PROTOTIPO ESPERADO.....	48
4.1.1	Crear nuevo servicio para el administrador web.....	48
4.1.2	Realizar actualización al servicio de limpieza del Aula virtual.	49
4.1.3	Actualizar el filtro que muestra los archivos compartidos en el Aula	50
4.1.4	Crear la opción de imprimir el formato PEP de cada asignatura	50
4.2	DIAGRAMAS DE CASOS DE USO	51
4.2.1	Gestionar los menús de todos los portales de las escuelas.	51
4.2.2	Actualizar el servicio de limpieza del Aula Virtual.	51
4.2.3	Actualizar el filtro que muestra los archivos compartidos en el aula.....	53
4.2.4	Descargar en formato PEP la información de cada asignatura del plan..	53
4.3	DOCUMENTACIÓN DE CASOS DE USO DEL SISTEMA.....	54
4.3.1	Gestionar los menús de todos los portales de las escuelas.	54
1.1.1	Actualizar el servicio de limpieza del Aula Virtual.	54
4.3.2	Actualizar el filtro que muestra los archivos compartidos en el aula.....	55
4.3.3	Descargar en formato PEP la información de cada asignatura del plan..	56
4.4	DISEÑO Y ANÁLISIS	56
4.4.1	Gestionar los menús de todos los portales de las escuelas.	56
4.4.2	Actualizar el servicio de limpieza del Aula Virtual.	57
4.4.3	Actualizar el filtro que muestra los archivos compartidos en el aula.....	64
4.4.4	Descargar en formato PEP la información de cada asignatura del plan..	64
4.5	DESCRIPCIÓN DE LAS ENTIDADES	65
4.5.1	Gestionar los menús de todos los portales de las escuelas	65
4.5.2	Actualizar el servicio de limpieza del Aula Virtual.	65
4.5.3	Actualizar el filtro que muestra los archivos compartidos en el aula.....	69

4.5.4	Descargar en formato PEP la información de cada asignatura del plan..	69
4.6	MODELO DE PROCESOS DEL SISTEMA	69
4.7	IMPLEMENTACIÓN, IMPLANTACIÓN Y PRUEBAS GENERALES	70
4.8	MANTENIMIENTO Y ADMINISTRACIÓN	71
4.8.1	Actividades de mantenimiento.....	71
4.8.2	Actividades de soporte a usuarios.....	71
4.8.3	Actividades de administración.	72
5.	PRUEBAS DEL SISTEMA.....	73
5.1	PRUEBAS DE VERIFICACIÓN	73
5.1.1	Pruebas por componentes.....	73
5.1.1.1	Gestionar los menús de todos los portales de las escuelas.	73
5.1.1.2	Actualizar el servicio de limpieza del Aula Virtual.	74
5.1.1.3	Actualizar el filtro que muestra los archivos compartidos en el aula.....	75
5.1.1.4	Descargar en formato PEP la información de cada asignatura.	75
6.	CONCLUSIONES	77
7	RECOMENDACIONES.....	79
	BIBLIOGRAFÍA.....	80
	ANEXOS.....	81

LISTA DE FIGURAS

	pág.
Figura 1. Modelo cliente/Servidor	21
Figura 2. Modelo de acceso a JSP	29
Figura 3. Prototipo evolutivo	39
Figura 4. Diagramas de casos de uso	43
Figura 5. Diagramas de secuencias.....	45
Figura 6. Diagrama de casos de uso: Gestionar menús de todas las escuelas.....	51
Figura 7. Diagrama de casos de uso: Realizar limpieza del aula virtual	52
Figura 8. Limpiar Aula.....	52
Figura 9. Diagrama de casos de uso: Archivos compartidos	53
Figura 10. Diagrama de casos de uso: Gestionar información de asignaturas	53
Figura 11. Diagrama E/R: Gestionar menús desde una hacia todas las escuelas	57
Figura 12. Diagrama E/R: Realizar limpieza del aula.....	58
Figura 13. Diagrama E/R: Limpiar archivos	58
Figura 14. Diagrama E/R: Limpiar Chat	59
Figura 15. Diagrama E/R: Limpiar mensajes en el aula.....	59
Figura 16. Diagrama E/R: Limpiar agenda.....	60
Figura 17. Diagrama E/R: Limpiar Foros	60
Figura 18. Diagrama E/R: Limpiar estudiantes incluidos	61
Figura 19. Diagrama E/R: Limpiar actividades.....	62
Figura 20. Diagrama E/R: Limpiar subgrupos	63
Figura 21. Diagrama E/R: Reiniciar aula.....	63
Figura 22. Diagrama E/R: Archivos compartidos	64
Figura 23. Diagrama E/R: Gestionar información de asignaturas	64
Figura 24. Diagrama de secuencia: Crear acción.....	81
Figura 25. Diagrama de secuencia: Editar acción.....	82

Figura 26. Diagrama de secuencia: Eliminar acción.....	83
Figura 27. Diagrama de secuencia: Ejecutar acción.....	84
Figura 28. Diagrama de secuencia: Limpiar aula.....	85
Figura 29. Diagrama de secuencia: Archivos compartidos.....	87
Figura 30. Diagrama de secuencia: Editar información.....	88

LISTA DE TABLAS

	pág.
Tabla 1. Casos de uso: Gestionar menús desde una hacia todas las escuelas	54
Tabla 2. Casos de uso: Realizar limpieza del aula	54
Tabla 3. Casos de uso: Archivos compartidos	56
Tabla 4. Casos de uso: Gestionar información de asignaturas.....	56
Tabla 5. Entidades: Gestionar menús desde una hacia todas las escuelas	65
Tabla 6. Entidades: Limpieza del aula	65
Tabla 7. Entidades: Archivos compartidos.....	69
Tabla 8. Entidades: Información de asignaturas	69
Tabla 9. Pruebas realizadas: Gestionar menús de todas las escuelas.....	73
Tabla 10. Pruebas realizadas: limpieza del aula.....	74
Tabla 11. Pruebas realizadas: Archivos compartidos	75
Tabla 12. Pruebas realizadas: Información de asignaturas.	75

LISTA DE ANEXOS

	pág.
Anexo A. Modelo de procesos de sistemas	81

RESUMEN

TÍTULO: Análisis, desarrollo, implementación y mantenimiento de módulos relacionados con los nuevos servicios del administrador web y renovación de servicios en el aula virtual de la plataforma coma.¹

AUTOR: Omar Sánchez Arciniegas ²

PALABRRAS CLAVE: Aula virtual, COMA, Servidor, gestión de servicios.

DESCRIPCIÓN

Actualmente las plataformas que conforman el proyecto COMA, Comunidad Académica, son las usadas por las escuelas como el principal canal de integración entre toda la comunidad de la Universidad Industrial de Santander. Permitiendo también a las escuelas la administración y control de todas las actividades académicas que se realizan dentro de estas, así como las actividades administrativas que se realizan las facultades y escuelas.

El grupo Calumet, actualmente brinda soporte al proyecto COMA razón por la que mantener a todos sus servicios actualizados y en óptimas condiciones es su principal objetivo del grupo, siempre buscando satisfacer las necesidades de los usuarios del proyecto como los son estudiantes, profesores, administrativos y entre otros. Por lo anterior, se hace necesario realizar cambios, modificaciones y actualizaciones como: Actualizar servicio de limpieza y la seguridad de los archivos compartidos dentro del módulo del Aula virtual; Crear un nuevo servicio para el administrador web que facilite la labor de este en todas las plataformas y actualizar la forma de como la comunidad puede obtener el contenido programático de cualquier asignatura subida a los planes de estudio.

¹ Trabajo de grado

² Facultad de Ingenierías Físico-mecánicas. Escuela de Ingeniería de Sistemas e Informática
Director: Msc. Luis Ignacio González Ramírez

ABSTRACT

TITLE: Analysis, development, implementation and maintenance of modules related to the new web administrator services and renewal of services in the virtual classroom of the coma platform¹

AUTHOR: Omar Sánchez Arciniegas ²

KEY WORDS: Virtual classroom, COMA, Server, service management.

DESCRIPTION

Currently the platforms that make up the COMA, Academic Community project, are those used by schools as the main channel of integration among the entire community of the Industrial University of Santander. Also allowing the schools the administration and control of all the academic activities carried out within them, as well as the administrative activities carried out by the faculties and schools. The Calumet group currently supports the COMA project, which is why keeping all its services updated and in optimal conditions is the main objective of the group always looking for satisfy the needs of this project users like students, teachers, administrators and others. Therefore, it is necessary to make changes, modifications and updates such as: Update cleaning service and the security of shared files within the Virtual Classroom module; Create a new service for the web administrator that facilitates the work of this on all platforms and update the way in which the community can obtain the programmatic content of any uploaded subject to the study plans.

¹ Degree Thesis

² Faculty of Physical-mechanical Engineering. School of Systems and Informatics Engineering.
Director: Msc. Luis Ignacio González Ramírez

INTRODUCCIÓN

Los Portales Web Comunidad Académica (COMA) son los principales canales de comunicación e integración entre los diferentes miembros de la comunidad que conforman las escuelas y miembros en general de la comunidad UIS. Actualmente los servicios se han extendido a más escuelas de las distintas facultades gracias a la aceptación de los usuarios por lo que se adopta el objetivo de mantener y mejorar los servicios que se ofrecen, además de crear nuevos servicios que satisfagan las necesidades crecientes de los usuarios de la comunidad académica.

El grupo CALUMET, grupo de desarrollo de software de la escuela de Ingeniería de Sistemas, se encarga de desarrollar los nuevos servicios y darle mantenimiento a los existentes de manera que su actualización responda al continuo cambio. Con el fin de llevar a su cumplimiento el objetivo principal del grupo y proporcionar portales web con contenido dinámico, se cuenta con herramientas software de libre distribución como Netbeans, Github, Sqlyog, JavaScript, jQuery, entre otras.

En este documento se presenta un soporte teórico, metodológico y técnico del desarrollo realizado en la sección de: administrador web de los portales para facilitar la gestión de los menús o servicios de las escuelas desde un módulo centralizado, planes de estudio y en su forma de mostrar la información de las asignaturas con el fin facilitar a la comunidad el fácil acceso y obtención de esta información cumpliendo los estándares del PEP y en el aula virtual para brindarle a los usuarios de esta, la seguridad al compartir información y la certeza de que el aula está en óptimo estado al iniciar un nuevo periodo académico.

1. PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA

1.1 DEFINICIÓN DE LA SITUACIÓN ACTUAL

En la actualidad las Escuelas y las Facultades de la UIS, cuentan con un sistema de información orientado a la Web que se encarga de la administración y control de las diferentes actividades tanto académicas como administrativas que se realizan dentro de la escuela y la Facultad, así como del control de usuarios y servicios que se les proporcionan.

Los servicios de los portales Web de las escuelas deben mejorar constantemente y adaptarse a los cambios que se presenten en su entorno, a su vez deben dar solución a los problemas y necesidades que surjan por parte de los usuarios del sistema para incrementar su tiempo de vida útil y no llegar a convertirse en un software obsoleto, razón por la cual las labores de mantenimiento y actualización se hacen indispensables.

Actualmente dentro del módulo del aula virtual que se tiene en los portales de las escuelas se encuentra desactualizado el servicio de limpieza del aula, el cual se hace necesario su mantenimiento para que el docente o director del aula pueda iniciar cada semestre o un nuevo periodo académico en las mejores condiciones del portal web. Al sistema de filtrado de los archivos que se comparten dentro de este mismo módulo se hace necesario hacerle mantenimiento para que solo muestre los archivos a los usuarios con los que fueron compartidos, logrando así que los estudiantes y directores del aula se sientan seguros en su entorno académico. Actualmente no existe una forma directa en donde el estudiante pueda obtener la información completa aprobada de cada asignatura en el Proyecto Educativo del Programa (PEP) por lo cual también se hace necesario actualizar la sección en donde esta información será ingresada al sistema. Actualmente, los

administradores de los portales web no cuenta con un servicio que le permita gestionar los menús de todos los portales de las otras escuelas, la razón por la que se hace necesario crear este nuevo servicio que permita gestionar y distribuir menús o servicios desde una escuela a las demás.

1.2 JUSTIFICACIÓN

El Portal Web Comunidad Académica (COMA), que desarrolla y mantiene el grupo Calumet, es una plataforma que provee lugares, adicionales a los ya existentes, de encuentro y comunicación a la comunidad de los programas de las diferentes escuelas. Esta plataforma contribuye a la realización de la misión en sus tres ejes: extensión, investigación y docencia. En ella, la Comunidad podrá mantener actualizada su hoja de vida, publicar noticias, proponer eventos, consultar y crear agendas, compartir documentos, enviar correos, y otros servicios que podrán hacer más fácil la realización de la misión institucional. Los servicios de los portales web deben actualizarse constantemente, mejorar y adaptarse a los cambios que se presenten en su entorno, satisfaciendo las necesidades de los usuarios del sistema. Esto permite aumentar el tiempo de vida útil del software evitando que pase a ser obsoleto con la creación de nuevos servicios, y con el mantenimiento realizado a los existentes.

Por lo anterior, se hace necesario llevar a cabo una reingeniería en los servicios de compartir archivos y limpieza del Aula Virtual, Crear un nuevo servicio para Administradores de los portales y para obtener el formato PEP en PDF:

- De este modo, atendiendo las necesidades de los administradores Web de las páginas de las escuelas se hace necesario crear un nuevo servicio que le permita al administrador crear, modificar, habilitar y distribuir menús en todos los servidores de manera sencilla. Actualmente, se hace necesario ingresar a

cada servidor para realizar estas tareas, labor que se torna bastante tediosa ya que se administran más de 30 servidores.

Con la implementación de este nuevo servicio, el usuario administrador que tenga la necesidad de crear un menú tendrá la posibilidad de elegir la escuela, de conceder los permisos, de establecer estado, el tipo de servicio y demás características que poseen los menús. De igual forma; podrá eliminar, desactivar y modificar los menús de todas las plataformas Web que gestiona y administra el Grupo Calumet. Todo esto el usuario administrador lo podrá realizar desde una sola escuela.

- El Aula Virtual se concibió como aquel lugar de alta interacción entre aquellos que hacen uso de ella, profesores y estudiantes. El Aula permite la comunicación de muchas formas y de manera sencilla: foros, correos, salas de conversación, mensajes, archivos y otros hacen parte de ella. Razón por la que se hace necesario actualizar:
 - El servicio de limpiar el aula, esta funcionalidad de permitir eliminar archivos, participaciones, calificaciones, foros y demás de tal forma que el proceso sea sencillo e incluya todos los recursos, evitando la eliminación de aquellos que pertenecen o fueron subidos por el docente. Al implementar la actualización el docente o director tendrá la certeza que al activar el servicio de Limpieza del Aula esta quedará lista y preparada para el inicio de un nuevo periodo académico.
 - Mejorar el servicio que permite compartir documentos y archivos con determinados usuarios del aula, o con equipos de trabajo y/o con grupos completos. A esta funcionalidad que complementa la labor de enseñanza se le hace necesario actualizar el filtro que es usado a la hora de mostrar los documentos que se comparten dentro de este

espacio académico. Con esta implementación los usuarios tendrán la seguridad y tranquilidad que al compartir un archivo solo se mostrará a quienes él decida.

Debido a la necesidad que presentan los estudiantes, egresados y administrativos al momento de acceder y obtener la información completa de cada asignatura del plan de estudio; se hace necesario actualizar la vista de como se muestra dicha información, cumpliendo de esta manera con el formato PEP y es indispensable crear la opción de descargar dicha información en formato PDF.

1.3 OBJETIVOS

1.3.1 Objetivo general. Crear nuevos servicios en los módulos del administrador web de las escuelas, implementación de mejoras significativas en distintos módulos de COMA, así como nuevas funcionalidades acorde a las necesidades de las escuelas de la Universidad Industrial de Santander asociadas a la plataforma COMA (comunidad académica).

1.3.2 Objetivos específicos.

1.3.2.1 Crear un nuevo servicio para los administradores Web de las plataformas de las escuelas que permita crear y administrar nuevos menús en estos portales. Debe crearse un nuevo servicio, que permita crear nuevos menús y que estos se puedan modificar, habilitar y distribuir desde el módulo del administrador web.

1.3.2.2 En el Aula Virtual de las escuelas, Se actualizará el servicio de limpieza dentro del Aula Virtual. Se realizará el soporte necesario para que la actualización de este servicio sea la más eficiente posible.

1.3.2.3 Actualizar el filtro que muestra a los usuarios del Aula Virtual los archivos compartidos dentro de esta. Se revisará y optimizará la función de filtro que se encarga de elegir los archivos que se le deben mostrar a los usuarios que corresponda.

1.3.2.4 Crear el servicio de imprimir en PDF la información de las asignaturas que están en el plan de estudio de las escuelas con el formato PEP. Para el correcto funcionamiento de este servicio no solo se debe actualizar cómo se mostrará la información, sino también hacer modificaciones en cómo se ingresa la información de las asignaturas de los planes de estudio al sistema. Esto, con el fin de cumplir el formato PEP.

1.4 IMPACTO Y VIABILIDAD

1.4.1 Impacto. Los portales web de las escuelas, han sido una herramienta útil para el manejo de la información, por lo tanto, es necesario realizar labores de administración y mantenimiento, para ofrecer al usuario un sitio más confiable. Se pretende que los procesos que se realizan en las escuelas cada día sean más ágiles, dinámicos, seguros y eficientes, permitiendo una mejor organización de la información, razón por la cual se crean nuevos servicios que se ajusten a las necesidades que puedan surgir.

1.4.2 Viabilidad. La administración del sitio, soporte a usuarios, mantenimiento y desarrollo de nuevos servicios es viable pues se usará software de libre distribución, recurso humano preparado para tal fin, servidores que marchan de manera legal en las escuelas, equipos disponibles y todo el soporte tecnológico necesario para el correcto funcionamiento del mismo.

Además, se cuenta con la supervisión por parte del director del proyecto y la colaboración del equipo de trabajo CALUMET, agentes de gran apoyo en la realización de este proyecto.

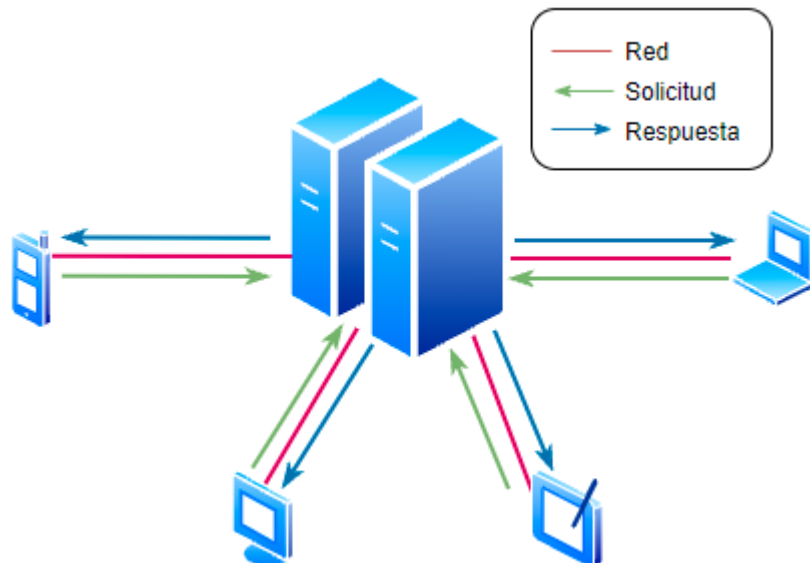
2. MARCO TEÓRICO

2.1 ARQUITECTURA CLIENTE/SERVIDOR

En la arquitectura cliente/servidor cada uno de los clientes produce un mensaje solicitando un determinado servicio a un servidor (hace una petición) estos envían uno o varios mensajes como respuesta (responden peticiones o provee un servicio). La mayoría del trabajo pesado (procesos de base de datos, procesar la lógica de la aplicación entre otros) está a cargo de los servidores, el cliente comúnmente se encarga de las funciones de administración de la interfaz de usuario, interacción con el usuario, recibir resultados del servidor, generar requerimientos de base de datos, entre otros.

Esta idea se puede aplicar tanto a programas que se están ejecutando en una sola máquina, pero es más ventajosa en un sistema operativo multiusuarios distribuidos a través de una red de computadores.

Figura 1. Modelo cliente/Servidor



2.1.1 Características de la arquitectura Cliente/Servidor. Las características básicas de una arquitectura Cliente / Servidor son:

- Es quien inicia solicitudes o peticiones, tienen por tanto un papel activo en la comunicación.
- El proceso del cliente da la interface entre usuarios y el resto del sistema, maneja recursos compartidos tales como bases de datos, impresoras, módems, etc.
- El cliente y el servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades independientes.
- Las tareas del cliente y el servidor tienen diferentes requerimientos como: velocidad del procesador, memoria o capacidad del disco, por tanto, la plataforma de hardware y el sistema operativo del cliente y del servidor no son siempre la misma y eso se conoce como ambiente heterogéneo.
- La escalabilidad horizontal permite agregar más estaciones de trabajo activas sin afectar el rendimiento y la escalabilidad vertical permite mejorar las características del servidor o agregar múltiples servidores. Se puede realizar independientemente cambios en las plataformas de los clientes o de los servidores, ya sea actualización o reemplazo tecnológico, de manera transparente para el usuario final

2.1.2 Clasificación de las arquitecturas Cliente/servidor. Los sistemas Cliente/Servidor se clasifican de acuerdo al nivel de abstracción del servicio que se ofrece. Se distinguen tres componentes básicos de software para la clasificación:

- **Presentación:** Presentación de resultados al usuario de forma comprensible.
- **Lógica de aplicación:** Esta capa es la responsable del procesamiento de la información que tiene lugar en la aplicación.
- **Base de datos:** Está compuesta por los archivos que contienen los datos persistentes de la aplicación.

2.1.2.1 Arquitectura Cliente/Servidor de dos capas. Consiste en una capa de presentación y lógica de la aplicación; y otra de la base de datos, cuando el cliente solicita recursos entonces el servidor responde directamente a la solicitud con sus propios recursos. Normalmente esta arquitectura es utilizada en las siguientes situaciones:

- Cuando se requiere poco procesamiento de datos en la organización.
- Cuando se tiene una base de datos centralizada en un solo servidor.
- Cuando la base de datos es relativamente estática.
- Cuando se requiere un mantenimiento mínimo.

2.1.2.2 Arquitectura Cliente/Servidor de tres capas. Define como organizar el modelo de diseño en capas, que pueden estar físicamente distribuidas, es decir que los componentes de una capa solo pueden hacer referencia a componentes en capas inferiores. Este patrón es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no conscientes de ningún detalle o interfaz de las superiores, está compuesta de:

- Un equipo cliente con una interfaz de usuario (habitualmente se utiliza un navegador web), que solicita los recursos.
- El servidor de aplicaciones (o software intermedio), cuya tarea es prestar los recursos solicitados, pero que requiere de otro servidor para hacerlo.
- El servidor de datos que almacena y proporciona al servidor de aplicaciones los datos que requiere.

2.1.3 Arquitectura Cliente/Servidor aplicada. En el desarrollo de este proyecto se recurre a arquitectura de tres capas, debido a las ventajas ofrecidas como: Escalabilidad, fácil mantenimiento y el manejo de un mayor número de usuarios que la ofrecida por la arquitectura C/S de dos capas. La arquitectura es aplicada de la siguiente forma:

- **Capa de Cliente:** Interfaz con el usuario, se usa un navegador web.
- **Capa Intermedia:** Para los servicios del negocio se utiliza un computador configurado como servidor web, el cual almacena el portal web conformado por páginas JSP y JavaBeans. Allí se realizan los procesos complejos, y se

solicitan los servicios del servidor de datos cuando es necesario acceder a la información almacenada en la base de datos.

- **Capa de Servidor:** Se utiliza el motor de bases de datos MySQL, el cual se encuentra en el mismo servidor web.

2.1.4 Ventajas del esquema Cliente/Servidor.

- La existencia de plataformas de software y hardware de varios fabricantes y cada vez más a económicas contribuye a la reducción de costos y favorece la flexibilidad en la implantación y actualización de soluciones.
- Este esquema facilita la integración entre sistemas heterogéneos y comparte información permitiendo que las máquinas existentes puedan ser utilizadas con interfaces amigables al usuario, de esta forma integrar los computadores con sistemas medianos y grandes, sin necesidad de que todos tengan que utilizar el mismo sistema operacional.
- Facilita a los diferentes departamentos de una organización soluciones locales, permitiendo la integración de la información principal totalmente.

:

2.1.5 Desventajas del esquema Cliente/Servidor.

- El mantenimiento de los sistemas es complejo pues implica la interacción de diferentes partes hardware y software de diferentes proveedores, lo cual dificulta el diagnóstico de fallas.

- Se cuenta con escasas herramientas para la administración y ajuste del desempeño de los sistemas, además se deben tener estrategias para el manejo de errores y para salvaguardar la consistencia de los datos.
- La seguridad del esquema C/S es preocupante, un ejemplo: las validaciones y verificaciones que se deben hacer tanto en el cliente como en el servidor.
- El desempeño es un aspecto a tener en cuenta en el esquema C/S, problemas de este estilo pueden presentarse por congestión en la red.

2.2 TECNOLOGÍAS DE DESARROLLO DE PÁGINAS WEB DINÁMICAS

Las páginas dinámicas aportan grandes beneficios porque permiten entrar a bases de datos para extraer información que pueda presentarse al usuario, dependiendo de algunos permisos y de la misma forma para almacenar información.

Existen diferentes tecnologías para el desarrollo de páginas dinámicas entre ellas están:

2.2.1 Código del Lado del Cliente (Client Side Scripts). Código ejecutado por los navegadores, el cual los computadores clientes tienen instalados. Las tecnologías más comunes de este tipo son:

- JavaScript: Lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Permite, crear ventanas, mostrar texto en movimiento y verificar las entradas a un formulario.
- Controles Activos: tecnología Microsoft que permite la creación de aplicaciones Windows, como pueden ser Visual Basic Script o Visual C. Es la respuesta de

Microsoft a los Applets de Java.

- Java Applets: Programas escritos en lenguaje de programación Java, se incrustan en HTML y se ejecutan en el navegador gracias a la Máquina Virtual de Java (JVM) que lleva éste incorporado.

2.2.2 Código del Lado del Servidor (Server Side Scripts). Código que se ejecuta en el servidor. Para su actividad el programa ejecuta y procesa los datos o peticiones que el usuario envía desde su navegador, para luego enviar los resultados del programa en una página HTML que el usuario verá normalmente en su navegador. Los más usados son:

- ASP (Active Server Pages): Permite crear dinámicamente páginas Web mediante HTML, scripts, y componentes de servidor ActiveX reutilizables, requiere de un computador configurado como Servidor Web de Microsoft (Microsoft Web Server), el navegador del cliente es indiferente pues el trabajo se realiza del lado del servidor. Da gran uso en la gestión de Bases de Datos ya que puede conectarse a SQL, Access, Oracle u otras.
- PHP (PHP Hypertext Pre-processor): Lenguaje de programación interpretado, diseñado para la creación de páginas web dinámicas. Es un lenguaje de código abierto (Open Source) y gratuito. Su gran potencia se encuentra en la interacción con los motores de bases de datos como Oracle y MySQL.
- JSP (Java Server Pages): tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo. Permiten la utilización de código Java mediante scripts.

2.2.3 Tecnología aplicada. La tecnología aplicada para la creación del portal web fue JSP, por lo tanto, los nuevos servicios son desarrollados con esta misma tecnología, ya que permite producir aplicaciones independientes de la plataforma y portables a otros sistemas operativos y servidores web.

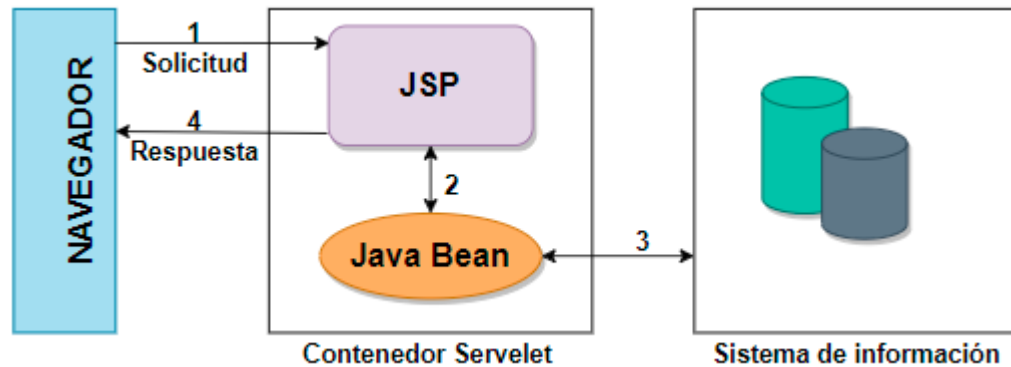
Las paginas JSP y servlets se ejecutan en la Máquina Virtual de Java, lo cual permite que se puedan usar en cualquier tipo de computador, siempre y cuando este instalada la Máquina Virtual de Java. Cada JSP se ejecuta en su propio contexto (llamado también hilo o hebra); pero no se comienza a ejecutar cada vez que recibe una petición, sino que persiste de una petición a la siguiente, de forma que no se pierde tiempo en invocarlo. Su persistencia permite hacer cosas de forma más eficiente como la conexión a bases de datos y manejo de sesiones.

Una página JSP se compila a una aplicación Java la primera vez que se invoca, y de esta aplicación Java se crea una clase que empieza a ejecutarse en el servidor como un servlet. Un JSP es una página web con etiquetas especiales y código Java incrustado, mientras que un servlet es un programa que recibe peticiones y genera a partir de ellas una página web.

2.2.3.1 Modelo de acceso a JSP.

- Un usuario en su navegador web cliente hace una petición que es enviada a un archivo JSP. Este archivo accede a componentes del servidor que generan contenido dinámico y lo presentan en el navegador.
- Después de recibir la petición del cliente, el archivo JSP pide información de un JavaBean si es necesario.
- El JavaBean en turnos puede pedir información de otro JavaBean o de una base de datos.
- Una vez el JavaBean genera el contenido, el archivo JSP puede consultar y presentar el contenido del JavaBean al navegador.

Figura 2. Modelo de acceso a JSP



La primera vez que un archivo JSP es invocado, este es compilado en un objeto, la respuesta del objeto es HTML estándar, el cual es interpretado por el navegador para ser presentado al usuario. Después de la compilación, el objeto de la página es almacenado en la memoria del servidor. En las peticiones posteriores a esta página, el servidor revisa si el archivo JSP ha cambiado. Si no ha cambiado, el servidor utiliza el objeto de la página compilada guardado en memoria para generar la respuesta al cliente, en caso contrario el servidor automáticamente compila el archivo de la página y reemplaza el objeto en la memoria.

2.3 BASES DE DATOS

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso, con una redundancia controlada y una estructura que refleja las interrelaciones y restricciones existentes en el mundo real. En la base de datos se almacena información considerada necesaria para una determinada organización o negocio.

Existen modelos que describen la estructura de una base de datos (entidades, atributos y relaciones), la mayoría de los modelos de datos poseen un conjunto de operaciones básicas como consultar y actualizar y eliminar.

2.3.1 Modelos de Bases de Datos. Las bases de datos se clasifican de acuerdo a su modelo de administración de datos. Algunos modelos utilizados con frecuencia son:

2.3.1.1 Base de Datos Jerárquica. Estas bases de datos almacenan su información en una estructura escalonada, organizando los datos en forma similar a un árbol (visto al revés), en donde un nodo padre de información puede tener varios hijos, el nodo que no tiene padres es llamado raíz, y a los nodos que no tienen hijos se les conoce como hojas. Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento. Esta limitado por su incapacidad de representar eficientemente la redundancia de datos.

2.3.1.2 Bases de Datos de Red. En este modelo se permite que un mismo nodo tenga varios padres. Ofrece una solución eficiente al problema de redundancia de datos; sin embargo, la dificultad para administrar los datos en una base de datos de red ha conllevado a que sea un modelo usado más por programadores que por usuarios finales.

2.3.1.3 Bases de Datos Relacional. Es el más utilizado para modelar problemas reales y administrar datos dinámicamente. Su fundamento es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos, también llamados tuplas. Cada relación es una tabla que está compuesta por registros (las filas de una tabla), que representan las tuplas, y campos (las columnas de una tabla). Los datos pueden ser recuperados o almacenados mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información. El lenguaje más habitual para construir las consultas a bases de datos relacionales es el Lenguaje Estructurado de Consultas (Structured Query Language, SQL), un estándar implementado por los principales manejadores de bases de datos relacionales.

2.3.2 Gestores de Bases de Datos. El sistema manejador de bases de datos es la porción más importante del software de un sistema de base de datos. Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica.

Las funciones principales de un DBMS son:

- Crear y organizar la Base de Datos.
- Establecer y mantener las trayectorias de acceso a la base de datos de tal forma que los datos puedan ser capturados rápidamente.
- Manejar los datos de acuerdo con las peticiones de los usuarios.
- Registrar el uso de las bases de datos.
- Interacción con el manejador de archivos a través de las sentencias en

Lenguaje Manipulador de Datos (Data Manipulation Language, DML) al comando del sistema de archivos.

- Respaldo y recuperación: Consiste en contar con mecanismos implantados que permitan la recuperación fácilmente de los datos en caso de ocurrir fallas en el sistema de base de datos.
- Control de concurrencia: consiste en controlar la interacción entre los usuarios concurrentes para preservar la consistencia de los datos.
- Seguridad e Integridad: consiste en contar con mecanismos que permitan el control de la consistencia de los datos evitando que estos se vean perjudicados por cambios no autorizados o previstos.

2.3.3 MySQL. Es un sistema de base de datos operacional considerado uno de los más importantes, utilizado por usuarios del medio para el diseño y programación de base de datos de tipo relacional. MySQL se usa como servidor a través del cual pueden conectarse múltiples usuarios y utilizarlo al mismo tiempo. La característica más interesante de MySQL es que permite recurrir a las bases de datos multiusuario a través de la web y en diferentes lenguajes de programación y diferentes plataformas que se adaptan a diferentes necesidades y requerimientos, además MySQL es conocida por desarrollar alta velocidad de búsqueda de datos e información, a diferencia de sistemas anteriores.

2.3.4 Ventajas de MySQL.

- El MySQL es un Open Source, o sea código abierto que puede ser usado y modificado.

- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consume puede ser ejecutado en una maquina con escasos recursos sin ningún problema.
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Su conectividad, velocidad, y seguridad hacen de MySQL altamente apropiado para acceder a bases de datos en internet.

2.4 NETBEANS

Es un entorno de desarrollo integrado (IDE), siendo una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas escritos en JAVA, pero puede servir para cualquier otro tipo lenguaje de programación. Netbeans es un producto libre y gratuito sin restricciones de uso.

- El Netbeans es un entorno de desarrollo integrado de código abierto escrito completamente en Java usando la plataforma Netbeans, soporta desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles).
- La versión actual es NetBeans IDE 8.1 Desde NetBeans IDE 6.5 se extienden las características existentes del Java EE (incluyendo Soporte a Persistencia, EEJB 3 y JAX-WS). Adicionalmente, el Netbeans Enterprise Pack soporta el desarrollo de Aplicaciones empresariales java EE 5, incluyendo herramientas de desarrollo visuales de SOA, herramientas de esquemas XML, orientación a web servicios (for BPEL), y modelado UML.

- Todas las funciones del IDE son provistas por módulos. Cada Módulo provee una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones. Netbeans contiene todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, permite al usuario comenzar a trabajar inmediatamente.

2.5 SISTEMAS DE CONTROL DE VERSIONES

Un sistema de control de versiones es un software que administra el acceso a un conjunto de ficheros, y mantiene un historial de cambios realizados. El control de versiones es útil para guardar cualquier documento que cambie con frecuencia, o el código fuente de un programa.

Normalmente consiste en una copia maestra en un repositorio central, y un programa cliente con el que cada usuario sincroniza su copia local. Además, el repositorio guarda registro de los cambios realizados por cada usuario, y permite volver a un estado anterior en caso de necesidad.

Existen multitud de sistemas de control de versiones, pero sin duda, el más popular es CVS (Concurrent Versions System). CVS tuvo el mérito de ser el primer sistema usado por el movimiento de código abierto para que los programadores colaboran remotamente mediante el envío de parches. Es de uso gratuito, código abierto, y emplea fusión de cambios. Subversión se creó para igualar y mejorar la funcionalidad de CVS, preservando su filosofía de desarrollo.

2.5.1 Subversión. Sistema de control de versiones iniciado por CollabNet Inc. Emplea licencia Apache/BSD. Se usa para mantener versiones actuales e históricas y los cambios de archivos tales como los de código fuente, páginas web y/o documentación. Esto permite recuperar versiones antiguas de los datos o examinar cómo han ido evolucionando esto. Su objetivo es ser un sucesor prácticamente compatible del ampliamente usado Concurrent Version System (CVS).

Subversión puede trabajar a través de redes, lo que permite que las personas que estén en diferentes computadores puedan usarlo, con la posibilidad de que varias personas modifiquen y gestionen el mismo conjunto de datos desde sus sitios promueve la colaboración, y como el trabajo está versionado, ya que, si se produce algún cambio incorrecto de los datos, sólo hace falta deshacerlo.

2.6 PROGRAMACIÓN UTILIZADA

Para el desarrollo de este proyecto se usó la Programación Orientada a Objetos (P.O.O.). La P.O.O. es una de las formas más populares de programas que usa objetos y sus interacciones para diseñar aplicaciones y programas de computador, intenta simular el mundo real a través del significado de objetos que contienen características y funciones; abstrae algunas características de sistemas naturales complejos como son:

- **Atributos:** estado del objeto.
- **Métodos:** comportamiento del objeto.
- **Herencia:** comportamientos comunes entre objetos relacionados para hallar relaciones de especialización y generalización de comportamientos.

2.6.1 Clases. Definición de todos los elementos de que esta hecho un objeto. Cuando se programa un objeto y se definen sus características y funcionalidades, realmente se programa una clase. Por lo tanto, para realizar la abstracción de sistemas naturales, observamos y analizamos un grupo de cosas con características comunes, el resultado de esta abstracción será válido para todas estas cosas.

2.6.2 Objetos. Cualquier cosa real o abstracta, que posee atributos y un conjunto de operaciones que manipulan esos atributos que da un comportamiento particular. Un objeto es una instancia de una clase, el estado del objeto se determina por el estado (valor) de sus propiedades o características (atributos).

2.6.3 Atributos. Características de un objeto siendo un conjunto de datos (valores) y calificadores para aquellos datos. Estos atributos pueden ser desde tipos de datos simples (enteros, caracteres, cadenas de texto) hasta otros objetos.

2.6.4 Métodos. Son funciones o procedimientos propios de la clase que pueden tener acceso a los atributos de la misma para realizar las operaciones para los que son programados.

2.6.5 Herencia. Se fundamenta en usar una clase ya creada para tomar sus características en clases más especializadas o derivadas de ésta para reutilizar el código que sea común con la clase base, y solamente definir nuevos métodos o redefinir algunos de los existentes para ajustarse al comportamiento particular de esta subclase.

2.6.6 Beneficios de la Programación Orientada a Objetos.

- Permite obtener aplicaciones modificables y fácilmente extensibles a partir de componentes reutilizables.
- Disminución en el tiempo de desarrollo gracias a la reutilización del código.
- El desarrollo del software es más intuitivo porque las personas piensan naturalmente en términos de objetos más que en términos de algoritmos de software.

A continuación, se presenta una breve descripción de Java, el lenguaje de programación orientado a objetos que se usó en el desarrollo de este proyecto:

2.6.7 Java y JDK (Java Development Kit). Java es un lenguaje desarrollado por Sun Microsystems, en el año 2009 fue adquirida por la compañía Oracle.

Permite escribir aplicaciones que puedan ejecutarse en casi cualquier plataforma. El lenguaje toma parte de la sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. Además, cuenta con una característica denominada “recolección de basura”, que examina la memoria y libera cualquier variable u objeto que no esté siendo usado. El JDK es un software que provee herramientas de desarrollo para la creación de programas en java.

Para trabajar con Java se necesita un kit de desarrollo que proporciona:

- Un compilador: javac.
- Un intérprete: java.

- Un generador de documentación: javadoc.
- Un visor de applet para generar sus vistas previas, ya que un applet carece de método main y no se puede ejecutar con el programa java: Appletviewer.

2.7 SERVIDORES WEB

Es un tipo de software que se encuentra a la espera de una petición hecha por una aplicación cliente y da respuesta a dicha petición a través de una página web. Para cada transacción el servidor debe realizar dos acciones básicas: integrar todos los componentes de la página (texto, imágenes, vídeo, scripts, etc.) y enviarla rápidamente al usuario. A continuación, se describe el servidor Web que se ajusta a la tecnología escogida para el proyecto.

2.7.1 Servidor Jakarta Tomcat. Servidor de aplicaciones Java basado en los estándares definidos por Sun Microsystems. Tomcat es desarrollado como parte del proyecto de código abierto Jakarta de la fundación de software Apache y es uno de los servidores de aplicaciones Java más utilizados, en especial porque es liviano, cumple con todos los estándares, sencillo de instalar, tiene muy buena documentación y es gratuito, además por ser escrito en Java funciona en cualquier sistema operativo que disponga de la Máquina Virtual de Java (JVM).

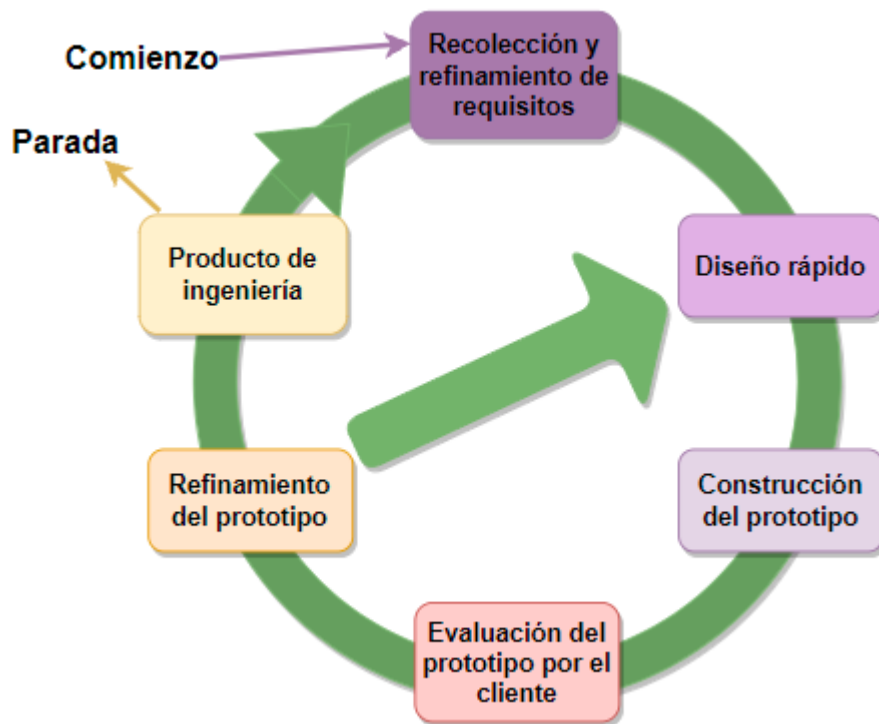
Es posible ejecutarlo desde la línea de comandos (consola o terminal), después de configurar algunas variables de entorno, sin embargo, configurar cada variable de entorno y seguir los parámetros de las líneas de comando usados por Tomcat es tedioso y expuesto a errores, en su lugar se proporciona código existente para arrancar y detener el servicio.

3. MARCO METODOLÓGICO

3.1 PROTOTIPO EVOLUTIVO

Para realizar los nuevos servicios para los portales web comunidad académica de las diferentes escuelas y facultades de la Universidad Industrial de Santander se propone como metodología de desarrollo el prototipo evolutivo.

Figura 3. Prototipo evolutivo



La elección de la metodología de prototipo evolutivo se debe a las siguientes razones:

- Es deseable tener un bosquejo de lo que se desee mejorar o crear para poder incorporar sugerencias de cambio por parte de los usuarios del portal de las escuelas en las etapas tempranas del desarrollo.
- Por otra parte, es necesario saber lo antes posible si hemos interpretado correctamente las especificaciones y las necesidades de las escuelas y de los profesores.
- En muchos casos los usuarios no tienen una idea definida de lo que desean, por lo tanto, debemos tomar decisiones y suponer qué es lo que el usuario quiere. Por este motivo, la emisión de los prototipos brinda la posibilidad de efectuar refinamientos de los requerimientos en forma sucesiva a fin de acercarse al producto deseado. Con el prototipo evolutivo se comienza diseñando y construyendo las partes más importantes de la aplicación en un prototipo que posteriormente se refinará y ampliará hasta que el prototipo se termine. Este prototipo será el software que se entregará al final.
- La decisión se fundamenta en la ventaja de la realización de los cambios en etapas tempranas y la posibilidad de emisión de varios prototipos evaluables durante el desarrollo, obteniéndose de este modo, y de forma paralela, una metodología integral también para el proceso de evaluación del programa.
- Esta metodología propicia un intercambio de conocimientos y de autocrítica al sistema, lo que conlleva a que se produzcan muchas pruebas antes de liberar una nueva versión, así como mejoras rápidas a problemas que puedan surgir durante su uso.

Procedimiento a seguir para la metodología planteada:

- Hacer un análisis de los requerimientos para la construcción de los prototipos.

- Desglosar los objetivos globales con el fin de tener una idea más detallada del software a realizar, mediante reuniones entre los desarrolladores y los usuarios, en las cuales se identifican los requerimientos de los usuarios y se concluyen los aspectos que requieren una mayor definición.
- Presentar al usuario el diseño de un prototipo enfocado en los aspectos visuales del software, métodos de entrada y formatos de salida, para proceder a la construcción del mismo.
- Evaluación del prototipo por parte del usuario para filtrar los requisitos del software a desarrollar.
- Se produce un proceso interactivo en el que el prototipo es depurado para satisfacer necesidades del usuario, de igual forma el desarrollador obtiene una mejor comprensión de lo que hay que hacer para la entrega del producto final de ingeniería requerido por el usuario.

3.2 LENGUAJE DE MODELADO UNIFICADO

El Lenguaje de Modelado Unificado o Unified Modeling Language (UML), es el más utilizado en la actualidad. Es un lenguaje gráfico estándar para visualizar, especificar, construir y documentar un sistema para describir un modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

UML no es un método de desarrollo porque no indica los pasos a seguir para llegar al código, es decir, no especifica como pasar del análisis al diseño y de este

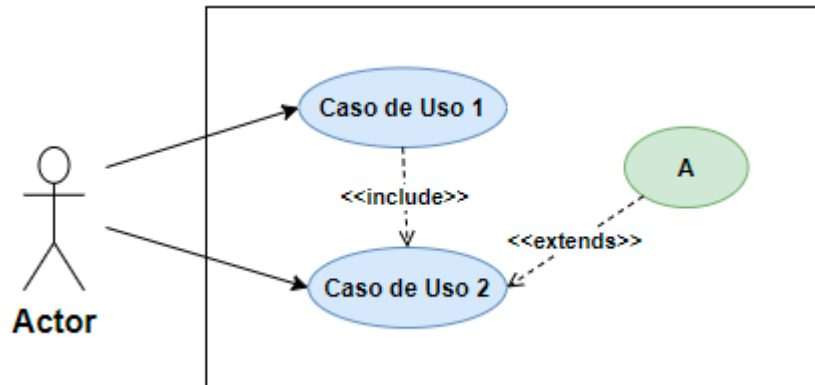
al código. Al no ser un método de desarrollo resulta ser independiente del ciclo de desarrollo que se siga, puede encajar en un ciclo en cascada, evolutivo, espiral o en métodos ágiles de desarrollo.

3.2.1 Diagramas de UML. Los diagramas UML utilizados en el desarrollo de este proyecto fueron diagramas de casos de uso y diagramas de secuencias. Estas son las razones principales por las cuales se prefirió UML como el lenguaje de modelado:

- Tiene una notación gráfica muy expresiva que permite representar todas las fases de un proyecto informático, desde el análisis con casos de uso, el diseño con diagramas de clases, objetos, etc.
- Facilita el entendimiento de la información, la función y el comportamiento de un sistema, haciendo fácil el análisis de los requerimientos, ya que sirve de apoyo en los procesos de análisis de un problema.
- Permite a los creadores de sistemas realizar diseños que faciliten la comunicación a otras personas de manera convencional.
- Permite generar un punto de comparación entre lo logrado y lo planificado.

3.2.2 Diagramas de casos de uso. Representación gráfica del entorno del sistema (actores) y su funcionalidad principal. Describe lo que hace el sistema desde el punto de vista de un observador externo, concentrándose en expresar lo que hace el sistema y no en dar respuesta de cómo lograr su comportamiento.

Figura 4. Diagramas de casos de uso



Actores: Un actor en un caso de uso representa un rol, que alguien o algo puede desempeñar dentro un sistema y no un alguien o algo específico.

En este proyecto se destacan tres clases de actores:

- **Administradores:** Son usuarios que además de pertenecer a la categoría de usuarios tienen un perfil de administrador, con el cual tiene permisos extras a los que tiene un usuario comúnmente dentro del sitio; alguno de estos son los auxiliares de administración del portal, los profesores, las secretarías de las escuelas, entre otras. Dentro de esta categoría se incluye también el súper administrador.
- **Súper Administrador:** Es el usuario que puede administrar, controlar y modificar los portales web de las escuelas, sus parámetros y sus usuarios.
- **Usuario Portal Web Comunidad Académica:** Es el tipo de usuario común de los portales web y a quien van dirigidas las páginas de servicio. Este usuario solo tiene control sobre sus servicios permitidos.

Inclusión (include–uses): Es una forma de interacción, un caso de uso dado puede "incluir" otro. Una inclusión es utilizada para indicar que un caso de uso

depende de otro, es decir, la funcionalidad de determinado caso de uso se requiere para realizar las tareas de otro. En la figura 4 el caso de uso “Caso de uso 1” depende de los casos de uso “A” y “B”.

Extensión (extend): Es otra forma de interacción, una extensión representa una variación de un caso de uso a otro, es decir, una dependencia específica entre los casos; a través de la cual un caso de uso puede extender a otro.

3.2.3 Diagramas de secuencias. Es aquel que muestra la forma en que los objetos interactúan entre sí al transcurrir el tiempo. Consta de objetos que se representan del modo usual: rectángulos con nombre (subrayado), mensajes representados por líneas continuas con una punta de flecha y el tiempo representado como una progresión vertical.

Objetos: Se ubican en la parte superior del diagrama de izquierda a derecha y se acomodan de manera que simplifiquen al diagrama. La línea que está debajo de cada objeto será una línea discontinua conocida como la línea de vida de un objeto.

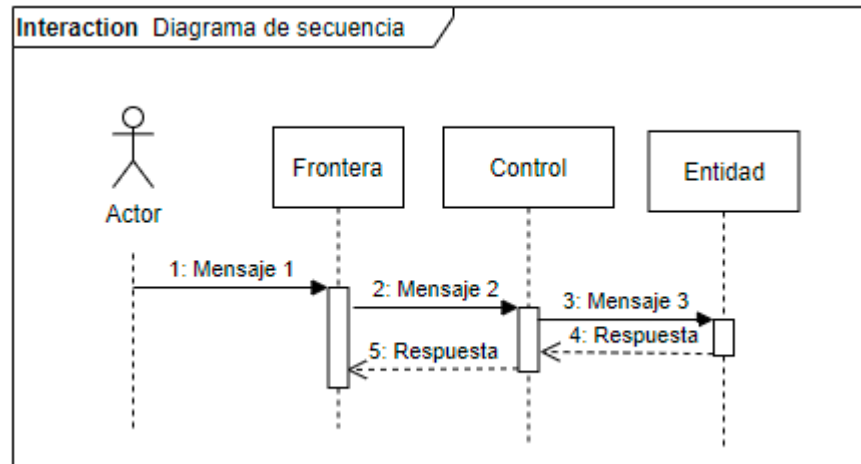
Con la línea de vida se encuentra un pequeño rectángulo conocido como activación, el cual representa la ejecución de una operación que realiza el objeto.

Mensaje: Un mensaje que va de un objeto a otro pasa la línea de vida de un objeto a otro. Un objeto puede enviarse un mensaje a sí mismo. Un mensaje puede ser simple, sincrónico o asincrónico.

Tiempo: El diagrama representa al tiempo en dirección vertical. Inicia en la parte superior y avanza hacia la parte inferior. Un mensaje que esté más cerca de la parte superior ocurrirá antes que uno que esté cerca de la parte inferior.

GUI: (Siglas en Ingles) La interfaz gráfica de usuario; es la interfaz de interacción del usuario y en la que más interactividades se presentan con otros objetos.

Figura 5. Diagramas de secuencias



3.3 ESTÁNDARES DE PROGRAMACIÓN

3.3.1 Modelo de datos. Es un lenguaje utilizado para la descripción de una base de datos, por lo general permite describir estructuras de datos de la base de datos (el tipo de datos que incluye la base y la forma en que se relacionan), las restricciones de integridad (las condiciones que los datos deben cumplir para reflejar correctamente la realidad deseada) y las operaciones de manipulación de los datos (agregar, borrar, modificar).

3.3.2 Nombres de las tablas. Los nombres de los campos, así como de las tablas de la base de datos, se escriben en minúsculas, exceptuando los prefijos TP, TR y TB que indican si la tabla es principal, relacional o básica, respectivamente. Después de los prefijos en los casos de los nombres de las tablas se usa la denominada notación camello donde la primera letra de cada palabra que conforme su nombre va en mayúscula; si es un nombre compuesto por dos o más palabras, los nombres tendrán en mayúscula la primera letra de cada palabra que la forma.

Como se mencionó, se han definido tres categorías para las diferentes tablas que conforman la base de datos. Dada la categoría de la tabla, se agrega un prefijo a su nombre que permita conocer la categoría a la que pertenece. Las categorías son:

- **Tabla básica:** Aquella cuyos registros son necesarios para el correcto funcionamiento de la base de datos. Estas tablas no experimentan muchos cambios en los datos. El prefijo a anteponer a los nombres de estas tablas es “TB_”, es decir, la tabla que almacena las categorías de clasificación de los usuarios del portal web es llamada “TB_Categorias”, por ejemplo.
- **Tabla de Relación:** Surge de la relación muchos a muchos de una o dos tablas cualquiera. Los nombres de las tablas de relación deben ser siempre descriptivos para cada relación. El prefijo a anteponer a los nombres de estas tablas es “TR_”, por ejemplo, la tabla “GuionTalleres” que relaciona una clase (Asignatura) con los talleres es conocida como “TR_GuionTalleres”.
- **Tabla Principal:** Aquella cuyo número de registros tiende a crecer en gran cantidad y que además no es posible clasificar como tabla básica o de relación. Un ejemplo de tabla principal es la tabla que almacena los usuarios del portal

EISIWeb. El prefijo por anteponer a los nombres de estas tablas es “TP_”, es decir la tabla “Usuarios”, es conocida como “TP_Usuarios”.

3.3.3 Clases. Para las nuevas clases que se van creando en el proyecto se sigue el mismo proceso que es utilizado en las tablas de las bases de datos. Por ejemplo, la tabla TP_MenuGestion en las clases se llama TP_MenuGestion.java.

3.3.4 Paginas JSP. Los nombres de las páginas JSP que componen los portales web comunidad académica también siguen la notación camello, son escritos de manera que la primera letra es una mayúscula seguido de letras minúsculas, en caso de que el nombre del JSP sea compuesto por dos o más palabras, entonces la primera de cada palabra debe ir en mayúscula, por ejemplo, NuevoAdminServicioEscuelas.jsp.

3.3.5 Organización de directorios. Los directorios del sitio están organizados de tal manera que los archivos que se almacenen en ellos correspondan a lo que describe el nombre del directorio.

4. DESARROLLO DE LA HERRAMIENTA, ADMINISTRACIÓN Y MANTENIMIENTO

Para el desarrollo de este proyecto se siguió la metodología de prototipo evolutivo. Al iniciar el proyecto se elaboró un primer prototipo durante la fase de requerimientos, el cual fue mejorado con la inclusión de nuevos requerimientos surgidos en la fase de desarrollo, a medida que se generaba un prototipo, el mismo era sometido a pruebas de funcionamiento y se le realizaban los refinamientos pertinentes a partir del resultado de dichas pruebas

4.1 PROTOTIPO ESPERADO

Al iniciar el proyecto aún no se contaba con una concepción clara de lo sería el producto final, sin embargo, en el transcurso del desarrollo y evolución de los prototipos, las pruebas y análisis del sistema se pudo comprobar que se estaba acercando a los requerimientos iniciales, esto con el fin de enfocar el desarrollo a la solución de las necesidades de los usuarios.

El objetivo específico inicial y los requisitos que surgieron se dieron gracias a la realización de prototipos y la realimentación con el cliente. Para cada objetivo se listaron los requerimientos detallados de este, los cuales se cumplieron para el prototipo final.

4.1.1 Crear nuevo servicio para el administrador web.

Objetivo inicial:

- Crear un nuevo servicio que permita gestionar los menús de todos los portales de las escuelas desde el módulo del Administrador Web.

Requisitos finales del objetivo:

- Nueva página de entrada desde donde se podrá crear un registro con una acción específica y seleccionar las escuelas en donde se requiera la ejecución.
- El administrador contará con la posibilidad de activar, desactivar, modificar eliminar y crear un menú desde una escuela y luego elegir a qué escuelas distribuirlo con los debidos permisos.

4.1.2 Realizar actualización al servicio de limpieza del Aula virtual.**Objetivo inicial:**

- Actualizar el servicio de limpieza que permita una limpieza interna más controlada y segura.

Requisitos finales del objetivo:

- El servicio de limpieza será actualizado tanto en la vista como en el código fuente.
- En el portal el usuario podrá ver mensaje de informe según el estado de limpieza de cada sección del Aula Virtual.
- Este servicio solo lo deben realizar los profesores o directores de una asignatura.

4.1.3 Actualizar el filtro que muestra los archivos compartidos en el Aula Virtual.

Objetivo inicial:

- Se revisará y optimizará la función de filtro que se encarga de elegir los archivos que se deben mostrar a los usuarios que correspondan.

Requisitos finales del objetivo:

- El servicio brindará la seguridad de privacidad al usuario que comparte un archivo, bien sea con un usuario, con un subgrupo, con todo el grupo o con los demás grupos de la asignatura o simplemente con el profesor.
- Los archivos solo se deben mostrar al usuario o al grupo de usuarios que se eligió para verlos.

4.1.4 Crear la opción de imprimir el formato PEP de cada asignatura

Objetivo inicial:

- Crear el servicio que permite descargar en PDF la información de las asignaturas que están en el plan de estudio de las escuelas con el formato PEP.

Requisitos finales del objetivo:

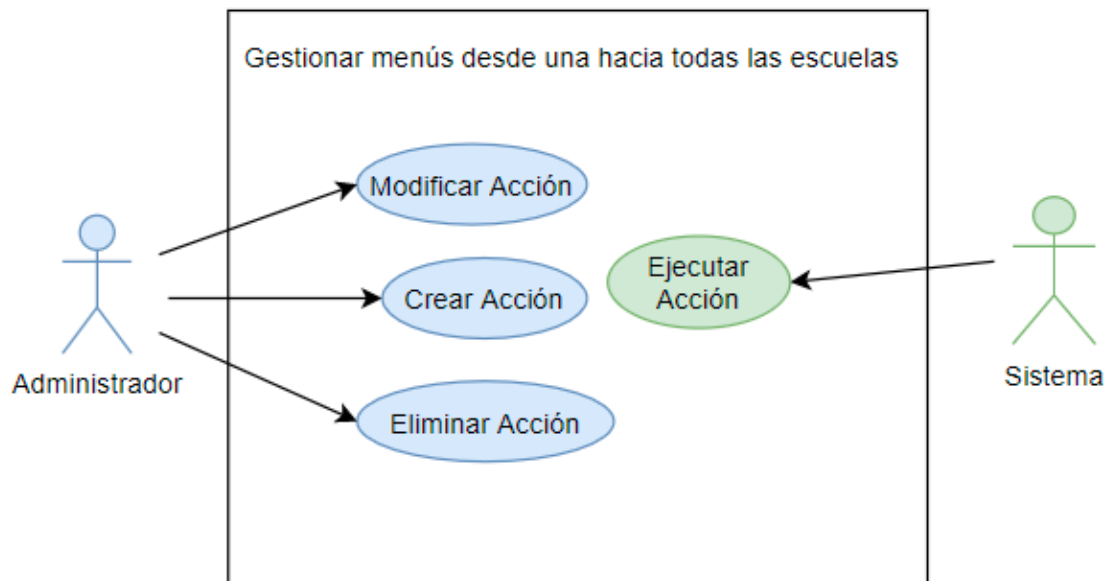
- Se hace necesario actualizar en cada escuela la información de sus planes de estudio e ingresar esta información cumpliendo el estándar del PEP.

- El portal mostrará a todo el público la información de las asignaturas que conforman cada plan de estudio dando la opción de descargarla.

4.2 DIAGRAMAS DE CASOS DE USO

4.2.1 Gestionar los menús de todos los portales de las escuelas desde el módulo del Administrador Web.

Figura 6. Diagrama de casos de uso: Gestionar menús desde una hacia todas las escuelas



4.2.2 Actualizar el servicio de limpieza del Aula Virtual.

Figura 7. Diagrama de casos de uso: Realizar limpieza del aula virtual

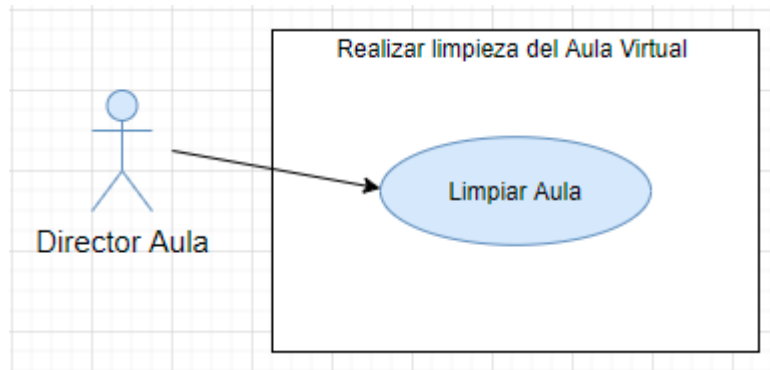
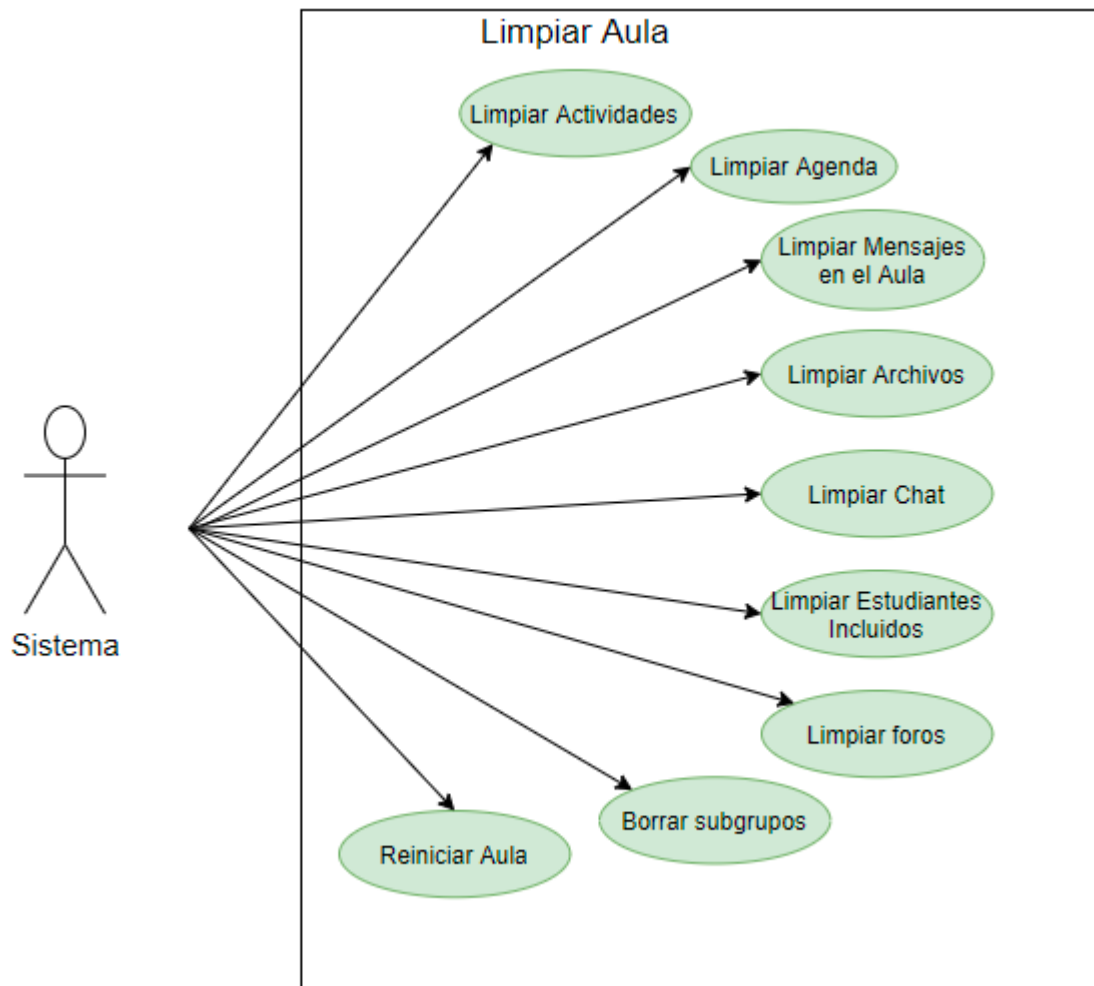
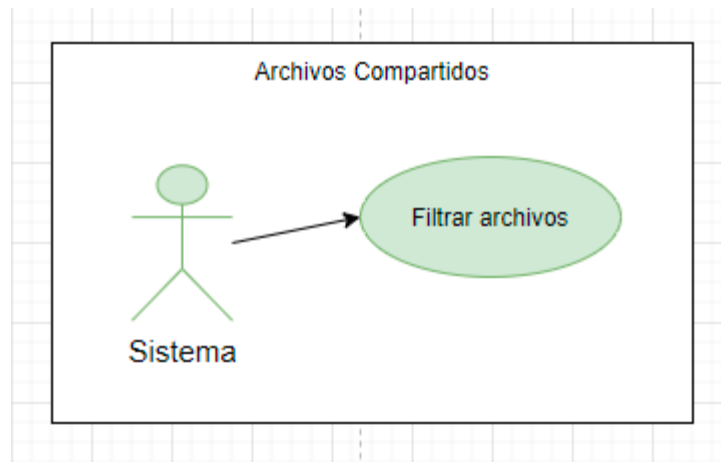


Figura 8. Limpiar Aula



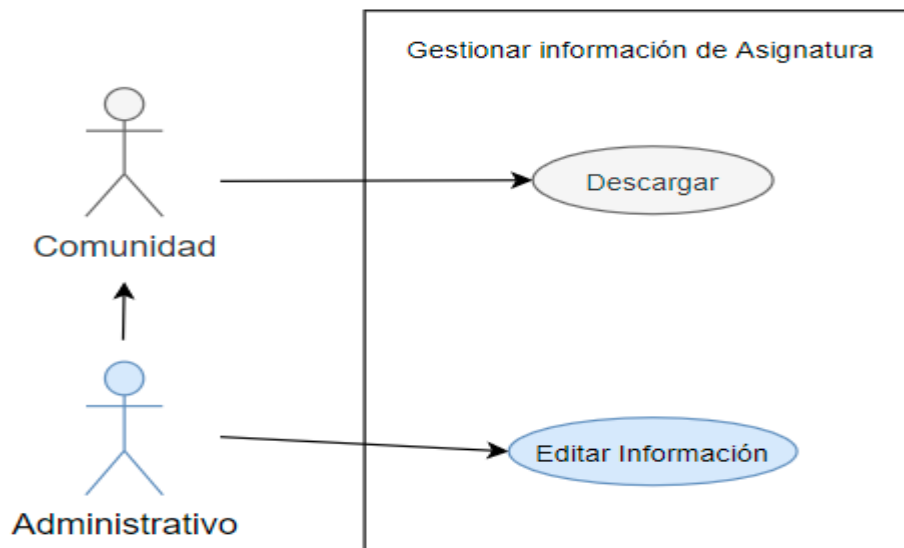
4.2.3 Actualizar el filtro que muestra los archivos compartidos en el aula virtual.

Figura 9. Diagrama de casos de uso: Archivos compartidos



4.2.4 Descargar en formato PEP la información de cada asignatura del plan de estudio.

Figura 10. Diagrama de casos de uso: Gestionar información de asignaturas



4.3 DOCUMENTACIÓN DE CASOS DE USO DEL SISTEMA

4.3.1 Gestionar los menús de todos los portales de las escuelas desde el módulo del Administrador Web.

Tabla 1. Casos de uso: Gestionar menús desde una hacia todas las escuelas

TÍTULO	FUNCIONES PRIMARIAS
Crear Acción	El administrador web crea un registro con una acción específica a realizar en las escuelas.
Modificar Acción	El administrador modifica el registro creado.
Eliminar Acción	Elimina el registro creado sin importar que se haya ejecutado o no.
Ejecutar Acción	El sistema valida si hay registro para la escuela y lo ejecuta.

1.1.1 Actualizar el servicio de limpieza del Aula Virtual.

Tabla 2. Casos de uso: Realizar limpieza del aula

TÍTULO	FUNCIONES PRIMARIAS
Limpiar Aula	El director del aula, efectúa la limpieza del aula cuando crea conveniente.
Limpiar Archivos	El sistema limpia solo los archivos compartidos por los estudiantes
Limpiar Chat	El sistema efectúa la limpieza del chat del aula, limpiando las participaciones de todos los miembros del aula.

Limpiar Mensajes en el Aula	El sistema lleva a cabo la limpieza de todos los mensajes: eliminando los creados por estudiantes y la participación de estos en los creados por el director (Profesor).
Limpiar Agenda	El sistema efectúa la limpieza eliminando los eventos creados o programados por el profesor.
Limpiar Foros	El sistema elimina todas las conversaciones y foros creados por estudiantes; de los foros creados por el profesor los limpian eliminando las calificaciones y participaciones de los estudiantes.
Limpiar Estudiantes Incluidos	El sistema elimina la lista de estudiantes incluidos al aula de manera manual.
Limpiar Actividades	El sistema efectúa la limpieza eliminando los archivos subidos y las participaciones que hayan realizado los estudiantes a las actividades.
Borrar Subgrupos	El sistema elimina todos los subgrupos de clase que se hayan creado en el Aula.
Reiniciar Aula	El sistema reinicia los parámetros del aula para que quede lista para un nuevo periodo académico.

4.3.2 Actualizar el filtro que muestra los archivos compartidos en el aula Virtual.

Tabla 3. Casos de uso: Archivos compartidos

TÍTULO	FUNCIONES PRIMARIAS
Filtrar archivos	El sistema solo carga y muestra los archivos que fueron compartidos con el usuario logueado o los que este subió.

4.3.3 Descargar en formato PEP la información de cada asignatura del plan de estudio.

Tabla 4. Casos de uso: Gestionar información de asignaturas

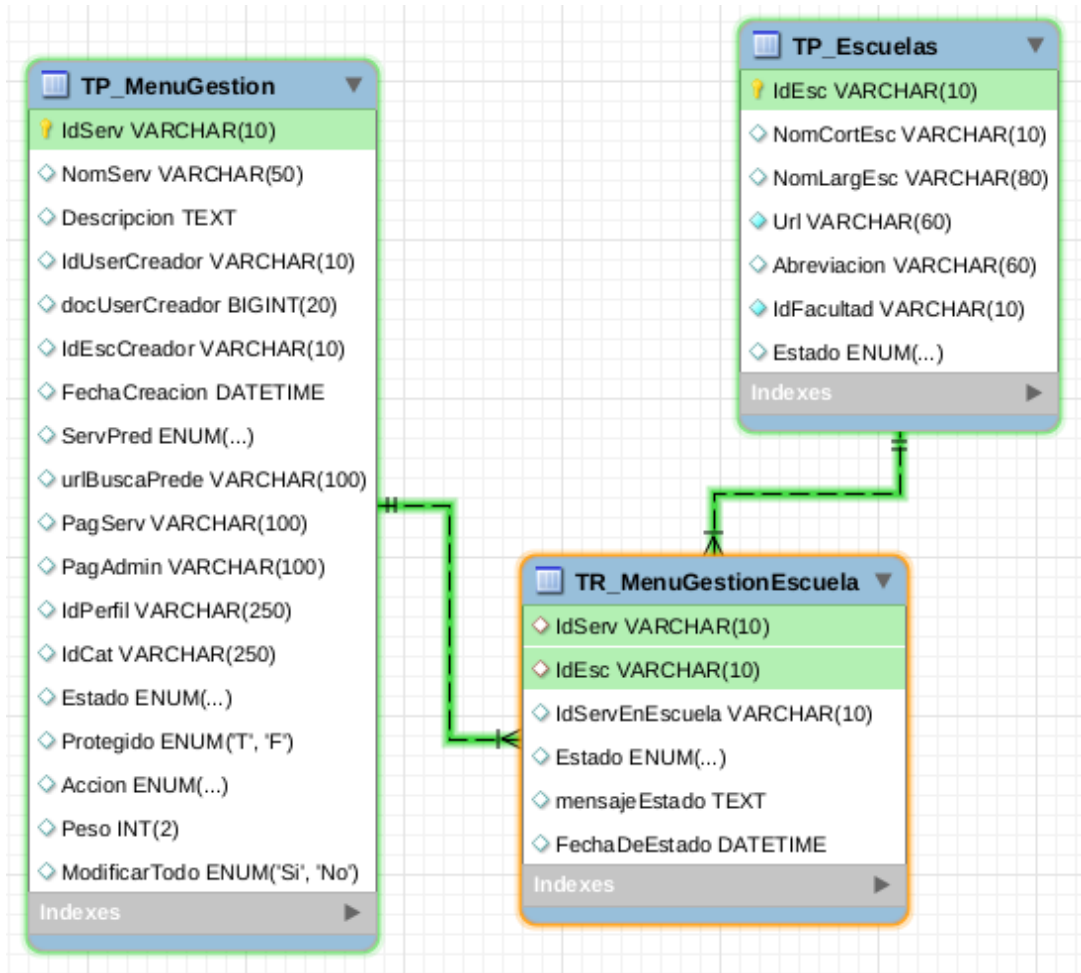
TÍTULO	FUNCIONES PRIMARIAS
Descargar	El usuario descarga la información en PDF de cada asignatura que está en los planes de las escuelas.
Editar Información	El administrativo modifica los datos ingresados de las asignaturas según lo estipula el formato PEP.

4.4 DISEÑO Y ANÁLISIS

Diagramas Entidad/Relación de los servicios desarrollados

4.4.1 Gestionar los menús de todos los portales de las escuelas desde el módulo del Administrador Web.

Figura 11. Diagrama E/R: Gestionar menú desde una hacia todas las escuelas



4.4.2 Actualizar el servicio de limpieza del Aula Virtual.

Figura 12. Diagrama E/R: Realizar limpieza del aula

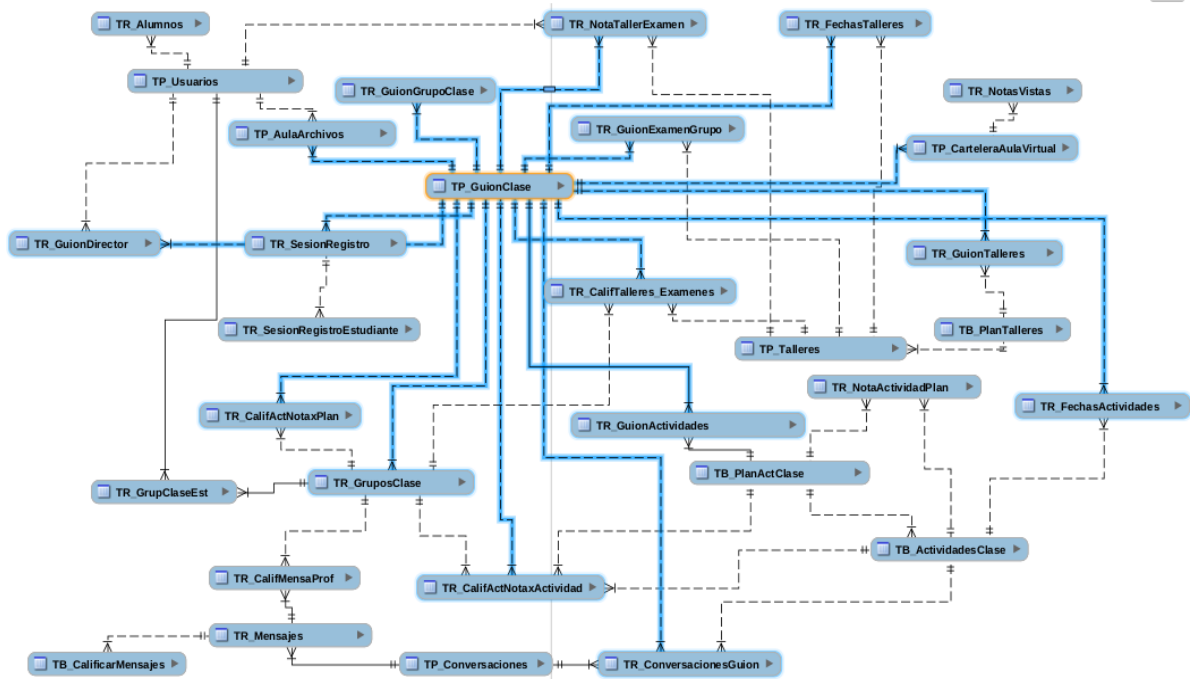


Figura 13. Diagrama E/R: Limpiar archivos

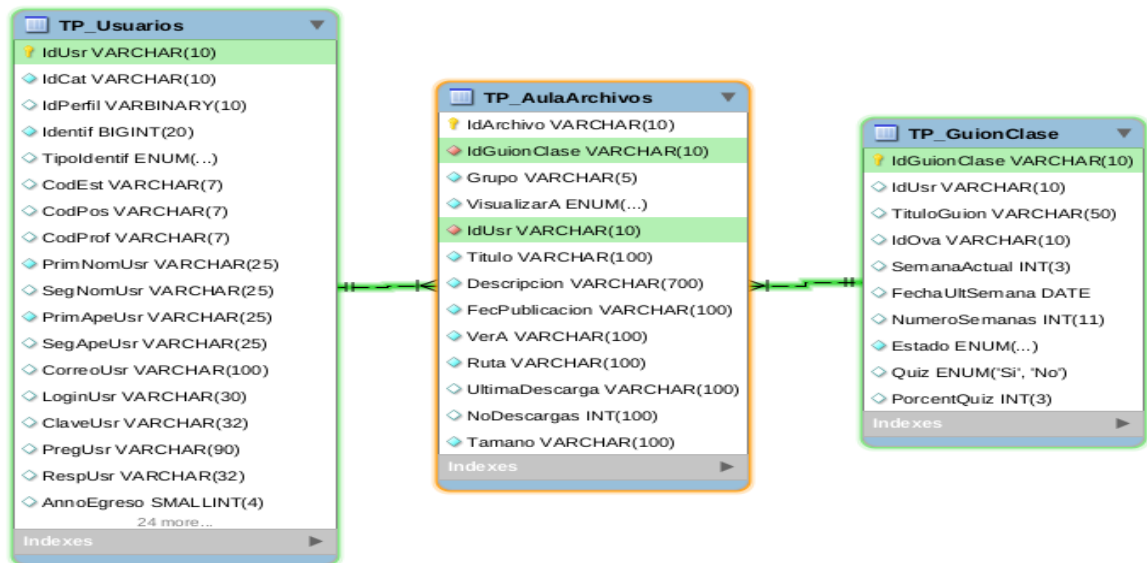


Figura 14. Diagrama E/R: Limpiar Chat

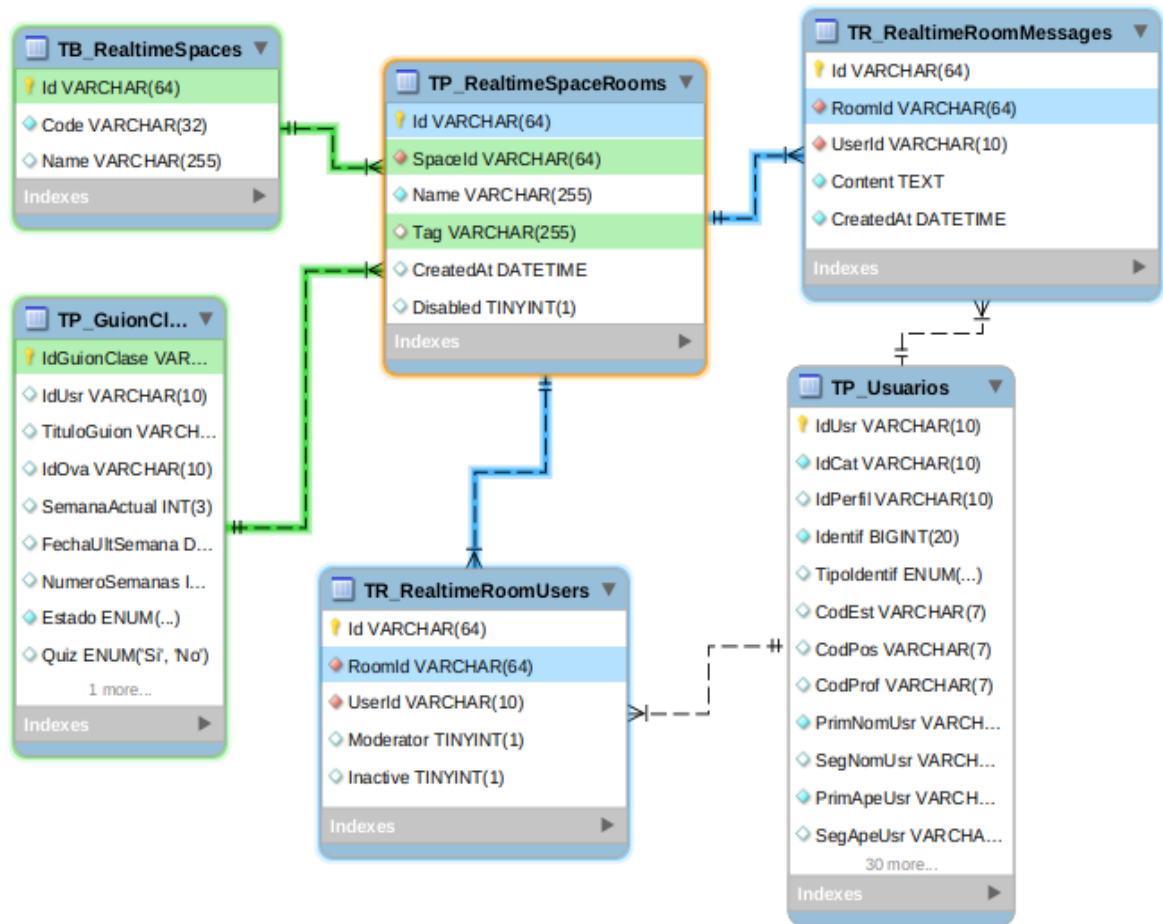


Figura 15. Diagrama E/R: Limpiar mensajes en el aula

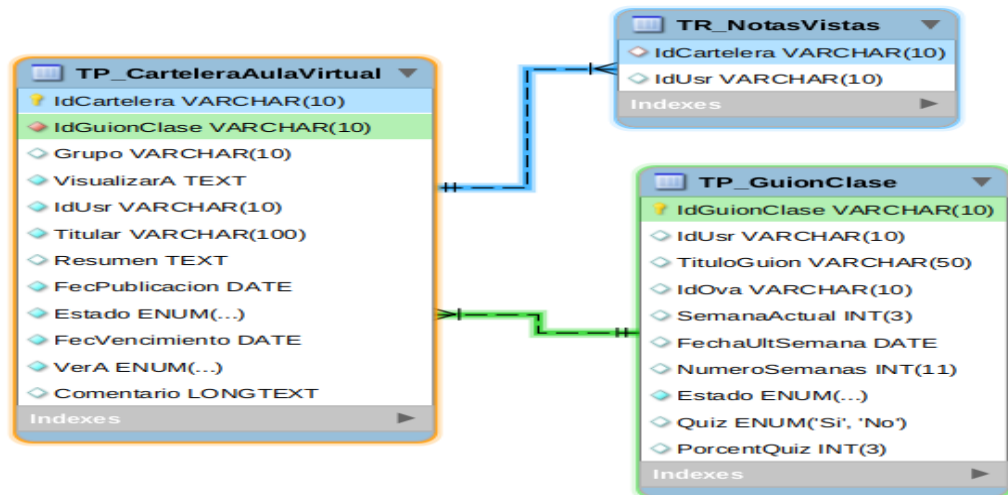


Figura 16. Diagrama E/R: Limpiar agenda

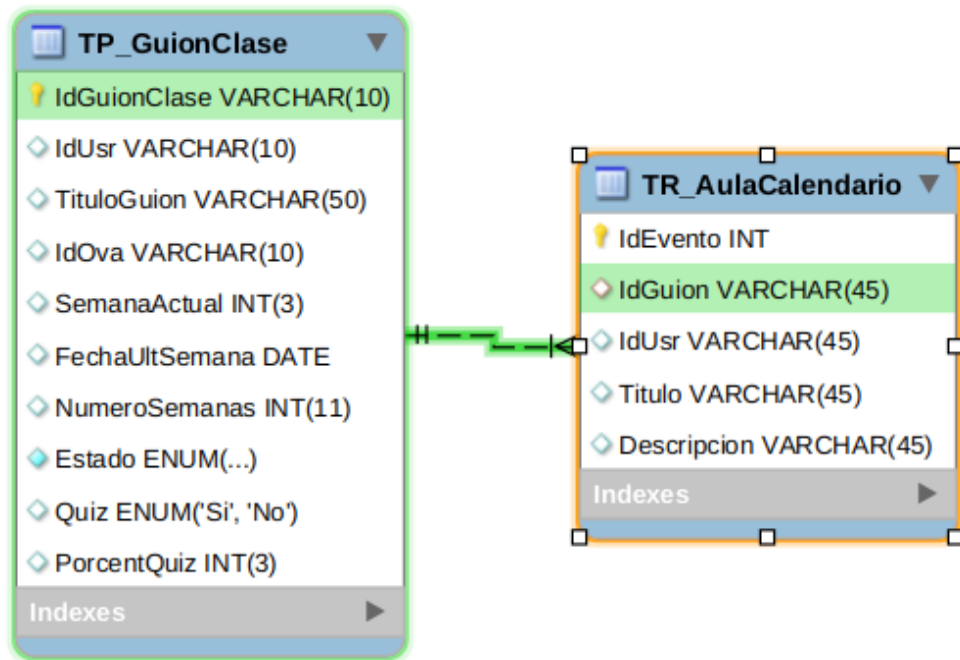


Figura 17. Diagrama E/R: Limpiar Foros

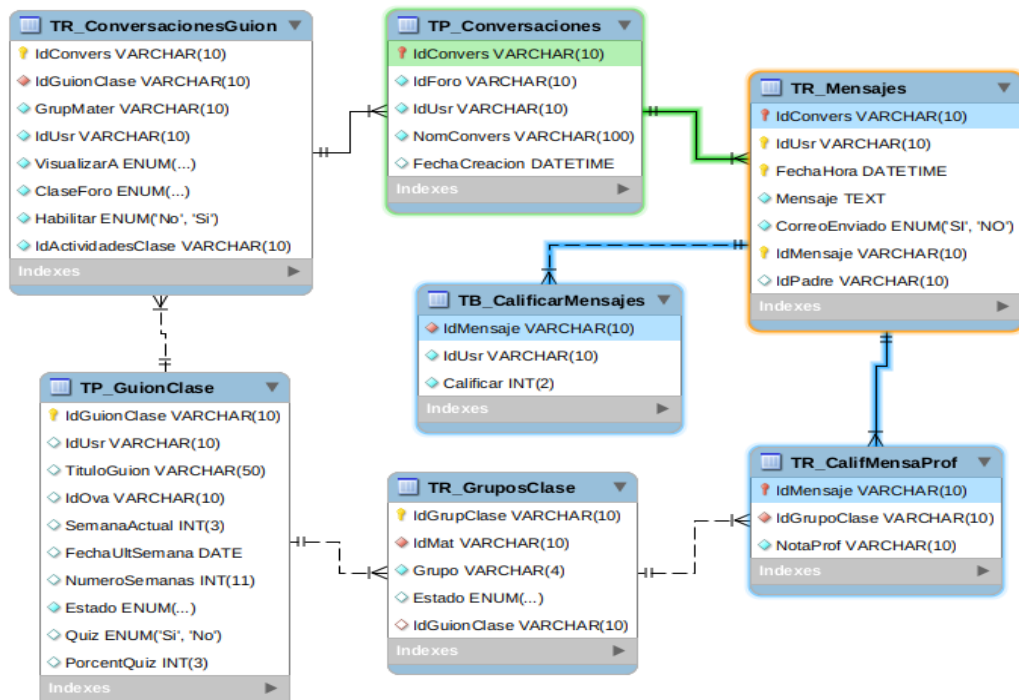


Figura 18. Diagrama E/R: Limpiar estudiantes incluidos

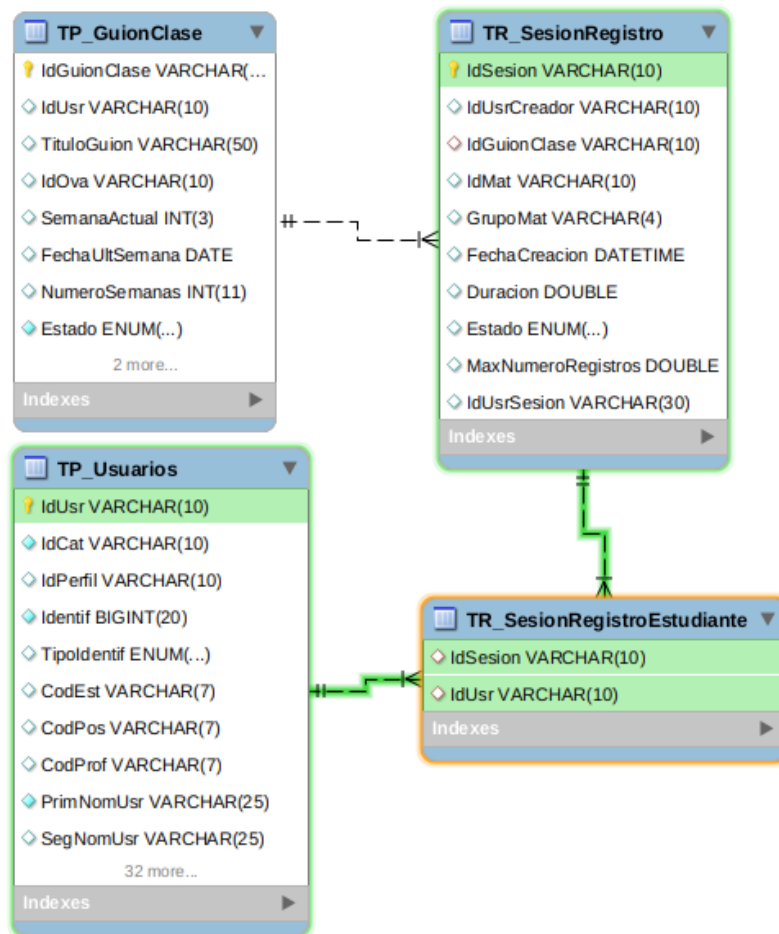


Figura 19. Diagrama E/R: Limpiar actividades

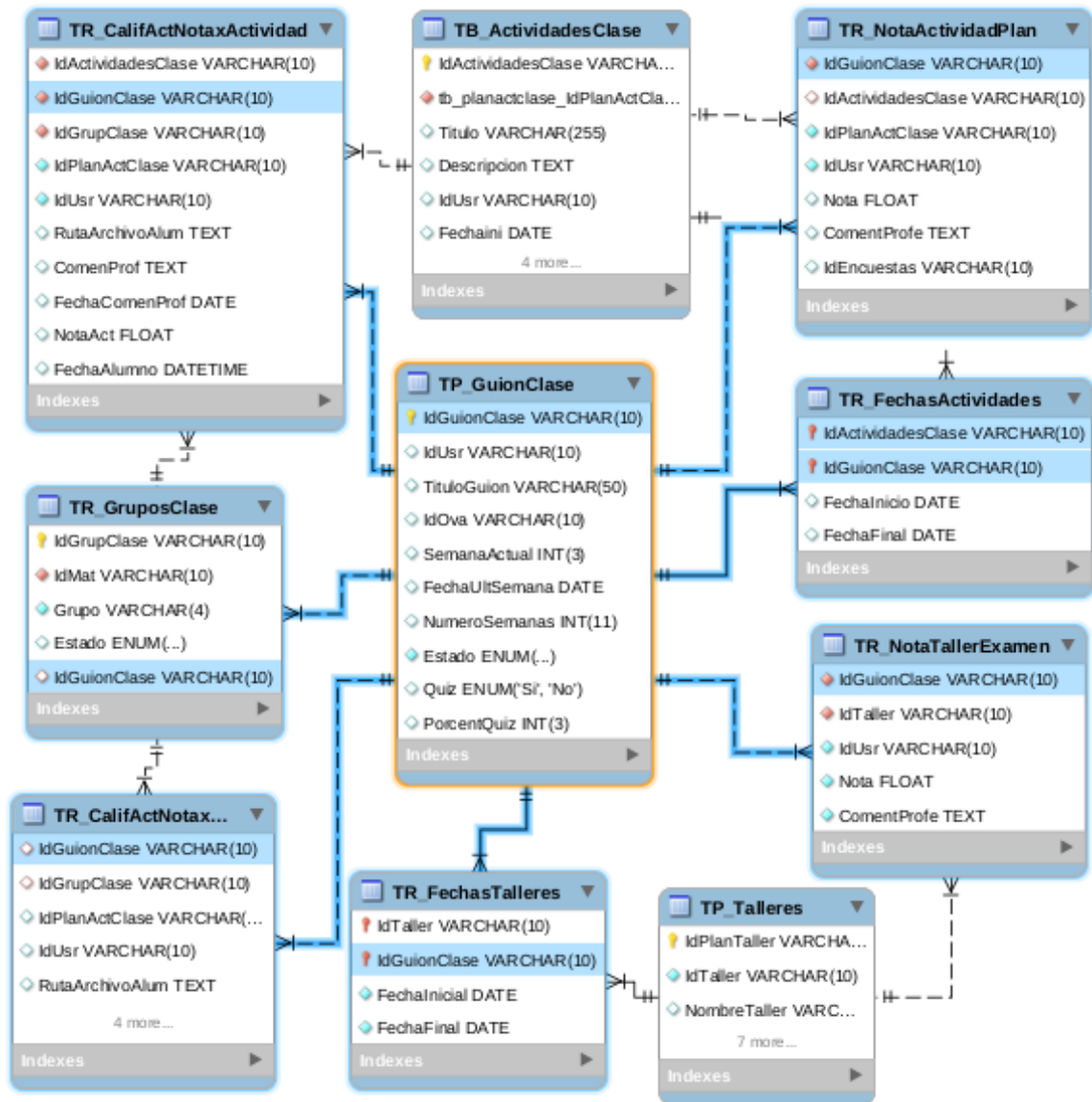


Figura 20. Diagrama E/R: Limpiar subgrupos

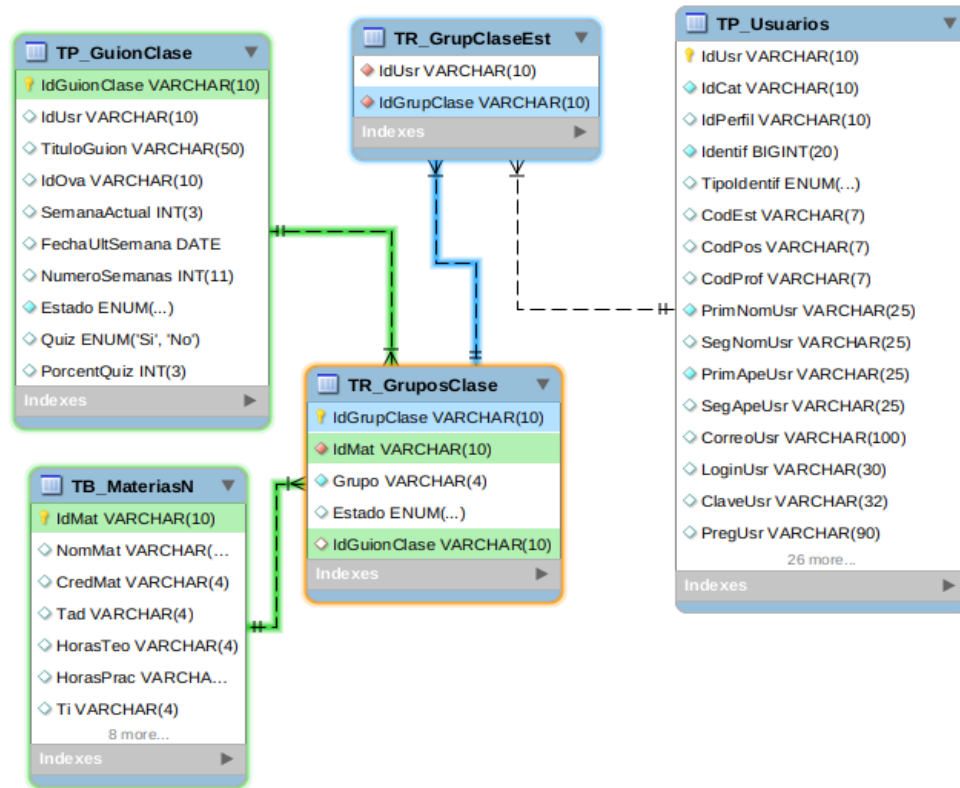
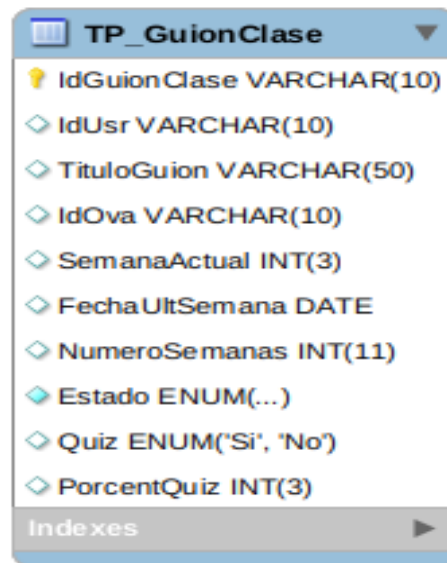
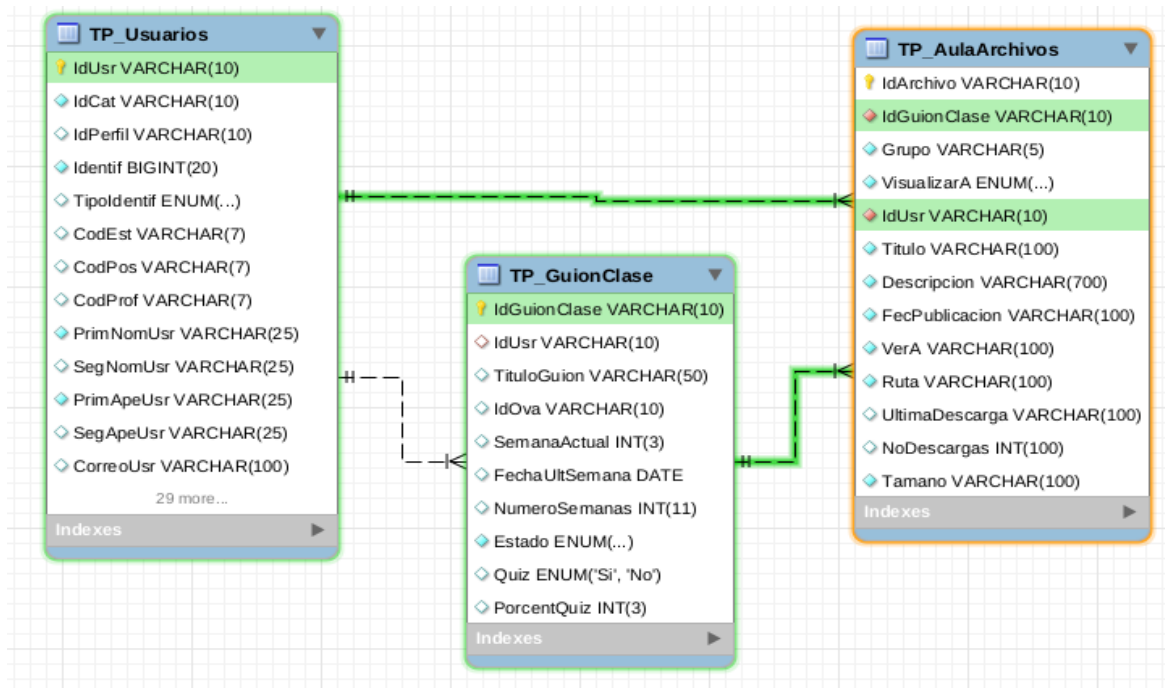


Figura 21. Diagrama E/R: Reiniciar aula



4.4.3 Actualizar el filtro que muestra los archivos compartidos en el aula Virtual.

Figura 22. Diagrama E/R: Archivos compartidos



4.4.4 Descargar en formato PEP la información de cada asignatura del plan de estudio.

Figura 23. Diagrama E/R: Gestionar información de asignaturas



4.5 DESCRIPCIÓN DE LAS ENTIDADES

Descripción de las entidades creadas y utilizadas en las bases de datos Diamante ubicada en el servidor de cada portal y Poseidón ubicada en el servidor del portal web de la Vicerrectoría Académica.

4.5.1 Gestionar los menús de todos los portales de las escuelas desde el módulo del Administrador Web.

Tabla 5. Entidades: Gestionar menús desde una hacia todas las escuelas

ENTIDAD	DESCRIPCIÓN
TP_Escuelas	Almacena la información base de cada una de las plataformas de las escuelas, así como las de las distintas facultades.
TP_MenuGestion	Almacena el registro que con toda la información necesaria que el sistema necesitará para crear un nuevo menú en cada escuela
TR_MenuGestionEscuela	Relaciona el registro con la escuela en la que se va ejecutar la acción de este registro.

4.5.2 Actualizar el servicio de limpieza del Aula Virtual.

Tabla 6. Entidades: Limpieza del aula

ENTIDAD	DESCRIPCIÓN
TB_ActividadesClase	Almacena la información base de todas las actividades que se realizan en el aula

	virtual. En esta tabla no se incluyen los Talleres, Exámenes y Quices.
TB_CalificarMensajes	Almacena la calificación promedio que se le da a un mensaje dentro de un foro. En otras palabras, almacena el favoritismo de un mensaje.
TB_MateriasN	Contiene la información completa de las asignaturas.
TB_Realtimespaces	Almacena los tipos de espacios que existen para los chats (Escuela y AulaVirtual).
TP_AulaArchivos	Almacena la información de los archivos que son compartidos dentro del aula virtual.
TP_CarteleraAulaVirtual	Contiene la información de los mensajes que son enviados entre los diferentes miembros del Aula.
TP_Conversaciones	Almacena las conversaciones que son creadas en los diferentes foros que se manejan en el aula (Evaluables y no evaluables).
TP_GuionClase	Almacena la información que una asignatura tiene en su espacio virtual.
TP_RealtimespaceRooms	Almacena la relación entre los espacios existentes para los chats y las asignaturas. Esta relación son las salas para los chats que hay disponible para cada una de las aulas virtuales.
TP_Talleres	Almacena la información de los talleres, exámenes y quices que siempre están disponibles para las determinadas aulas.

TP_Usuarios	Contiene toda la información de todos los usuarios registrados en el portal.
TR_AulaCalendario	Contiene la información de todos los eventos creados en el aula.
TR_CalifActNotaxActividad	Contiene la información que relaciona a los subgrupos de trabajo y las actividades; como lo son los archivos subidos por el alumno y las apreciaciones del profesor.
TR_CalifActNotaxPlan	Contiene la información que relaciona a los estudiantes y los planes de actividades; como lo son los archivos subidos por el alumno y las apreciaciones del profesor.
TR_CalifMensaProf	Guarda la calificación que el profesor asigna a un mensaje en las actividades de foros evaluables.
TR_ConversacionesGuion	Relaciona el Guion, el aula virtual, con las conversaciones creadas por los usuarios de dicha aula.
TR_FechasActividades	Almacena todas las fechas en las que son programadas las distintas actividades de clase excepto la de los talleres, exámenes y quices.
TR_FechasTalleres	Almacena todas las fechas en las que son programados los talleres, exámenes y quices.
TR_GrupClaseEst	Relaciona los estudiantes que han pertenecido o pertenecen a los subgrupos de trabajo.
TR_GruposClase	Relaciona los subgrupos de trabajo con una

	asignatura y su determinado grupo, también relaciona al aula virtual que conforman.
TR_Mensajes	Contiene la información de los mensajes que son enviados por los usuarios dentro de las conversaciones que conforman un foro.
TR_NotaActividadPlan	Contiene las calificaciones individuales obtenidas de los estudiantes en las distintas actividades excepto las de los talleres, exámenes y quices.
TR_NotasVistas	Registra si los mensajes que se le envían directo a los usuarios han sido vistos por este.
TR_NotaTallerExamen	Almacena las calificaciones que los estudiantes obtienen en los talleres, exámenes y quices
TR_RealttimeRoomMessages	Almacena la relación entre el usuario y los mensajes enviados a una determinada sala de chat.
TR_RealttimeRoomUsers	Relaciona la sala del chat y a los usuarios participantes.
TR_SesionRegistro	Almacena todos los datos de los estudiantes agregados a un aula virtual de manera manual.
TR_SesionRegistroEstudiante	Almacena la relación de los estudiantes agregados manualmente con la tabla de usuario del portal.

4.5.3 Actualizar el filtro que muestra los archivos compartidos en el aula Virtual

Tabla 7. Entidades: Archivos compartidos

ENTIDAD	DESCRIPCIÓN
TP_AulaArchivos	Tiene la información completa de los archivos que son compartidos dentro del aula virtual de clase.
TP_GuionClase	Almacena toda la información de una Aula Virtual (Una Asignatura).
TP_Usuarios	Almacena la información base de todos los usuarios.

4.5.4 Descargar en formato PEP la información de cada asignatura del plan de estudio.

Tabla 8. Entidades: Información de asignaturas

ENTIDAD	DESCRIPCIÓN
TB_MateriasN	Contiene toda la información de cada asignatura según el PEP

4.6 MODELO DE PROCESOS DEL SISTEMA

Para una mejor interpretación de los modelos de los procesos del sistema se realizaron los diagramas de secuencia necesarios para cada caso de uso en los que se explica con detalle los pasos para el funcionamiento de cada uno de los servicios (**Ver Anexo A**).

4.7 IMPLEMENTACIÓN, IMPLANTACIÓN Y PRUEBAS GENERALES

Para la implementación de los servicios se utilizaron las siguientes herramientas:

- Lenguaje de marcado para la elaboración de páginas web, HTML.
- Lenguaje de programación orientado a la web, JSP.
- Lenguaje de programación interpretado orientado a objetos, JavaScript.
- Hojas de estilo en cascada, CSS.
- Lenguaje Java.
- NetBeans, IDE para desarrollar las clases de Java y los archivos JSP.
- Servidor Jakarta Tomcat.
- Manejador de Base de datos, MySQL 5.0.

Haciendo uso de las anteriores herramientas, y con la asesoría y seguimiento del director de proyecto se diseñaron los nuevos servicios y se estructuraron los datos para el primer prototipo. También se tomaron en cuenta sugerencias recibidas por miembros del Grupo Calumet, para realizar un posterior refinamiento a las interfaces.

Para el desarrollo se trabajó con la base de datos "Diamante" ubicada en todos los servidores de los portales de las escuelas, y con la base de datos centralizada "Poseidón" ubicada en el servidor del portal web de la Vicerrectoría Académica. En las cuales se crearon las tablas necesarias para el funcionamiento de los servicios. Se llevaron a cabo pruebas para cada subsistema propuesto, verificando que el resultado correspondiera con lo esperado, de esta forma, se evidenció el correcto funcionamiento en la captura de datos, selección de ítems y almacenamiento de información.

La implantación de los servicios se realizó en primer lugar en el portal web de pruebas del Grupo Calumet, donde se realizan las pruebas a todos los componentes. Finalmente se implementan en todos los portales web a los cuales el Grupo Calumet presta soporte.

4.8 MANTENIMIENTO Y ADMINISTRACIÓN

4.8.1 Actividades de mantenimiento. Dentro de las funciones que se realizan en la administración y mantenimiento de los servicios de los portales web se encuentra la tarea de corregir errores, las cuales se denominan incidencias, que se corrigen durante la primera fase como integrantes del Grupo Calumet.

4.8.2 Actividades de soporte a usuarios. Las escuelas cuentan en sus portales con un servicio de consultas y sugerencias, a través del cual se responden preguntas y se resuelven problemas de los usuarios. Diariamente se reciben consultas de estudiantes y profesores, que necesitan asesoría en el funcionamiento de servicios o soporte en el manejo de su usuario y contraseña. También se resuelven problemas de manera presencial, donde el usuario acude directamente a las oficinas del Grupo Calumet y es atendido directamente por alguno de sus miembros, que le prestan la asesoría necesaria. Algunas de las labores realizadas comúnmente son:

- Restablecimiento de contraseña a usuarios.
- Dar respuesta y solución a las inquietudes y problemas que tengan los usuarios de los portales respecto al uso de los servicios.

- Facilitar orientación a los estudiantes nuevos de las escuelas a las cuales presta sus servicios el grupo Calumet, en cuanto al registro en el portal y el uso de sus principales servicios.

4.8.3 Actividades de administración. Dentro de las actividades realizadas por los miembros del Grupo Calumet, se encuentra la tarea de administrar los servidores de los portales de las escuelas, cada semestre se asignan nuevos administradores, los cuales cuentan con un usuario dentro de uno de los portales a los cuales presta soporte el grupo. Como administrador las tareas que se realizan comúnmente son:

- Realizar periódicamente copias de las bases de datos Diamante y División.
- Dar aval a las solicitudes de publicación de los usuarios en la cartelera para que puedan ser accedidas por la comunidad.
- Atender las sugerencias hechas por los usuarios del sistema a través del servicio de consultas y sugerencias.
- Actualizar periódicamente las bases de datos con respecto a la información que ofrece la División de Servicios de Información de la Universidad Industrial de Santander, para que el portal cuente con información actualizada.

5. PRUEBAS DEL SISTEMA

Para garantizar el correcto desarrollo de los nuevos servicios creados y a los que se les hizo reingeniería, se realizaron las siguientes pruebas:

5.1 PRUEBAS DE VERIFICACIÓN

Esta prueba es una de las más utilizadas en desarrollo de software mediante esta se aplican diferentes técnicas para detectar errores en el sistema antes de ser utilizado.

Se efectúa ejecutando paso a paso el proceso del servicio de manera que se explora cada funcionalidad que tiene el módulo desarrollado, realizando verificaciones de validación, los campos que son obligatorios no pueden quedar vacíos, por ejemplo.

Las siguientes tablas describen las pruebas de cada caso de uso de los servicios que fueron desarrollados.

5.1.1 Pruebas por componentes. Esta prueba se realizó para cada caso de uso de cada servicio desarrollado y descrito anteriormente:

5.1.1.1 Gestionar los menús de todos los portales de las escuelas desde el módulo del Administrador Web.

Tabla 9. Pruebas realizadas: Gestionar menús desde una hacia todas las escuelas

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Crear Acción	Ingresar como administrador en Administrar Servicios, allí seleccionar	

	Gestionar Servicio, allí llena los campos y elige la acción que va a realizar el registro, seleccionando también las escuelas y los permisos si corresponde.
Modificar Acción	En la sección de Distribuir Servicio podrá seleccionar y/o escoger un registro ya creado y modificarlo en su totalidad.
Eliminar Acción	En esta misma sección, podrá eliminar un registro independientemente si ya fue o no distribuido en las demás escuelas.
Ejecutar Acción	El registro lo ejecuta el sistema según la acción en un determinado momento.

5.1.1.2 Actualizar el servicio de limpieza del Aula Virtual.

Tabla 10. Pruebas realizadas: limpieza del aula.

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Limpiar Aula	El director y/o el profesor en su espacio del aula virtual efectúa la limpieza del aula.	
Limpiar Actividades	El sistema limpia todas las actividades; sus fechas, notas y relación con estudiantes.	
Limpiar Agenda	Limpia de la agenda todos los eventos creados por estudiantes.	
Limpiar Mensajes en el Aula	Elimina todos los mensajes creados por estudiantes y limpia todas las participaciones en los del profesor.	
Limpiar Archivos	Elimina todo archivo subido por estudiantes.	

Limpiar Chat	Limpiar el chat del aula eliminando todas las participaciones.
Limpiar Estudiantes Incluidos	Elimina los registros de los estudiantes agregados al aula manualmente.
Limpiar Foros	Elimina todas las conversaciones creadas por los estudiantes y la participación de estos en los foros y conversaciones creadas por el profesor.
Borrar subgrupos	Elimina todos los subgrupos de trabajos que se hubiesen creado en el aula.
Reiniciar Aula	Reinicia los parámetros del aula virtual.

5.1.1.3 Actualizar el filtro que muestra los archivos compartidos en el aula virtual.

Tabla 11. Pruebas realizadas: Archivos compartidos

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Filtrar archivos	Dentro del aula virtual se comparte y se muestra el archivo solo al o a los usuarios que se le indicó al momento de publicarlo.	

5.1.1.4 Descargar en formato PEP la información de cada asignatura del plan de estudio.

Tabla 12. Pruebas realizadas: Información de asignaturas.

CASO DE USO	PRUEBA REALIZADA	RESULTADO
-------------	------------------	-----------

Editar Información	Modificar o actualizar la información de las asignaturas que conforman los planes de estudios.
Descargar	Entrando desde la página principal y entrando en la sección de pregrado todo usuario podrá descargar la información de cada asignatura que compone los planes de estudios

6. CONCLUSIONES

- Se efectuó la actualización del servicio de limpieza del aula virtual, que consiste en una nueva vista con información más explicativa para que el director o docente tenga plena certeza de cómo funciona este servicio y en cómo afectará al aula. Por otro lado, el sistema le informa mediante mensajes tipo de notificaciones el estado de cada sección del aula al usuario que ejecute la limpieza.
- En este mismo módulo, Aula virtual, se llevó acabo la actualización y corrección de la parte del sistema en cargada de filtrar los archivos que se comparten dentro del aula. Con esta implementación los miembros de este espacio académico tendrán la seguridad de que sus archivos compartidos solo se mostrarán con los usuarios que él haya elegido para hacerlo.
- Se efectuó la implementación del nuevo servicio en donde el super administrador de los portales de escuelas podrá gestionar los menús de todas las escuelas. Lo que en otras palabras significa que el administrador web podrá crear, modificar eliminar y cambiar de estado cualquier servicio o menús que en todas las escuelas.
- Por otro lado, también podrá modificar los permisos de cada servicio o menú que ya exista en las escuelas. Este nuevo servicio es de gran relevancia, ya que facilita en gran medida uno de los objetivos del grupo Calumet como administrador de los portales de las escuelas; la pronta solución a las necesidades que presentan los usuarios de los por tales. Pues ya no es necesarios ir escuela por escuela, teniendo en cuenta que cada portal es un servidor distinto, creando un menú o servicio que se requiere en todo o algunos portales. Por lo anterior, el administrador puede elegir en que escuelas

gestionar dichos menús o servicios.

- Se efectuó acabo el desarrollo para poder ofrecer la opción de descargar la información de las asignaturas pertenecientes a un plan de estudio. Con esta implementación se facilita la gestión padecían los estudiantes de posgrados a la hora de matricular una asignatura o los administrativos a al momento en que cualquier miembro de la comunidad solicitara dicha información, pues no era posible acceder a esta información de manera rápida y que además cumpliera con el formato especificado en el proyecto educativo del programa, PEP.

7. RECOMENDACIONES

Solicitar a los administrativos la importancia de actualizar toda la información de las asignaturas según el PEP de cada carrera, ya que esto permitirá la óptima distribución y disponibilidad de esta información para la comunidad que la requiera.

Teniendo en cuenta lo anterior, se hace importante socializar con los administrativos de las escuelas acerca de la nueva opción con la que cuentan los portales de las mismas, en donde cualquier persona podrá visualizar y descargar la información completa de cada asignatura que conforma los planes de estudio.

Con la implementación del nuevo servicio Distribuidor de menús, los administradores de los portales web deben tener plena certeza de que registro van a crear sin importar la acción que elijan, ya que una mala decisión podría afectar la integridad y funcionalidad de los portales de las escuelas. Este nuevo servicio desafortunadamente también cuenta con ciertas limitaciones. Para crear o modificar un servicio en las demás escuelas se requiere conocer al menos la url de algún servicio vecino, padre o hijo del servicio que queremos modificar o crear. En la sección de limpieza del aula, se llevó acabo el desarrollo de la limpieza de aula de tal forma que el sistema haga por el momento una limpieza total y automática evitando esto que el usuario interfiera con la limpieza. Aunque cabe mencionar que se dejó establecido las bases un prototipado para que la limpieza se puede llevar por secciones y no solo de manera completa y automática.

BIBLIOGRAFÍA

CALUMET ESTANDAR. “Estandarización de los proyectos COMA y RealTime” [En línea] [26 junio de 2021] disponible en: <http://cormoran.uis.edu.co/eisi/Calumet/Estandar/>.

CODIGO XULES. “java itextpdf – creando un pdf en java con itext” [En línea] [12 mayo 2021] disponibles en: <https://codigoxules.org/java-itext-pdf-creando-pdf-java-itext/>.

FOSSATI, Matias. Todo sobre MySQL: Libro ideal para ingresar en el mundo de la base de datos MySQL. Natsys, 2014.

MANUALES. Manuales básicos de JSP. [En línea] Disponible en: <http://www.desarrolloweb.com/manuales/73/>.

PIATTINI VELTHUIS, Mario, et al. Análisis y diseño detallado de Aplicaciones Informáticas de Gestión. Madrid: RA-MA S.A. Editorial y Publicaciones, 2007, 480. ISBN-10: 8478977767

SCHMULLER, Joseph. Aprendiendo UML en 24 horas. México: Alhambra Mexicana S.A, 2000, 387.

SOMEBOOKS. “Arquitectura Cliente Servidor”. [En línea] [20 agosto 2021] disponible en: <http://somebooks.es/arquitectura-clienteservidor/>>.

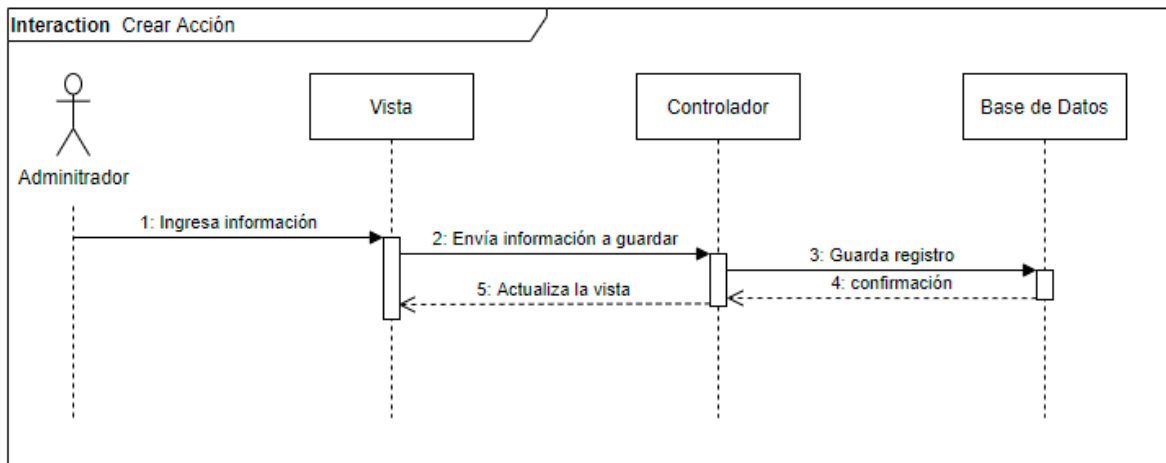
WEITZENFELD, Alfredo. Ingeniería de Software Orientada a Objetos con UML, JAVA e Internet. México: Thomson International, 2003, 678.

ANEXOS

Anexo A. Modelo de procesos de sistemas

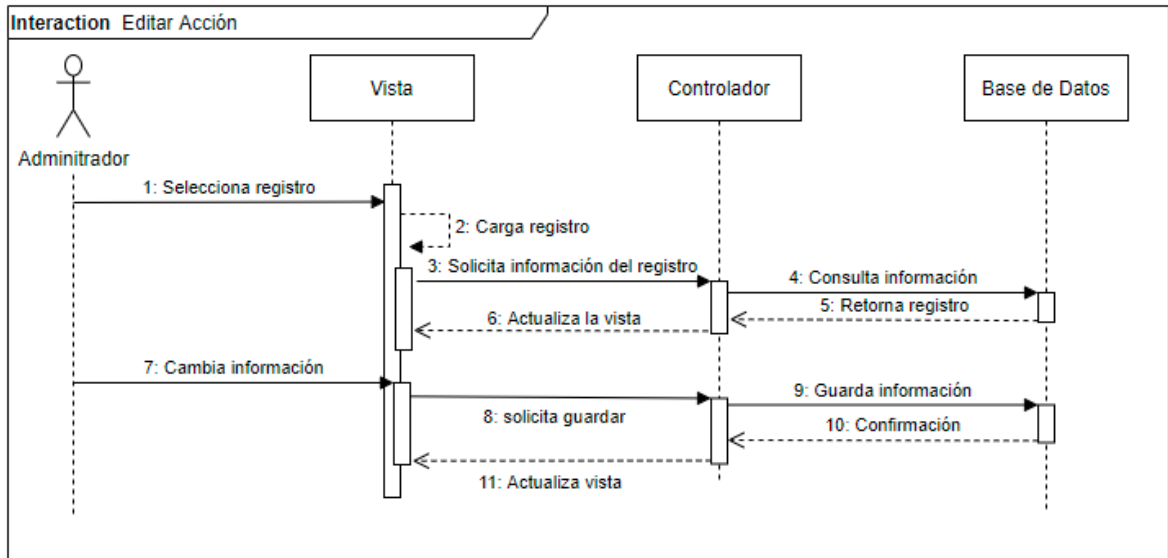
Módulo de gestionar servicios de todos los portales de las escuelas.

Figura 24. Diagrama de secuencia: Crear acción.



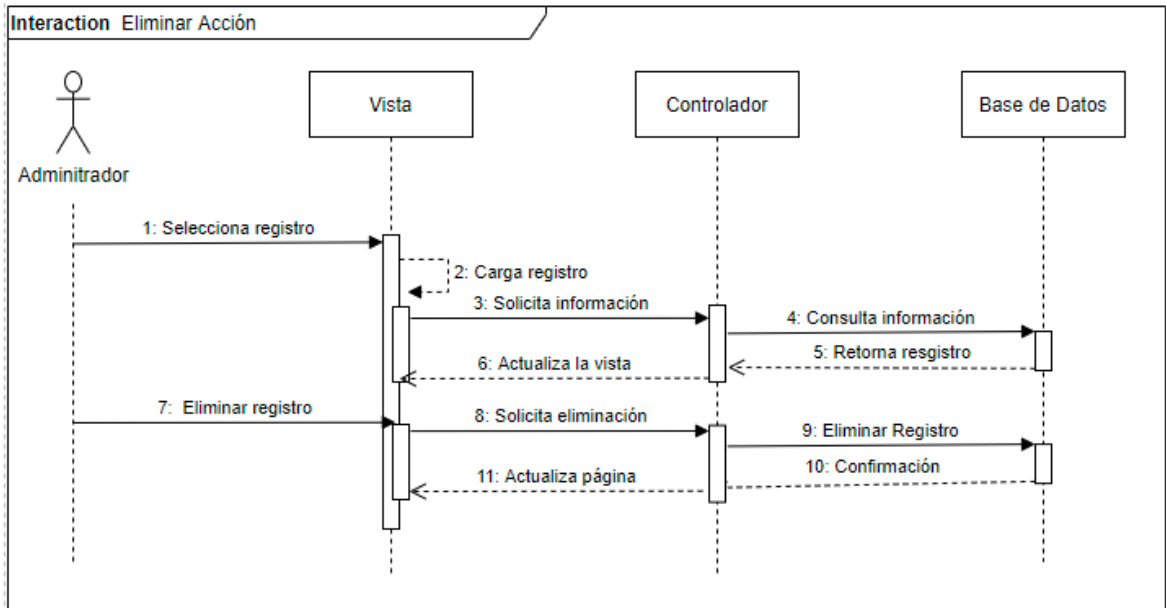
1. El administrador ingresa la información requerida según la acción a crear.
2. La vista solicita al controlador guardar la información.
3. El controlador crea la conexión con la base de datos para guardar el registro.
4. La base de datos guarda el registro y retorna la confirmación.
5. El controlador envía a la vista la confirmación y esta actualiza mostrando el registro que se acaba de crear.

Figura 25. Diagrama de secuencia: Editar acción



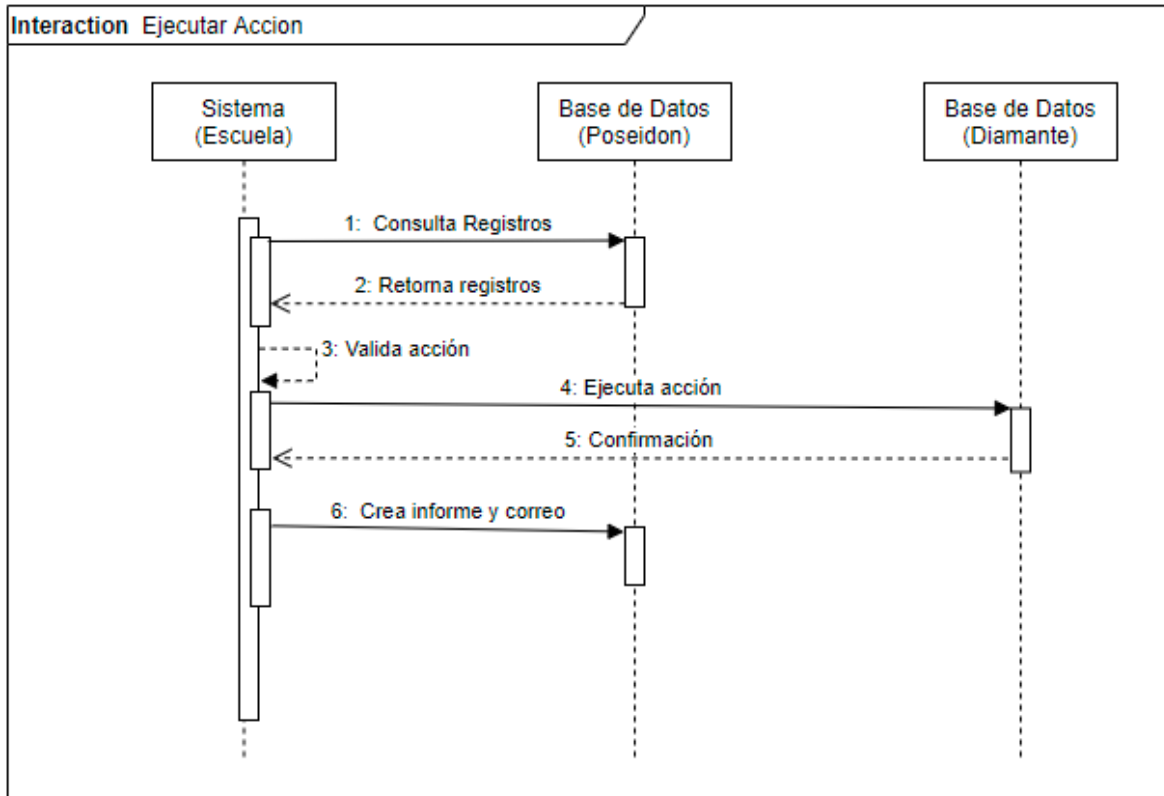
1. El administrador selecciona un registro ya creado con una acción.
2. La vista se recarga para mostrar la información del registros seleccionado.
3. La vista solicita al controlador la información de registro.
4. El controlador consulta a la base de datos la información de dicho registro.
5. La base de datos retorna la información de registro seleccionado.
6. La vista recibe la información del registro que envía el controlador y la muestra.
7. El administrador cambia la información de dicho registro.
8. La vista solicita al controlador guardar la nueva información ingresada.
9. La base de datos guarda la información del registro seleccionado.
10. La base de datos retorna la confirmación.
11. El controlador retorna a la vista la confirmación para que esta se actualice mostrándole al usuario el registro con la nueva información ingresada.

Figura 26. Diagrama de secuencia: Eliminar acción.



1. El administrado selecciona un registro ya creado.
2. La vista se llama a sí misma para mostrar la información del registro seleccionado.
3. La vista solicita la información al controlador.
4. El controlador consulta a la base de datos la información del registro.
5. La base de datos retorna toda la información que conforma el registro.
6. La vista muestra toda la información del registro seleccionado.
7. El administrador solicita eliminar el registro que había seleccionado.
8. La vista solicita al controlador la eliminación del registro.
9. La base de datos efectúa la eliminación del registro.
10. La bases de datos retorna la confirmación.
11. La vista al recibir la confirmación desde el controlador recarga y deja la vista lista para crear un nuevo registro.

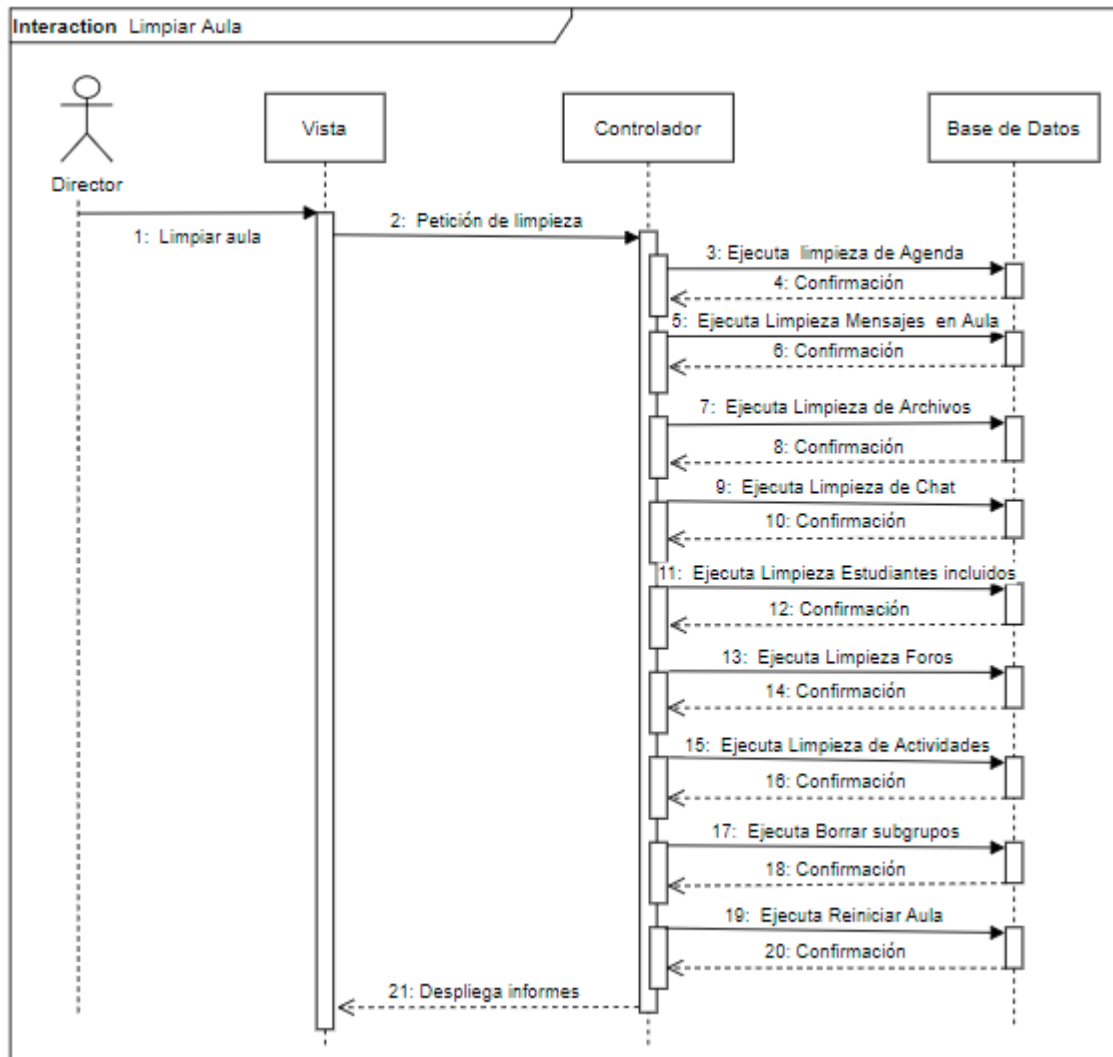
Figura 27. Diagrama de secuencia: Ejecutar acción.



1. El sistema (cada escuela) en un determinado momento consulta a la base de datos Poseidon los registros para la escuela.
2. La BD le retorna al sistema todos los registros disponible para dicha escuela.
3. El sistema efectúa la validación de la acción con la que se haya creado el registro.
4. El sistema en la base de datos Diamante la acción que tenga dicho registro.
5. La base de datos de diamante retorna la confirmación de la acción ejecutada.
6. El sistema actualiza el estado del registro en poseidon según su ejecución de la acción en la base de datos de diamante y crea un informe programado para ser enviado por correo al usuarios creador del reistro.

Servicio de limpieza del aula virtual.

Figura 28. Diagrama de secuencia: Limpiar aula.



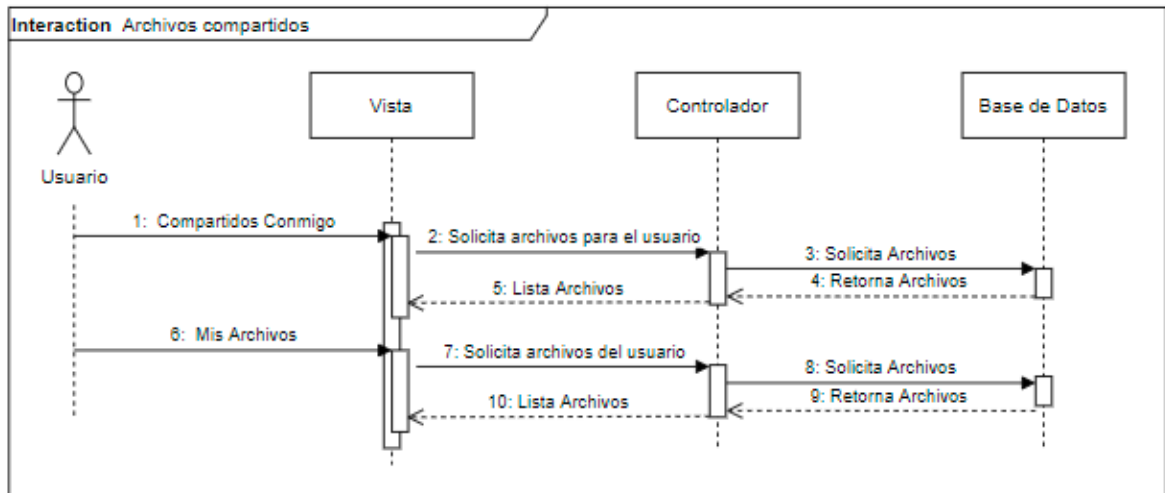
1. El director del aula solicita limpiar el aula virtual.
2. La vista remite dicha solicitud al controlador para que ese efectúe la limpieza de las distintas sección del aula.
3. El controlador ejecuta la limpieza la agenda del aula.
4. La base de datos retorna la confirmación para que el controlador arme el informe de limpieza.
5. El controlador ejecúa la conexión para que la base de datos lleve acabo la limpieza de los mensajes internos del aula.

6. El controlador recibe la confirmación y continua armando el informe.
7. El controlador hace el llamado para la limpieza de los archivos que se comparten en el aula entre los distintos usuarios.
8. La base de datos retorna la confirmación.
9. El controlador sigue armando el informe mientras efectúa la limpieza del chat del aula.
10. La base retorna la confirmación de haber ejecutado las sentencia de la limpieza del cha.
11. El controlador solicita a la base eliminar los registros de los estudiantes incluido manualmente al aula de clase.
12. La base de datos realiza la confirmación de la limpieza.
13. El controlador hace el llamado a las clase que se conectan con la base de datos para efectuar la limpieza de los foros del aula.
14. El controlador recibe la connformación de la limpieza realizada y la sigue agregando al informe.
15. Para llevar acabo la limpieza de las actividades realizada en el aula virtual el controlador efectúa la conexión a base de datos para limpiar esta información.
16. La base de datos retorna la confirmación de la limpieza realizada.
17. El controlador hace el llamado a las clases necesarias para efectuar la eliminación de los subgrupos de clase.
18. En controlador recibe la confirmación de la limpieza realizada en la base de datos y la agrega al ainforme.
19. Por ultimo, el controlador efectua el reinicio del aula virtual. El cual consiste en cambiar o reinicar los parametros de esta para que entre en un estado detenido.
20. La base de datos retorna la confirmación de la actualización de los parametros del aula.

21. El controlador despues de haber recibido todos los informes los envía a la vista para que el usuario mire el estado de la limpieza que se ejecutó en el aula.

Servicio de archivos compartidos en el aula virtual.

Figura 29. Diagrama de secuencia: Archivos compartidos.

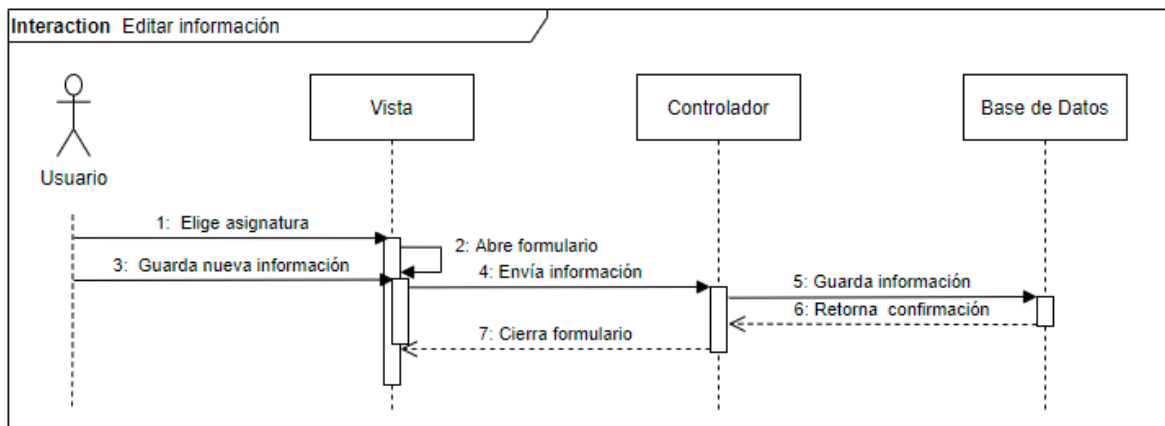


1. El usuario del aula virtual solicita ver los archivos compartidos él.
2. La vista solicita al controlador la lista de archivos compartidos con el usuario logeado.
3. El controlador consulta a la base de datos la lista de archivos.
4. La base de datos retorna la lista de archivos para el usuario.
5. La vista recibe la lista de archivos compartidos con el usuario logeado y la muestra.
6. El usuario solicita ver los archivos que el ha compartido con otros usuarios del aula.
7. La vista solicita al controlador los archivos que el usuario logeado ha compartido dentro del aula virtual.
8. El controlador consulta a base de datos los archivos solicitados desde la vista.
9. La base de datos retorna los archivos compartidos por el usuario.

10. La vista recibe la respuesta de la solicitud hecha al controlador y lista los archivos del usuario.

Servicio de gestionar asignaturas del plan de estudio

Figura 30. Diagrama de secuencia: Editar información.



1. El usuario administrativo elige la asignatura a la cual le desea agregar o editar la información.
2. La vista despliega el formulario con dichos campos a modificar.
3. El usuario solicita guardar la información de la asignatura.
4. La vista hace el envío de la información al controlador.
5. El controlador solicita a la base de datos guardar la información de la asignatura.
6. La base de datos retorna la confirmación de la información guardada.
7. La vista cierra formulario al recibir la confirmación del controlador dejando ver las demás asignaturas al usuario.