

**ADMINISTRACIÓN, SOPORTE A USUARIOS, MANTENIMIENTO DEL PORTAL
WEB, ANÁLISIS, DESARROLLO E IMPLEMENTACIÓN DE NUEVOS
SERVICIOS PARA EL PORTAL WEB DE LA ESCUELA DE DISEÑO
INDUSTRIAL Y LA DECANATURA DE INGENIERÍAS FISCOQUÍMICAS**

NESTOR IVAN RODRIGUEZ GOMEZ

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECAÓNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2015

**ADMINISTRACIÓN, SOPORTE A USUARIOS, MANTENIMIENTO DEL PORTAL
WEB, ANÁLISIS, DESARROLLO E IMPLEMENTACIÓN DE NUEVOS
SERVICIOS PARA EL PORTAL WEB DE LA ESCUELA DE DISEÑO
INDUSTRIAL Y LA DECANATURA DE INGENIERÍAS FISICOQUÍMICAS**

NESTOR IVAN RODRIGUEZ GOMEZ

Trabajo de grado para optar al título de Ingeniero de Sistemas

Director

Msc. LUIS IGNACIO GONZÁLEZ RAMÍREZ

Magíster en Informática

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE INGENIERÍAS FISICOMECAÑICAS

ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

BUCARAMANGA

2015

TABLA DE CONTENIDO

INTRODUCCIÓN

1 DESCRIPCIÓN DEL PROYECTO.....	16
1.1 INFORMACION DE LA SITUACIÓN ACTUAL.....	16
1.2 JUSTIFICACIÓN.....	17
1.3 OBJETIVOS.....	17
1.3.1 Objetivo general.....	17
1.3.2 Objetivos específicos.....	18
2 MARCO TEORICO.....	20
2.1 ARQUITECTURA CLIENTE/SERVIDOR.....	20
2.2 NETBEANS.....	21
2.2.1Definision.....	21
2.2.2 Tecnologías que soporta.....	21
2.2.3 Características.....	22
2.2.4 Puntos fuertes.....	22
2.2.5 Puntos débiles.....	22
2.3 BASES DE DATOS.....	23
2.3.1 Definición.....	23
2.3.2 Características.....	23
2.3.3 Sistemas de gestión de bases de datos.....	23
2.3.4 Ventajas de bases de datos.....	23
2.3.5 Desventajas de bases de datos.....	24
2.3.6 Sistemas de gestión de bases de datos utilizado MySQL	24
2.4 TECNOLOGIAS DE DESARROLLO DE PAGINAS WEB DINAMICAS....	25
2.4.1Páginas web dinámicas.....	25
2.4.2 Ventajas de páginas web dinámicas.....	25

2.4.3 Tipos de tecnologías para desarrollo de páginas web dinámicas.....	25
2.4.4 Tecnología aplicada para desarrollar el proyecto.....	26
2.5 SISTEMA DE CONTROL DE VERSIONES.....	27
2.5.1 Definición.....	27
2.5.2 Características.....	27
2.5.3 Tipos de sistemas de control de versiones.....	28
2.5.4 Sistema de control de versiones aplicado en el proyecto.....	28
2.6 SERVIDORES WEB.....	29
2.6.1 Definición.....	29
2.6.2 Tipos de servidores.....	30
2.6.3 Servidores web utilizado en el desarrollo del proyecto.....	31
2.7 PROGRAMACION UTILIZADA.....	31
2.7.1 Definición de la programación orientada a objetos.....	31
2.7.2 Conceptos fundamentales.....	31
2.7.3 Características.....	32
2.7.4 Lenguajes orientados a POO.....	33
2.7.5 Lenguaje utilizado para el desarrollo del proyecto JAVA.....	33
3 MARCO METODOLOGICO.....	34
3.1 PROTOTIPO EVOLUTIVO.....	34
3.2 LENGUAJE DE MODELADO UNIFICADO.....	36
3.2.1 Tipos de diagramas UML.....	37
3.2.2 Diagramas de casos de uso.....	37
3.2.3 Diagramas de secuencia.....	38
3.3 ESTANDARES DE PROGRAMACION.....	39
3.3.1 Modelo de datos.....	39
3.3.2 Nombres de las tablas.....	39
3.3.3 Pagina JSP.....	40
3.3.4 Clases.....	41

3.3.5 Organización de directorios.....	41
4. DESARROLLO DE LA HERRAMIENTA, ADMINISTRACION Y MANTENIMIENTO.....	42
4.1 PROTOTIPO ESPERADO.....	42
4.1.1 Diagramas de casos de uso.....	49
4.1.1.1 Subir plan PDF.....	49
4.1.1.2 Imprimir nota.....	53
4.1.1.3 Mantenimiento y administración de los portales EdiWEB y FIFQuiWEB.....	54
4.1.2 Descripción de los casos de uso.....	55
4.1.2.1 Subir plan PDF.....	55
4.1.2.2 Imprimir nota.....	59
4.1.2.3 Mantenimiento y administración de portales EdiWEB y FIFQuiWEB.	60
4.1.3 Diseño y análisis.....	62
4.1.3.1 Diagrama entidad/relación.....	62
4.1.3.2 Descripción de las entidades.....	68
4.1.3.3 Modelo de procesos de sistema.....	71
4.1.3.4 Implementación, implantación y pruebas generales.....	76
4.2 SOPORTE Y ADMINISTRACION.....	78
4.2.1 Actividades de soporte o usuarios.....	78
4.2.2 Actividades de administración.....	78
5 PRUEBAS DEL SISTEMA.....	80
5.1 PRUEBAS DE VERIFICACION.....	80
5.1.1 Pruebas por componente.....	80
5.2 PRUEBAS DE INTEGRACION.....	81
5.3 PRUEBAS DE VALIDACION.....	81
6 CONCLUSIONES.....	82
7 RECOMENDACIONES.....	83
BIBLIOGRAFIA.....	84
ANEXOS.....	86

LISTA DE FIGURAS

	Pág.
Figura 1. Modelo Cliente/servidor	20
Figura 2. Esquema Cliente-Servidor	30
Figura 3. Prototipo Evolutivo	34
Figura 4. Diagramas de casos de uso: Subir Plan	49
Figura 5. Diagramas de casos de uso: Solicitar Evaluador.....	50
Figura 6. Diagramas de casos de uso: Concepto Plan.	50
Figura 7. Diagramas de casos de uso: Solicitar Calificador.....	51
Figura 8. Diagramas de casos de uso: Asignar Evaluador.	51
Figura 9. Diagramas de casos de uso: Asignar Calificadores.....	52
Figura 10. Diagramas de casos de uso: Evaluar Plan.	52
Figura 11. Diagramas de casos de uso: Buscador Imprimir Nota.....	53
Figura 12. Diagramas de casos de uso: Imprimir Nota.	53
Figura 13. Diagrama de Casos de Uso: Subsistema de Mantenimiento y Administración.	54
Figura 14. Diagrama Entidad/Relación: Subir Plan Pdf	62
Figura 15. Diagrama Entidad/Relación: Imprimir Nota.....	63
Figura 16. Diagrama Entidad/Relación: No mostrar Nota	64
Figura 17. Diagrama de secuencias: Subir Plan	71
Figura 18. Diagrama de secuencias: Imprimir Nota.	72
Figura 19. Diagrama de secuencias: Buscador Imprimir Nota.....	73
Figura 20. Diagrama de secuencias: No mostrar decisiones dia actual.....	75

LISTA DE TABLAS

	Pág.
Tabla 1. Casos de Uso: Subir Plan.....	55
Tabla 2. Casos de Uso: Solicitar Evaluador.....	55
Tabla 3. Casos de uso: Concepto Plan.....	56
Tabla 4. Casos de uso: Solicitar Calificador.	56
Tabla 5. Casos de uso: Asignar Evaluador.....	57
Tabla 6. Casos de uso: Asignar Calificadores.	57
Tabla 7. Casos de uso: Evaluar Plan.....	58
Tabla 8. Casos de uso: Buscador Imprimir Nota.....	59
Tabla 9. Casos de uso: Imprimir Nota.....	59
Tabla 10. Casos de Uso: Subsistema de Mantenimiento y Administración.	60
Tabla 11. Descripción de las entidades..	68

LISTA DE ANEXOS

	Pág.
Anexo A. Arquitectura cliente/servidor	86

RESUMEN

TÍTULO: Administración, soporte a usuarios, mantenimiento del portal web, análisis, desarrollo e implementación de nuevos servicios para el portal web de la escuela de Diseño Industrial y la Decanatura de Ingenierías Físicoquímicas*.

AUTOR: Néstor Iván Rodríguez Gómez**.

PALABRAS CLAVE: Trabajos de grado, subir plan, consultas t.g., EdiWEB(Escuela de diseño industrial), FIFQuiWEB(Decanatura de fisicoquimicas).

DESCRIPCION:

En la actualidad la escuela de Diseño Industrial y facultad de Ingenierías Físicoquímicas, 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 las escuelas, 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. El Portal web de las escuelas soporta la gestión de los trabajos de grado, actualmente existe un servicio que permite ingresar la nota de los trabajos de grado, para complementar este servicio se agregó la posibilidad de imprimir la nota del trabajo de grado en el formato especificado por parte de la universidad. En el módulo de trabajos de grado se agregó la opción de subir el plan del proyecto en formato pdf por parte de los autores, este plan puede ser visto y descargado por parte del director del proyecto, los miembros del comité de trabajos de grado y los calificadores del proyecto, por consiguiente la revisión del plan y la toma de decisiones sobre este serán más cómoda. Se hizo correcciones en los siguientes servicios: Mi proyecto, Mis Proyectos Dirigidos, y Buscador

Privado, dicha corrección se realizó para que no se mostrara las decisiones del comité del día actual, debido a que las decisiones tomadas por parte del comité de trabajos de grado pueden ser modificadas en el transcurso del día, hasta las 11:59:59 del día en que se realizó la sesión.

*Trabajo de grado Modalidad Trabajo de Investigación.

**Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Luis Ignacio González Ramirez.

ABSTRACT

TITLE: Administration, users support, website maintenance, analysis, development and implementation of new services for the web portal of the school of industrial design and the faculty of physical chemical engineering*.

AUTHOR: Néstor Iván Rodríguez Gómez**

KEYWORDS: Jobs grade, up plan, t.g consultations, EdiWEB(school of Industrial Design), FIFQuiWEB(faculty of Physical chemical engineering).

DESCRIPTION:

At present the school of Industrial Design and faculty of Physical chemical engineering, they rely on an information system orientated to the Web that he takes charge of the administration and control of the different activities so much academic as administrative officers who carry out inside the schools, as well as of the users' control and services that provide them. The services of the Web portals of the schools must improve constant and adapt to the changes that they present in his environment, in turn they must give solution to the problems and needs that arise on the part of the users of the system to increase his time of useful life and not to manage to turn into an obsolete software, reason by which the labors of maintenance and update become indispensable.

The web Portal of the schools supports the management of the works of degree, nowadays there exists a service that allows depositing the note of the works of degree, to complement this service I add the possibility of stamping the note of the work of degree on the format specified on the part of the university. In the module of works of degree I add the option to raise the plan of the project in format pdf on the part of the authors, this plan can be seen and unloaded on the part of the director of the project, the members of the committee of works of degree and the censors of the project, consequently the review of the plan and the capture of decisions on this one will be more comfortable. Corrections were done in the following services: My project, My Directed Projects, and Private Seeker, the above

mentioned correction I realize in order that one was not showing the decisions of the committee of the current day, due to the fact that the decisions taken on the part of the committee of works of degree can be modified in the course of the day, until 11:59:59 of the day in which I realize the session.

*Degree Work: Research Paper.

**Faculty of Physical Mechanical Engineering, School of System Engineering and Computing Science. Director: M.Sc. Luis Ignacio González Ramírez

INTRODUCCION

El grupo de desarrollo de software calumet se encarga de desarrollar, administrar y dar soporte a los portales EdiWEB y FIFQuiWEB pertenecientes a la escuela de diseño industrial y la decanatura de físicoquímicas respectivamente, estos portales se han convertido en uno de los principales canales de comunicación e integración entre los miembros de la comunidad de cada escuela. Debido a la aceptación y utilización por parte de los usuarios surge el objetivo fundamental de fortalecer y mejorar los servicios que se ofrecen, además de la creación de nuevos servicios para suplir las necesidades crecientes de los miembros de cada escuela.

Para suplir las necesidades de la comunidad a los portales EdiWEB y FIFQuiWEB se les realizan continuos cambios y actualizaciones para hacer de estos una herramienta cada vez más útil, eficiente, agradable y de fácil uso. Con el objeto de llevar a cabo esta labor y proporcionar páginas con contenido dinámico se ha contado con herramientas software de libre distribución como JSP, Java, JavaScript, jQuery y MySQL. El soporte que se brinda a los procesos que se realizan en la Escuela y en la Facultad hace que 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 ajustan a las necesidades que puedan surgir. Como lo es la creación en el módulo de trabajos de grado de un servicio que permita subir el documento del plan del proyecto de grado por parte de los autores, para que este plan pueda ser visto por el director, evaluador, comité y calificadores de proyecto de grado, el ingreso de la nota de los proyectos de grado por parte de los administradores de los portales y hacer correcciones para que no sean visibles las decisiones que se toman en la sesión del comité de trabajos de grado en el día actual. En el presente documento se presenta el soporte teórico, metodológico y técnico del desarrollo web de los servicios previamente mencionados, así como de las labores de administración, mantenimiento y soporte a usuarios desempeñadas durante el desarrollo del proyecto.

1. DESCRIPCION DEL PROYECTO

1.1 INFORMACIÓN DE LA SITUACIÓN ACTUAL

El Portal web de las escuelas soporta la gestión de los trabajos de grado dentro de las funcionalidades está la posibilidad de ingresar la calificación final de un proyecto por parte del calificador, o bien, por parte del administrador del sistema, para complementar esta funcionalidad se debe permitir la impresión de la nota en el formato especificado inmediatamente cuando el administrador ingrese la nota, y además se podrá imprimir la nota desde un nuevo servicio que realizara la búsqueda para los trabajos de grado ya finalizados.

El módulo de trabajos de grado soporta la gestión de los proyectos por parte de autores, directores y comité de trabajos de grado por consiguiente para que sea más cómodo el proceso se debe agregar a trabajos de grado la posibilidad de subir el plan en pdf. Esta posibilidad se dará a los autores que tengan aprobado el tema, pero no el plan. Además los directores y el comité deben tener acceso al plan para que puedan tomar decisiones de este.

El módulo de trabajos de grado permite a los usuarios del sistema consultar toda la información de un trabajo de grado. Esto incluye la información del proyecto como tal, y las decisiones del Comité de Trabajos de Grado. Esta información es accesible desde diferentes servicios: Mi Proyecto, Mis Proyectos Dirigidos, Buscador Privado, y otros, para que la información mostrada sea la definitiva se deben corregir estos servicios para NO mostrar las decisiones del comité del día actual. Esto incluye el módulo de solicitudes de los autores. Lo anterior, debido a que el comité de trabajos de grado puede cambiar sus decisiones en el transcurso del día, y hasta las 11:59:59 pm, del día en que sesionó.

1.2 JUSTIFICACION

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.

En los portales de las escuelas se encuentra en módulo de trabajos de grado el cual contiene servicios para gestionar, visualizar y realizar diversas opciones con los trabajos de grado, para complementar este módulo se agregó un servicio para que los autores de los proyectos de grado puedan subir un documento en formato pdf con su plan de proyecto de grado, este documento puede ser visto por los directores, evaluador, comité y calificadores de proyecto de grado, se realizó una corrección para que no se puedan ver las decisiones que se toman por parte del comité que se encuentra en sesión en el día actual, estas decisiones son visibles después de las 11:59:59 pm del día en que se realizó la sesión en el comité, y se agregó la posibilidad de ingresar las notas a los proyectos de grado por parte de la persona administradora de los respectivos portales de las escuelas.

1.3 OBJETIVOS

1.3.1 Objetivo general Realizar las funciones de administración, soporte a usuarios, mantenimiento del portal web, análisis, desarrollo e implementación de nuevos servicios para el portal web de la escuela de Diseño Industrial y la Decanatura de Ingenierías Físicoquímicas para hacer óptimos, eficientes y sostenibles los trámites dentro de las escuelas y el acceso a la información de las mismas.

1.3.2 Objetivos específicos

- Efectuar labores de Administración del Portal web de la Escuela de Diseño Industrial y Facultad de Ingenierías Físicoquímicas:
 - Hacer seguimiento del uso que hacen los usuarios a los servicios de Foros, Eventos Propuestos, Cartelera, Archivos y Mi perfil dentro de los portales EdiWEB y FIFQuiWEB para detectar usos indebidos o incorrectos por parte de éstos.
 - Actualizar periódicamente las Bases de Datos con el objeto de mantener al día la información referente a matrículas, horarios, estados y categorías de los usuarios.
 - Atender las consultas y sugerencias que realicen los usuarios para proponerlas como mejoramiento y ofrecimiento de nuevos servicios en la próxima versión de los portales WEB.

- Llevar a cabo labores de mantenimiento a los portales EdiWEB y FIFQuiWEB, en los que podemos encontrar:
 - Hacer seguimiento del funcionamiento de los portales para corregir posibles defectos generados por errores en el código fuente que se puedan presentar.
 - Revisar y depurar la estructura de directorios y archivos de los portales Web.

- Administración, soporte a usuarios, mantenimiento del portal web, análisis, desarrollo e implementación de nuevos servicios tales como:
 - Crear un servicio en el módulo de trabajos de grado, en el cual se permitirá subir el plan en pdf por parte de los autores que tengan

aprobado el tema, pero no el plan. Además se debe dar acceso a los directores y comité para ver el plan en los diferentes servicios.

- Agregar al módulo de trabajos de grado un servicio que permita realizar la búsqueda de los proyectos ya finalizados, para realizar la impresión de la nota en el formato especificado.
 - Permitir al administrador la impresión de la nota del proyecto en el respectivo formato, inmediatamente se ingrese la nota.
 - Aplicar reingeniería a los servicios de Mi proyecto, Mis proyectos dirigidos y Buscador privado, que se encuentran en el módulo de trabajos de grado, también al módulo de solicitudes de los autores, para que no se muestre en ellos las decisiones del comité del día actual, debido a que el comité de trabajos de grado puede cambiar sus decisiones en el transcurso del día que sesiono hasta las 11:59:59 pm.
- Desempeñar labores de soporte a los usuarios de los portales EdiWEB y FIFQuiWEB, brindando así solución a sus diferentes necesidades o situaciones que se puedan presentar. Entre estas labores se destacan:
 - Capacitar usuarios y estudiantes del primer nivel en el uso de servicios dentro de los portales EdiWEB y FIFQuiWEB, promoviendo así su utilización.
 - Atender usuarios por olvido de la contraseña, creación de grupos, solicitudes de propuestas de eventos o de cartelera en el índice y creación de agendas con eventos y foros con sus respectivas conversaciones.
 - Crear usuarios de forma manual, para personas no pertenecientes a la escuela o la facultad y que por algún motivo necesitan registrarse en el sitio.

2 MARCO TEORICO

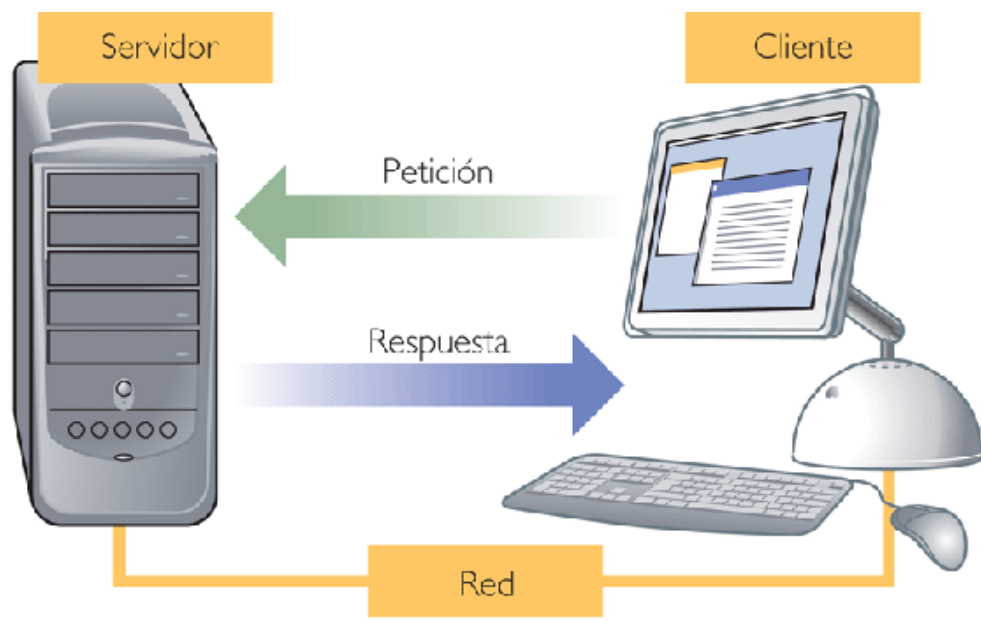
2.1 ARQUITECTURA CLIENTE/SERVIDOR

La parte funcional de la computación Cliente/Servidor, se define como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información de forma transparente aún en entornos multiplataforma.

El funcionamiento de la arquitectura Cliente/servidor se ve en los siguientes pasos:

- 1) El cliente solicita algún tipo de información al servidor.
- 2) El servidor recibe la petición que el cliente ha enviado.
- 3) El servidor procesa y maneja la solicitud del cliente.
- 4) El servidor envía el resultado obtenido al cliente.
- 5) El cliente recibe el resultado y lo procesa obteniendo alguna respuesta.

Figura 1. Modelo Cliente/servidor



Fuente: Wiki. [en línea]. Disponible en internet:

http://es.sandramarramirez.wikia.com/wiki/Arquitectura_Mono%ADtica_y_Arquitectura_Cliente_-_Servidor

La arquitectura Cliente/Servidor se puede ver como una extensión de programación modular ya que lo que se busca es separar una gran pieza de software en módulos logrando así hacer más fácil el desarrollo y mejorar su mantenimiento.

La arquitectura Cliente/Servidor presenta varios tipos, modelos y componentes los cuales permiten la aplicación correcta de este modelo. (Ver Anexo A).

2.2 NETBEANS

2.2.1 Definición NetBeans es un entorno de desarrollo integrado libre, el cual está hecho principalmente para el lenguaje de programación Java, pero soporta muchos más lenguajes de programación. Es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Es un producto libre y gratuito sin restricciones de uso.

2.2.2 Tecnologías que soporta Netbeans soporta muchas tecnologías para desarrollar diferentes proyectos, algunas de esas tecnologías son:

- C, C + +
- Fortran
- PHP
- HTML5
- JavaFX
- JavaScript
- Ajax
- GUI Builder
- JSP

2.2.3 Características Las principales características de NetBeans son:

- Sintaxis Resaltada.
- Completa el Código y Análisis de Típo.
- Ofrece la posibilidad de refactorizar código.
- Registro de servidores MySQL.
- Ver, crear y borrar bases de datos.
- Depurador JavaScript.
- Extensión JavaFX.
- Soporta control de versiones con la herramienta gratuita CVS.
- Permite la integración de la herramienta Ant de Apache.

2.2.4 Puntos fuertes

- Se puede decir que es el más completo, estable y fiable de los entornos de desarrollo integrado que se encuentran disponibles en el mercado.
- La gestión de paquetes y sus avanzadas detecciones de errores (incluso antes de compilar) resultan más cómodas e intuitivas que en los otros entornos.
- Si alguien necesita programar en su vida profesional y ha aprendido con NetBeans, podrá enfrentarse con confianza a cualquier entorno presente en la empresa, ya que todos son muy parecidos entre sí.

2.2.5 puntos débiles

- Su consumo de recursos es significativamente mayor que el de las otras alternativas disponibles.
- Enfrentarse a un entorno algo complejo como NetBeans puede desanimar o confundir al usuario, especialmente si no tiene nociones de programación.
- Sus múltiples ayudas al programador pueden no ser adecuadas para iniciarse en el lenguaje Java.

2.3 BASES DE DATOS

2.3.1 Definición Una base de datos se define como un conjunto o una serie de datos, que pertenecen a mismo contexto, están organizados y relacionados entre sí, estos datos son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

2.3.2 Características Las principales características de las base de datos que podemos mencionar son:

- Se pueden optimizar las consultas complejas.
- Independencia lógica y física de los datos.
- Integridad de los datos.
- Mínima Redundancia.
- Seguridad de acceso y auditoría.
- Puede realizarse acceso concurrente por parte de múltiples usuarios.
- Se puede tener respaldo y copias de recuperación.
- Acceso a través de lenguajes de programación estándar.

2.3.3 Sistema de gestión de base de datos Los Sistemas de Gestión de Base de Datos son un tipo de software muy específico, el cual se dedica a cumplir la función de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Este se conforma por varios componentes que son: un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de consulta.

2.3.4 Ventajas de las bases de datos

- Permite realizar control sobre la redundancia de los datos.
- Garantiza consistencia de los datos.
- En una misma empresa o entidad permite compartir datos.
- Como existe validez y consistencia de los datos, mejora la integridad.
- Se puede mejorar la seguridad de los datos.

2.3.5 Desventajas de las bases de datos

- Los sistemas de gestión de bases de datos pueden ser complejos.
- Para el mantenimiento de las bases de datos y los sistemas de gestión puede ser necesario algunos equipos especiales lo cual generara gastos adicionales.
- Pueden ser vulnerables a fallos, ya que todo se encuentra centralizado en el sistema de gestión de bases de datos, por eso se deben tener copias de seguridad.

2.3.6 Sistema de gestión de bases de datos utilizado

MySQL Para el desarrollo del proyecto se usó MySQL porque es un sistema de gestión de base de datos que se considerado uno de los mejores y más importantes, es bastante utilizado para el diseño y programación de base de datos de tipo relacional, también se usa bastante para el desarrollo de aplicaciones web.

Este sistema de gestión de bases de datos presenta algo importante y es que permite emplear las bases de datos multiusuario a través de la web, en diversos lenguajes de programación y varias plataformas que se adaptan a diferentes necesidades y requerimientos.

Ventajas de MySQL

- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Tiene alta velocidad, y seguridad que lo hacen altamente apropiado para acceder a bases de datos en internet.
- Posee un bajo costo en requerimientos para la elaboración de bases de datos, debido a que es un software que puede ser ejecutado en una maquina con escasos recursos.
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.

- Tiene una versión en Open Source bastante completa, la cual es código abierto que puede ser usado sin pagar ningún tipo de licencia.

2.4 TECNOLOGIAS DE DESARROLLO DE PAGINAS WEB DINAMICAS

2.4.1 Páginas web dinámicas Se puede definir las páginas web dinámicas como aquellas en las que la información mostrada es generada a partir de una petición por parte del usuario de la página. Caso diferente a lo que sucede con las páginas estáticas, en las cuales su contenido ya está predeterminado, mientras que en las páginas dinámicas la información aparece inmediatamente después de que se realiza una solicitud por el usuario.

Este procedimiento se hace posible ya que una página web dinámica tiene asociada una Base de Datos desde la que se permite visualizar la información contenida en ella por medio de una petición.

2.4.2 Ventajas de las páginas web dinámicas

- El proceso de actualización es sencillo, ya que no hay necesidad de entrar al servidor.
- Se puede tener la información actualizada y al día fácilmente.
- Existe un gran número de posibilidades para su diseño, desarrollo y su mantenimiento.
- Hay una mayor interactividad con el usuario, ya que este puede realizar cambios en el diseño, contenidos y hasta la presentación de la página.
- Permite la presentación del contenido en diversos formatos y dispositivos.

2.4.3 Tipos de tecnologías para desarrollar páginas web dinámicas Algunas de las más importantes tecnologías para el desarrollo de páginas web dinámicas son:

- ASP.NET
- JSP

- PHP
- PYTHON

2.4.4 Tecnología aplicada en el desarrollo del proyecto

JSP El significa de sus siglas en ingles es JavaServerPages y en español su significado es Paginas de Servidor Java, JSP es una tecnología que permite a los desarrolladores de software la creación de páginas web dinámicas basadas en lenguajes como HTML, XML, y muchos más, un requisito que se tiene es para poder utilizar y correr JSP, se necesita de un servidor web compatible con contenedores servlet como Apache Tomcat o Jetty.

La tecnología de JSP permite a los desarrolladores y a los diseñadores de páginas web elaborar de forma rápida sus proyectos, a su vez hace fácil realizar las tareas de mantenimiento. La tecnología de JSP separa la interfaz del usuario de la parte lógica del contenido dando así la oportunidad a los diseñadores cambiar a su gusto las plantillas de la interfaz sin alterar el contenido dinámico subyacente.

Características de JSP Las características más importantes de JSP son:

- JSP es portable a muchos servidor y sistemas operativos.
- Como su parte dinámica está escrita lo hace más poderoso para desarrollar aplicaciones que requieran componentes reutilizables
- Permite crear aplicaciones web dinámicas de una forma ágil y sencilla.
- Es capaz de soportar contenido dinámico que refleja condiciones del mundo real.
- Además de poderse utilizar con HTML, puede usarse con XML y WML sin ningún problema

Modelo de acceso a JSP

1. 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.

2. Después de recibir la petición del cliente, el archivo JSP pide información de un JavaBean si es necesario.
3. El JavaBean en turnos puede pedir información de otro JavaBean o de una base de datos.
4. Una vez el JavaBean genera el contenido, el archivo JSP puede consultar y presentar el contenido del JavaBean al navegador.

2.5 SISTEMA DE CONTROL DE VERSIONES

2.5.1 Definición Se define como sistema de control de versiones a una herramienta que registra todos los cambios que se realizan a uno o más proyectos, almacenando las versiones del producto durante todas sus fases del desarrollo. Las versiones son como capturas que registran el estado del proyecto en ese momento del tiempo y se van guardando a medida que se hacen modificaciones al código fuente.

Los sistemas de control de versiones nacen por la necesidad de llevar un registro de los cambios que se realizan sobre los elementos de algún producto o una configuración del mismo.

2.5.2 Características Las principales características de un sistema de control de versiones son:

- Poseer un mecanismo de almacenamiento de los elementos que se deben gestionar como lo son: archivos de texto, imágenes, documentación y otros.
- Tener la posibilidad de realizar cambios sobre los elementos almacenados, estos cambios pueden ser: borrar, añadir, renombrar o realizar modificaciones parciales.

- Llevar un registro histórico de todas las acciones que se han realizado con cada uno de los elementos.

2.5.3 Tipos de sistemas de control de versiones Algunos ejemplos de este tipo de herramientas son:

- CVS
- Subversion
- SourceSafe
- Git
- Mercurial
- Bazaar

2.5.4 Sistema de control de versiones aplicado en el proyecto Este sistema de control de versiones salió al mercado en el año 2000 bajo una licencia Apache/BSD. Subversion se considera como uno de los sistemas más legendarios, es una herramienta open source basada en un repositorio cuyo funcionamiento se asemeja enormemente al de un sistema de ficheros.

Subversion utiliza el concepto de revisión para almacenar los cambios que se producen en el repositorio. Entre dos revisiones sólo guarda el conjunto de modificaciones, permitiendo así optimizar al máximo el uso de espacio en disco. También permite al usuario copiar, crear y borrar carpetas con la misma flexibilidad con la que lo haría si estuviese en su disco duro local.

Subversion puede acceder al repositorio a través de redes, lo que le permite que pueda ser usado por personas que se encuentran en distintas computadoras, permitiendo así que se puede progresar más rápidamente sin un único conducto por el cual deban pasar todas las modificaciones.

Ventajas de Subversion

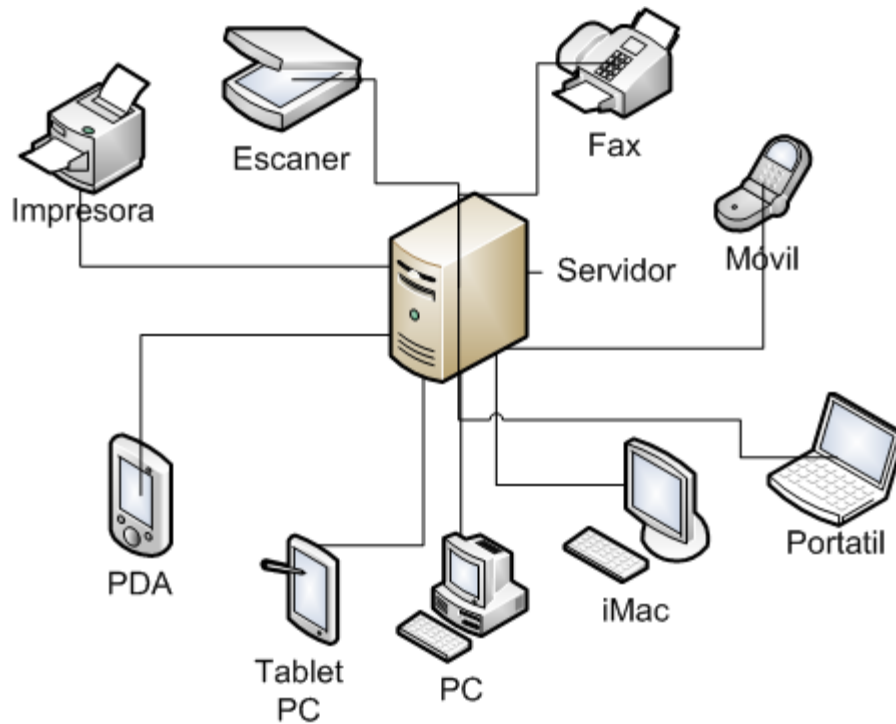
- Se guarda un histórico de los archivos y directorios a través de copias y renombrados.
- Envía solo las diferencias que encuentra entre el archivo anterior y el nuevo o modificado, no envía archivo completo.
- Permite manejar eficientemente archivos binarios.
- Cuando es usado integrado a Apache permite utilizar todas las opciones que este servidor provee a la hora de autenticar archivos, como lo son: SQL, LDAP, PAM, y otros.
- Las modificaciones pueden ser a un solo archivo o ser a varios archivos.

2.6 SERVIDORES WEB

2.6.1 Definición Un servidor, un software que está al servicio de otras máquinas, ordenadores o personas llamadas clientes y que le suministran a estos, todo tipo de información.

Este software se encarga de procesar una aplicación del lado del servidor, realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente y así generar una respuesta en cualquier lenguaje o Aplicación del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web.

Figura 2. Esquema Cliente-Servidor



Fuente: apr. [en línea]. Disponible en internet: <http://aprenderaprogramar.com/index.php>

2.6.2 Tipos de servidores Los tipos de servidores más habituales son:

- Servidor proxy.
- Servidor Web.
- Servidor de correo.
- Servidor de bases de datos.
- Servidores clúster.
- Servidores dedicados.
- Servidores de imágenes
- Servidores de aplicaciones.

2.6.3 Servidor utilizado en el desarrollo del proyecto

Apache tomcat Apache Tomcat es un servidor de aplicaciones el cual funciona como un contenedor de servlets este fue desarrollado bajo el proyecto Jakarta en la fundación de software Apache.

Apache tomcat es uno de los servidores de aplicaciones Java más utilizados, las características que hacen que lo prefirieran son: es sencillo de instalar, cumple con todos los estándares, es liviano, tiene muy buena documentación y es software gratuito, además por ser escrito en Java no presenta problemas de compatibilidad y funciona en cualquier sistema operativo que disponga de la Máquina Virtual de Java (JVM).

2.7 PROGRAMACION UTILIZADA

2.7.1 Definición de la programación orientada a objetos La programación utilizada para el desarrollo de este proyecto fue la Programación Orientada a Objetos con siglas (POO). Esta es una de las formas más populares de programar, ya que usa objetos y sus interacciones para diseñar aplicaciones y programas de computador, esta forma de programar es lo que sería más cercano a como expresaríamos las cosas en la vida real, a través del significado de objetos que contienen características y funciones.

2.7.2 Conceptos fundamentales

- **Clase:** una clase contiene las definiciones de las propiedades y comportamiento de un tipo de objeto concreto. Lo que significa que la definición de un objeto es la clase.
- **Herencia:** La herencia nos permite crear objetos que incorporen propiedades y métodos de otros objetos. Es así como se pueden construir unos objetos a partir de otros sin tener que reescribirlo todo.

- **Objeto:** Es un ejemplar de una clase. Es una entidad conformada de un conjunto de propiedades y de un comportamiento, los cuales se verán representados en eventos.
- **Métodos:** Son las funcionalidades asociadas a los objeto, es lo que el objeto puede hacer. Un método puede llegar producir cambios en las propiedades del objeto.
- **Evento:** Es un acontecimiento en el sistema. El sistema maneja el evento enviando el mensaje adecuado al objeto pertinente. Puede definirse como evento a la acción generado por un objeto.
- **Atributos:** son las características que posee un objeto, siendo estas un conjunto de datos y calificadores para aquellos datos. Los atributos pueden ser desde datos simples hasta objetos.

2.7.3 Características Las principales características de la POO son:

- **Abstracción:** indica las características esenciales que tiene un objeto.
- **Encapsulamiento:** Permite reunir todos los elementos que pueden considerarse pertenecientes a una misma entidad.
- **Modularidad:** podemos subdividir una aplicación en partes más pequeñas, las cuales pueden ser independientes de la aplicación principal y de las otras partes.
- **Polimorfismo:** Un conjunto de objetos puede contener objetos de diferentes tipos, y a la hora de llamarlos no existe problema ya que el comportamiento será el del objeto referenciado.
- **Principio de ocultamiento:** Así protegeremos las propiedades de un objeto contra su modificación por parte de quien no tenga derecho a acceder a ellas.

2.7.4 Lenguajes orientados a POO A continuación se presenta algunos de los más importantes lenguajes orientados a objetos:

- C++.
- Java.
- VB.NET.
- JavaScript.
- C#.
- GObject.
- PHP.
- Fortran.
- Python.

2.7.5 Lenguaje utilizado para el desarrollo del proyecto Java El lenguaje que se utilizó en el desarrollo del proyecto fue java el cual es uno de los mejores y más usados en el mundo para aplicaciones de Cliente/Servidor, este lenguaje es de propósito general y orientado a objetos, fue creado para que tenga pocas dependencias de implementación lo que nos permite que una aplicación pueda ser ejecutada desde cualquier dispositivo.

Características

- Orientado a Objetos.
- Independencia de la plataforma.
- Seguro.
- Tiene recolector de basura.
- Portable.
- Posee alto rendimiento.
- Es dinámico.

3. MARCO METODOLÓGICO

3.1 PROTOTIPO EVOLUTIVO

Para la realización de los nuevos servicios para el portal web de la escuela de Diseño Industrial y la facultad de Ingenierías Físicoquímicas de la Universidad Industrial de Santander, se utilizó como metodología de desarrollo el prototipo Evolutivo.

Figura 3. Prototipo Evolutivo



Fuente: wfrancor. [en línea]. [Citado el 27 de Junio, 2014]. Disponible en internet: <http://wfrancor.blogspot.com/2012/12/modelo-de-prototipos.html>

La elección de esta metodología se debe a las siguientes razones:

- Se desea tener un bosquejo de las cosas que se desean mejorar o de las cosas que se crearan, para así poder incorporar sugerencias de cambio por parte de los usuarios del portal de la Escuela y la Facultad en etapas tempranas del desarrollo.
- La escuela de Diseño Industrial y la facultad de Ingenierías Físicoquímicas, deben estar actualizando su portal Web mejorando los servicios existentes y agregar nuevos servicios en un tiempo considerablemente corto debido a que el sistema no es un producto final sino que al contrario es sometido a una permanente reconstrucción.
- 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.
- 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.
- Con esta metodología podremos saber lo antes posible si hemos interpretado correctamente las especificaciones y las necesidades de la Escuela y la Facultad.
- La decisión se fundamenta en la ventaja que presenta la realización de los cambios en las etapas tempranas y la posibilidad de realizar 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.

Procedimiento a seguir para la metodología utilizada:

1. La construcción de prototipos comienza con la Recolección de los Requisitos.
2. El desarrollador y usuario se reúnen y definen los objetivos globales para el software, identifican todos los requisitos conocidos y perfilan las áreas en donde será necesaria una mayor definición.
3. Luego se produce el Diseño del prototipo que se enfoca sobre la representación de los aspectos del software visibles al usuario (por ejemplo, métodos de entrada y formatos de salida) y se prosigue a su construcción.
4. El prototipo es evaluado por el usuario y se utiliza para refinar los requisitos del software a desarrollar.
5. Se produce un proceso interactivo en el que el prototipo es “afinado” (Refinamiento del prototipo) para que satisfaga las necesidades del usuario, al mismo tiempo que le facilita al desarrollador una mejor comprensión de lo que hay que hacer y poder entregar el producto final requerido o Producto de Ingeniería.

3.2 LENGUAJE DE MODELADO UNIFICADO

Lenguaje Unificado de Modelado o UML, por sus siglas en inglés, Unified Modeling Language, Es un lenguaje gráfico con el cual podremos especificar, visualizar,

construir y documentar un sistema. UML ofrece un estándar para describir un esbozo del sistema, en el cual se incluyen aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación y esquemas de bases de datos.

3.2.1 Tipos de diagramas UML Existen tres tipos de diagramas UML que son:

- **De estructura**
 - Diagrama de clases.
 - Diagrama de objetos.
 - Diagrama de componentes.
 - Diagrama de estructura compuesta.
 - Diagrama de paquetes.
 - Diagrama de despliegue.

- **De comportamiento**
 - Diagrama de casos de uso.
 - Diagrama de actividades.
 - Diagrama de estado.

- **De interacción**
 - Diagrama de secuencia.
 - Diagrama de colaboración UML.
 - Diagrama de tiempo.
 - Diagrama de interacción.

3.2.2 Diagramas de casos de uso Los diagramas de casos de uso sirven para describir las relaciones entre un grupo de casos de uso y los actores participantes en el proceso. Los diagramas de caso de uso se encargan de mostrar lo que hace el sistema desde el punto de vista de un observador externo, nos muestran que es lo que hace el sistema pero no como lo hace.

Caso de uso Los casos de uso nos describen las interacciones típicas entre los usuarios de un sistema y del mismo sistema. Especifican qué requisitos de funcionamiento debe tener este.

Actor Un actor es una entidad externa al sistema la cual interacciona con el sistema participando en un caso de uso. Los actores pueden ser gente real usuarios del sistema, otros ordenadores o eventos externos. Hay que tener en cuenta que los actores no representan personas físicas si no su rol con el sistema.

3.2.3 Diagramas de secuencia Los diagramas de secuencia muestran el intercambio de mensajes en un momento dado. Los diagramas de secuencia ponen especial énfasis en el orden y el momento en que se envían los mensajes a los objetos.

Línea de vida Es una línea vertical que representa la secuencia de eventos que se producen en un participante durante una interacción, mientras el tiempo avanza. Este participante puede ser una instancia de una clase, componente o actor.

Objetos Los 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.

Mensaje Un mensaje se desplaza de un objeto a otro a través de la línea de vida de un objeto a otro. Un objeto también puede enviarse un mensaje a sí mismo. Un mensaje puede ser simple, sincrónico o asincrónico.

Tiempo En el diagrama de secuencia se representa al tiempo en dirección vertical. Este 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.

3.3 ESTANDARES DE PROGRAMACION

3.3.1 Modelo de datos Un modelo de datos es un lenguaje enfocado a las Bases de Datos. En otras palabras un modelo de datos permite describir las estructuras de datos contenidos en las base de datos, el tipo de los datos que hay en la base de datos y la forma en que se relacionan, también las restricciones de integridad las cuales son un conjunto de condiciones que deben cumplir los datos para reflejar correctamente la realidad deseada y finalmente las operaciones de manipulación de los datos como lo son las operaciones de agregado, borrado, modificación y recuperación de los datos de la base.

Clasificación de los modelos de datos Los modelos de datos se clasifican de acuerdo al nivel de abstracción que presentan, esos niveles son:

- **Modelos de Datos Conceptuales** Estos son orientados a la descripción de estructuras de datos y a las restricciones de integridad. Se usan principalmente durante la etapa de Análisis de un problema dado y están orientados a representar los elementos que intervienen en ese problema y sus relaciones. Un ejemplo de este tipo de modelo es el Modelo Entidad-Relación.
- **Modelos de Datos Lógicos** Estos se orientan a las operaciones más que a la descripción de una realidad. Se implementan en algunos Manejadores de Base de Datos. Un ejemplo es el Modelo Relacional.
- **Modelos de Datos Físicos** Estas Son estructuras de datos a bajo nivel las cuales se implementan dentro del propio manejador.

3.3.2 Nombres de las tablas En el grupo calumet se maneja un estándar en el cual los nombres de los campos de las tablas, así como los nombres de las tablas de la base de datos, se escriben en minúsculas, a excepción de la primera letra de cada palabra que conforme su nombre; en el caso de que sea un nombre compuesto por dos o más palabras, los nombres tendrán en mayúscula la primera letra de cada palabra que la forma.

También se han definido tres categorías para las diferentes tablas que conforman la base de datos. Dependiendo la categoría de la tabla, se antepone un prefijo a su nombre con lo cual se podrá identificar a que categoría pertenece. Las categorías son:

- **Tabla básica:** Una tabla básica es aquella cuyos registros son necesarios para el correcto funcionamiento de la base de datos. En estas tablas no se realizan muchos cambios en los datos. El prefijo que se antepone a los nombres de las tablas de esta categoría es “TB_”, un ejemplo de una tabla básica es “TB_Estados”, la cual contiene la información de todos los posibles estados en los que puede estar un trabajo de grado.
- **Tabla Principal:** Una tabla principal es aquella en la cual la cantidad de registros tiende a crecer en gran cantidad. Para identificar una tabla principal se antepone a su nombre el prefijo “TP_”, un ejemplo de una tabla de esta categoría es “TP_Actas”, en esta tabla es donde se almacena información de la sesión del comité de trabajos de grado.
- **Tabla de Relación:** La tabla de relación es aquella que surge de la relación muchos a muchos de una o dos tablas cualquiera. Los nombres de las tablas de relación deberán ser descriptivos para cada relación. El prefijo que se antepone a las tablas de esta categoría es “TR_”, y como ejemplo de esta categoría tenemos la tabla “TR_ComiteEvaluadores”, esta tabla se guarda la información de los evaluadores de los planes de los proyectos de grado, así como cierta información del plan y del acta.

3.3.3 Páginas JSP Al igual que con las tablas se tiene un estándar para las páginas JSP en el cual los nombres de las páginas JSP que componen los portales EdiWEB y FIFQuiWEB serán escritos inicializando en mayúscula precedido de letras minúsculas, en caso de que el nombre del JSP compuesto por dos o más palabras, entonces la primera de cada palabra interna debe ir en mayúscula, como ejemplo tenemos ImprimirNota.Jsp.

3.3.4 Clases Los nombres de las clases deben ser sustantivos en plural, la primera letra de cada palabra debe ser mayúscula. Estos deben ser simples y descriptivos como por ejemplo Archivos.java.

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. Por ejemplo:

- En el sitio existe un directorio para las imagenes llamado “images”, este contiene todos los archivos .jpg, .gif, .png.
- El sitio cuenta con un directorio llamado Script, donde están todos los archivos de JavaScript “.js” que son requeridos para el buen funcionamiento del portal.
- En el sitio hay un directorio para la parte de trabajos de grado, el cual contiene otros directorios con las paginas jsp que permiten el funcionamiento del módulo de trabajos de grado.

4. DESARROLLO DE LA HERRAMIENTA, ADMINISTRACIÓN Y MANTENIMIENTO

En el marco metodológico se mostró claramente que para el desarrollo de este proyecto se siguió la metodología de prototipo evolutivo.

Por consiguiente durante la fase de requerimientos se elaboró un prototipo, en el cual se buscaba satisfacer todas las necesidades iniciales, este prototipo inicial poco a poco fue mejorando con la inclusión de nuevos requerimientos que surgieron durante la fase de desarrollo, cada vez que se creaba un nuevo prototipo, era sometido a pruebas de funcionamiento y se realizaban los ajustes que se requerían a partir de los resultados de las pruebas.

4.1 PROTOTIPO ESPERADO

En el inicio del proyecto no se tenía una clara interpretación de cómo sería el producto finalizado, pero durante el desarrollo del proyecto, se veía como poco a poco se acercaba cada vez más al objetivo inicial, gracias a la evolución de los prototipos, las pruebas realizadas y al análisis del sistema se logró desarrollar exitosamente la solución a las necesidades de los usuarios.

Teniendo como base el objetivo específico inicial, los requisitos que surgieron durante la creación, pruebas y análisis de los nuevos prototipos y la realimentación con el cliente. Para cada uno de los objetivos se listan los requerimientos detallados, los cuales se cumplieron para lograr el prototipo final.

- 1. Añadir al módulo de trabajos de grado un servicio que permita a los autores de los proyectos subir el documento que contiene la información del plan de su proyecto de grado.**

Objetivo inicial:

- Crear un servicio en el módulo de trabajos de grado, en el cual se permitirá subir el plan en pdf por parte de los autores que tengan aprobado el tema, pero no el plan. Además se debe dar acceso a los directores y comité para ver el plan en los diferentes servicios.

Requisitos finales del objetivo:

- Los autores de los proyectos de grado podrán subir el documento en pdf de su plan de proyecto de grado, siempre y cuando tengan aprobado su tema, y se encuentre dentro de las fechas establecidas para la entrega del plan.
- los autores de los proyectos de grado podrán subir su plan de proyecto de grado, si su evaluador de plan, o si el comité de trabajos de grado dan la opción de cambiarlo o corregirlo.
- Los directores, evaluadores y el comité de trabajos de grado podrán ver y descargar el documento en pdf que contiene el plan de trabajo de grado.
- El evaluador, y el comité de trabajos de grado podrá sugerir a los autores que suban un nuevo documento con correcciones, en caso de que el autor suba varios documentos, el evaluador o el comité de

trabajos de grado dependiendo el caso podrán escoger el que cumpla sus expectativas, y podrán ver el nuevo plan y el plan anterior.

- Enviar un correo informando que se ha subido un nuevo plan, el correo se enviara cada vez que un autor suba un documento con su plan de proyecto de grado, el correo se enviara dependiendo el caso al que corresponda, si sube el plan para que revisión de director de proyecto, el correo se enviará al director y a los autores, si el autor sube un plan a sugerencia del evaluador, el correo se enviara

2. Añadir un servicio al módulo de trabajos de grado en el cual se puedan buscar los proyectos finalizados e imprimir la respectiva nota en el formato especificado.

Objetivo inicial:

- Agregar al módulo de trabajos de grado un servicio que permita realizar la búsqueda de los proyectos ya finalizados, para realizar la impresión de la nota en el formato especificado.

Requisitos finales del objetivo:

- Cuando se muestre el formato establecido para la nota de proyecto de grado, debe permitir al administrador del portal ingresar la nota, el número de créditos de la materia requisito de grado de forma

manual, esto para los proyectos en los cuales los autores no estén activos.

- Crear un buscador el cual liste los proyectos finalizados, para cada proyecto que se liste como resultado de la búsqueda debe tener la opción de imprimir la cual permitirá ver el documento de la nota de proyecto de grado para realizar su impresión.

3. Crear un servicio en el módulo de trabajos de grado que permita a los administradores de los portales imprimir la nota en el formato especificado.

Objetivo inicial:

- Permitir al administrador la impresión de la nota del proyecto en el respectivo formato, inmediatamente se ingrese la nota.

Requisitos finales del objetivo:

- Tan pronto se ingrese la nota al sistema por parte del administrador del portal se debe mostrar en pantalla el respectivo formato para la nota del proyecto de grado con todos sus datos y con la opción de imprimir.
- Si el proyecto tiene varios autores se mostrara el formato de la nota del proyecto con toda la información de cada uno de los autores de forma independiente, al igual que la opción de imprimir dicho formato.

4. Corregir el código de los servicios de Mi proyecto, Mis proyectos dirigidos y Buscador privado, para que no se muestre en ellos las decisiones tomadas por lo el comité de trabajos de grado en el día actual.

Objetivo inicial:

- Aplicar reingeniería a los servicios de Mi proyecto, Mis proyectos dirigidos y Buscador privado, que se encuentran en el módulo de trabajos de grado, también al módulo de solicitudes de los autores, para que no se muestre en ellos las decisiones del comité del día actual, debido a que el comité de trabajos de grado puede cambiar sus decisiones en el transcurso del día que sesiono hasta las 11:59:59 pm.

Requisitos finales del objetivo:

- Realizar una verificación en la cual se determine si la última acta se encuentra en sesión en el día actual.
- No mostrar los cambios o decisiones que tome el comité de trabajos de grado para los temas, los planes y las solicitudes de los autores en la sesión del día actual.
- Las decisiones que se tomen en el comité de trabajos de grado deben ser visibles después de las 11:59:59 pm del día en el que se realizó la sesión.

Administración.

Objetivos iniciales:

- Hacer seguimiento del uso que hacen los usuarios a los servicios de Foros, Eventos Propuestos, Cartelera, Archivos y Mi perfil dentro de los portales EdiWEB y FIFQuiWEB para detectar usos indebidos o incorrectos por parte de éstos.
- Actualizar periódicamente las Bases de Datos con el objeto de mantener al día la información referente a matrículas, horarios, estados y categorías de los usuarios.

- Atender las consultas y sugerencias que realicen los usuarios para proponerlas como mejoramiento y ofrecimiento de nuevos servicios en la próxima versión de los portales WEB.

Requisitos finales del objetivo:

- Solucionar incidencias las cuales eran fallos que afectaban el buen funcionamiento del sistema.
- Realizar mantenimiento a las bases de datos creando tablas para el correcto desarrollo del proyecto, y eliminar las tablas que se encontraban en desuso
- Realizar una revisión de la estructura de directorios que hacen parte del portal, para así borrar los archivos que ya no se encuentren en uso y agregar los nuevos servicios.
- Mantener los JavaBeans actualizados y los portales de la escuela de diseño industrial y la decanatura de fisicomecánicas para su correcto funcionamiento.

Estas actividades se llevaron a cabo durante el tiempo en el que se realizó el proyecto.

Mantenimiento.

Objetivos iniciales:

- Hacer seguimiento del funcionamiento de los portales para corregir posibles defectos generados por errores en el código fuente que se puedan presentar.
- Revisar y depurar la estructura de directorios y archivos de los portales Web.

Requisitos finales del objetivo:

- Realizar copias periódicas de las bases de datos División y Diamante y de los directorios de los portales EdiWEB y FIFQuiWEB.

- Realizar actualizaciones de la base de datos Diamante para mantener los portales al día.
- Revisar regularmente el historial de cambios para supervisar el uso correcto que los usuarios le dan a los servicios.

Estas actividades se llevaron a cabo durante el tiempo en el que se realizó el proyecto.

Soporte a Usuarios.

Objetivo inicial:

- Capacitar usuarios y estudiantes del primer nivel en el uso de servicios dentro de los portales EdiWEB y FIFQuiWEB, promoviendo así su utilización.
- Atender usuarios por olvido de la contraseña, creación de grupos, solicitudes de propuestas de eventos o de cartelera en el índice y creación de agendas con eventos y foros con sus respectivas conversaciones.
- Crear usuarios de forma manual, para personas no pertenecientes a la escuela o la facultad y que por algún motivo necesitan registrarse en el sitio.
- Modificar los estados de los usuarios de acuerdo a la relación con la escuela o la facultad (activo, inactivo, suspendido)

Requisitos finales del objetivo:

- Estudiar y conocer el correcto funcionamiento de los portales EdiWEB y FIFQuiWEB, con el fin de conocer dónde y cómo atacar los errores para así darles una pronta respuesta.
- Realizar procesos de Investigación para poder dar soluciones a todos los problemas que surgieron con el funcionamiento de los servicios.

- Realizar el registro y la inducción del uso de los portales a los estudiantes de primer nivel.

Cada una de las actividades mencionadas en el objetivo anterior se llevó a cabo durante la práctica en repetidas ocasiones. Además, se dio orientación a los usuarios sobre el uso de determinados servicios.

4.1.1 Diagramas de casos de uso

4.1.1.1 Subir Plan Pdf

Figura 4. Diagramas de casos de uso: Subir Plan

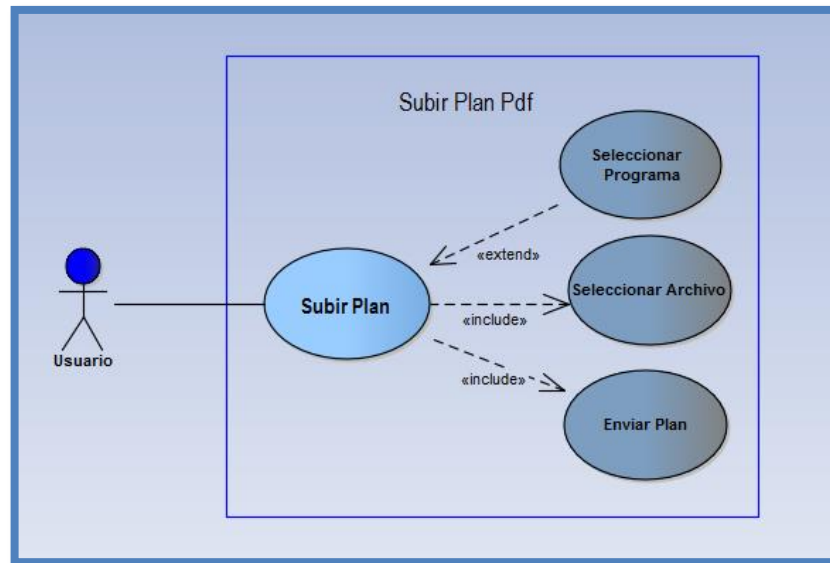


Figura 5. Diagramas de casos de uso: Solicitar Evaluador

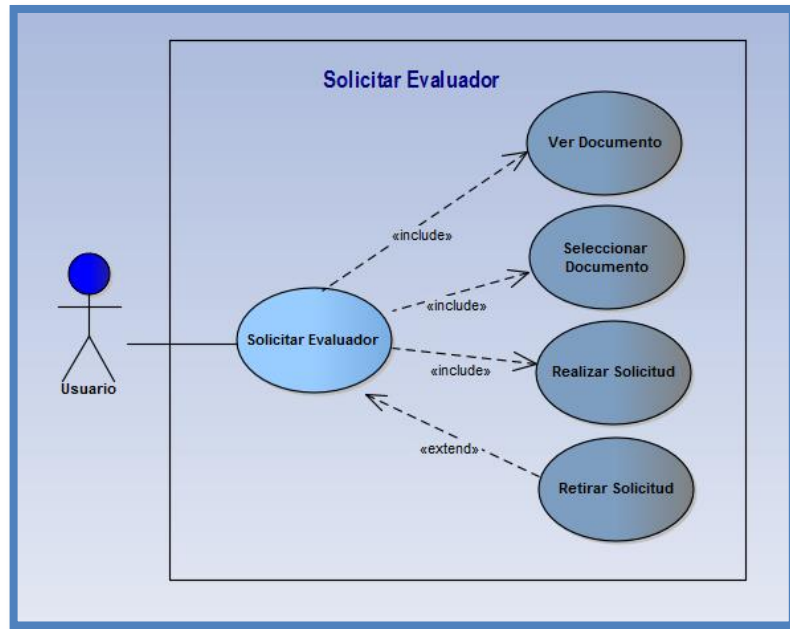


Figura 6. Diagramas de casos de uso: Concepto Plan

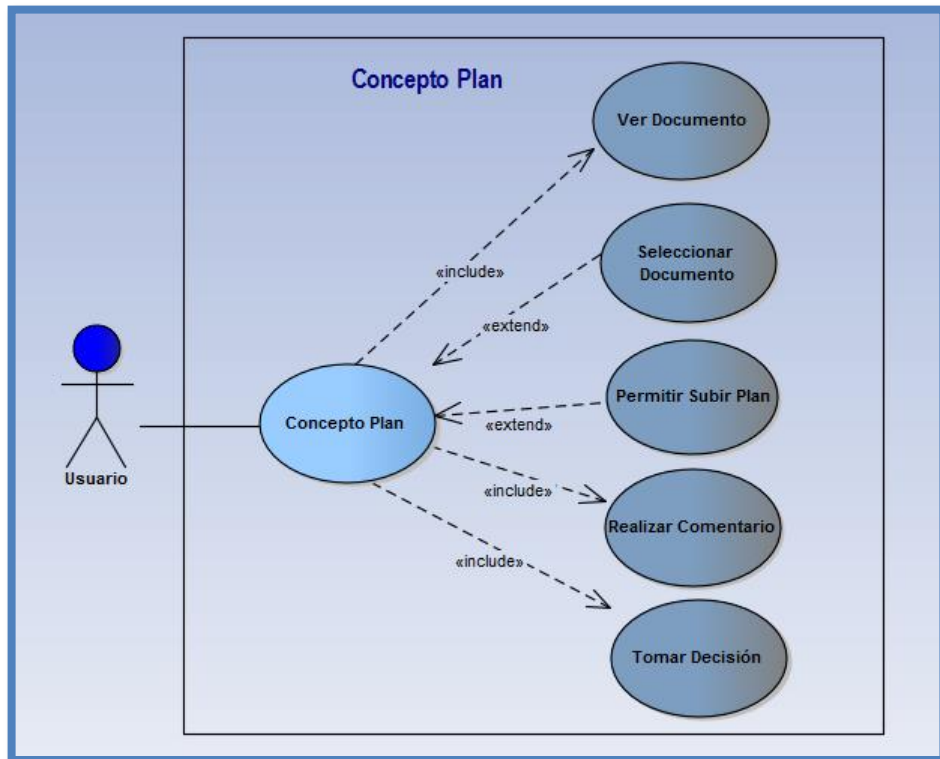


Figura 7. Diagramas de casos de uso: Solicitar Calificador

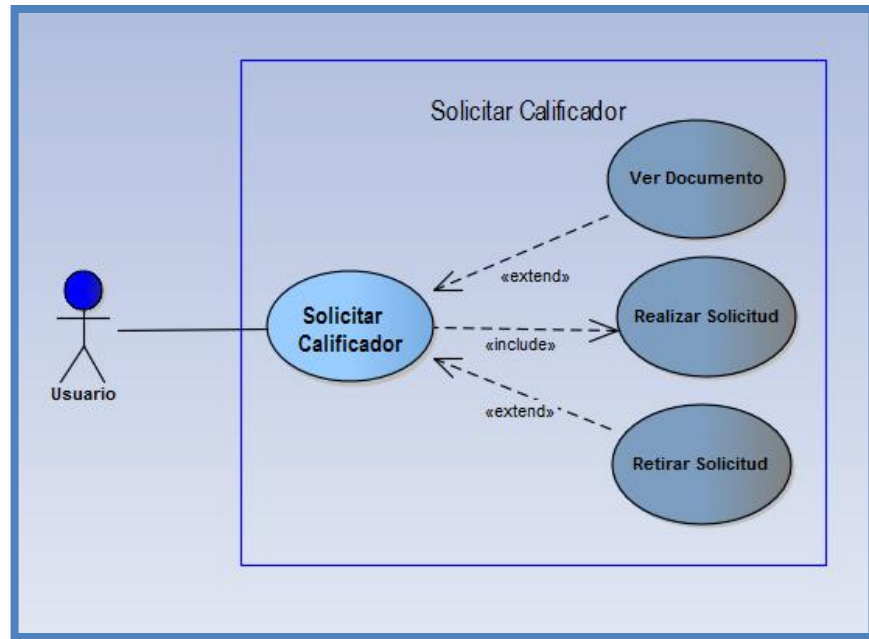


Figura 8. Diagramas de casos de uso: Asignar Evaluador

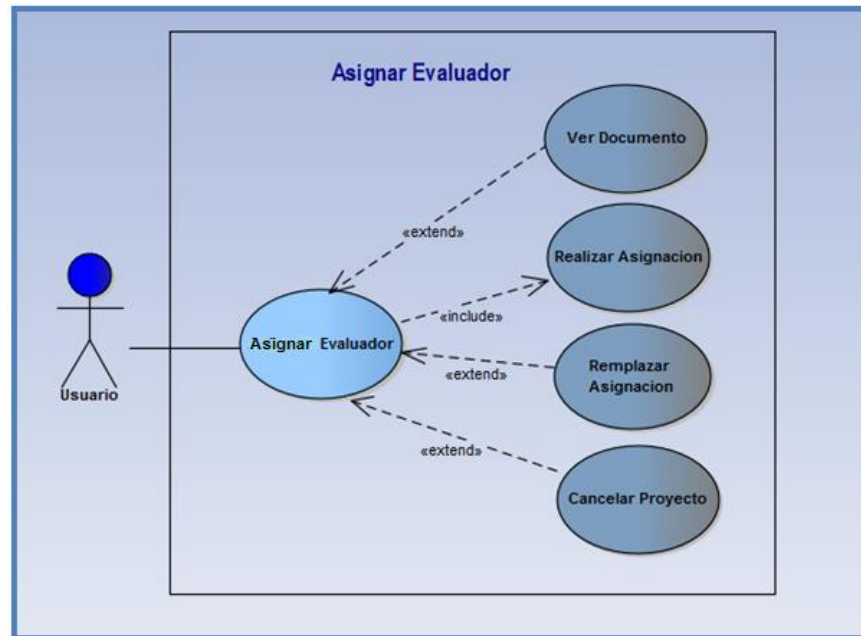


Figura 9. Diagramas de casos de uso: Asignar Calificadores

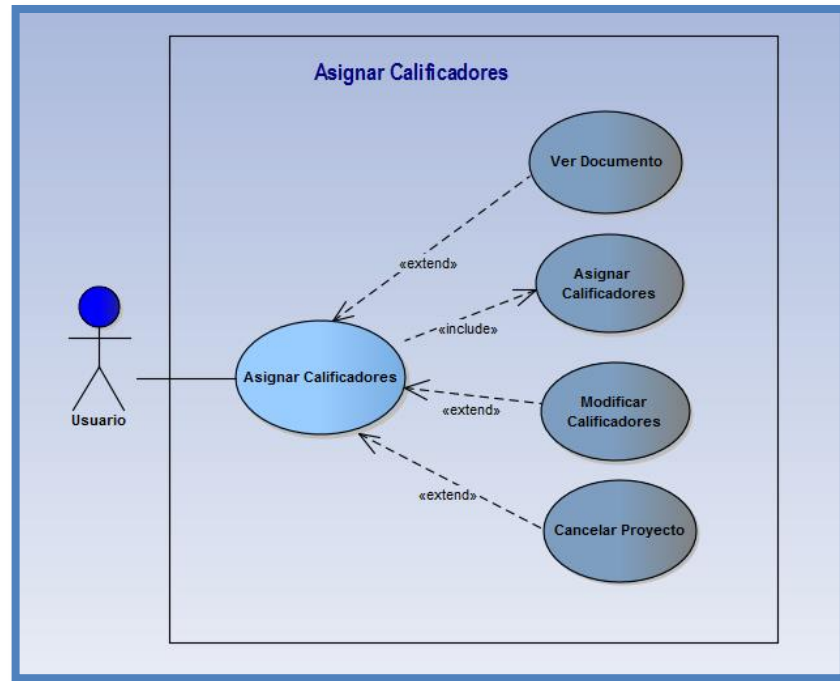
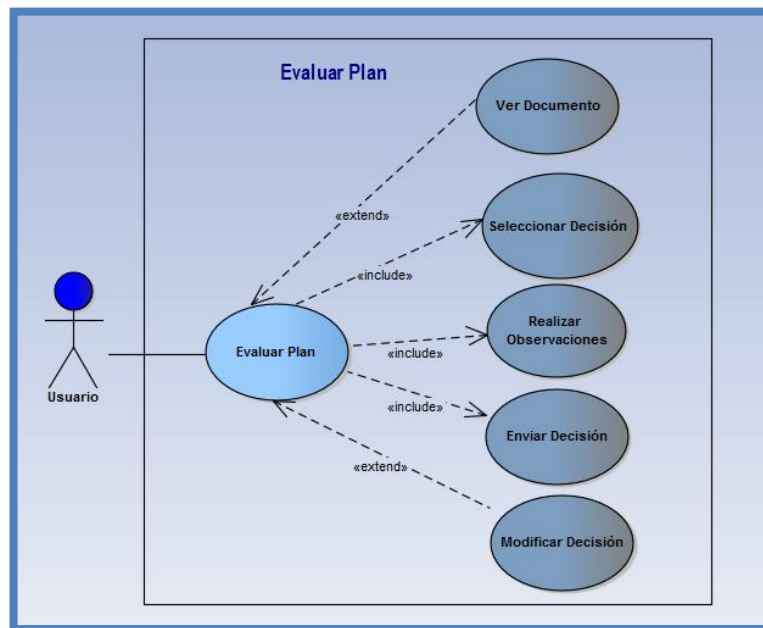


Figura 10. Diagramas de casos de uso: Evaluar Plan



4.1.1.2 Imprimir Nota

Figura 11. Diagramas de casos de uso: Buscador Imprimir Nota

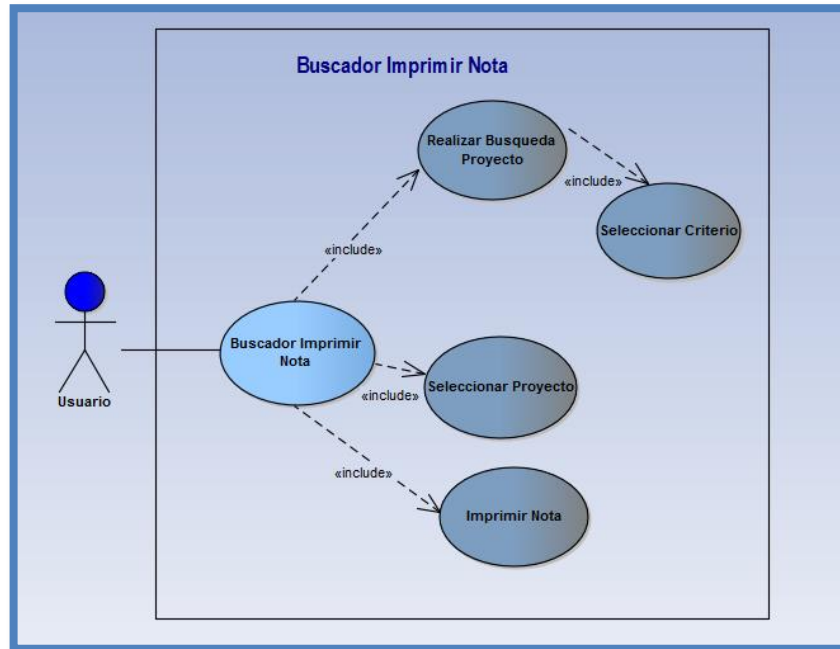
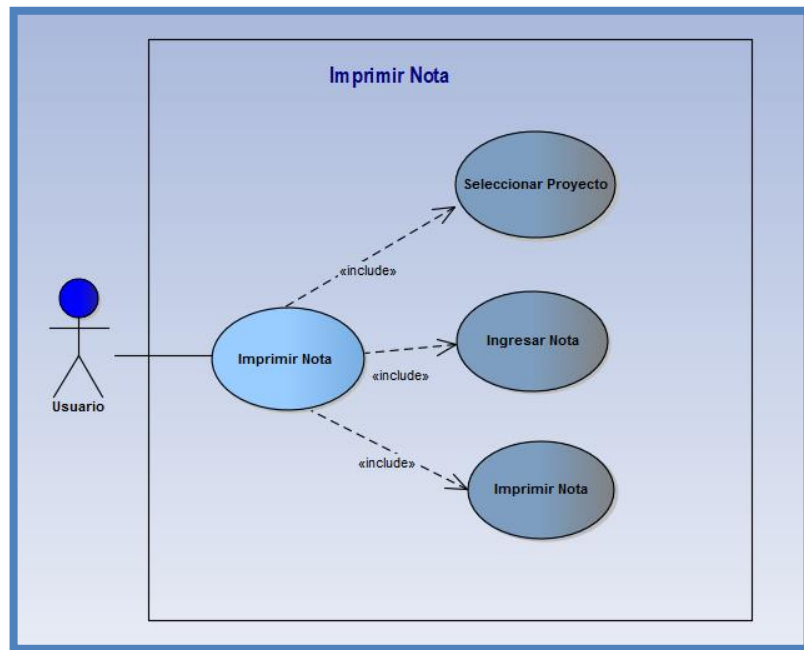
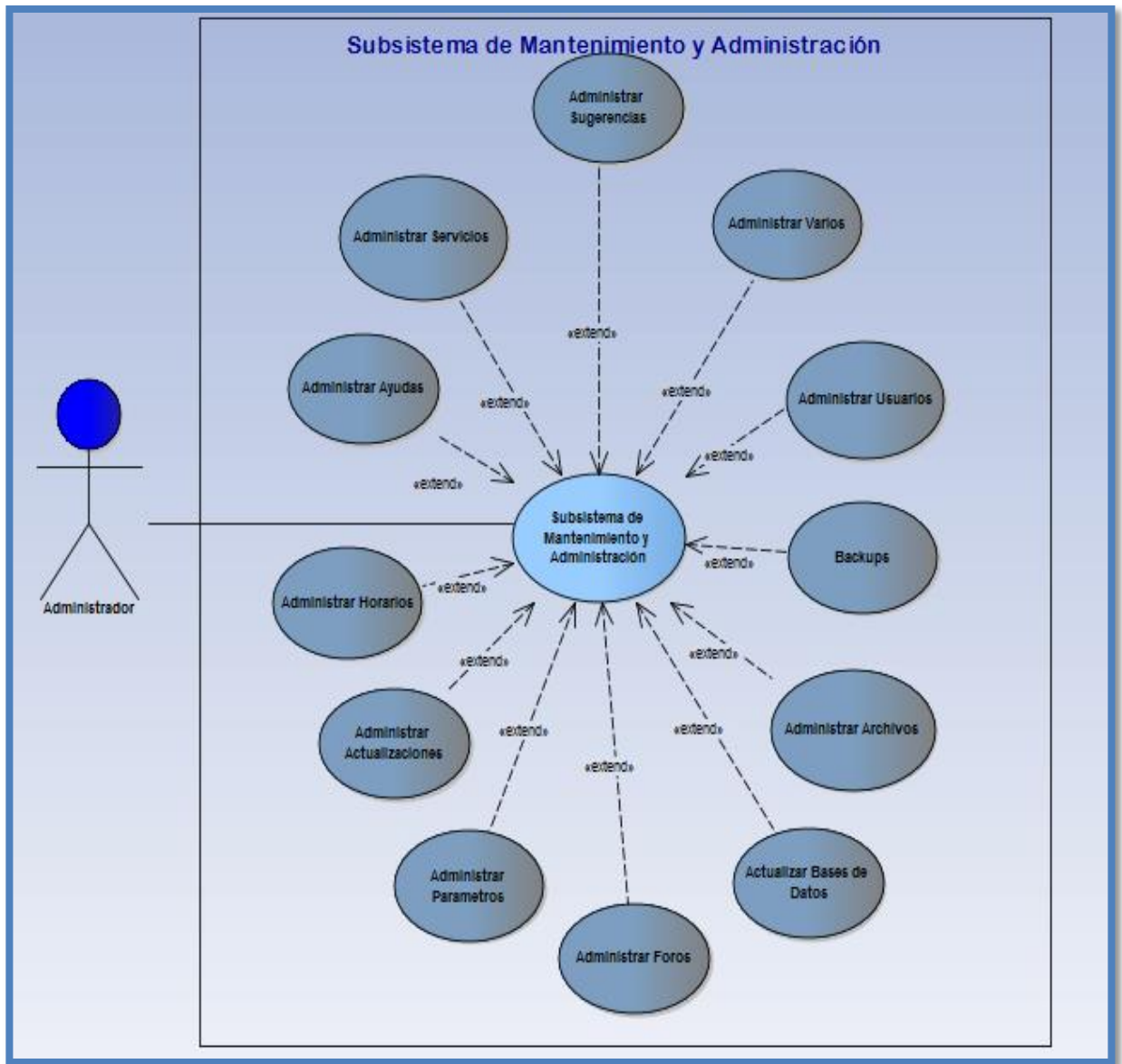


Figura 12. Diagramas de casos de uso: Imprimir Nota



4.1.1.3 Mantenimiento y administración de los portales EdiWEB y FIFQuiWEB.

Figura 13. Diagrama de casos de Uso: Subsistema de Mantenimiento y Administración.



4.1.2 Descripción de los casos de uso

4.1.2.1 Subir Plan Pdf

Tabla 1. Casos de Uso: Subir Plan.

TÍTULO CASO DE USO	FUNCIONES DE LOS CASOS DE USO
Subir Plan	Permite ingresar al servicio para subir el documento del plan en formato Pdf.
Seleccionar Programa	Si se encuentra inscrito en 2 programas académicos permite seleccionar a cual subir el documento.
Seleccionar Archivo	Permite realizar la búsqueda y selección del documento a subir desde su equipo.
Enviar Plan	Permite subir el documento seleccionado a el servidor del respectivo portal

Tabla 2. Casos de Uso: Solicitar Evaluador.

TÍTULO CASO DE USO	FUNCIONES DE LOS CASOS DE USO
Solicitar Evaluador	Permite ingresar al servicio para solicitar Evaluador para el plan de proyecto de grado.
Ver Documento	Permite ver el documento del plan por parte del director para solicitar evaluador o pedir correcciones.
Seleccionar Documento	Permite seleccionar un documento en el caso de que el autor subiera varios.
Realizar Solicitud	Permite al director del proyecto solicitar evaluador.

Retirar Solicitud	Permite al director del proyecto retirar la solicitud de evaluador en caso de que lo requiera.
-------------------	--

Tabla 3. Casos de Uso: Concepto Plan.

TÍTULO CASO DE USO	FUNCIONES DE LOS CASOS DE USO
Concepto Plan	Permite al evaluador del plan dar su concepto acerca de este.
Ver Documento	Permite ver el documento del plan por parte del evaluador para tomar su decisión y realizar sugerencias.
Seleccionar Documento	Permite seleccionar un documento en el caso de que el evaluador permitiera al autor subir nuevos planes.
Permitir subir Plan	Permite al director del proyecto solicitar evaluador.
Realizar Comentario	Este espacio permite al director del proyecto dar su opinión acerca del plan.
Tomar Decisión	Permite tomar alguna decisión acerca del plan, ya sea aprobar, aprobar o cancelar.

Tabla 4. Casos de Uso: Solicitar calificador.

TÍTULO CASO DE USO	FUNCIONES DE LOS CASOS DE USO
Solicitar Calificador	Permite ingresar al servicio para solicitar calificadores por parte del director de proyecto.
Ver Documento	Permite ver el documento final del plan de proyecto.

Realizar solicitud	Permite realizar la solicitud para la asignación de calificadoros para el proyecto.
Retirar Solicitud	Permite retirar la solicitud de calificadoros en caso de que lo requieras.

Tabla 5. Casos de Uso: Asignar Evaluador.

TÍTULO CASO DE USO	FUNCIONES DE LOS CASOS DE USO
Asignar Evaluador	Permite al comité de trabajos de grado ingresar al servicio para asignar evaluador para el plan de proyecto de grado.
Ver Documento	Permite ver el documento del plan por parte de los miembros del comité de trabajos de grado.
Realizar Asignación	Permite realizar la asignación del evaluador del plan.
Reemplazar Asignación	Permite reemplazar el evaluador del plan si es requerido.
Cancelar proyecto	Permite cancelar el proyecto si es necesario.

Tabla 6. Casos de Uso: Asignar calificador.

TÍTULO CASO DE USO	FUNCIONES DE LOS CASOS DE USO
Asignar Calificadores	Permite al comité de trabajos de grado ingresar al servicio para asignar calificadoros para el proyecto.

Ver Documento	Permite ver el documento del plan por parte de los miembros del comité de trabajos de grado.
Asignar Calificadores	Permite realizar la asignación del evaluador del plan.
Modificar Calificadores	Permite reemplazar los calificadores del proyecto si es requerido.
Cancelar proyecto	Permite cancelar el proyecto si es necesario.

Tabla 7. Casos de Uso: Evaluar Plan.

TÍTULO CASO DE USO	FUNCIONES DE LOS CASOS DE USO
Evaluar Plan	Permite al comité de trabajos de grado ingresar al servicio para evaluar el plan del proyecto.
Ver Documento	Permite ver el documento del plan al comité.
Seleccionar Decisión	Permite tomar decisiones acerca del plan, ya sea aplazar, aprobar y cancelar.
Realizar Observaciones	Permite realizar observaciones y sugerencias al plan.
Modificar Decisiones	Permite cancelar el proyecto si es necesario.

4.1.2.2 Imprimir Nota

Tabla 8. Casos de Uso: Buscador Imprimir Nota.

TÍTULO CASO DE USO	FUNCIONES DE LOS CASOS DE USO
Buscador Imprimir Nota	Permite ingresar al buscador de proyectos terminados.
Realizar Búsqueda	Permite buscar los proyectos terminados.
Seleccionar Criterio	Permite seleccionar el criterio de búsqueda del proyecto.
Seleccionar Proyecto	Permite seleccionar el proyecto de la lista que se obtiene de la búsqueda.
Imprimir Nota	Permite imprimir la nota en su respectivo formato.

Tabla 9. Casos de Uso: Imprimir Nota.

TÍTULO CASO DE USO	FUNCIONES DE LOS CASOS DE USO
Imprimir Nota	Permite ingresar al servicio para imprimir la nota del proyecto de grado.
Seleccionar Proyecto	Permite seleccionar un proyecto de la lista de los proyectos en espera de nota.
Ingresar Nota	Permite ingresar la nota del proyecto de grado.
Imprimir Nota	Permite imprimir la nota del proyecto en el formato especificado.

4.1.2.3 Mantenimiento y administración de los portales EdiWEB y FIFQuiWEB.

Tabla 10. Diagrama de Casos de Uso: Subsistema de Mantenimiento y Administración.

TÍTULO CASO DE USO	FUNCIONES PRIMARIAS DE LOS CASOS DE USO
Administrar Servicios	<p>Crear, modificar y eliminar servicios del portal.</p> <p>Establecer parámetros.</p>
Administrar Sugerencias	<p>Revisar todas las sugerencias para atender o eliminar.</p>
Administrar Varios	<p>Ejecutar labores de mantenimiento que no clasifican en otras categorías.</p> <p>Llevar el contador de visitas a cero.</p> <p>Insertar, cambiar y eliminar banner.</p>
Administrar Foros	<p>Ver información y eliminar conversaciones de los usuarios en foros.</p>
Administrar Usuarios	<p>Crear usuarios manualmente.</p> <p>Cambiar parámetros.</p> <p>Cambiar clave.</p> <p>Autorizar servicios a categorías y perfiles.</p> <p>Crear, modificar, eliminar categorías y perfiles.</p> <p>Enviar Correos.</p>
Administrar Ayudas	<p>Crear, modificar y eliminar las ayudas en el portal de</p>

	ayudas de las escuelas.
Administrar Parámetros	Crear, modificar y eliminar los diferentes parámetros del portal.
Backups	Generar copias de la base de datos División y Diamante.
Administrar Archivos	Ver información, descargar y eliminar archivos cargados por los usuarios.
Administrar Horarios	Cambiar, restaurar, ver incoherencias y cambios de horarios. Crear grupo de materia.
Administrar Actualizaciones	Ver historial y estadísticas de cambios del portal web.
Actualizar Base de Datos	Actualizar tablas de datos de División y Diamante. Ver contadores de registros.

4.1.3 Diseño y análisis

4.1.3.1 Diagrama Entidad/Relación

Figura 14. Diagrama Entidad/Relación: Subir Plan Pdf

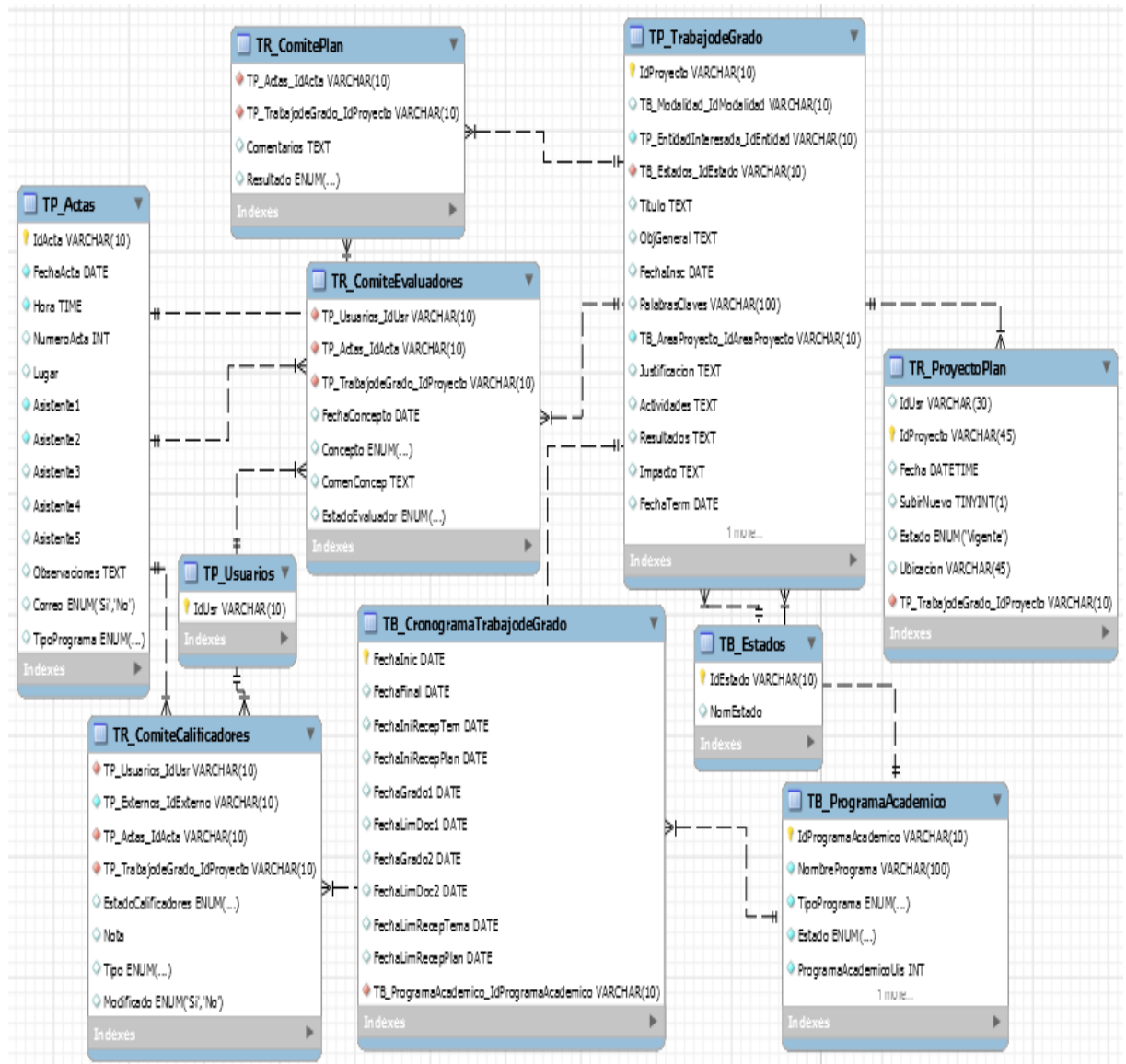


Figura 15. Diagrama Entidad/Relación: Imprimir Nota

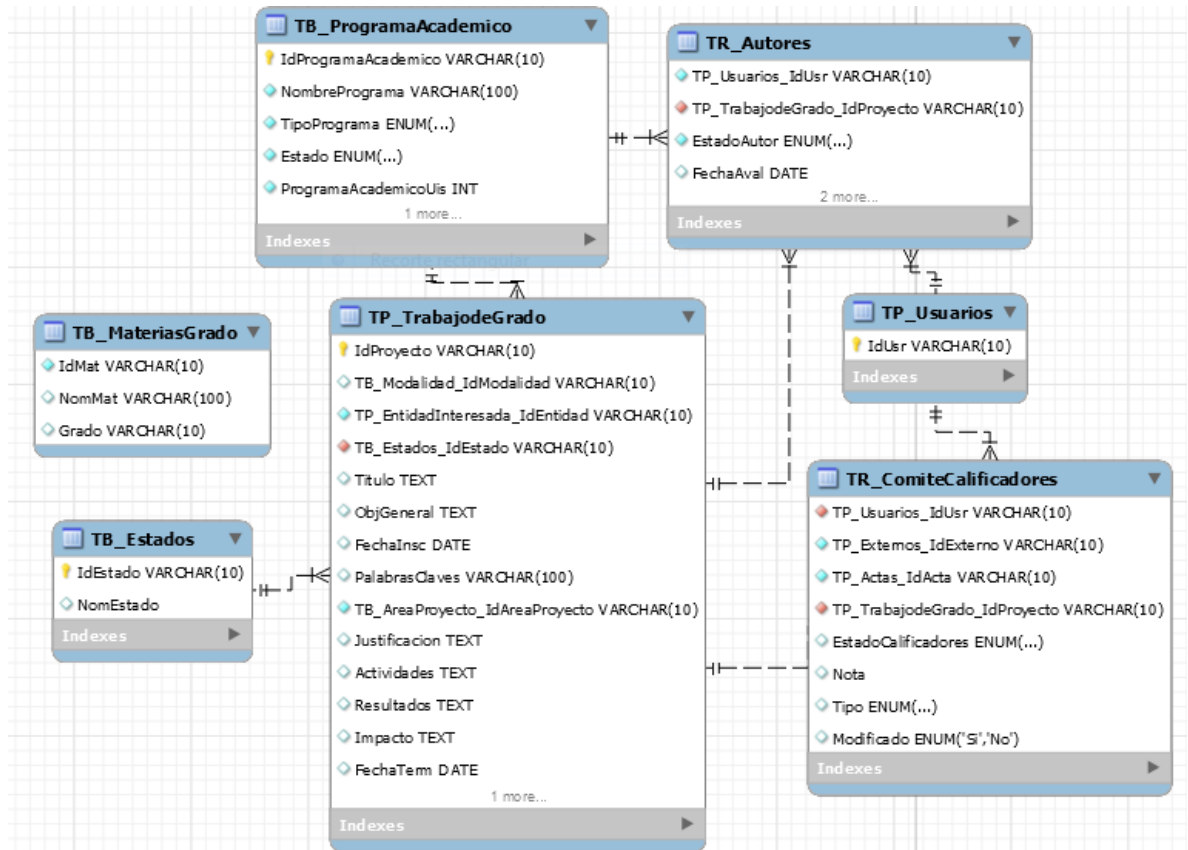
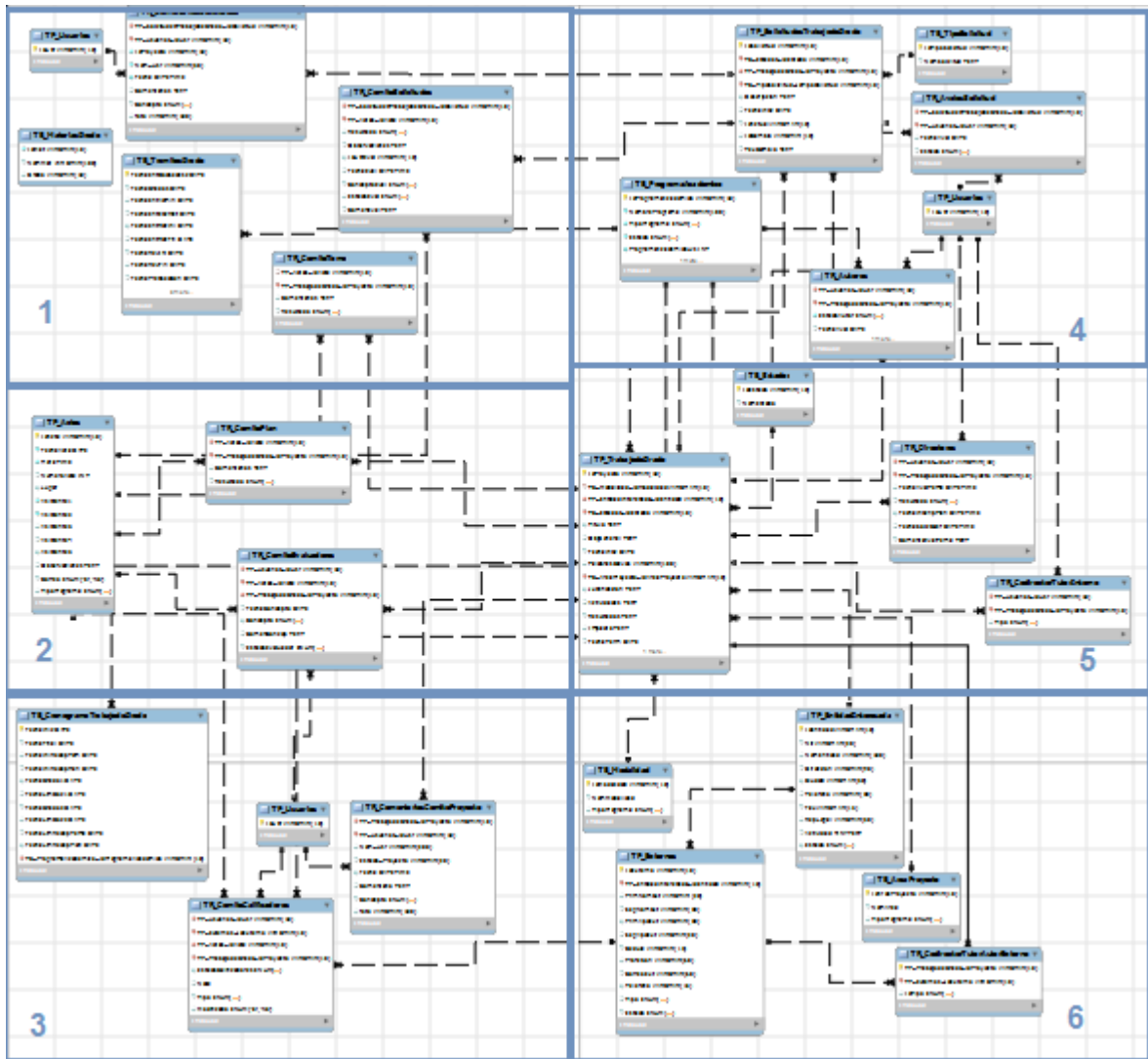
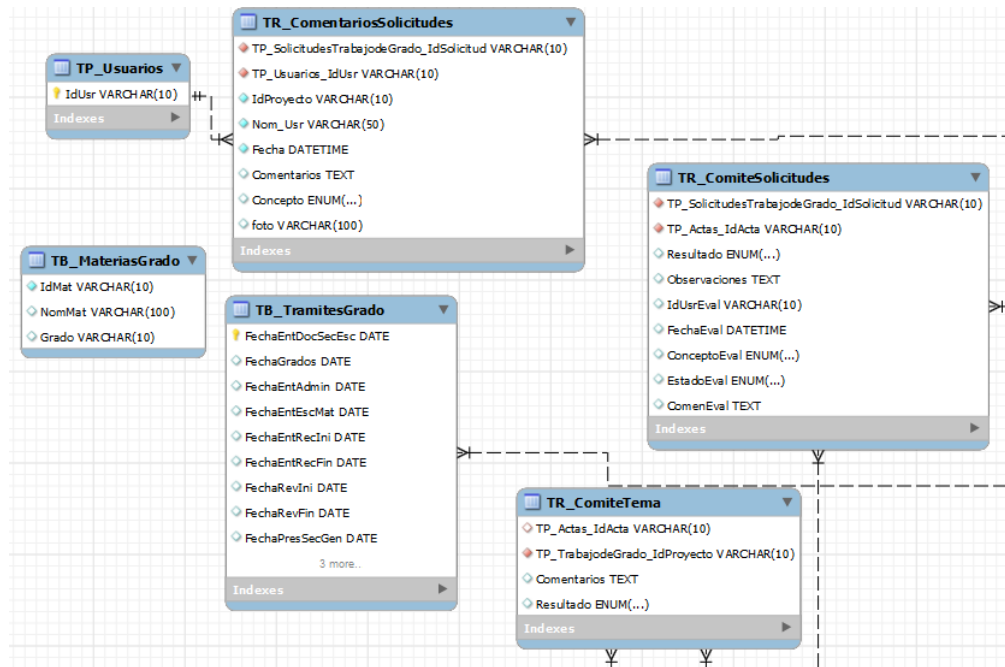


Figura 16. Diagrama Entidad/Relación: No mostrar decisiones día actual

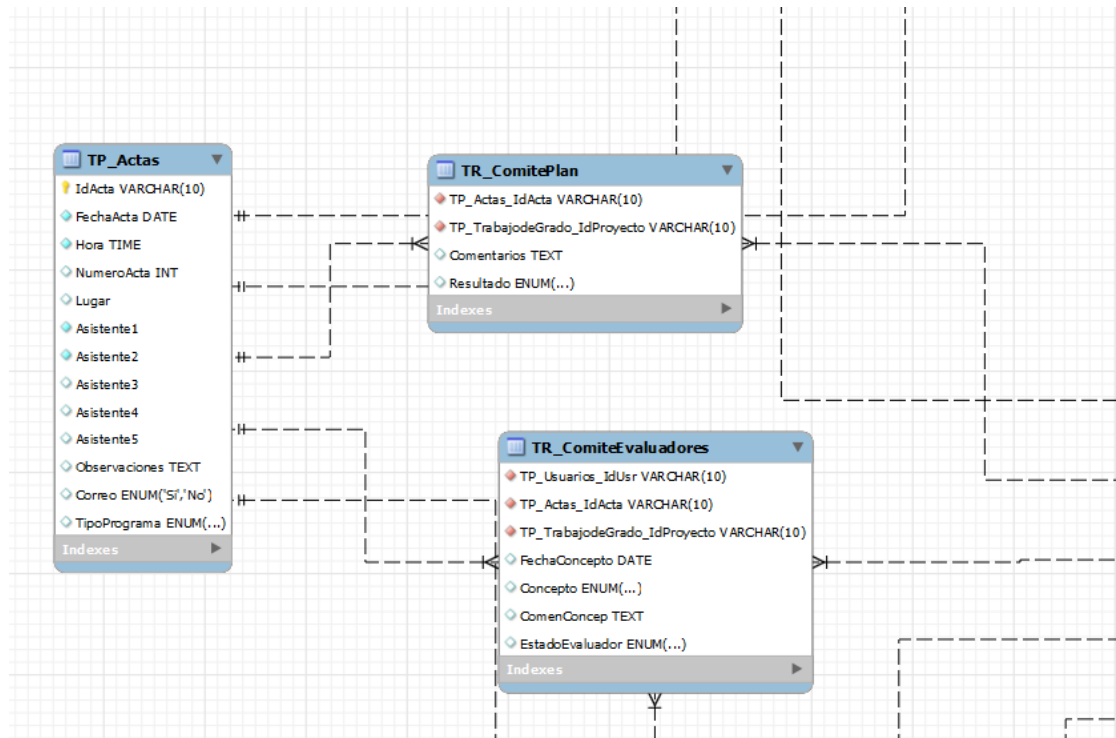
Imagen completa del diagrama



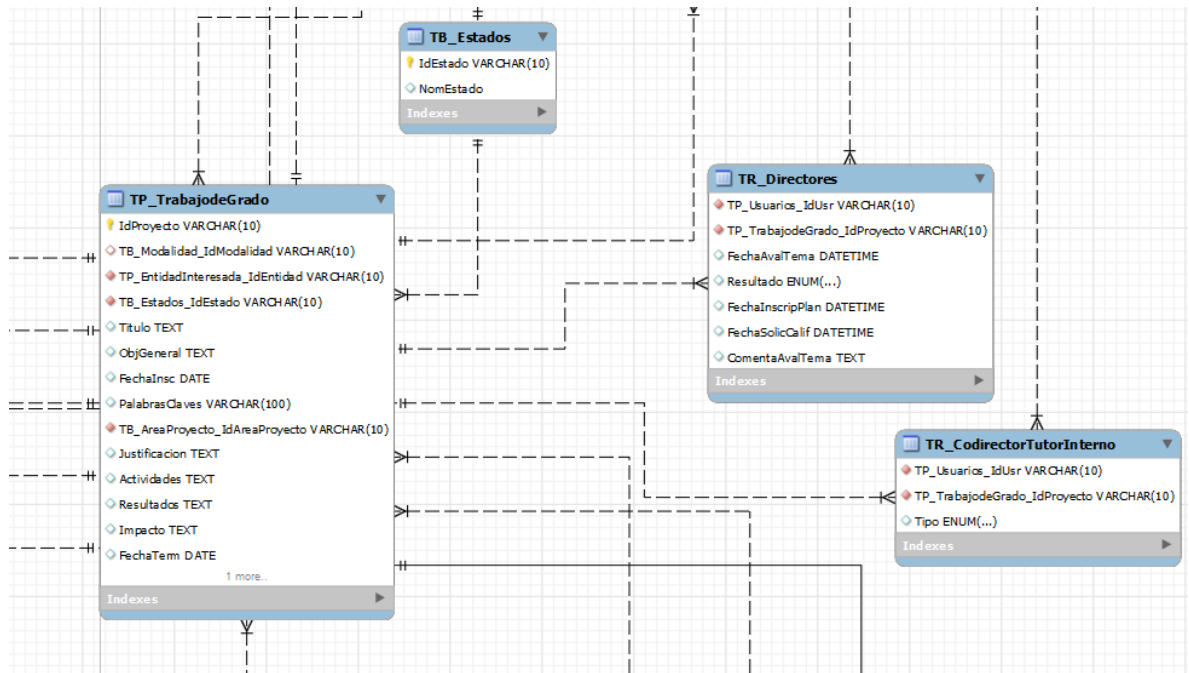
Sección 1



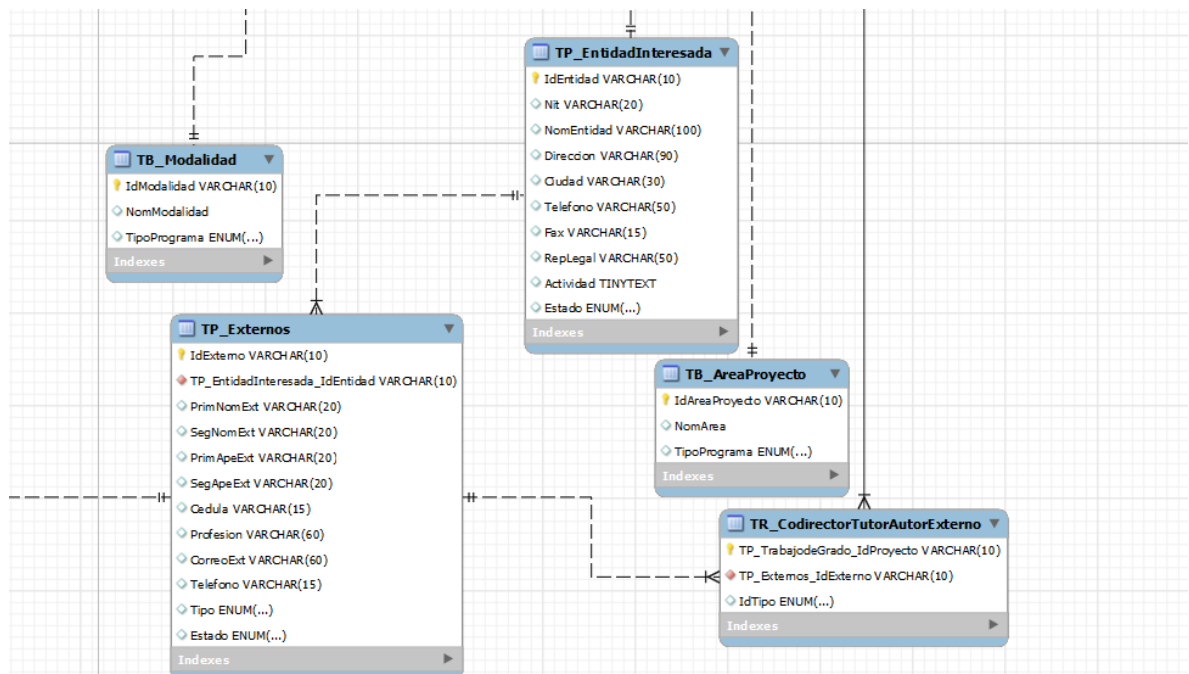
Sección 2



Sección 5



Sección 6



4.1.3.2 Descripción de las entidades A continuación se presentan las entidades que conforman los diagramas de Entidad/Relación que hacen parte del correcto funcionamiento de los servicios creados para los portales de la escuela de Diseño industrial y la decanatura de fisicoquimicas.

Tabla 11. Descripción de las entidades.

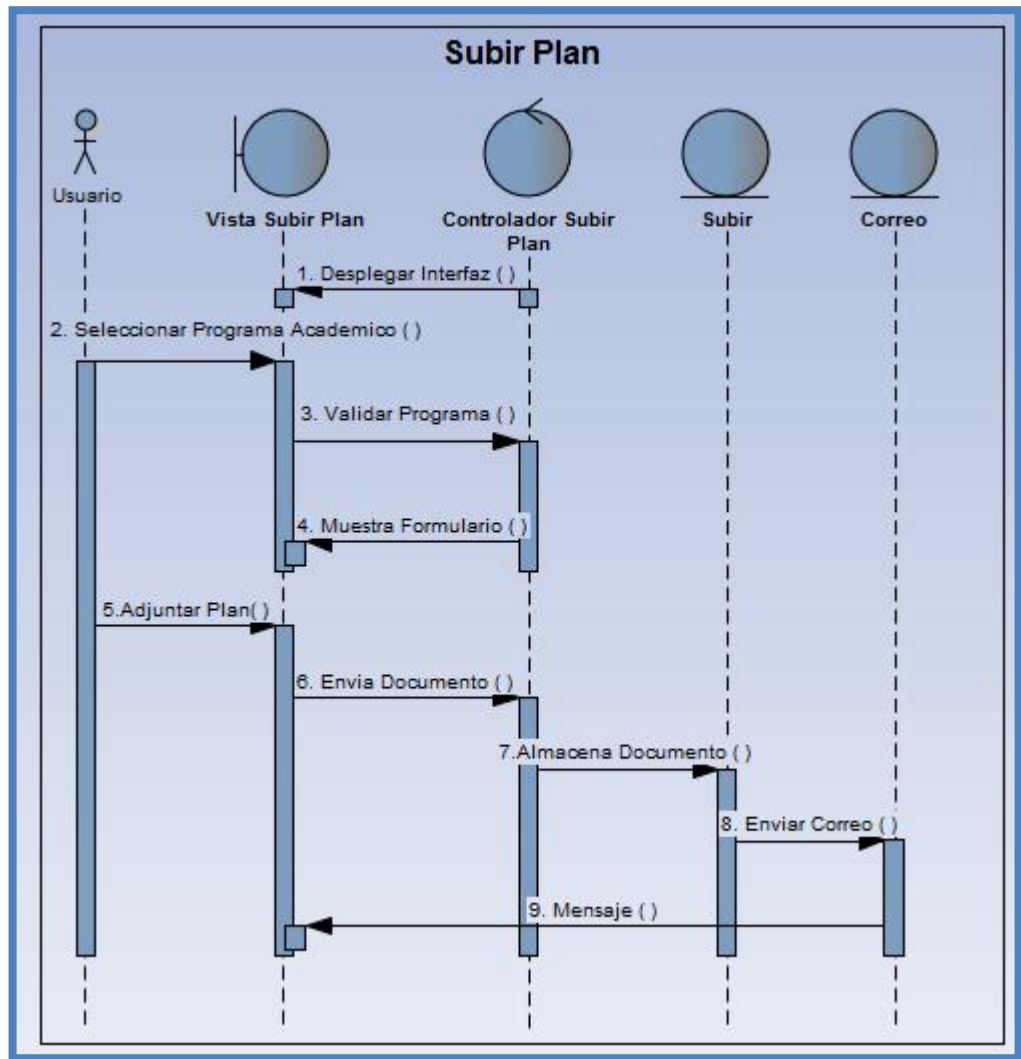
ENTIDAD	DESCRIPCIÓN
TB_ProgramaAcademico	Contiene la información de los programas académicos.
TR_Autores	Contiene la información para relacionar los autores de proyecto de grado
TB_Estados	Contiene el identificador y el nombre de los posibles estados que puede tener un proyecto.
TB_MateriasGrado	Contiene la información de de las materias requisitos para grado.
TP_Usuarios	Contiene toda la información de los usuarios de los portales.
TR_ComiteCalificadores	Permite relacionar la información de los calificadores asignados con el trabajo de grado.
TP_TrabajosDeGrado	Contiene la información correspondiente a los trabajos de grado.
TP_Actas	Contiene la información del acta generada por la sesión del comité de trabajos de grado
TR_ComitePlan	Permite relacionar la información del plan con el acta en el cual se toma decisión sobre el

	plan.
TR_ComiteEvaluadores	Permite relacionar la información del evaluador con el acta.
TR_ProyectoPlan	Contiene la información para relacionar el documento del plan con la información del proyecto de grado.
TR_ComiteSolicitudes	Permite relacionar la información de las solicitudes y la decisión del comité sobre la misma.
TP_SolicitudesTrabajoGrado	Contiene la información de las solicitudes realizadas por los autores de los proyectos de grado.
TP_Externos	Contiene la información de los tutores, autores y entidades externas a la universidad.
TR_AvalesSolicitud	Contiene la información que permite relacionar las solicitudes con los usuarios.
TB_TipoSolicitud	Contiene la información de todas las posibles solicitudes que pueden realizar los autores.
TR_Directores	Contiene la información que permite relacionar la información del proyecto con el director de este.
TP_EntidadInteresada	Contiene la información de la entidad en la cual se realizara el proyecto.
TB_AreaProyecto	Contiene la información del área en la que se desarrollara el proyecto.

TB_Modalidad	Contiene la información del tipo de modalidad en la que se realizara el proyecto.
TR_ComiteTema	Contiene la información que permite relacionar el tema con las decisiones que el comité toma sobre este.
TR_ComentariosComiteProyecto	Contiene la información que permite relacionar los comentarios que toma el comité con el respectivo proyecto.
TR_CodirectorTutorInterno	Permite relacionar la información de un codirector con su respectivo proyecto.
TR_CodirectorTutorAutorExterno	Permite relacionar la información de un tutor o autor externo con su respectivo proyecto

4.1.3.3 Modelo de procesos del sistema

Figura 17. Diagrama de secuencias: Subir Plan

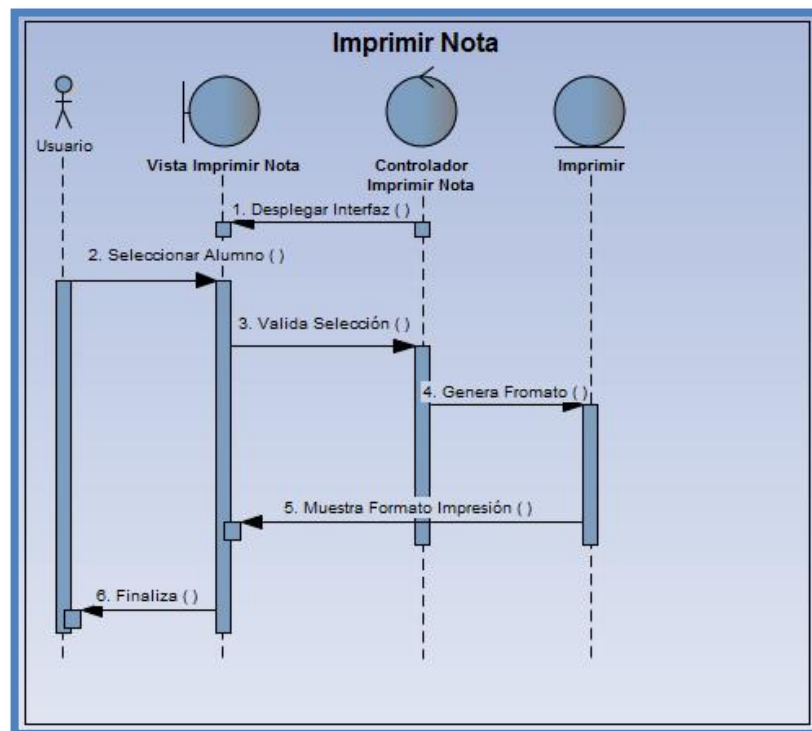


Descripción del diagrama de secuencias: Subir Plan

1. El controlador Subir Plan despliega la interfaz vista subir plan la cual es percibida por el usuario.
2. El Usuario puede seleccionar a que programa académico desea subir el plan si se encuentra matriculado en más de un programa académico.
3. El controlador subir plan valida la selección realizada por el Usuario acerca de a qué programa académico pertenece.

4. El controlador subir plan muestra el formulario para subir el documento del plan de proyecto al usuario.
5. La vista subir plan permite al Usuario seleccionar desde su pc el documento del plan para adjuntarlo y subirlo.
6. La vista subir plan una vez contiene el documento lo envía al controlador subir plan.
7. El controlador subir plan envía toda la información a la base de datos subir quien almacenara la respectiva información del documento.
8. La base de datos subir envía un correo a los implicados con el respectivo plan, director, autores, evaluador, comité o calificadores.
9. La base de datos correo envía un mensaje satisfactorio o no dependiendo la situación a la vista subir plan.

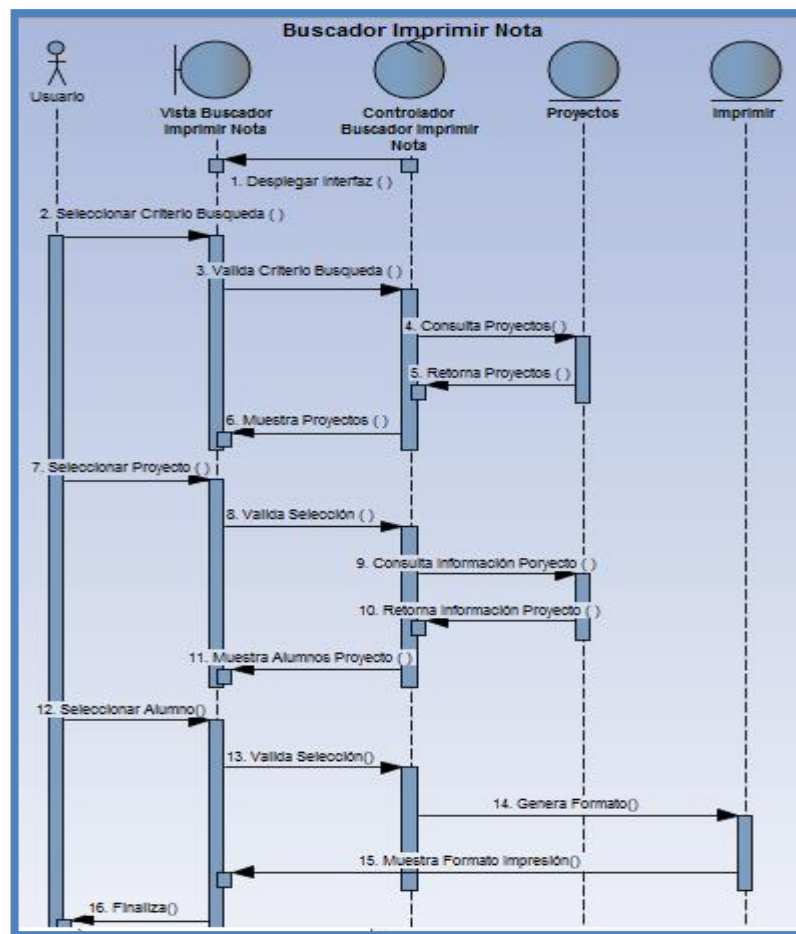
Figura 18. Diagrama de secuencias: Imprimir Nota



Descripción del diagrama de secuencias: Imprimir Nota

1. El controlador Imprimir Nota despliega la interfaz al Usuario donde puede ver el formato de la nota del plan para cada uno de los autores.
2. La vista Imprimir Nota permite al Usuario escoger cuál nota de los dos autores desea imprimir.
3. El controlador Imprimir Nota valida la selección realizada por el Usuario.
4. El controlador Imprimir Nota genera el formato para presentar la información del proyecto.
5. La vista Imprimir Nota muestra al usuario el formato con la nota y la información del proyecto.
6. El Usuario realiza la impresión y finaliza el proceso.

Figura 19. Diagrama de secuencias: Buscador Imprimir Nota

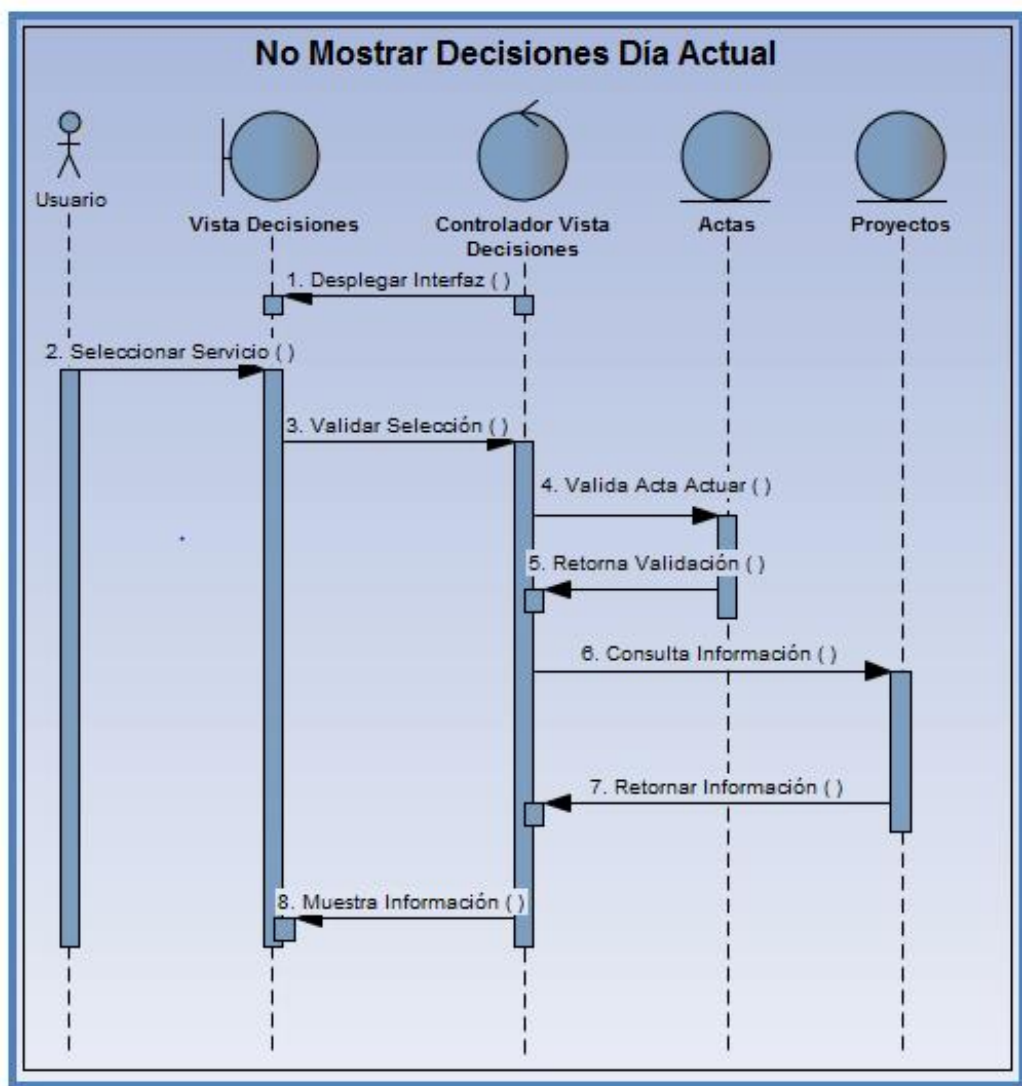


Descripción del diagrama de secuencias: Buscador Imprimir Nota

1. El controlador Buscador Imprimir Nota despliega la interfaz del buscador de proyectos al Usuario.
2. El Usuario realiza la selección del criterio por el cual desea realizar la búsqueda del proyecto.
3. El controlador Buscador Imprimir Nota recibe la selección del usuario por medio de la vista Buscador Imprimir Nota y valida la selección del criterio.
4. El controlador Buscador Imprimir de acuerdo con el criterio de búsqueda realiza una consulta en la base de datos Proyecto.
5. La base de datos Proyecto retorna al Controlador Buscador Nota los proyectos que cumplen con el criterio de búsqueda.
6. El controlado despliega la vista Buscador Imprimir Nota con los proyectos de interés al usuario.
7. El Usuario selecciona el proyecto que es de su interés en la Vista buscador Imprimir Nota.
8. La Vista buscador Imprimir Nota envía al Controlador la información del Proyecto seleccionado, el Controlador buscador Imprimir Nota Valida la selección realizada por el Usuario.
9. El controlador Buscador Imprimir Nota consulta en la base de datos proyecto la información del proyecto seleccionado.
10. La base de datos proyecto retorna al Controlador Buscador Imprimir Nota la información del proyecto seleccionado.
11. El Controlador Buscador imprimir Nota despliega la Vista Buscador Imprimir Nota con el formato que contiene la nota del proyecto para cada uno de los autores.
12. El Usuario selecciona de la Vista Buscador Imprimir Nota el autor al que desea imprimir la nota del proyecto.
13. El Controlador Buscador Imprimir Nota valida la selección realizada por el Usuario desde la Vista Buscador Imprimir Nota.

14. El controlador Buscador Imprimir Nota genera el formato de impresión de la respectiva nota.
15. El Controlador Buscador Imprimir Nota muestra en la Vista Buscador Imprimir Nota el formato con la información de la respectiva nota del proyecto.
16. El Usuario realiza la impresión del respectivo formato y finaliza el proceso.

Figura 20. Diagrama de secuencias: No mostrar decisiones día actual



Descripción del diagrama de secuencias: No mostrar Decisiones Día Actual

1. El Controlador Vista Decisiones despliega la interfaz vista Decisiones al Usuario.
2. El usuario selecciona el servicio para ver la información del proyecto.
3. El Controlador Vista Decisiones valida la selección realizada por parte del Usuario en la Vista Decisiones.
4. El controlador Vista Decisiones valida la información de la base de datos Actas, en la cual se encuentra la información de las sesiones del comité,
5. La base de datos Actas retorna al Controlador Vista Decisiones la validación de la para las actas con fechas diferentes al día actual.
6. El Controlador Vista Decisiones consulta en la base de datos proyecto la información correspondiente a este.
7. La base de datos Proyecto retorna la información del proyecto al Controlador Vista Decisiones.
8. El controlador Vista Decisiones despliega la interfaz vista Decisiones con la información del proyecto al usuario.

4.1.3.4 Implementación, Implantación y Pruebas Generales Para desarrollar la correcta implementación de los nuevos servicios creados se utilizaron las siguientes herramientas:

- Lenguaje de programación interpretado orientado a objetos, JavaScript.
- Lenguaje Java.
- Librería Javascript JQuery y AJAX.
- Manejador de Base de datos.
- Lenguaje de programación orientado a la web, JSP.
- Lenguaje de marcado para la elaboración de páginas web, HTML.
- NetBeans, IDE para desarrollar las clases de Java y los archivos JSP.
- Servidor Apache Tomcat.

- Estándares de diseño con, CSS presentados en ELISE.

Haciendo uso de estas herramientas, y con la asesoría y seguimiento del director de proyecto se realizó el diseño de los nuevos servicios y se estructuraron los datos que se utilizarían para la generación del primer prototipo. Para refinar las interfaces se tuvo en cuenta los comentarios y sugerencias de los demás miembros del grupo Calumet para así manejar un estándar de estilos y de colores que identifiquen cada uno de los portales.

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 Escuela de Ingeniería Civil. 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 los resultados correspondieran 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.

4.2 SOPORTE Y ADMINISTRACIÓN

4.2.1 Actividades de soporte a usuarios Los portales de la escuela de diseño industrial y la decanatura de fisicoquímicas poseen un servicio de consultas y sugerencias, por medio del cual se responden preguntas y se resuelven los problemas que se presentan a los usuarios. Diariamente se reciben consultas de estudiantes, profesores y usuarios en general, que necesitan asesoría en el funcionamiento de servicios o soporte en el manejo de su usuario y contraseña. En ocasiones estos problemas son resueltos de manera presencial, cuando el usuario acude directamente a las oficinas del Grupo Calumet y es atendido directamente para resolver sus inquietudes e inconvenientes prestando 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.

4.2.2 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.
- Realizar copias de seguridad periódicas de las bases de datos Diamante y División.

- 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

Con el fin de verificar y garantizar el correcto desarrollo y funcionamiento de los servicios realizados en los portales de la escuela de diseño industrial y la decanatura de fisicoquimicas, se aplicaron las pruebas que se muestran a continuación:

5.1 PRUEBAS DE VERIFICACIÓN

Con esta actividad se tiene como objetivo realizar la completa verificación del sistema, modulo por modulo, componente por componente por medio de diferentes técnicas, para así poder detectar fallas en el mismo, antes de su puesta en marcha.

La forma en que se realizan las pruebas de verificación es elaborando actividades para cada uno de los servicios desarrollados, estas pruebas consisten en: probar que los campos que exijan campos obligatorios no se acepten vacíos, examinar que los campos contengan el tipo de dato que le corresponde, validar que si se permite cierto formato para algún documento, de no ser el formato permitido lo restrinja.

5.1.1 Pruebas por componente Con esta prueba como su nombre lo indica se examina componente por componente, por consiguiente se realizó para cada caso de uso de cada servicio desarrollado y descrito anteriormente. De igual manera en las mejoras a los servicios ya existentes.

Al obtener unos resultados satisfactorios se logra comprobar el correcto desarrollo y funcionamiento de los servicios creados y de los arreglos realizados en los portales, confirmando así el cumplimiento de los objetivos del proyecto.

5.2 PRUEBAS DE INTEGRACIÓN

Las pruebas de integridad son pruebas a los métodos, funciones y sentencias utilizados en el desarrollo del proyecto, después de realizadas estas pruebas se comprobó el correcto funcionamiento de todos los servicios creados y corregidos, estas pruebas también se realizaron para las consultas a las bases de datos y se aseguró de que los datos fueran siempre los requeridos.

5.3 PRUEBAS DE VALIDACIÓN

Con las pruebas de validación se realiza un proceso de revisión con el cual se verifica que los servicios creados en el desarrollo del proyecto cumplen con las especificaciones y requisitos planteados en los objetivos, durante el desarrollo del proyecto al realizar las pruebas de validación se presentaron novedades las cuales fueron tratadas y solucionadas cumpliendo así con el correcto funcionamiento de los servicios planteados en los objetivos en su totalidad.

6. CONCLUSIONES

- Con el servicio creado para imprimir la nota del proyecto de grado, se logra agilizar este proceso y se aumenta la confiabilidad ya que los datos contenidos en el respectivo documento donde se encuentra la nota del proyecto eran ingresados uno a uno por parte del administrador, ahora el formato es generado de forma automática con la información que existe en las bases de datos.
- Gracias a los arreglos realizados en los servicios que muestran la información del proyecto de grado, aumento la confiabilidad, ya que los cambios de las decisiones que se realicen durante la sesión del comité de trabajos de grado no se mostraran, la información será visible cuando la sesión finalice.
- El servicio desarrollado para subir el plan en formato pdf, el cual se agregó al módulo de trabajos de grado ayudo a complementar los portales de la escuela de diseño industrial y la decanatura de ingeniería físicoquímicas, ya que la entrega del plan es un requisito indispensable, y con este servicio se facilita el proceso de evaluación y manejo del plan proyecto por parte del director, el evaluador y el comité de trabajos de grado.
- Como conclusión final he aportado al mejoramiento de los portales de la escuela de diseño industrial y la decanatura de ingenierías físicoquímicas con el proyecto que realice ya que con los servicios creados se ha generado satisfacción en el cliente.

7. RECOMENDACIONES

- Motivar a toda la comunidad a ver la información de los proyectos de grado en sus respectivos portales, desde el servicio de mis proyectos dirigidos, mi proyecto o Buscador privado dependiendo quien lo utilice, desde allí pueden estar informados de todo lo que sucede con respecto a su trabajo de grado, como lo es la información del tema, del plan, de los evaluadores, de los calificadores y estarán informados de las decisiones que se tomen del proyecto y de las solicitudes que realicen a los proyectos.
- Promover la utilización del servicio de imprimir la nota de proyecto de grado en el formato especificado, ya que este servicio agiliza y le da más confianza al desarrollo del proceso, y facilita esta labor a los administradores de los portales web.
- Promover la utilización del servicio de Subir el plan, debido a que este servicio facilita todo el proceso de la evaluación del plan de proyecto de grado, ya que el director, el evaluador y los miembros del comité pueden ver el documento del plan de proyectos en todo momento para tomar decisiones sobre este.

BIBLIOGRAFIA

- Arquitectura cliente-servidor. [en línea]. [Citado el 22 de agosto, 2014]. Disponible en internet: <http://g701giadar.wikispaces.com/Arquitctura+Cliente+Servidor>.
- Elise Library. [en línea]. [Citado el 20 de Septiembre, 2014]. Disponible en internet: <http://cormoran.uis.edu.co/eisi/Estandar>.
- Java. [en línea]. [Citado el 12 de Noviembre, 2014]. Disponible en internet: <http://docs.oracle.com/javase>.
- JOHNSON, James. Bases de datos: Modelos lenguajes y diseño. 1ª ed. Oxford, 2000.
-
- Libros desarrollo web. [en línea]. [Citado el 10 de Octubre, 2014]. Disponible en internet: <http://librosweb.es/>.
- Manuales básicos de JSP. [en línea]. [Citado el 14 de Noviembre, 2014]. Disponible en internet: <http://manuales.dgsca.unam.mx/jsp>.
- MySQL. [en línea]. [Citado el 19 de Octubre, 2014]. Disponible en internet: <http://www.mysql.com>.
- SCHMULLER, Joseph. Aprendiendo UML en 24 horas. 1ª ed. México: Alhambra Mexicana S.A, 2000.
- STALLINGS, William. Sistemas Operativos: Aspectos Internos y Principios de Diseño. Madrid: Pearson Prentice Hall, 2005.

- Tutoriales HTML. [en línea]. [Citado el 25 de Septiembre, 2014]. Disponible en internet: <http://www.htmlquick.com/es/tutorials.html>.
- Tutoriales Java. [en línea]. [Citado el 27 Junio, 2014]. Disponible en internet: <http://www.programacion.com/java/tutorial>.
- WEITZENFELD, Alfredo. Ingeniería de Software Orientada a Objetos con UML, JAVA e Internet. Thomson International, 2003.

ANEXOS

Anexo A. Arquitectura Cliente/Servidor

1. TIPOS DE ARQUITECTURA CLIENTE SERVIDOR

Para asumir las diferentes necesidades existen varios tipos de arquitectura cliente servidor los cuales se ajustan para responder de la mejor forma a las estadísticas, restricciones y requerimientos del problema en cuestión, los tipos de esta arquitectura son:

1.1 Por tamaño de componentes El tipo de clasificación por tamaño de componentes se basa en los grados de libertad que brinda el modelo Cliente/Servidor para balancear la carga de procesos entre los diferentes niveles: presentación, aplicación y base de datos. Dependiendo de qué segmento de las capas de software tenga que soportar la mayor o menor carga de procesamiento se habla de unos subtipos que son:

- **Fat client** En este tipo de arquitectura el peso de la aplicación es ejecutada en el cliente, es decir, el nivel de presentación y el nivel de aplicación corren en un único proceso cliente, y el servidor solo tendrá que realizar las funciones que provee un administrador de base de datos.
- **Fat server** En este tipo de arquitectura se presenta el caso opuesto al anterior, el proceso cliente es restringido solamente a la presentación de la interfaz de usuario, mientras que todo el peso de la aplicación corre por el lado del servidor de aplicación.

Como consecuencia este tipo de arquitectura presenta una flexibilidad mayor para desarrollar una gran variedad de aplicaciones, incluyendo los sistemas que tengan que realizar muchas transacciones con servidores.

1.2 Por naturaleza de servicio

Dependiendo la naturaleza de servicio existen varios subtipos de arquitectura que son:

- **Servidores de ficheros** Con un servidor de archivos, un cliente lo que hace es requerimientos de los mismos sobre una red. Esta es una forma muy primitiva de servicios de datos, la cual necesita intercambio de muchos mensajes sobre una red para hallar el dato requerido. Los servidores de archivos usan recursos compartidos sobre la red y son necesarios para crear repositorios de documentos, imágenes y archivos grandes sobre la red.
- **Servidores de bases de datos** En el subtipo de arquitectura de servidores de bases de datos, se tienen en cuenta las distintas normas de SQL las cuales dicen que el servidor debe proveer un acceso compartido a los datos con los mecanismos de protección necesarios, así como proveer mecanismos para seleccionar resultados dentro de un conjunto de datos, posibilitando un ahorro en procesos de comunicación. El servidor debe también proveer mecanismos de concurrencia, seguridad y consistencia de datos, basados principalmente en el concepto de transacción en el que todo se realiza, y por lo tanto se hace permanente, anularse la transacción en caso de fallos y errores.
- **Servidores de transacciones** Con un servidor de transacciones el proceso cliente llama a funciones, procedimientos o métodos que residen en el servidor, ya sea que se trate de un servidor de bases de datos o un servidor de aplicaciones. Lo importante es que el intercambio a través de la red se realiza mediante un único mensaje de solicitud/respuesta, es decir, independientemente de que se necesite ejecutar una o más funciones, una

o más instrucciones o sentencias SQL, estas son agrupadas en una unidad lógica llamada transacción.

- **Servidores de objetos** Con un servidor de objetos, las aplicaciones Cliente/Servidor son escritas como un conjunto de objetos que se comunican. Los objetos cliente se comunican con los objetos servidores usando un Object Request Broker (ORB). El cliente invoca un método de un objeto remoto. El ORB localiza el método del objeto en el servidor, y lo ejecuta para devolver el resultado al objeto cliente. Los servidores de objetos deben soportar concurrencia.
- **Servidores web** La primera aplicación cliente servidor que cubre todo el planeta es el World Wide Web. Este nuevo modelo consiste en clientes simples que hablan con servidores Web. Un servidor Web devuelve documentos cuando el cliente pregunta por el nombre de los mismos. Los clientes y los servidores se comunican usando un protocolo basado en RPC, llamado HTTP. Este protocolo define un conjunto simple de comandos, los parámetros son pasados como cadenas y no provee tipos de datos. La Web y los objetos distribuidos están comenzando a crear un conjunto muy interactivo de computación Cliente/Servidor.

2. MODELOS DE LA ARQUITECTURA CLIENTE/SERVIDOR

Una de las clasificaciones mejor conocidas de las arquitecturas Cliente/Servidor se basa en la idea de planos, la cual es una variación sobre la división o clasificación por tamaño de componentes. Esto se debe a que se trata de definir el modo en que las prestaciones funcionales de la aplicación serán asignadas, y en qué proporción, tanto al cliente como al servidor.

2.1 Nivel de software Este enfoque o clasificación es el más generalizado y el que más se ajusta a los enfoques modernos, dado que se fundamenta en los componentes lógicos de la estructura Cliente/Servidor y en la madurez y popularidad de la computación distribuida. Por ejemplo, esto permite hablar de servidores de aplicación distribuidos a lo largo de una red, y no tiene mucho sentido identificar a un equipo de hardware como servidor, sino más bien entenderlo como una plataforma física sobre la cual pueden operar uno o más servidores de aplicaciones.

- **modelo cliente/servidor 2 capas** Esta estructura se caracteriza por la conexión directa entre el proceso cliente y un administrador de bases de datos.
- **modelo cliente/servidor 3 capa** Esta estructura se caracteriza por elaborar la aplicación en base a dos capas principales de software, más la capa correspondiente al servidor de base de datos. Al igual que en la arquitectura de dos capas, y según las decisiones de diseño que se tomen, se puede balancear la carga de trabajo entre el proceso cliente y el nuevo proceso correspondiente al servidor de aplicación.

2.2 Nivel de hardware Esta clasificación del modelo Cliente/Servidor se basa igualmente en la distribución de los procesos y elementos entre sus componentes, pero centrándose en la parte física del mismo, en el que la administración de la interfaz gráfica se asocia a los clientes PC y la seguridad e integridad de los datos quedan asociados a ambientes mainframe o por lo menos a servidores locales y/o centrales.

- **modelo cliente / servidor 2 capas** Los clientes son conectados vía LAN a un servidor de aplicaciones local, el cual, dependiendo de la aplicación puede dar acceso a los datos administrados por él.
- **modelo cliente / servidor 3 capas** Los clientes son conectados vía LAN a un servidor de aplicaciones local, el cual a su vez se comunica con un servidor central de bases de datos. El servidor local tiene un comportamiento dual, dado que actúa como cliente o servidor en función de la dirección de la comunicación.