

**EXPLORACIÓN DEL FRAMEWORK VAADIN PARA EL DESARROLLO
DE UN GESTOR DE TAREAS PARA LA ASIGNATURA
INGENIERÍA DEL SOFTWARE I**

**MARÍA FERNANDA LASPRILLA SUÁREZ
SEBASTIAN NIÑO PEÑALOZA**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
BUCARAMANGA**

2011

**EXPLORACIÓN DEL FRAMEWORK VAADIN PARA EL DESARROLLO
DE UN GESTOR DE TAREAS PARA LA ASIGNATURA
INGENIERÍA DEL SOFTWARE I**

**MARÍA FERNANDA LASPRILLA SUÁREZ
SEBASTIAN NIÑO PEÑALOZA**

**Proyecto de grado presentado como requisito parcial
Para optar al título de Ingeniero de Sistemas**

**Director
MSc Fernando Antonio Rojas Morales**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
BUCARAMANGA**

2011

A Dios por darme sabiduría y fortaleza para lograr este triunfo.

A mi madre Claudia por todo su amor y enseñanzas.

A mi padre Luis Alejandro por su apoyo y ayuda a lo largo de mi carrera.

A mis hermanos Alejandro y Camilo por su comprensión y apoyo.

A mi tía María del Carmen por la ayuda que me ha brindado para salir adelante.

A mis amigos que siempre han brindado su cariño, comprensión y me han ayudado en todos los momentos.

A mi compañero de grado Sebastian por todos los momentos que compartimos.
Gracias por tu cariño.

María Fernanda Lasprilla Suárez.

A Dios por darme sabiduría y fortaleza para lograr mis metas.

A mi madre Yolanda por su gran apoyo y recomendaciones

A mi padre Juan Carlos por su incondicional ayuda y soporte.

A mi hermana Andrea por su comprensión.

A mi compañera de grado María Fernanda, gracias por todo tu apoyo, cariño y comprensión durante todo este tiempo. Sin ti, nada de esto hubiera sido posible.

Sebastian Niño Peñaloza

TABLA DE CONTENIDO

INTRODUCCIÓN	17
1. ESPECIFICACIONES DEL PROYECTO	18
1.1 TÍTULO	18
1.2 MODALIDAD	18
1.3 DIRECTOR	18
1.4 AUTORES	18
2. PLANTEAMIENTO DEL PROYECTO	19
2.1 OBJETIVOS	19
2.1.1 OBJETIVO GENERAL	19
2.1.2 OBJETIVOS ESPECÍFICOS	19
2.2 JUSTIFICACIÓN	20
2.2.1 IMPACTO	21
2.2.2 VIABILIDAD	22
3. MARCO TEÓRICO	23
3.1 APLICACIONES WEB	23
3.1.1 APLICACIONES WEB ENRIQUECIDAS (RIA)	23
3.1.2 ARQUITECTURA DE LAS RIA	24
3.1.3 CUADRO COMPARATIVO ENTRE APLICACIÓN WEB Y RIA	25
3.2 FRAMEWORK	26
3.2.1 FRAMEWORK WEB	27
3.2.2 CARACTERÍSTICAS DE LOS FRAMEWORK WEB	27
3.3 PATRÓN DE DISEÑO MVC	30
3.3.1 ARQUITECTURA DEL MODELO VISTA CONTROLADOR	31
3.3.2 FLUJO DE CONTROL DEL MODELO MVC	32
3.3.3 VENTAJAS Y DESVENTAJAS DEL MODELO MVC	33
3.4 VAADIN	34
3.4.1 CARACTERÍSTICAS DE VAADIN	35
3.4.2 BIBLIOTECA DE VAADIN	36

3.4.3 ARQUITECTURA DE VAADIN	36
3.4.3.1 AJAX (ASYNCHRONOUS JAVA SCRIPT AND XML).....	39
3.4.3.2 GWT (GOOGLE WEB TOOLKIT)	40
3.4.3.3 JSON	40
3.4.3.4 APLICACIONES COMO JAVA SERVLET SESSIONS.....	41
3.4.3.5 MOTOR DEL CLIENT-SIDE	41
3.4.3.6 EVENTS Y LISTENERS	41
3.4.4 COMPONENTES DE LA INTERFAZ DE USUARIO.....	42
3.4.4.1 MAIN WINDOW	43
3.4.4.2 CARACTERÍSTICAS COMUNES DE LOS COMPONENTES.....	44
3.4.4.3 DESARROLLO DE COMPONENTES PROPIOS.....	45
4. METODOLOGÍA	46
5. DESCRIPCIÓN E IMPLEMENTACIÓN DE LA APLICACIÓN.....	49
5.1 ESPECIFICACIONES DEL SOFTWARE	49
5.2 SEGURIDAD DE LA APLICACIÓN	52
5.2.1 Disponibilidad	53
5.2.2 Integridad.....	53
5.2.3 Confidencialidad	53
5.3.1 GUI APLICACIÓN WEB GESTAR.....	54
5.3.1.1 PÁGINA DE INICIO DE SESIÓN.....	54
5.3.1.2 PÁGINA DE INICIO	56
5.3.1.3 PÁGINA DE ESTUDIANTE	57
5.3.1.3.1 Clase estudiante	57
5.3.1.3.2 Proyecto estudiante	61
5.3.1.4 PÁGINA DE PROFESOR	66
5.3.1.4.1 Clase profesor	66
5.3.1.4.2 Proyecto profesor	69
5.3.1.4.3 Editar notas.....	73
5.3.1.5 PÁGINA DE TUTORIAL	76
5.3.1.6 PÁGINA DE ACERCA DE	76

5.3.1.7 CERRAR SESIÓN	77
5.3.1.8 CAMBIAR CONTRASEÑA	78
5.4 PRUEBAS REALIZADAS A LA APLICACIÓN WEB GESTAR	80
5.4.1 PRUEBAS DURANTE EL DESARROLLO	80
5.4.2 PRUEBAS DE USABILIDAD	81
5.4.3 PRUEBAS FUNCIONALES	81
6. GUIA DIDACTICA.....	82
7. CONCLUSIONES	83
8. RECOMENDACIONES.....	85
BIBLIOGRAFIA.....	86

LISTA DE FIGURAS

Figura 1. Modelo Vista Controlador	31
Figura 2. Herencia jerárquica de las clases e interfaces de los componentes de interfaz de usuario.	43
Figura 3. Ciclo de vida del Proceso Ágil Unificado	48
Figura 4. Casos de uso aplicación web GESTAR	51
Figura 5. Página de Inicio de sesión.....	55
Figura 6. Página de inicio de sesión (error en los datos escritos)	55
Figura 7. Página de Inicio	56
Figura 8. Página de Inicio (ventana de programa).....	57
Figura 9. Página de Estudiante (sub-pestaña clase)	58
Figura 10. Página de Estudiante (sub-pestaña clase) descargar material de clase.....	59
Figura 11. Página de Estudiante (sub-pestaña clase) enviar comentarios.	60
Figura 12. Página de Estudiante (sub-pestaña clase) enviar comentarios (error).	60
Figura 13. Página de Estudiante (sub-pestaña proyecto) registro de equipo.	62
Figura 14. Página de Estudiante (sub-pestaña proyecto) registro de equipo (notificación exitosa).....	62
Figura 15. Página de Estudiante (sub-pestaña proyecto) enviar tareas (opción 1).....	63
Figura 16. Página de Estudiante (sub-pestaña proyecto) enviar tareas (ventana ayuda).	64
Figura 17. Página de Estudiante (sub-pestaña proyecto) enviar tareas (opción 2).....	64
Figura 18. Página de Estudiante (sub-pestaña proyecto) enviar tareas (opción 2 notificación).	65
Figura 19. Página de Estudiante (sub-pestaña proyecto) enviar tareas (opción 2 error).....	65
Figura 20. Página de Profesor (sub-pestaña clase)	66
Figura 21. Página de Profesor (sub-pestaña clase) registro de estudiantes.	67

Figura 22. Página de Profesor (sub-pestaña clase) registro de estudiantes (notificación).....	68
Figura 23. Página de Profesor (sub-pestaña clase) registro de estudiantes (error).	68
Figura 24. Página de Profesor (sub-pestaña proyecto)	70
Figura 25. Página de Profesor (sub-pestaña proyecto) descargar archivos. ...	71
Figura 26. Página de Profesor (sub-pestaña proyecto) eliminar archivos.	71
Figura 27. Página de Profesor (sub-pestaña proyecto) Administrar tareas. ...	72
Figura 28. Página de Profesor (sub-pestaña notas)	73
Figura 29. Página de Profesor (sub-pestaña notas) visualizar notas grupo. ..	74
Figura 30. Página de Profesor (sub-pestaña notas) editar notas grupo.	74
Figura 31. Página de Profesor (sub-pestaña notas) guardar notas grupo.	75
Figura 32. Página de Profesor (sub-pestaña notas) ver notas grupo cambiadas.....	75
Figura 33. Página de Tutorial	76
Figura 34. Página de Acerca de	77
Figura 35. Cerrar sesión	78
Figura 36. Cambiar contraseña	79
Figura 37. Cambiar contraseña (notificación)	79
Figura 38. Cambiar contraseña (error)	80
Figura 39. Casos de uso GESTAR.....	93

LISTA DE TABLAS

Tabla 1. Cuadro comparativo entre aplicación web y aplicación web enriquecida (RIA)	25
Tabla 2. Características de los Frameworks web	28
Tabla 3. Ventajas y desventajas del modelo MVC	33
Tabla 4. Principales partes de la arquitectura de Vaadin y sus funciones	37
Tabla 5. Descripción de actores involucrados en el sistema.....	49
Tabla 6. Caso de uso GESTAR: Publicar Noticias.....	94
Tabla 7. Caso de uso GESTAR: Registrar Estudiantes	94
Tabla 8. Caso de uso GESTAR: Editar Notas.....	95
Tabla 9. Caso de uso GESTAR: Subir Material de Clase	95
Tabla 10. Caso de uso GESTAR: Descargar Tareas.....	96
Tabla 11. Caso de uso GESTAR: Ver Comentarios	97
Tabla 12. Caso de uso GESTAR: Visualizar Tutorial	97
Tabla 13. Caso de uso GESTAR: Visualizar Noticias	98
Tabla 14. Caso de uso GESTAR: Iniciar Sesión.....	98
Tabla 15. Caso de uso GESTAR: Enviar Comentarios	99
Tabla 16. Caso de uso GESTAR: Registrar Equipo.....	99
Tabla 17. Caso de uso GESTAR: Subir Tareas.....	100

LISTA DE ANEXOS

ANEXO A.....	89
---------------------	-----------

Resumen

TÍTULO: EXPLORACIÓN DEL FRAMEWORK VAADIN PARA EL DESARROLLO DE UN GESTOR DE TAREAS PARA LA ASIGNATURA INGENIERÍA DEL SOFTWARE I*

AUTORES: LASPRILLA, María, NIÑO, Sebastian**

PALABRAS CLAVE: Framework, Java, Vaadin, aplicación web enriquecida, MVC, AUP.

DESCRIPCIÓN:

El contenido del presente documento es la investigación y puesta en práctica del Framework Vaadin para el desarrollo de una aplicación web, concretamente un gestor de tareas para la asignatura Ingeniería del Software I. El principal objetivo de esta investigación fue la exploración y extensión del estado del arte de este Framework con el fin de realizar una transferencia tecnológica a la Escuela de Ingeniería de Sistemas e Informática, apropiándola de una nueva tecnología y dejándola en un estado cómodo y accesible a toda la comunidad mediante la elaboración de una guía didáctica.

La aplicación web desarrollada, permite principalmente administrar y gestionar una asignatura específica durante un semestre y es de uso para el profesor y sus estudiantes. Ofrece servicios distinguidos para cada tipo de usuario, proporcionando al profesor una manera eficiente de llevar la asignatura y a los estudiantes, una excelente interacción con el profesor y el aula de clase. Fue desarrollada con base en la metodología de desarrollo de software AUP (The Agile Unified Process) y sobre la plataforma Java Enterprise Edition.

Durante el desarrollo del proyecto se realizaron las pruebas necesarias con el fin de comprobar el correcto funcionamiento del software. El resultado de dichas pruebas se encuentra al final del presente libro.

La elaboración de la guía didáctica como uno de los objetivos de este proyecto, es una compilación de las mejores características del Framework Vaadin, una completa explicación del funcionamiento del Framework y su marco teórico. Todo esto con el fin de dejar de la mejor manera el uso del Framework a la comunidad.

* Trabajo de Investigación

** Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática.
Director ROJAS MORALES, Fernando Antonio.

Abstract

TITLE: VAADIN FRAMEWORK EXPLORING TO DEVELOPMENT A TASK MANAGER TO SOFTWARE ENGINEERING COURSE I*

AUTHORS: LASPRILLA, María, NIÑO, Sebastian**

KEYWORDS: Framework, Java, Vaadin, rich web application, MVC, AUP.

DESCRIPTION:

The content of this document is the research and the application of the Framework Vaadin for the development of a web application, more exactly, a task manager for the Software Engineering I subject. The main goal of this research was the exploration and extension of the state of the art of this Framework with the purpose of making a technological transfer to the School of Engineering and Computer Systems, appropriating new technology and leaving it in a comfortable state and accessible to the community through the development of a tutorial.

The development web application, allows administer and manage a specific subject for a semester and is used for the teacher and students. It offers outstanding services for each user, giving the teacher an efficient way to take the course and to the students, an excellent interaction with the teacher and the classroom. The software was development based on the software development methodology AUP (The Agile Unified Process) and Java Platform Enterprise Edition.

It was done a series of test during the development of the project to verify the correct operation of the software. The result of test is at the end of this book.

The development of the tutorial as one of the goals of this project is a compilation of the best features of Vaadin framework, a complete explanation of its behavior and its theoretical framework. All this in order to make the best use of the Framework to the community.

* Research Work

** Faculty of Physics-Mechanics Engineering, Systems of Engineering and Computer Systems. Director ROJAS MORALES, Fernando Antonio.

INTRODUCCIÓN

En este proyecto se pretende abordar y explorar la tecnología Vaadin¹ (Marco de Trabajo), descubriendo su funcionalidad, potencial, servicios y sobre todo su facilidad a la hora de ponerlo en la práctica, todo esto con el fin de realizar una transferencia tecnológica a la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander, es decir, apropiarse de una nueva tecnología estudiándola a fondo, dejándola en un estado cómodo y accesible a toda la comunidad, mediante la elaboración de una guía didáctica, la cual es uno de los objetivos de este proyecto.

El Framework Vaadin 6.4.7 disponible desde el 25 de Octubre de 2010², es la versión de mantenimiento para la versión estándar Vaadin 6.4. El elemento clave de Vaadin es la biblioteca de Java diseñada para facilitar la creación y el mantenimiento para interfaces de usuario web.

El enfoque de Vaadin es claro, esta herramienta está diseñada para la creación de aplicaciones web, no para la creación de sitios web.

Mediante el desarrollo de una aplicación web dirigida a la asignatura Ingeniería del Software I, concretamente un gestor de tareas, permitirá poner en práctica los conocimientos adquiridos durante la exploración de dicha tecnología, y a su vez mostrando al estudiante interesado en Vaadin su facilidad para el desarrollo de aplicaciones web.

¹ Vaadin: Framework para el desarrollo de aplicaciones web.

² Release Notes for Vaadin 6.4.7

<http://vaadin.com/download/release/6.4/6.4.7/release-notes.html>

1. ESPECIFICACIONES DEL PROYECTO

1.1 TÍTULO

EXPLORACIÓN DEL FRAMEWORK VAADIN PARA EL DESARROLLO DE UN GESTOR DE TAREAS PARA LA ASIGNATURA INGENIERÍA DEL SOFTWARE I.

1.2 MODALIDAD

Proyecto de investigación.

1.3 DIRECTOR

Nombre: Fernando Antonio Rojas Morales
Profesión: Ingeniero de Sistemas
Institución: Universidad Industrial de Santander
Cargo: Profesor planta

1.4 AUTORES

Nombre: María Fernanda Lasprilla Suárez
Código: 2061461
Carrera: Ingeniería de Sistemas

Nombre: Sebastian Niño Peñaloza
Código: 2060578
Carrera: Ingeniería de Sistemas

2. PLANTEAMIENTO DEL PROYECTO

2.1 OBJETIVOS

2.1.1 OBJETIVO GENERAL

Extender el Framework Vaadin para realizar la implementación de una aplicación web – Un gestor de tareas para la asignatura de Ingeniería del Software I

2.1.2 OBJETIVOS ESPECÍFICOS

- Realizar el estado del arte de los Frameworks para el desarrollo de aplicaciones web enriquecidas, fundamentando la selección de Vaadin como la herramienta para el desarrollo de este proyecto.
- Realizar un artículo con la fundamentación obtenida.
- Establecer los requisitos del gestor de tareas por medio de la metodología de modelado UML.
- Seleccionar características del Framework Vaadin relevantes para el desarrollo de una aplicación web - un gestor de tareas para la asignatura de Ingeniería del Software I
- Elaborar una guía didáctica para el aprendizaje del Framework Vaadin, utilizando como IDE NetBeans 6.9

2.2 JUSTIFICACIÓN

El área de la ingeniería del software se ha visto envuelta en cambios que han transformado parcialmente la forma de pensar, analizar y resolver las dificultades de los sistemas tradicionales, adaptándolos a los nuevos estándares y empleando metodologías para agilizar tiempos y disminuir costos, con tecnologías que permitan la reutilización de código cumpliendo con los requisitos de calidad necesarios.

Sumado a esto, el alto nivel de rotación de la tecnología que se evidencia diariamente, pone a prueba la capacidad tanto del estudiante como del ingeniero de sistemas de explorar nuevas tecnologías que complementen su perfil profesional, permitiéndole ser más competitivo laboralmente y a su vez estar actualizado.

Actualmente existen diferentes marcos de trabajo (Frameworks) para el desarrollo de RIA¹ donde las exigencias demandan la portabilidad, escalabilidad y funcionalidad. Vaadin es un Framework que cumple con las características necesarias para el desarrollo de este proyecto entre las cuales se destacan:

- Su curva de aprendizaje es baja.
- Es un Framework open source.
- Utiliza Java como único lenguaje de programación.
- Es intuitivo y de agradable presentación al usuario.
- Es soportado por la mayoría de navegadores web conocidos sin necesidad de plugins.

¹ Rich Internet Applications

La exploración de la tecnología Vaadin permitirá el desarrollo de una aplicación web (gestor de tareas para la asignatura de Ingeniería del Software I), como pretexto para consolidar el aprendizaje adquirido durante el desarrollo del proyecto. No se pretende formar expertos en Vaadin, sino la capacidad de realizar aplicaciones web en con los elementos básicos que Vaadin ofrece y sobretodo, en muy poco tiempo.

2.2.1 IMPACTO

A nivel científico:

Motivar en futuros desarrolladores de proyectos a la investigación de nuevas herramientas de desarrollo software, permitiendo explorarlas a fondo para generar ideas innovadoras en la creación de aplicaciones de software, aportando este conocimiento a la Escuela de Ingeniería de Sistemas e Informática.

A nivel de la práctica:

Generar nuevos espacios de conocimiento en el cual el estudiante puede acceder a la información en cualquier momento y lugar.

Permitir a los estudiantes el uso de una tecnología nueva con la cual puede complementar su proceso de aprendizaje.

A nivel social:

Animar a los estudiantes a la investigación y al uso de nuevas tecnologías como apoyo al proceso de formación, permitiendo dar un paso más en las técnicas de aprendizaje.

2.2.2 VIABILIDAD

Viabilidad económica:

El proyecto será realizado con herramientas de software libres, para lo cual los costos se conforman de la conexión a internet del usuario para el uso de la aplicación.

Viabilidad técnica:

Los recursos necesarios para el desarrollo del proyecto consisten en computadores que se encuentran en la universidad, el computador de los autores y el acceso a internet.

Las pruebas se realizarán de manera local en los computadores de los autores.

Viabilidad social:

Actualmente la demanda del uso de internet es tal que la mayoría de estudiantes cuentan con este o la universidad provee de este servicio; así cada persona vinculada en el proceso de aprendizaje académico puede tener acceso a la aplicación.

Por esto se considera que este proyecto es viable.

3. MARCO TEÓRICO

3.1 APLICACIONES WEB

En el campo de la ingeniería del software se designa aplicación web a todas aquellas aplicaciones que los usuarios pueden manipular accediendo a un servidor web a través de internet mediante un navegador web.

Entre las ventajas de una aplicación web se tiene el ahorro de tiempo, permitiendo desarrollar tareas sencillas sin necesidad de instalar ningún programa. No ocupan espacio en el disco duro. No tienen problemas de compatibilidad solo se necesita de un navegador web. Son multiplataforma y portables. [1]

Los navegadores cada vez ofrecen más y mejores funcionalidades para crear aplicaciones web enriquecidas (RIA)¹.

Usualmente se necesitan Frameworks para ejecutar una aplicación RIA; esta debe ser instalada en el computador antes de ejecutar la aplicación. El Framework² se encarga de descargar, actualizar, verificar y ejecutar el RIA.

3.1.1 APLICACIONES WEB ENRIQUECIDAS (RIA)

Las aplicaciones web enriquecidas surgen de las aplicaciones web y de las aplicaciones de escritorio. Estas buscan mejorar la experiencia del usuario ofreciendo las siguientes características: [2]

¹ Rich Internet Applications

² Provee una estructura y una metodología de trabajo.

- **Accesibilidad:** Evitar la recarga completa de páginas web cada vez que el usuario da clic sobre un enlace. Evitando el tráfico entre el cliente y servidor. Con el uso de AJAX³ nativo en los navegadores web.
- **Instalación y mantenimiento:** Algunas requieren instalación de plugins. Las actualizaciones son automáticas, mientras se mantenga el navegador actualizado.
- **Multiplataforma:** Se pueden utilizar desde cualquier computador con conexión a internet sin depender del sistema operativo que este utilice.
- **Agilidad:** El usuario interactúa directamente con el servidor, sin necesidad de recargar la página, lo que genera mayor capacidad de respuesta.
- **Experiencia visual:** Hacen el uso de la aplicación sencillo, ofrecen mejoras en la conectividad y despliegue instantáneo de la aplicación, agilizando su acceso, garantizan la desvinculación de la capa de presentación, es decir, el acceso a la aplicación se puede realizar desde cualquier computador en cualquier lugar del mundo.

3.1.2 ARQUITECTURA DE LAS RIA

Se basa en la arquitectura Cliente-Servidor. El cliente sabe acerca de si mismo y el tipo de datos que está solicitando y únicamente solicita los datos que necesita sin ninguna otra información. [3]

³ Asynchronous JavaScript And XML (JavaScript asíncrono y XML)

Ciente: Interacción entre el usuario y la interfaz de usuario, el usuario invoca comandos, actualiza vistas y carga datos. El usuario mantiene el estado de la aplicación, maneja las peticiones de datos hacia el servidor y controla como se presentan los datos.

Servidor: El servidor maneja y procesa todas las peticiones del cliente y delega las acciones tales como: guardar datos en la base de datos, actualizar archivos del sistema, retornar datos al servidor, etc. El servidor determina y le da formato a los datos que son retornados al cliente.

3.1.3 CUADRO COMPARATIVO ENTRE APLICACIÓN WEB Y RIA

Tabla 1. Cuadro comparativo entre aplicación web y aplicación web enriquecida (RIA)

APLICACIÓN WEB	APLICACIÓN WEB ENRIQUECIDA (RIA)
Las aplicaciones web se ejecutan en su totalidad desde el navegador.	Aplicaciones que funcionan desde el navegador pero además necesitan la instalación de un software en el computador para poder usarse, por este motivo es que varias de las funcionalidades, los navegadores web no las ofrecen.

Actualizaciones automáticas, el software lo gestiona el propio desarrollador, al conectarse a internet siempre se está usando la última versión que se haya lanzado.	No necesitan actualización solo es necesario mantener actualizado el navegador web.
--	---

Experiencia del usuario simple y sencilla	Buscan mejorar la experiencia del usuario.
---	--

Ofrecen menos funcionalidades que una aplicación de escritorio.	Ofrecen características no nativas de los navegadores web.
---	--

Poca capacidad multimedia. Uso necesario de programas externos.	Capacidad multimedia total gracias a reproductores internos, sin necesidad de programas del sistema operativo del usuario.
---	--

3.2 FRAMEWORK

El concepto de Framework se utiliza en varios ámbitos del desarrollo de sistemas software, no solo en el desarrollo de aplicaciones web. Se pueden encontrar Frameworks para el desarrollo de aplicaciones médicas, de visión por computador, para el desarrollo de juegos, para aplicaciones aeronáuticas, entre otros.

Un Framework es una estructura de soporte definida y configurable, a la cual se le añaden piezas para organizar y desarrollar un proyecto software concreto. Los objetivos de un Framework son: Agilizar el proceso de desarrollo y reutilización de código. [4]

En otras palabras, un Framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.

3.2.1 FRAMEWORK WEB

Es una estructura definida y reusable en el que sus componentes facilitan la creación de aplicaciones Web. Este conjunto de componentes tales como clases en Java y archivos de configuración en XML⁴, forman un diseño para facilitar la reusabilidad, acelerando los tiempos de desarrollo y mantenimiento. La mayoría de Frameworks web se encargan de ofrecer una capa de controladores de acuerdo al modelo MVC⁵. [5]

3.2.2 CARACTERÍSTICAS DE LOS FRAMEWORK WEB

La siguiente tabla muestra las características relevantes de los Frameworks web. [6]

⁴ eXtensible Markup Language (lenguaje de marcas extensible)

⁵ Modelo Vista Controlador

Tabla 2. Características de los Frameworks web

CARACTERÍSTICA	DESCRIPCIÓN
Abstracción de URLs y sesiones	No es necesario manipular directamente las URLs ni las sesiones, el Framework ya se encarga de hacerlo.
Acceso a los datos	Incluyen las herramientas e interfaces necesarias para integrarse con herramientas de acceso a datos, en BBDD, XML, etc.
Controladores	La mayoría de Frameworks implementa una serie de controladores para gestionar eventos, como una introducción de datos mediante un formulario o el acceso a una página. Estos controladores suelen ser fácilmente adaptables a las necesidades de un proyecto concreto.
Autenticación y Control	Identificación de usuarios mediante un login y un password, restringen acceso a páginas a determinados usuarios.
Internacionalización	Maneja una librería de propiedades.
Reutilización de Código	Están orientados a permitir la reutilización

	de código o paquetes. Definen clases abstractas para que los desarrolladores hagan sus propios componentes.
Facilidad de uso y ahorro de tiempo	Una vez aprendido es fácil de usar lo cual ayuda a ahorrar tiempo en el desarrollo de una aplicación.
Validación de entradas	Provee métodos para la validación de las entradas en un formulario.
Código abierto	Licencias
Manejo de eventos	El manejo de eventos se realiza mediante listeners
Siguen una arquitectura, patrón o modelo	Modelo Vista Controlador (MVC), Cliente-Servidor, Modelo de tres capas, etc.

Fuente: Autores

3.3 PATRÓN DE DISEÑO MVC

Un patrón de diseño se refiere a un problema que ocurre habitualmente en el campo del desarrollo de software y su respectiva solución; puede ser empleado varias veces, en diferentes contextos, sin tener que duplicar el diseño. [7]

Un patrón de diseño se compone de cuatro elementos:

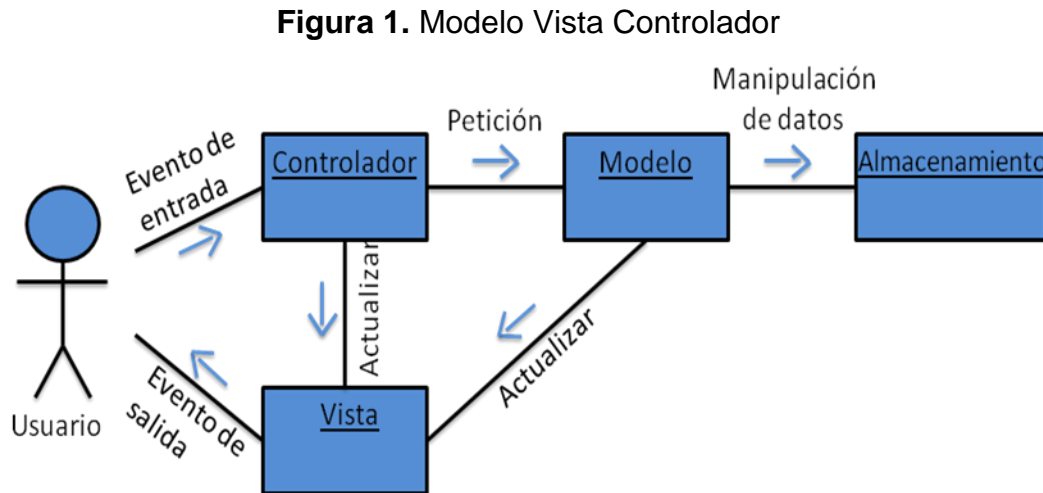
- El nombre del patrón: Es aquel que se utiliza para describir el problema de diseño, sus soluciones y consecuencias en una o dos palabras.
- El problema: Describe cuando aplicar el patrón. Especifica el problema y su contexto. Debe describir los problemas de diseño específico así como su representación conceptual como objetos.
- La solución: Describe los elementos que construyen el diseño, sus responsabilidades y colaboraciones. La solución no describe un problema de diseño en particular, porque un patrón es una especie de plantilla que puede ser aplicada en diferentes situaciones.
- Consecuencias: Son los resultados de aplicar el patrón.

El modelo vista controlador (MVC) es un patrón para el diseño de arquitecturas de aplicaciones que tienen gran interactividad con usuarios. Este resuelve el problema de mezclar el código de tres elementos: acceso a datos, lógica del negocio y la presentación. [8]

MVC consiste de tres tipos de objetos. La lógica de negocio que son los objetos de la aplicación, también conocida como lógica de la aplicación. La vista especifica la visualización de los datos, conocida como lógica de la presentación. Finalmente el

controlador es el coordinador entre estos dos anteriores, es decir, define la manera en como la interfaz de usuario reacciona ante la entrada de usuario.

3.3.1 ARQUITECTURA DEL MODELO MVC (MODELO VISTA CONTROLADOR)



Fuente: Autores

Los tres elementos que componen el Modelo Vista Controlador son: [9]

- **Modelo:** Es el objeto que representa los datos del programa. Maneja los datos y controla todas las transformaciones. El modelo no tiene conocimiento específico de los controladores o de las vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene la responsabilidad de mantener enlaces entre el Modelo y sus Vistas y notificar a las Vistas cuando cambia el modelo.
- **Vista:** Es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el modelo a través de una referencia al propio modelo.

- **Controlador:** Es el objeto que proporciona significado a las ordenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio modelo

3.3.2 FLUJO DE CONTROL DEL MODELO MVC

Se pueden encontrar diferentes implementaciones de MVC, pero usualmente tienen el siguiente flujo de control: [10]

1. El usuario interactúa con la interfaz sea mediante un click en un enlace, presionando un botón, etc.
2. El controlador recibe (por parte de los objetos la interfaz vista) la notificación de la acción solicitada por el usuario.
3. El controlador accede al modelo, posiblemente actualizando los datos enviados por el usuario.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario.
5. La vista usa el modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo.
6. En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.

7. La interfaz espera por nuevas interacciones de usuario para iniciar nuevamente el ciclo.

3.3.3 VENTAJAS Y DESVENTAJAS DEL MODELO MVC

La siguiente tabla muestra las ventajas y desventajas del modelo – vista - controlador [11]

Tabla 3. Ventajas y desventajas del modelo MVC

VENTAJAS	DESVENTAJAS
Existe una clara separación entre los componentes de una aplicación; lo cual permite implementarlos por separado.	La distribución de componentes obliga a crear y mantener un mayor número de archivos.
La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.	La curva de aprendizaje para nuevos desarrolladores se estima mayor.
Sencillez para crear distintas representaciones de los mismos datos.	Existen problemas difíciles de resolver utilizando el patrón MVC.
Facilidad para la realización de pruebas unitarias de los componentes, así como de desarrollo guiado por pruebas.	Dependencia del código fuente de una aplicación de un Framework. Si se quiere cambiar a otro Framework, debe reescribirse la mayoría del

código.

Reutilización de los componentes.

Simplicidad en el mantenimiento de los sistemas.

Facilidad para desarrollar prototipos de manera rápida y escalable.

Fuente: Autores

3.4 VAADIN

Vaadin es un Framework para el desarrollo de aplicaciones web, diseñado para la creación de aplicaciones interactivas en un navegador web sin necesidad de plugins. Vaadin se basa en componentes reutilizables para hacer más simple el desarrollo de aplicaciones web y mejorar la seguridad de las mismas. No necesita de programación en HTML⁶, XML⁷ o JavaScript⁸.

Vaadin está diseñado para la construcción de aplicaciones web, no de sitios web. El modelo de programación se ajusta al tradicional con eventos y listeners en lugar de solicitudes y respuestas. [12]

⁶ HyperText Markup Language (Lenguaje de Marcado de Hipertexto)

⁷ eXtensible Markup Language (Lenguaje de marcas extensible)

⁸ Lenguaje de programación interpretado

3.4.1 CARACTERÍSTICAS DE VAADIN

Vaadin tiene varias características que lo diferencian de los demás Frameworks web, haciendo de este uno muy tentativo. Las siguientes son las características más importantes de Vaadin: [13]

- Framework con componentes sofisticados: Decenas de componentes integrados a la interfaz de usuario, diseños dinámicos en Java, enlace de datos con el modelo MVC, validación de entradas, planillas basadas en páginas HTML.
- Comunicación con el servidor: aplicaciones web de una sola página, actualización automática de componentes de seguimiento.
- Desarrollo de aplicaciones web optimizadas: Java como único lenguaje de programación, no necesita JavaScript, no necesita configuraciones XML.
- Diferentes navegadores web: Internet Explorer, Mozilla Firefox, Safari, Opera, Google Chrome.
- Aplicaciones seguras: Lógica de la aplicación ejecutada en el servidor.
- Aspecto personalizable: Componentes basados en estilos de CSS⁹ (Cascading Style Sheets), personalización de temas.

⁹ Cascading Style Sheets (hojas de estilo en cascada)

3.4.2 BIBLIOTECA DE VAADIN

La biblioteca de Vaadin está diseñada para facilitar la creación y mantenimiento de interfaces de usuario de aplicaciones web. La biblioteca de Vaadin define una clara separación entre la interfaz de usuario y la parte lógica, permitiendo desarrollarlas por separado. [14]

La biblioteca de Vaadin debe actualizarse en cada proyecto web que se desarrolle, algunos de los componentes que se incluyen son:

- Paquetes para importar en el desarrollo del código con los elementos usados en la interfaz de usuario como: Button, TextField, Form, Panel, Link, etc.¹⁰
- Themes (temas) que controlan el aspecto visual de la aplicación web. El paquete viene con 2 temas Runo y Reindeer, pero igualmente se pueden añadir otros ya sea descargándolos y creándolos a través de CSS.

3.4.3 ARQUITECTURA DE VAADIN

Vaadin consiste en una Interfaz de Programación de Aplicaciones Web (API)¹¹, tiene a su disposición muchos componentes para realizar interfaces de usuario, temas por defecto y creación de estos mismos para mejorar notablemente la apariencia de la aplicación y un modelo de datos que permite la integración de los componentes de la interfaz de usuario con los datos directamente. Vaadin también usa un “adaptador terminal” el cual recibe las peticiones del explorador web y da como respuesta una renderización a la página en cuestión. [15]

¹⁰ Componentes de la interfaz de usuario de Vaadin

¹¹ Application Programming Interface (Programación de Interfaz de Aplicaciones)

Uno de los fuertes más atractivos de Vaadin es que hace invisible al desarrollador tecnologías como AJAX, JSON y JavaScript que forman parte de su arquitectura. Las aplicaciones hechas en Vaadin se ejecutan como servlets. [13]

Principales partes de la arquitectura y sus funciones:

Tabla 4. Principales partes de la arquitectura de Vaadin y sus funciones

PARTE	DESCRIPCIÓN
Componentes de la Interfaz de usuario	La interfaz de usuario consiste de componentes que se crean y se muestran en la aplicación. Cada componente del lado del servidor tiene una contra parte por el lado del cliente, con la que el usuario interactúa.
Client-Side Engine	El motor del lado del cliente de Vaadin gestiona la representación en el navegador web a través de Google Web Toolkit (GWT). Este comunica la interacción del usuario y los cambios en la interfaz de usuario con el adaptador terminal del lado del servidor utilizando la User Interface Definition Language (UIDL), un lenguaje basado en JSON.
Adaptador de Terminal	Los componentes de la interfaz de usuario no se representan directamente en una página web, sino con el uso de un adaptador terminal. La capa de abstracción permite a los usuarios utilizar las aplicaciones realizadas en Vaadin con cualquier navegador web.

Cuando el usuario realiza una acción en la aplicación web, estos eventos son comunicados al adaptador de terminal a través del servidor web con peticiones asíncronas de AJAX.

El adaptador terminal entrega los eventos del usuario a los componentes de la interfaz de usuario, quienes se lo entregan a la lógica de estos componentes.

Themes La interfaz de usuario separa entre la presentación y la lógica. Mientras que la interfaz de usuario es manejada en código java, la presentación es definida en temas en CSS. Vaadin provee temas por defecto. Estos temas además de ser hojas de estilo en CSS, incluir plantillas en HTML que definen diseños personalizados y otros recursos.

UIDL El adaptador terminal dibuja la interfaz de usuario en la aplicación web y cualquier cambio usando User Interface Definition Language (UIDL).

Las comunicaciones UIDL se realizan usando JSON (JavaScript Object Notation) que es un formato de intercambio de datos ligero, especialmente eficaz para las interfaces con JavaScript basadas en AJAX.

Events Los eventos son creados de la interacción del

usuario con los componentes de la interfaz de usuario, estos son primero procesados en el lado del cliente con JavaScript y luego pasan por el servidor HTTP, el adaptador terminal y la capa de componentes de usuario de la aplicación.

Modelo de Datos Vaadin proporciona un modelo de datos para interconectar los datos presentados en los componentes de la interfaz de usuario. Usando el modelo de datos, los componentes de la interfaz de usuario pueden actualizar los datos de la aplicación directamente sin necesidad de ningún código de control.

Todos los componentes de la interfaz de usuario utilizan este modelo de datos internamente, pero se pueden separar las fuentes de los datos de igual manera.

Fuente: Autores

3.4.3.1 AJAX (ASYNCHRONOUS JAVA SCRIPT AND XML)

Es una tecnología para el desarrollo de aplicaciones Web enriquecidas usando como respuesta la interacción del usuario, similar a las aplicaciones de escritorio tradicionales. Dichas aplicaciones se ejecutan en el browser (cliente) haciendo simultáneamente comunicación asíncrona con el servidor, permitiendo así que el usuario realice cambios en la página sin necesidad de recargarla. Esto aumenta considerablemente la interactividad, velocidad y usabilidad en las aplicaciones. Todo esto se logra mediante el uso de tecnologías como: XHTML¹², CSS, DOM¹³,

JavaScript, XMLHttpRequest y XML. Mientras que Vaadin 4 usaba XML para intercambio de datos, la versión 5 utiliza el eficiente JSON¹⁴. [13]

3.4.3.2 GWT (GOOGLE WEB TOOLKIT)

Es un Framework creado por Google para el fácil desarrollo de aplicaciones web del lado del cliente, sin tener que usar JavaScript u otra tecnología directamente. Una de sus desventajas radica en la comunicación con el servidor mediante llamadas a procedimiento remoto (RPC) ¹⁵ y la serialización que hace con cualquier dato. Vaadin evidentemente “esconde” todas las comunicaciones cliente-servidor. Google Web Toolkit es utilizado por Vaadin para renderizar las interfaces de usuario en el browser, sin embargo, el GWT¹⁶ es también totalmente invisible en Vaadin. Las aplicaciones hechas con este Framework son desarrolladas con Java y compiladas con JavaScript. [13]

3.4.3.3 JSON

Es un formato ligero para el intercambio de datos el cual puede llegar a ser cien veces más rápido que XML. El Client-Side Engine de Vaadin usa JSON a través de Google Web Toolkit. El uso de JSON, igual que otras tecnologías mencionadas, es completamente invisible para el desarrollador de Vaadin. [13]

¹² eXtensible Hypertext Markup Language (lenguaje extensible de marcado de hipertexto)

¹³ Document Object Model (Modelo de Objetos del Documento)

¹⁴ JavaScript Object Notation (formato ligero para el intercambio de datos)

¹⁵ Remote Procedure Call (Llamada a Procedimiento Remoto)

¹⁶ Google Web Toolkit

3.4.3.4 APLICACIONES COMO JAVA SERVLET SESSIONS

Vaadin hace todo su desarrollo en la parte superior de Java Servlet API, con el adaptador terminal que es la capa de nivel más bajo para el manejo de las peticiones del contenedor web.

Cuando el contenedor web recibe la primera solicitud de una dirección URL registrada por una aplicación, se crea una instancia de la clase `ApplicationServlet` en el Framework Vaadin, que hereda la clase `HttpServlet` definida en Java Servlet API. De esto se deduce sesiones utilizando la interfaz `HttpSession`, asociando la aplicación con cada sesión. Durante el tiempo de la sesión el Framework transmite las acciones del usuario a la aplicación indicada y además a un componente de la interfaz de usuario. [13]

3.4.3.5 MOTOR DEL CLIENT-SIDE

En el Framework Vaadin la tecnología Client-Side, generalmente no es utilizada a menos que se desarrolle con componentes personalizados de Google Web Toolkit (GWT). El motor del lado del cliente se basa en Google Web Toolkit (GWT), que permite el desarrollo de los componentes del motor y del Client-Side únicamente con Java.

La comunicación con el servidor se hace mediante el UIDL, usando los JSON mensajes, en un formato de intercambio de conexión HTTP. [13]

3.4.3.6 EVENTS Y LISTENERS

La aplicación necesita saber las acciones que el usuario realiza. La mayoría de Frameworks basados en Java, trabajan con un observador diseñado para comunicar la entrada del usuario con la lógica de la aplicación. Este diseño consiste en dos tipos de elementos un objeto y el número de observadores que detectan eventos sobre el objeto. Cuando se produce un evento relacionado con el

objeto, el observador recibirá una notificación sobre este. En la mayoría de los casos solo hay un observador definido en la lógica de la aplicación.

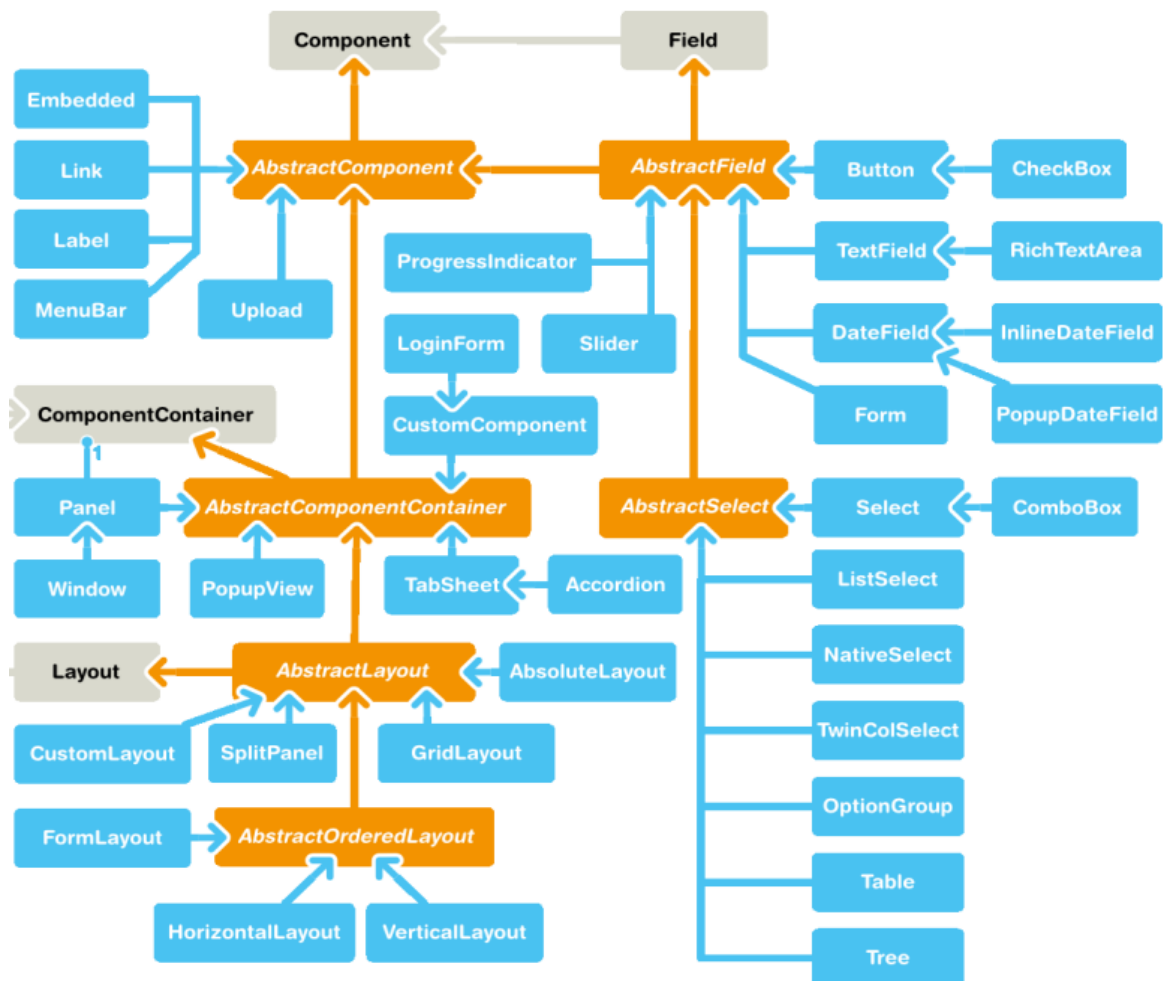
En Vaadin, el objetivo de los eventos es mantener la interacción del usuario con la interfaz de usuario. La administración de sesiones, el tiempo de espera, las bases de datos y otros tipos de comunicación asíncrona, requieren de eventos especiales. Para recibir los eventos la aplicación debe incluir la clase que implementa la interfaz del correspondiente listener. Los listeners son administrados por la clase `AbstractComponent`. [13]

3.4.4 COMPONENTES DE LA INTERFAZ DE USUARIO

Vaadin proporciona un conjunto completo de componentes de interfaz de usuario y también permite personalizar componentes propios. A continuación se nombran los principales componentes de la interfaz de usuario con sus características:

La siguiente imagen ilustra la herencia jerárquica de las clases e interfaces de los componentes de interfaz de usuario:

Figura 2. Herencia jerárquica de las clases e interfaces de los componentes de interfaz de usuario.



Fuente: Grönroos, Marko. "Book of Vaadin". Vaadin Ltd. 2010

3.4.4.1 MAIN WINDOW

- **Una por aplicación:** Cada instancia `com.vaadin.Application` tiene una `mainWindow` (ventana principal). Para asignar esta ventana como ventana principal de la aplicación se utiliza el método `setMainWindow ()`

- **Una por usuario:** Los usuarios pueden acceder a la aplicación web a través de múltiples pestañas. De forma predeterminada, desde la versión 6 de Vaadin, la ventana principal es compartida por todas las fichas de los usuarios. Si el usuario muestra algo en una pestaña y luego se mueve a otra pestaña y da algún click, probablemente recibirá una notificación roja con el mensaje de error "Out of Sync", necesitará actualizar la página y probablemente logearse de nuevo.

3.4.4.2 CARACTERÍSTICAS COMUNES DE LOS COMPONENTES

Los componentes de la interfaz de usuario proporcionan un gran número de características. Las más importantes se nombran a continuación: [13]

- **Caption:** Un Caption (leyenda) es una etiqueta explicativa de texto que acompaña a un componente de la interfaz de usuario, por lo general se muestra arriba, a la izquierda o dentro de los contenidos de los componentes.
- **Description and Tooltips:** Todos los componentes tienen una descripción aparte del Caption. Esta descripción usualmente se muestra una información cuando el puntero del mouse se posición sobre el componente por un corto tiempo.
- **Enabled:** Esta propiedad controla si el usuario puede o no usar un componente. Un componente Enabled es visible, pero de otro color lo cual indica que esta desactivado. Los componentes están activados por defecto.
- **Icon:** Un icono es una etiqueta gráfica explicativa que acompaña a un componente de la interfaz de usuario.

- **Read-Only:** Esta propiedad define si el usuario puede cambiar el valor de un componente. Sólo los componentes con campos de escritura pueden tener un valor que es editado por los usuarios.
- **Style Name:** Esta propiedad define el nombre de estilo de la clase CSS para la personalizar cada componente. Para agregar y remover nombres de estilo a componentes individuales se usa los métodos `addStyleName ()` y `removeStyleName ()`.
- **Visible:** Los componentes pueden ocultarse utilizando la propiedad visible como falsa, esto incluye que el caption, icon y las demás características de los componentes quedan ocultas. El contenido de los componentes que se establecen como no visibles, no se comunica con el navegador en lo absoluto. Esta es una característica importante en la seguridad, si se tienen componentes que contienen información que no debe ser visible por todos en la aplicación. El método usado es `setVisible ()` para cada componente.
- **Sizing Components:** Los componentes de Vaadin se pueden dimensionar. Permitiendo manipular su altura y anchura en unidades específicas, o simplemente dejándoles un tamaño indefinido. Los métodos para dimensionar un componente son `setWidth ()` y `setHeight ()` ancho y alto respectivamente. Para especificar las unidades, se utiliza un segundo parámetro en este método (Size Units).

3.4.4.3 DESARROLLO DE COMPONENTES PROPIOS

La facilidad de crear nuevos componentes de interfaz de usuario es una de las ventajas de Vaadin. Consiste en combinar componentes ya creados para producir nuevos componentes. Para crear un componente propio se necesita heredar de la clase `CustomComponent`.

4. METODOLOGÍA

En el presente capítulo se describe con detalle la metodología empleada en el desarrollo de la aplicación web GESTAR. A partir de su seguimiento dependió del éxito del mismo.

La metodología de desarrollo, marca las diferentes maneras de realizar las fases de creación de un proyecto informático, particularmente las relacionadas con el análisis y diseño.

La metodología utilizada en el desarrollo de la aplicación web GESTAR, es el proceso unificado ágil o AUP, su selección se debió a que esta describe una manera fácil y simple de desarrollar aplicaciones software por medio del uso de técnicas ágiles tales como el Desarrollo Dirigido por Pruebas (test driven development – TDD), el Modelado Ágil, la Gestión de Cambios Ágil, y la Refactorización de Bases de Datos para mejorar la productividad.

Las disciplinas de AUP son ejecutadas en una forma iterativa, definiendo las actividades que el equipo de desarrollo ejecuta para construir, validar y liberar software funcional, el cual cumple con las necesidades del usuario. [16]

La metodología de desarrollo AUP cuenta con cuatro fases las cuales lo describen:

- **Creación:** El objetivo es identificar el alcance inicial del proyecto, elaborar la arquitectura de la aplicación y la aceptación de las partes interesadas.
- **Elaboración:** El objetivo de esta fase es probar la arquitectura de la aplicación.

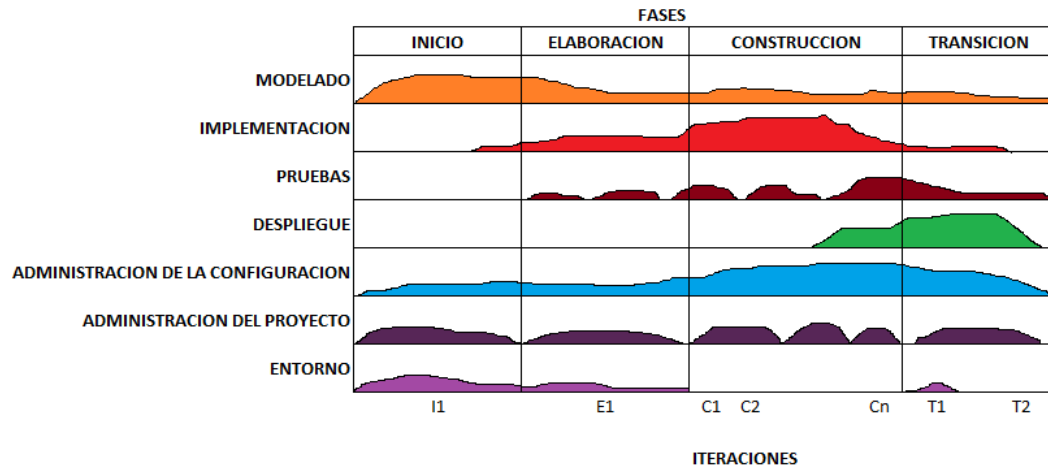
- **Construcción:** En esta fase se realiza la construcción del software, trabajando en forma regular y gradual, respondiendo a las necesidades de mayor prioridad de las partes interesadas en el proyecto.
- **Transición:** El objetivo es validar y desplegar el sistema en su entorno de producción.

Disciplinas:

Las disciplinas de AUP se realizan de manera iterativa, los miembros del equipo de desarrollo definen las actividades a realizar, se realiza la validación y despliegue del software para comprobar que se satisfagan las necesidades de las partes interesadas. Las disciplinas son:

- **Modelo:** El objetivo de esta disciplina es entender el negocio de la organización, el dominio del problema que el proyecto ocupa y determinar una solución viable.
- **Implementación:** El objetivo es transformar el modelo en código ejecutable y se realizan las pruebas de unidad.
- **Pruebas:** Esta disciplina consta de la realización de una evaluación objetiva para asegurar la calidad. Esto incluye encontrar defectos, validar el sistema y verificar que se cumplan los requisitos.
- **Despliegue:** Esta disciplina elabora el plan para la entrega del sistema y el plan para que el sistema esté a disposición de los usuarios finales.
- **Entorno:** La disciplina garantiza la orientación adecuada del proceso (normas) y que las herramientas (hardware, software, etc.) estén disponibles para el equipo según sea necesario.

Figura 3. Ciclo de vida del Proceso Ágil Unificado



Fuente: Autores

Filosofía de AUP:

El proceso ágil unificado se basa en los siguientes principios: [17]

- **El equipo sabe que lo hace:** El equipo no requiere de orientación y entrenamiento.
- **Simplicidad:** Todo está descrito de manera concisa.
- **Agilidad:** El proceso ágil UP se ajusta a los valores y principios de la alianza ágil.
- **Centrada en actividades de alto valor:** Se centra en actividades que realmente cuentan.
- **Independencia de herramientas:** Se puede utilizar cualquier conjunto de herramientas que se desee con AUP.

5. DESCRIPCIÓN E IMPLEMENTACIÓN DE LA APLICACIÓN

En esta sección se describe el desarrollo de la aplicación web GESTAR teniendo en cuenta los objetivos planteados.

Primero se describen los requerimientos de la aplicación obtenidos a partir de casos de uso, luego se mostrará los componentes de la interfaz de usuario.

5.1 ESPECIFICACIONES DEL SOFTWARE

En esta sección se describe la especificación de requisitos de la aplicación, necesaria para el desarrollo del software, aquí se presentan las funcionalidades de la aplicación representadas por medio de casos de uso con su respectiva descripción, se identifican a los usuarios finales de la aplicación y se realiza un análisis del hardware necesario para el funcionamiento de la aplicación.

Listado de actores:

Basado en las características que presenta la aplicación web GESTAR y los servicios que se le implementaron, se clasificaron los actores que se describen en la siguiente tabla:

Tabla 5. Descripción de actores involucrados en el sistema

ACTOR	DESCRIPCIÓN	RESPONSABILIDADES (PAPEL QUE JUEGA)
Estudiante	Representa a las personas inscritas en el curso de Ingeniería del Software I.	Suministra información personal. Crea perfil de Estudiante. Visualiza noticias y eventos en la cartelera. Visualiza notas

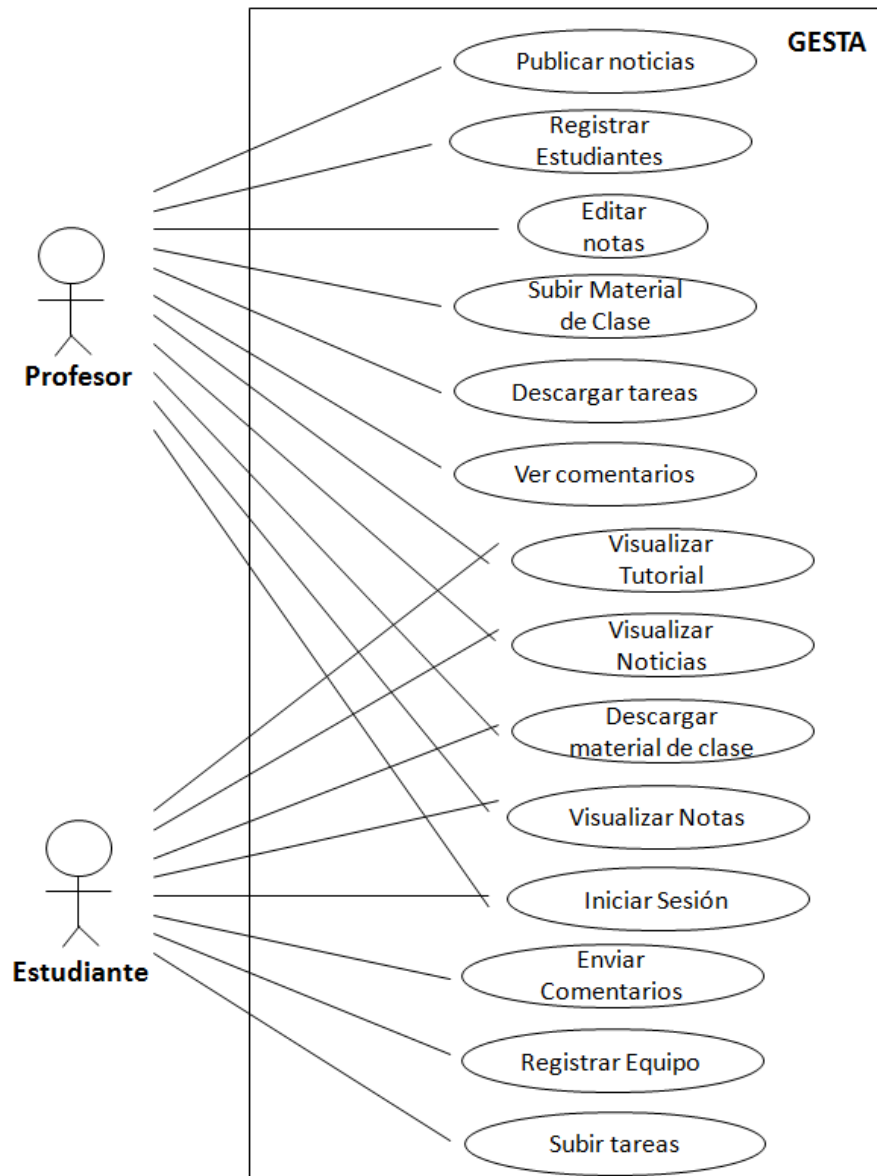
		<p>Envía comentarios respecto a las notas publicadas al profesor.</p> <p>Descarga material de clase.</p> <p>Registra equipo de estudiantes para el trabajo durante la asignatura.</p> <p>Sube tareas.</p> <p>Visualiza el tutorial de Vaadin.</p>
Profesor	Representa a la persona encargada de dictar las clases y administrar la aplicación.	<p>Publica noticias o eventos en la cartelera.</p> <p>Registra grupos de la asignatura con sus respectivos estudiantes.</p> <p>Sube material de clase.</p> <p>Administra los archivos subidos.</p> <p>Descarga tareas.</p> <p>Edita las notas de los estudiantes.</p> <p>Visualiza el tutorial.</p>

Fuente: Autores

Diagrama de casos de uso:

El diagrama que se representa en la siguiente figura corresponde a las funcionalidades de la aplicación web GESTAR y tiene como actores al Estudiante y al Profesor.

Figura 4. Casos de uso aplicación web GESTAR



Fuente: Autores

A continuación se muestra una tabla de un caso de uso en particular que indica los pasos de cómo llevarse a cabo una funcionalidad. La siguiente es la tabla para el caso de uso de publicar noticias en la cartelera.

Tabla 6. Caso de uso GESTAR: Publicar Noticias

Nombre:	Publicar Noticias
Actor(es):	Profesor
Descripción:	El profesor edita noticias relacionadas con la materia publicándolas en la cartelera, permitiendo que los estudiantes las puedan visualizar.
Precondición(es):	Se necesita validación correcta del usuario (profesor).
Flujo Principal:	Se presenta al usuario una pantalla donde inicia sesión con un usuario y contraseña. Una vez validado el usuario, este accede a la sección de cartelera, donde publica la noticia o aviso respectivo.
Pos-condición(es):	Las noticias quedan disponibles en el sistema
Sub-flujos:	Ninguno
Excepciones:	Ninguno

Fuente: Autores

En el anexo A se detalla el documento de especificación de requisitos de software.

5.2 SEGURIDAD DE LA APLICACIÓN

Durante el uso de la aplicación web GESTAR es posible que se presenten problemas respecto a la seguridad de la información que se maneja.

La aplicación cuenta con los principios básicos de seguridad de la información: disponibilidad, integridad y confiabilidad.

5.2.1 Disponibilidad

La disponibilidad hace referencia a que los recursos y la información siempre estén al alcance del usuario en cualquier momento que él lo necesite.

En la aplicación web GESTAR un usuario puede acceder a la información a través de internet. GESTAR no está disponible para dispositivos móviles.

5.2.2 Integridad

La integridad se refiere a que la información sólo sea modificada por el personal autorizado para tal fin.

En el caso de la aplicación descrita en este libro la integridad se maneja con diferentes perfiles de usuario que acceden a la aplicación, cada uno con privilegios diferenciados de acuerdo al perfil que poseen. De la misma forma cada usuario debe ser previamente autenticado para poder acceder a la información y realizar cambios sobre ella.

5.2.3 Confidencialidad

La confidencialidad se refiere a que la información sólo pueda ser leída por los usuarios autorizados para esto.

En la aplicación web GESTAR el usuario debe iniciar sesión con su nombre de usuario y contraseña para poder visualizar la información. Cada usuario sólo puede interactuar con su información y no con la de otro usuario. El administrador de la aplicación puede modificar la información de todos los usuarios.

5.3 INTERFAZ GRÁFICA DE LA APLICACIÓN WEB GESTAR

La interfaz gráfica de usuario (GUI) consta de una aplicación web desarrollada en Java en donde el estudiante y el profesor pueden interactuar con ella para llevar a cabo la asignatura.

En esta sección sólo se mostrarán los componentes principales de la interfaz gráfica más no los algoritmos y demás implementaciones que se desarrollaron para el correcto funcionamiento de la aplicación.

5.3.1 GUI APLICACIÓN WEB GESTAR

A continuación se describe cada una de las secciones que contiene la aplicación web GESTAR.

5.3.1.1 PÁGINA DE INICIO DE SESIÓN

En esta página se le presenta al usuario una pantalla para iniciar sesión por medio de un código y una contraseña tanto para el estudiante como para el profesor, posteriormente se realiza una autenticación y en caso de ser exitosa se le permite al usuario acceder a los servicios que ofrece la aplicación, los cuales se explicarán más adelante.

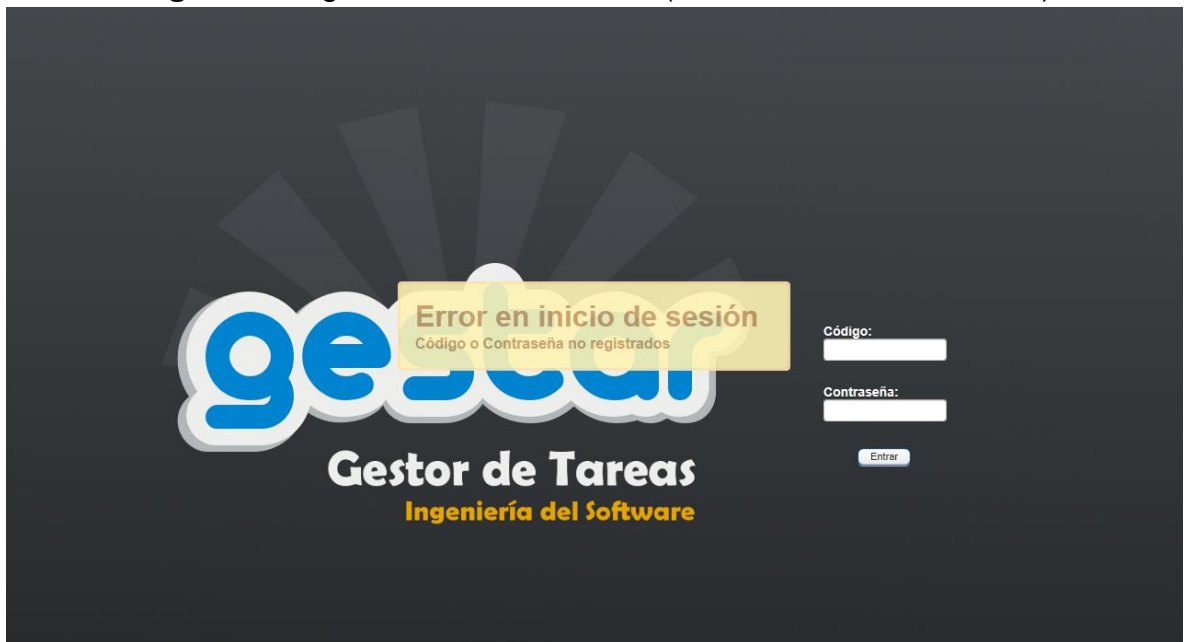
Si el inicio de sesión no es exitoso, se le muestra al usuario un mensaje de error, indicándole que introduzca de nuevo los datos.

Figura 5. Página de Inicio de sesión



Fuente: Autores

Figura 6. Página de inicio de sesión (error en los datos escritos)



Fuente: Autores

5.3.1.2 PÁGINA DE INICIO

Al iniciar sesión se muestra a la izquierda un panel el cual contiene un mensaje de bienvenida con el nombre del usuario. Se muestra también un calendario con el mes actual. A la derecha se encuentra un panel llamado cartelera el cual contiene una tabla con las noticias o eventos que el profesor publica y elimina, el estudiante sólo puede visualizarlas.

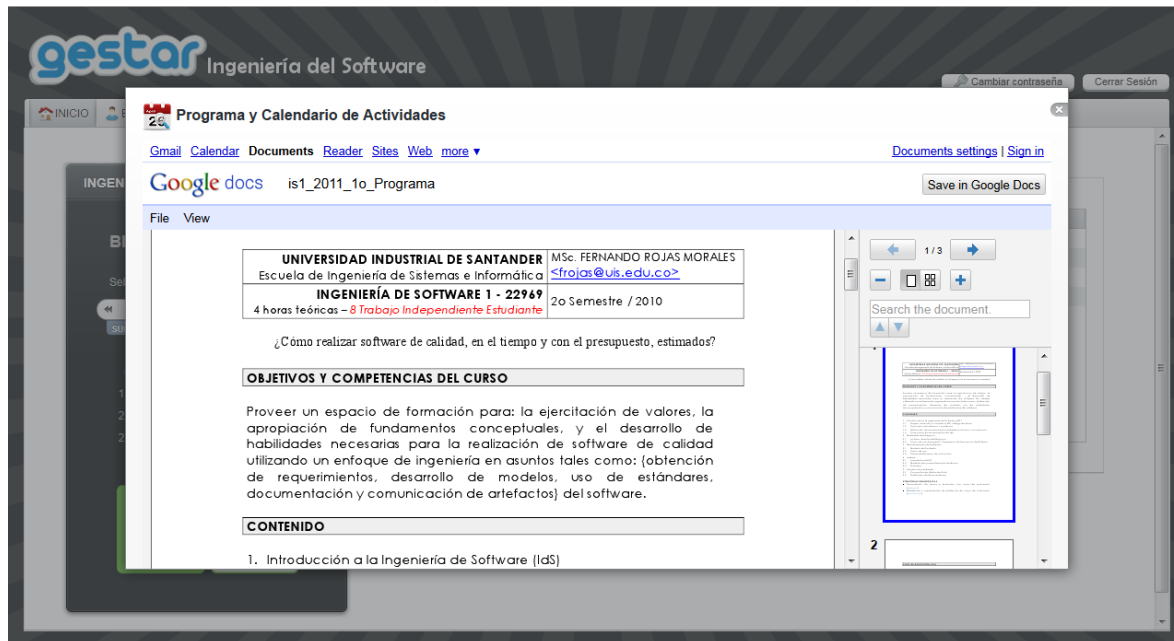
En la parte inferior se halla un botón que al presionarlo abre una ventana con el programa y contenido de actividades, todo el contenido de la aplicación se bloquea hasta que se cierre la ventana recién abierta.

Figura 7. Página de Inicio



Fuente: Autores

Figura 8. Página de Inicio (ventana de programa)



Fuente: Autores

5.3.1.3 PÁGINA DE ESTUDIANTE

En la pestaña estudiante se encuentran tres sub-pestañas las cuales se describen a continuación.

5.3.1.3.1 Clase estudiante

Esta sub-pestaña es la que se muestra siempre por defecto cuando el usuario hace click en la pestaña estudiante. En esta sub-pestaña, el estudiante interactúa con todo lo relacionado al aula de clase como lo son las descargas de material didáctico, sus respectivas notas y comentarios o sugerencias hacia el profesor. La organización de esta sub-pestaña cuenta con tres paneles.

Figura 9. Página de Estudiante (sub-pestaña clase)

The screenshot shows the 'gestar' student interface. At the top, there is a navigation bar with 'INICIO', 'ESTUDIANTE', 'TUTORIAL', and 'ACERCA DE'. Below this, there are tabs for 'Clase' and 'Proyecto'. The main content area is divided into three sections:

- DESCARGAR MATERIAL DE CLASE:** A dark sidebar containing a list of downloadable files with their extensions: 'Ciclos de Vida del Software.pdf', 'Code And Fix.pdf', 'Fundamentos Conceptual.pdf', 'Que es la Ing-Sw.pdf', 'SE came and gone.pdf', 'Tarea 0.docx', 'Tarea 1.docx', 'Tarea 2.docx', and 'Tarea 3.docx'.
- NOTAS ESTUDIANTE:** A table showing student grades for a specific student.
- COMENTARIOS:** A text input area for students to provide feedback or comments on their grades.

CODIGO	NOMBRE	APELLIDO	T0	T1	T2	T3	EXP	ASIS	P1	P2	DEF
2060578	Sebastian	Niño Peñalosa	4.0	4.3	3.6	3.8	4.0	4.0	4.5	3.8	3.9

COMENTARIOS

En el siguiente espacio escriba el comentario o solicitud respecto a las notas publicadas. Este mensaje será enviado al profesor y unicamente será visto por él.

Buenas tardes profesor. Me gustaría saber si es posible revisar la nota de la "T2". Gracias.

Enviar

Fuente: Autores

Descargar material de clase

Este panel contiene un listado de los archivos (material de clase) previamente subidos al servidor por el profesor. Estos archivos están disponibles para ser descargados en cualquier momento por el estudiante haciendo click en el botón que los representa. Los archivos están nombrados con su extensión.

Figura 10. Página de Estudiante (sub-pestaña clase) descargar material de clase.



Fuente: Autores

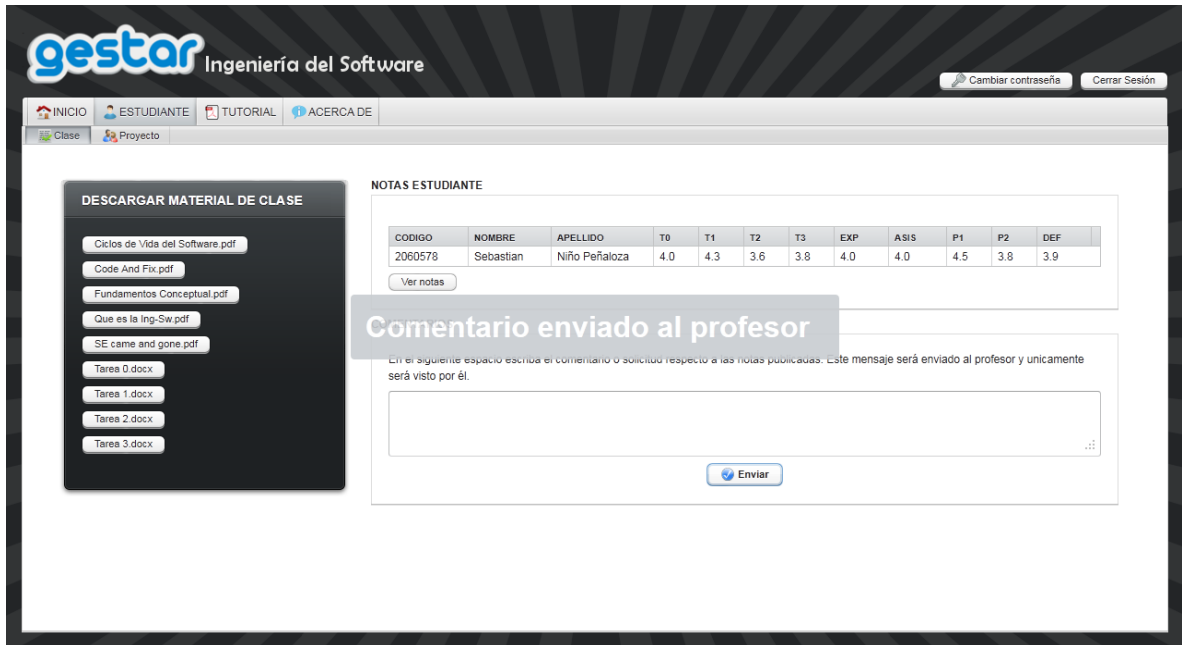
Notas estudiante

Este panel tiene almacenado en una tabla las respectivas notas del estudiante previamente actualizadas por el profesor. Las notas solo se hacen visibles haciendo click en el botón “Ver notas”.

Comentarios

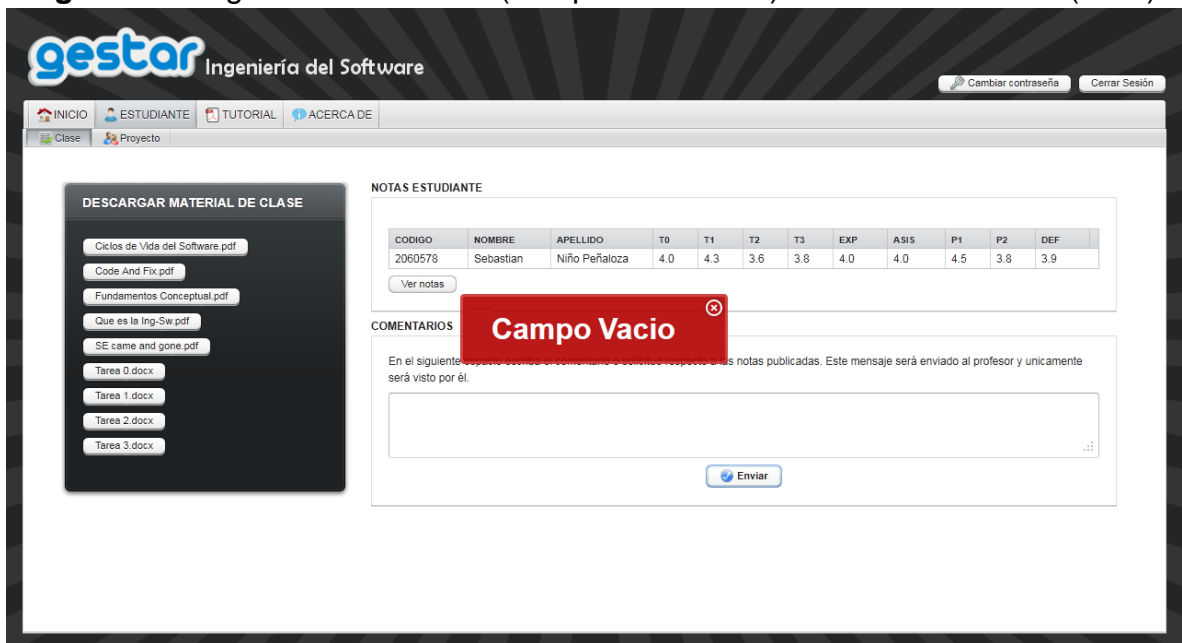
En este panel hay un área de texto en la cual el estudiante tiene la posibilidad de hacer algún comentario o sugerencia respecto a sus notas. Los comentarios se mandan al profesor haciendo click en el botón “Enviar”. Si el área de texto se encuentra vacía y el estudiante hace click en el botón “Enviar”, se notifica con mensaje tipo error.

Figura 11. Página de Estudiante (sub-pestaña clase) enviar comentarios.



Fuente: Autores

Figura 12. Página de Estudiante (sub-pestaña clase) enviar comentarios (error).



Fuente: Autores

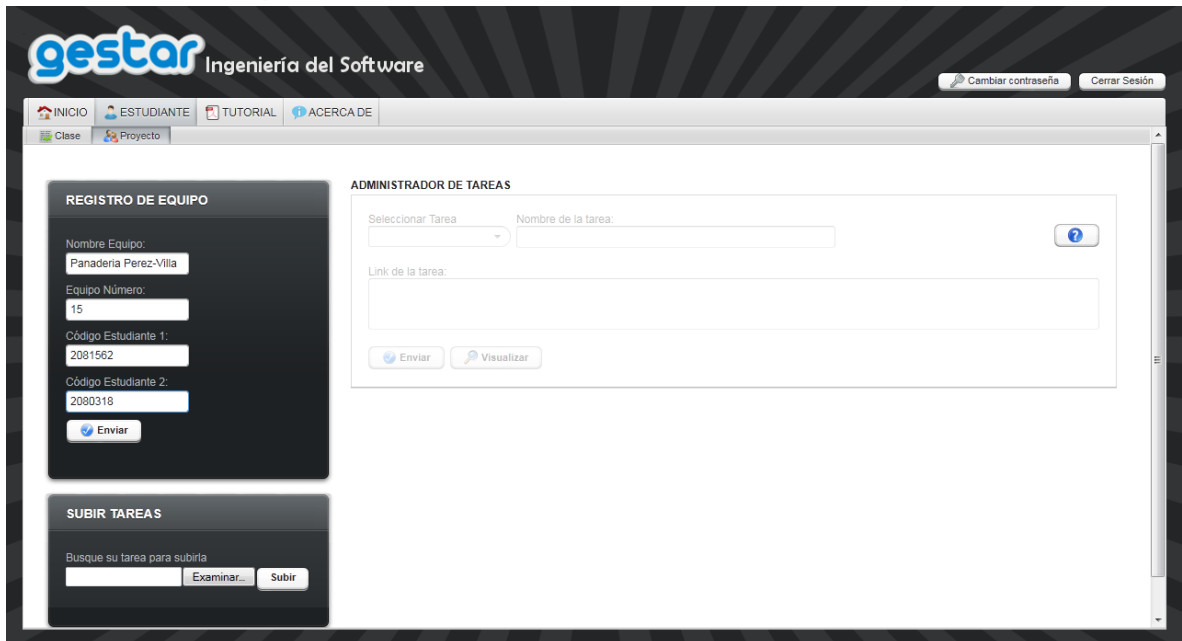
5.3.1.3.2 Proyecto estudiante

Es la segunda sub-pestaña de la pestaña estudiante. En esta sub-pestaña el estudiante registra el equipo de trabajo de todo el semestre y cuenta con dos opciones para realizar el envío de sus tareas, una forma es por medio de una subida directa al servidor y la otra es por medio de Google Docs. La organización de esta sub-pestaña cuenta con tres paneles.

Registro de equipo

Este panel contiene con tres campos de texto y un botón “Enviar”. En el primer campo, el estudiante ingresa el número del equipo asignado por el profesor, si este campo contiene letras o está vacío, se informa de esta anomalía al estudiante mediante un mensaje tipo error. En el segundo campo, se muestra el código del estudiante que inicio sesión. En el tercer campo, si es la primera vez que se va a usar este panel, este campo aparecerá vacío para el estudiante que inicio sesión, el cual ingresará el código de su compañero. Si es el código ya está en otro equipo o es un código inválido, se informa de esta anomalía al estudiante mediante un mensaje tipo error. Los datos de este panel solamente se envían haciendo click en el botón “Enviar”, el cual también cuenta con validaciones para los tres campos mencionados.

Figura 13. Página de Estudiante (sub-pestaña proyecto) registro de equipo.



Fuente: Autores

Figura 14. Página de Estudiante (sub-pestaña proyecto) registro de equipo (notificación exitosa).



Fuente: Autores

Subir tareas

Este panel cuenta con la primera opción que se brinda al estudiante para enviar sus tareas. El estudiante selecciona su tarea y la envía mediante el botón “Subir”. Se implementa una subida directa de las tareas al servidor.

Figura 15. Página de Estudiante (sub-pestaña proyecto) enviar tareas (opción 1).



The screenshot shows the 'gestar' web application interface. At the top, there is a navigation bar with the 'gestar' logo and 'Ingeniería del Software' text. Below the navigation bar, there are tabs for 'Inicio', 'Estudiante', 'Tutorial', and 'Acerca de'. The main content area is divided into several sections:

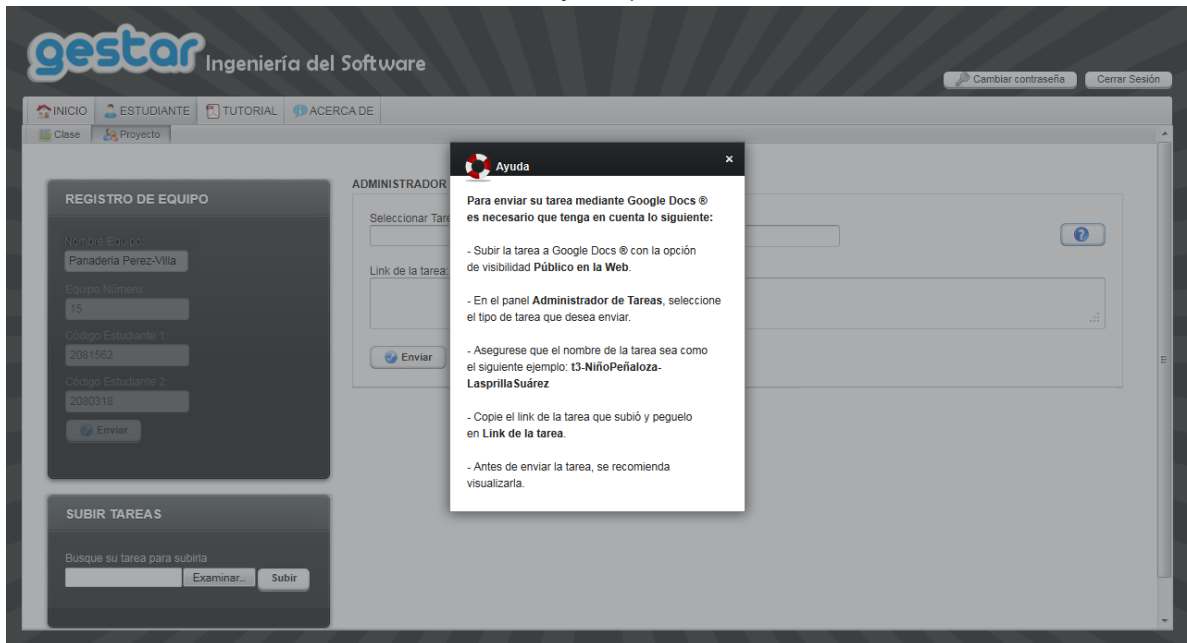
- REGISTRO DE EQUIPO:** A dark-themed panel containing fields for 'Nombre Equipo' (Panaderia Perez-Villa), 'Equipo Número' (15), 'Código Estudiante 1' (2081562), and 'Código Estudiante 2' (2080318). There is an 'Enviar' button at the bottom.
- SUBIR TAREAS:** A dark-themed panel with a search bar labeled 'Busque su tarea para subir' and buttons for 'Examinar...' and 'Subir'.
- ADMINISTRADOR DE TAREAS:** A light-themed panel with a 'Seleccionar Tarea' dropdown menu, a 'Nombre de la tarea:' text input field, and a 'Link de la tarea:' text input field. There are 'Enviar' and 'Visualizar' buttons at the bottom.

Fuente: Autores

Administrador de tareas

Este panel es la segunda opción que se brinda al estudiante para enviar sus tareas. El estudiante selecciona el tipo de tarea que desea enviar y seguidamente el nombre de la tarea. En el área de texto de la parte inferior, el estudiante copia el link de su tarea previamente subida a Google Docs. Este panel cuenta con un botón que muestra una sub-ventana la cual ayuda a inquietudes del estudiante con respecto al envío de su tarea. El panel cuenta con un botón “Visualizar” que permite al estudiante mirar su tarea antes de enviarla. La tarea solamente se envía haciendo click en el botón “Enviar”.

Figura 16. Página de Estudiante (sub-pestaña proyecto) enviar tareas (ventana ayuda).



Fuente: Autores

Figura 17. Página de Estudiante (sub-pestaña proyecto) enviar tareas (opción 2).



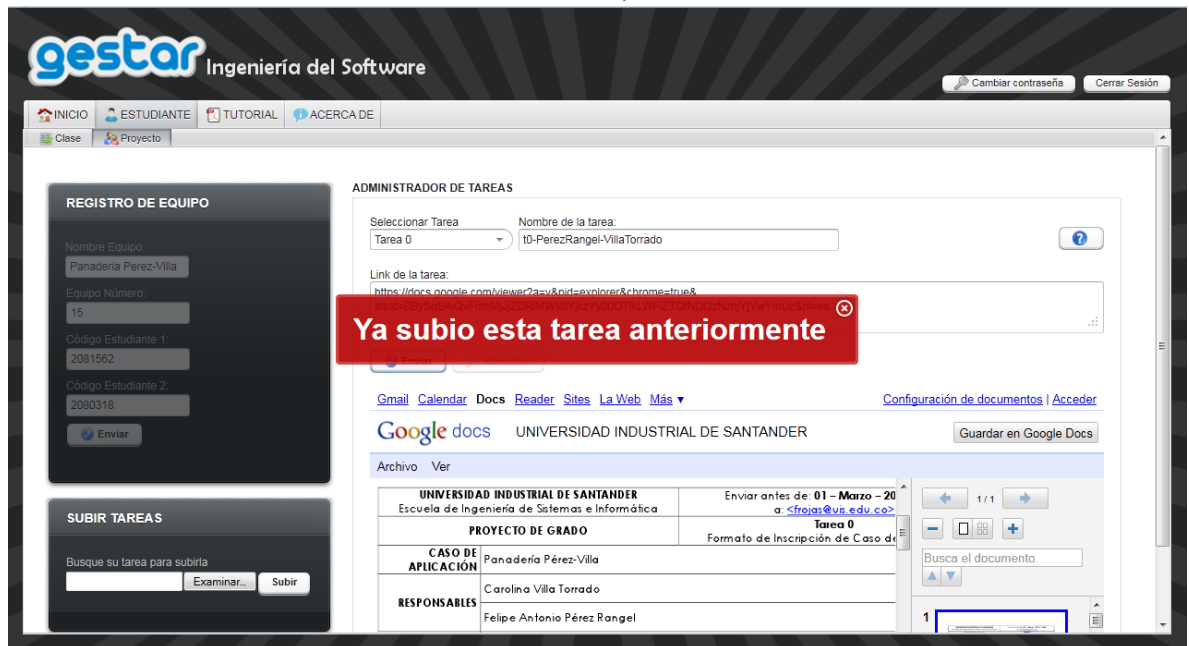
Fuente: Autores

Figura 18. Página de Estudiante (sub-pestaña proyecto) enviar tareas (opción 2 notificación).



Fuente: Autores

Figura 19. Página de Estudiante (sub-pestaña proyecto) enviar tareas (opción 2 error).



Fuente: Autores

5.3.1.4 PÁGINA DE PROFESOR

La pestaña profesor sólo es visible para el profesor. En esta pestaña se encuentran tres sub-pestañas las cuales se describen a continuación.

5.3.1.4.1 Clase profesor

Esta sub-pestaña es la que se muestra siempre por defecto cuando el usuario hace click en la pestaña profesor. En esta sub-pestaña el profesor registra los grupos y los estudiantes, sube el material de clase y puede revisar los comentarios y sugerencias que los estudiantes le han mandado. La organización de esta sub-pestaña cuenta con tres paneles.

Figura 20. Página de Profesor (sub-pestaña clase)



Fuente: Autores

Registro

Este panel contiene con cuatro campos de textos y un botón “Enviar”. En el primer campo el profesor ingresa el grupo de la materia. Los demás campos se llenan con los datos de los estudiantes que pertenecen al grupo de esa materia. Una vez se guarda el primer estudiante, el campo de texto donde se ingresa el grupo se inactiva, ya que se sobre entiende que los siguientes estudiantes a ser agregados pertenecen al mismo grupo. Si algún campo se encuentra vacío o ya se ha registrado el estudiante, se informa esta anomalía mediante un mensaje tipo error. Sólo se guardan los estudiantes haciendo click en el botón “Enviar”.

Figura 21. Página de Profesor (sub-pestaña clase) registro de estudiantes.

The screenshot shows the 'gestar' web application interface. The header includes the logo 'gestar Ingeniería del Software' and navigation links for 'INICIO', 'ESTUDIANTE', 'PROFESOR', 'TUTORIAL', and 'ACERCA DE'. The 'PROFESOR' tab is active, and the 'Clase' sub-tab is selected. The main content area is divided into two sections: 'REGISTRO' and 'COMENTARIOS'.

REGISTRO

Grupo: B No: 1 Código: 2061389

Nombre: Laura Juliana

Apellidos: Gomez Martinez

SUBIR MATERIAL DE CLASE

Busque el material para subirlo

COMENTARIOS

CODIGO	COMENTARIO
2061461	Buenos dias profesor. Quisiera revisar mis notas del parcial 1.
2060578	Buenas tardes profesor no tengo aun la nota de mi tarea 0. gracias
2060579	Profe, no tengo la nota de la tarea 2 actualizada aun. gracias
2061461	buenas tardes profesor, mi nota de la exposicion no es la correcta. gracias

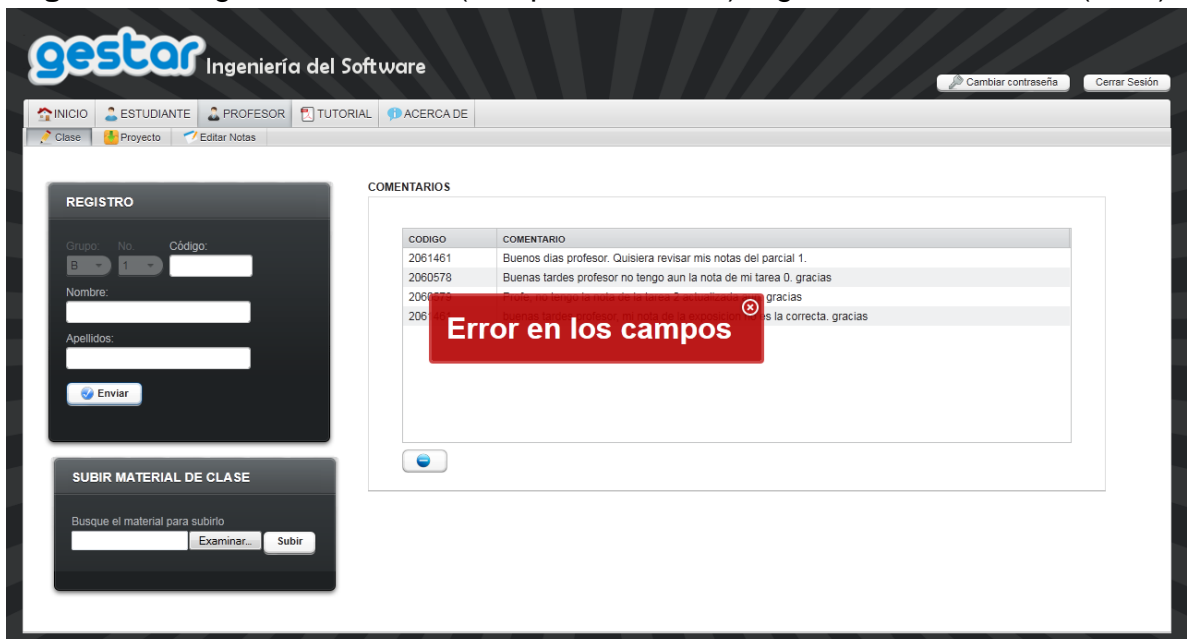
Fuente: Autores

Figura 22. Página de Profesor (sub-pestaña clase) registro de estudiantes (notificación).



Fuente: Autores

Figura 23. Página de Profesor (sub-pestaña clase) registro de estudiantes (error).



Fuente: Autores

Subir material de clase

En este panel el profesor selecciona los archivos que son material de la clase y los envía mediante el botón “Subir”. Se implementa una subida directa de estos archivos al servidor. Sólo se envían los archivos haciendo click en el botón “Subir”.

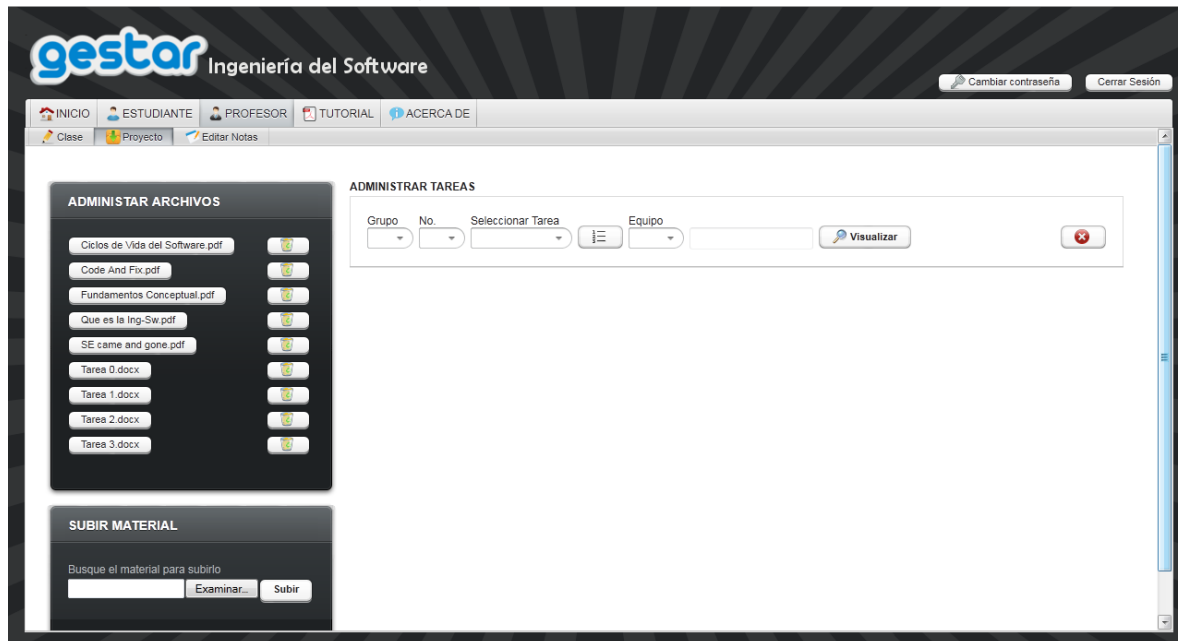
Comentarios

Este panel contiene una tabla en donde el profesor puede leer los comentarios o sugerencias que los estudiantes han hecho respecto a sus notas. En la parte inferior hay un botón que permite borrar los comentarios una vez se hayan seleccionado de la tabla.

5.3.1.4.2 Proyecto profesor

Es la segunda sub-pestaña de la pestaña profesor. En esta sub-pestaña el profesor puede descargar y borrar los archivos que el o los estudiantes han subido, subir material de clase respecto al proyecto y visualizar las tareas que los estudiantes han subido mediante Google Docs. La organización de esta sub-pestaña cuenta con tres paneles.

Figura 24. Página de Profesor (sub-pestaña proyecto)



Fuente: Autores

Administrar archivos

Este panel contiene un listado de los archivos (material de clase) y las tareas previamente subidos al servidor por el profesor y el estudiante respectivamente. Estos archivos están disponibles para ser descargados en cualquier momento por el profesor haciendo click en el botón que los representa y también pueden ser borrados con su respectivo botón. Los archivos están nombrados con su extensión.

Figura 25. Página de Profesor (sub-pestaña proyecto) descargar archivos.



Fuente: Autores

Figura 26. Página de Profesor (sub-pestaña proyecto) eliminar archivos.



Fuente: Autores

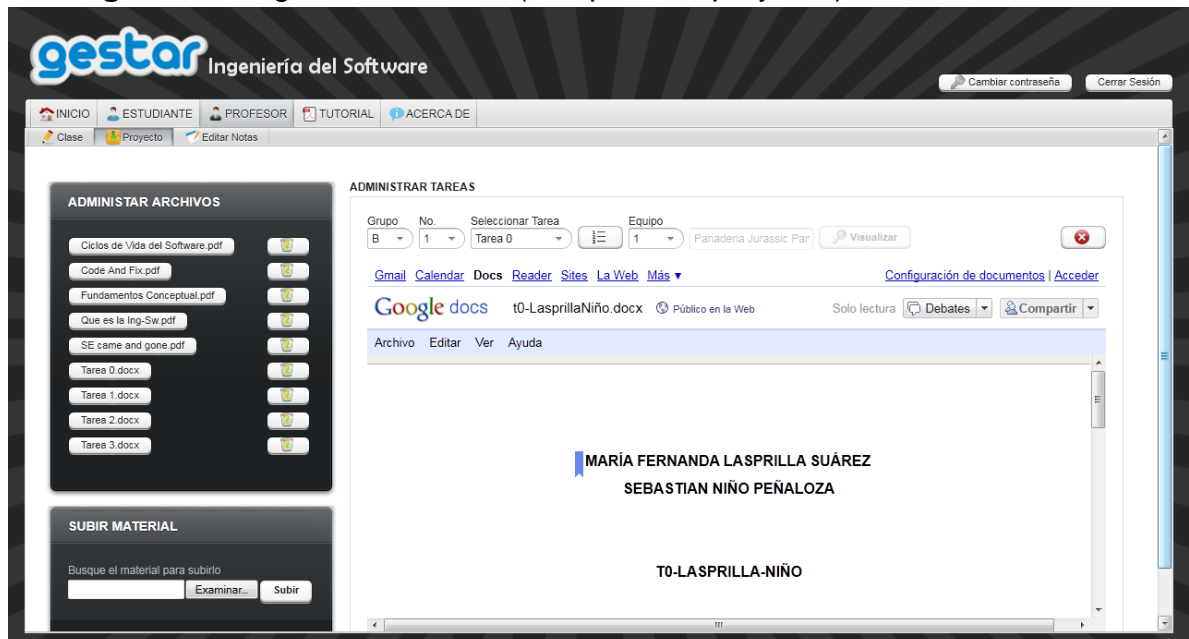
Subir material

En este panel el profesor selecciona los archivos que son relevantes para los estudiantes en el desarrollo del proyecto clase y los envía mediante el botón “Subir”. Se implementa una subida directa de estos archivos al servidor. Sólo se envían los archivos haciendo click en el botón “Subir”.

Administrar tareas

En este panel el profesor selecciona el tipo de tarea que desea visualizar y seguidamente se hace click en un botón para listarlas. El profesor elige cual archivo desea ver y para visualizarlo mediante Google Docs el panel cuenta con el botón “Visualizar”. Para poder visualizar otro archivo, en la parte superior derecha hay un botón que permite cerrar el archivo que se está visualizando actualmente para así poder ver otro.

Figura 27. Página de Profesor (sub-pestaña proyecto) Administrar tareas.



The screenshot shows the 'gestar' web application interface for a professor. The top navigation bar includes 'INICIO', 'ESTUDIANTE', 'PROFESOR', 'TUTORIAL', and 'ACERCA DE'. The main content area is titled 'ADMINISTRAR TAREAS' and features a sidebar on the left with 'ADMINISTRAR ARCHIVOS' and 'SUBIR MATERIAL' sections. The main area displays a Google Docs document titled 't0-LasprillaNiño.docx' with the text 'MARIÁ FERNANDA LASPRILLA SUÁREZ SEBASTIAN NIÑO PEÑALOZA' and 'T0-LASPRILLA-NIÑO'. The interface also includes a search bar for uploading material and a 'Visualizar' button for viewing the task.

Fuente: Autores

5.3.1.4.3 Editar notas

Es la tercera sub-pestaña de la pestaña profesor. En esta sub-pestaña el profesor puede visualizar y editar las notas de los estudiantes de los diferentes grupos. La organización de esta sub-pestaña cuenta con un sólo panel.

Notas

En este panel el profesor selecciona el grupo de la asignatura de la cual desea ver las notas. Las notas se muestran en una tabla, para editar la nota de algún estudiante el profesor presiona el botón “Editar” y luego selecciona el estudiante al que desea cambiarle las notas. Una vez estén editadas las notas presiona el botón “Guardar”, el cual envía las notas a la base de datos y quedan almacenadas.

Figura 28. Página de Profesor (sub-pestaña notas)

The screenshot shows the 'gestar' web application interface. At the top, there is a logo for 'gestar Ingeniería del Software' and two buttons: 'Cambiar contraseña' and 'Cerrar Sesión'. Below the logo is a navigation menu with options: 'INICIO', 'ESTUDIANTE', 'PROFESOR', 'TUTORIAL', and 'ACERCA DE'. Underneath the navigation menu is a sub-menu with 'Clase', 'Proyecto', and 'Editar Notas'. The main content area is titled 'NOTAS' and contains a form for selecting a group. It has a 'Grupo:' label, a dropdown menu, and a 'No.' input field. Below these are three buttons: 'Ver Notas', 'Editar', and 'Guardar'. The main part of the page is a table with the following columns: CODIGO, NOMBRE, APELLIDO, T0, T1, T2, T3, EXP, ASIS, P1, P2, and DEF. The table is currently empty. At the bottom of the page, there are several empty input fields.

Fuente: Autores

Figura 29. Página de Profesor (sub-pestaña notas) visualizar notas grupo.

The screenshot shows the 'gestar' web application interface. At the top, there is a navigation menu with options: INICIO, ESTUDIANTE, PROFESOR (selected), TUTORIAL, and ACERCA DE. Below this, there are sub-tabs: Clase, Proyecto, and Editar Notas. The main content area is titled 'NOTAS' and contains a form with 'Grupo: B' and 'No: 1'. Below the form is a table of student grades.

CODIGO	NOMBRE	APELLIDO	T0	T1	T2	T3	EXP	ASIS	P1	P2	DEF
2060578	Sebastian	Niño Peñalozca	4.0	4.3	3.6	3.8	4.0	4.0	4.5	3.8	3.9
2060579	Julian Fernando	Uribe Sabogal	2.0	3.8	4.3	3.6	3.9	4.0	4.6	4.1	3.8
2061389	Laura Juliana	Gomez Martinez	0	0	0	0	0	0	0	0	0
2061461	Maria Fernanda	Lasprilla Suarez	3.0	4.2	3.6	3.7	3.5	4.0	4.6	3.9	4.0
2061462	Angie Vanessa	Gomez Chaparro	4.0	2.7	3.7	3.9	4.4	4.0	4.5	3.9	4.1
2070489	Andrea Carolina	Serrano Figueroa	0	0	0	0	0	0	0	0	0
2070713	Miguel	Castillo Jerez	0	0	0	0	0	0	0	0	0

Fuente: Autores

Figura 30. Página de Profesor (sub-pestaña notas) editar notas grupo.

The screenshot shows the same 'gestar' web application interface as Figure 29, but in edit mode. The 'NOTAS' page displays the same table of student grades. The row for student 2061389 (Laura Juliana Gomez Martinez) is highlighted in blue. Below the table, there is a form for editing the grades for this student, with input fields for each grade column.

CODIGO	NOMBRE	APELLIDO	T0	T1	T2	T3	EXP	ASIS	P1	P2	DEF
2060578	Sebastian	Niño Peñalozca	4.0	4.3	3.6	3.8	4.0	4.0	4.5	3.8	3.9
2060579	Julian Fernando	Uribe Sabogal	2.0	3.8	4.3	3.6	3.9	4.0	4.6	4.1	3.8
2061389	Laura Juliana	Gomez Martinez	0	0	0	0	0	0	0	0	0
2061461	Maria Fernanda	Lasprilla Suarez	3.0	4.2	3.6	3.7	3.5	4.0	4.6	3.9	4.0
2061462	Angie Vanessa	Gomez Chaparro	4.0	2.7	3.7	3.9	4.4	4.0	4.5	3.9	4.1
2070489	Andrea Carolina	Serrano Figueroa	0	0	0	0	0	0	0	0	0
2070713	Miguel	Castillo Jerez	0	0	0	0	0	0	0	0	0

Below the table, the edit form for student 2061389 is visible:

2061389 Laura Juliana Gomez Martinez 4.0 3.0 2.8 3.6 3.7 3.9 4.3 3.2 3.6

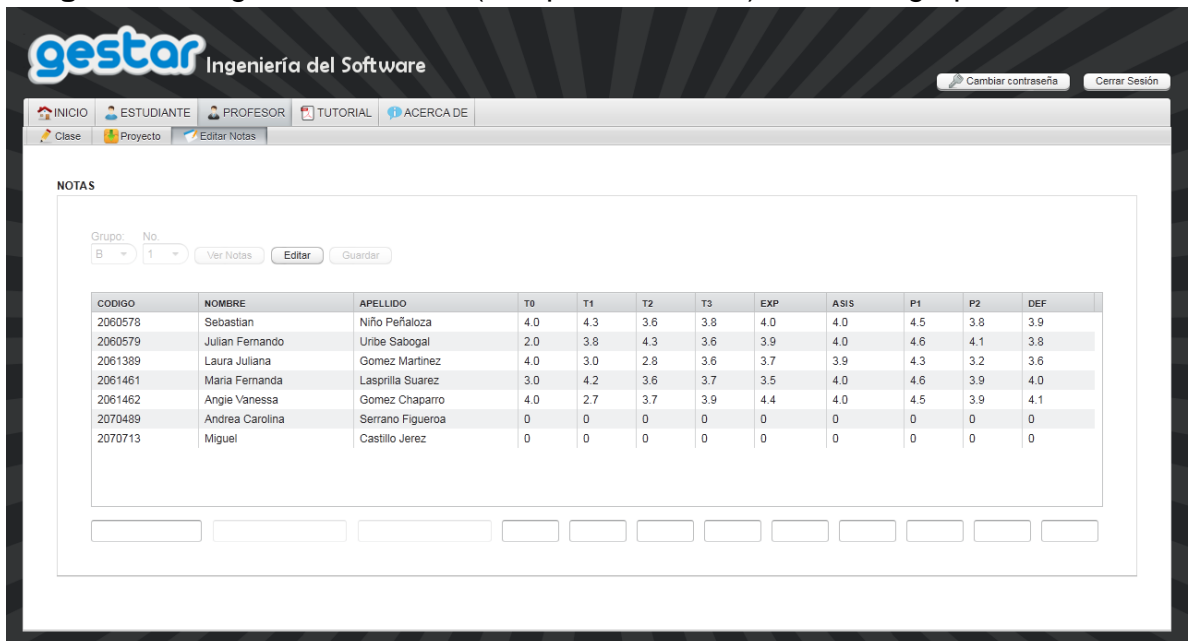
Fuente: Autores

Figura 31. Página de Profesor (sub-pestaña notas) guardar notas grupo.



Fuente: Autores

Figura 32. Página de Profesor (sub-pestaña notas) ver notas grupo cambiadas.

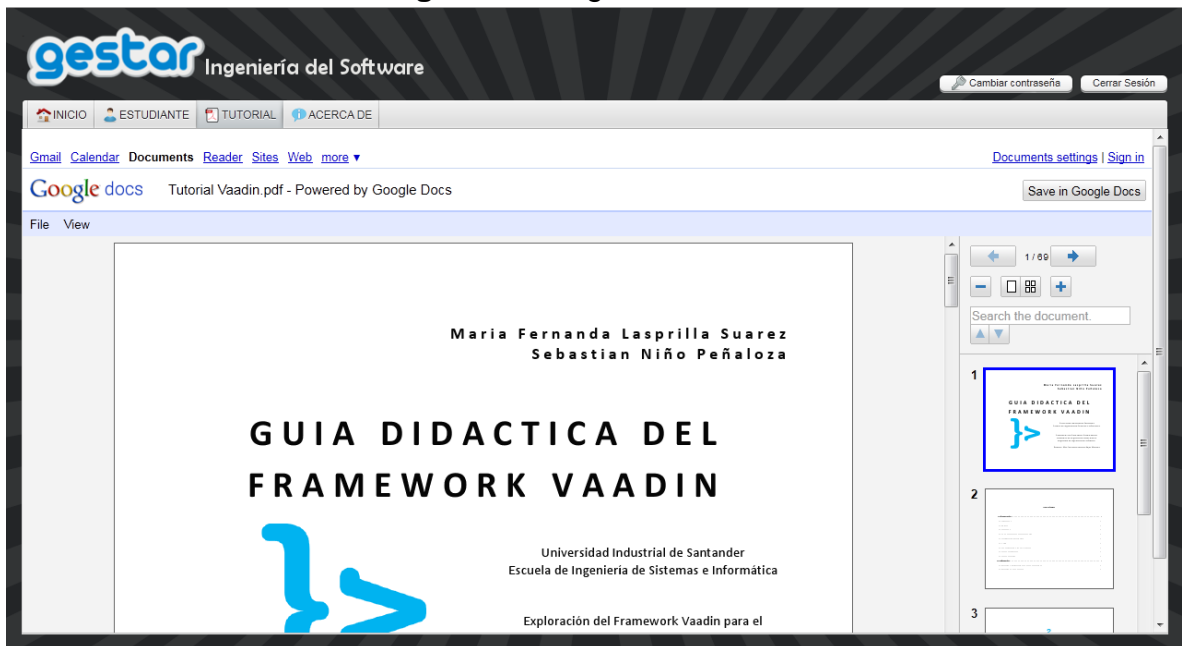


Fuente: Autores

5.3.1.5 PÁGINA DE TUTORIAL

La pestaña tutorial muestra la guía didáctica del Framework Vaadin desarrollada como uno de los objetivos específicos planteados en este libro. Esta guía didáctica se visualiza mediante Google Docs y se puede descargar para su aprendizaje.

Figura 33. Página de Tutorial



Fuente: Autores

5.3.1.6 PÁGINA DE ACERCA DE

La pestaña acerca de contiene una sub-ventana en donde se muestran los propósitos de GESTAR, la versión de GESTAR y los desarrolladores.

Figura 34. Página de Acerca de

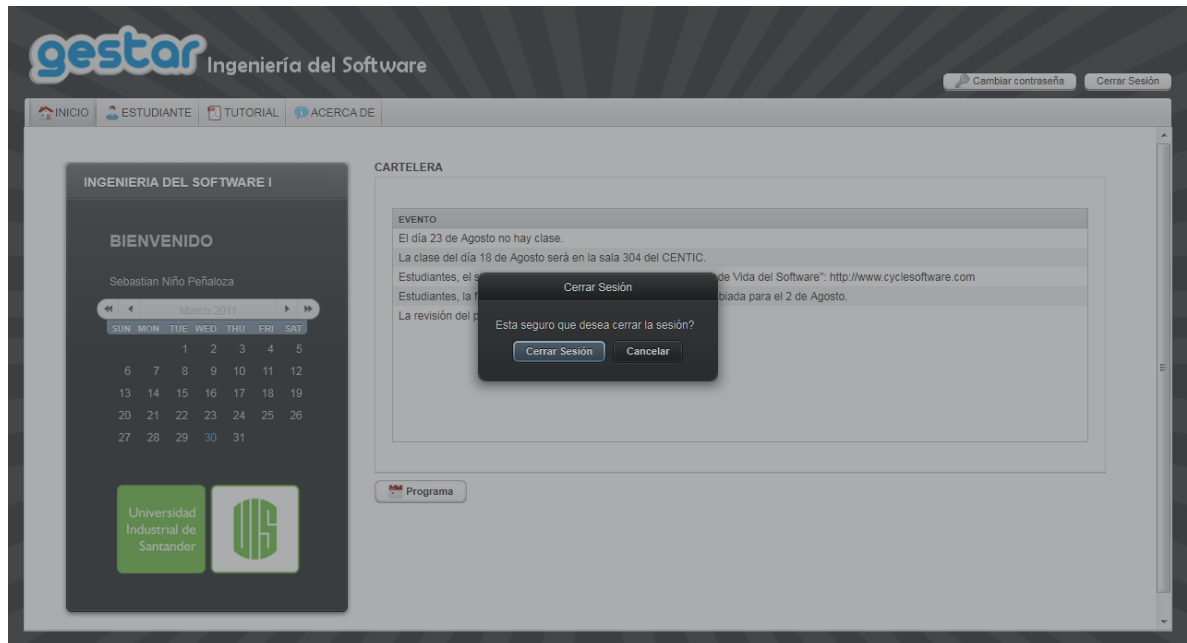


Fuente: Autores

5.3.1.7 CERRAR SESIÓN

Es un botón que se encuentra en la parte superior derecha y tiene como función cerrar la sesión actual del usuario que inicio sesión en GESTAR. Al ser presionado, se abre una sub-ventana en donde se pregunta al usuario si realmente desea cerrar la sesión. Esta sub-ventana tiene dos botones.

Figura 35. Cerrar sesión



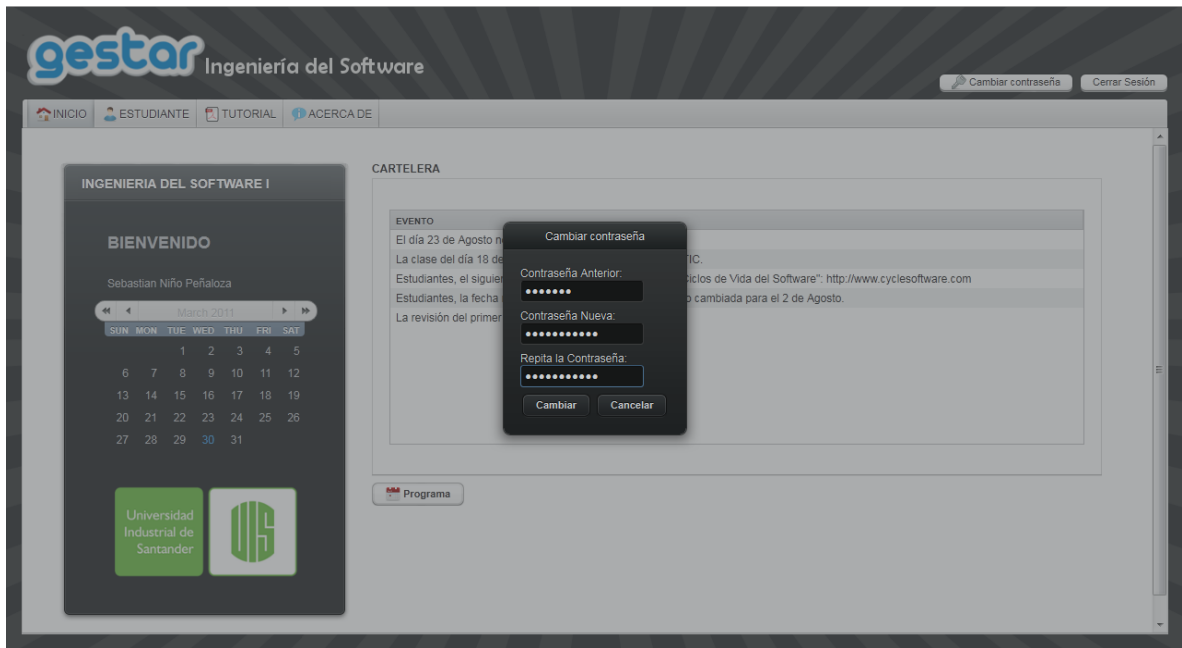
Fuente: Autores

5.3.1.8 CAMBIAR CONTRASEÑA

Es un botón que se encuentra al lado izquierdo del botón cerrar sesión y tiene como función cambiar la contraseña del usuario por una nueva. Al ser presionado, se abre una sub-ventana que contiene tres campos de texto y dos botones. El primer campo exige al usuario ingresar la contraseña actual. El segundo y el tercer campo exigen al usuario una nueva contraseña y estos dos campos tienen que ser iguales. En la parte de abajo se encuentra el botón “Cambiar” el cual tiene la función de realizar correctamente el cambio de la contraseña y cerrar esta sub-ventana.

Si alguno de los campos ha sido ingresado incorrectamente, se informa al usuario esta anomalía mediante un mensaje tipo error. El botón “Cancelar” cierra esta sub-ventana sin realizar ningún cambio.

Figura 36. Cambiar contraseña



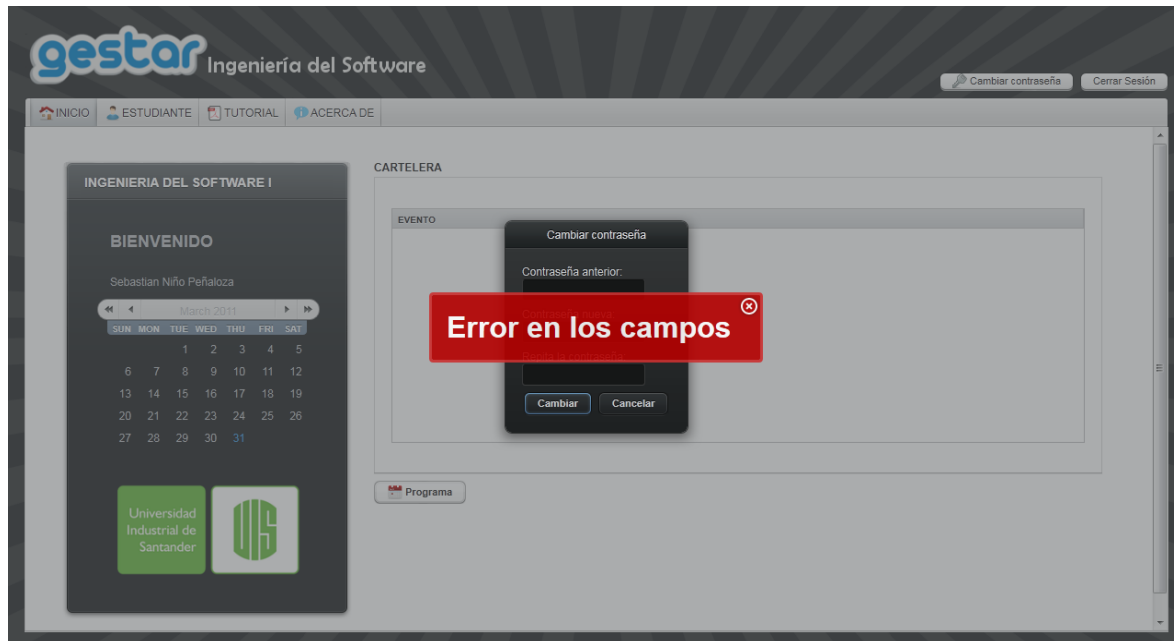
Fuente: Autores

Figura 37. Cambiar contraseña (notificación)



Fuente: Autores

Figura 38. Cambiar contraseña (error)



Fuente: Autores

5.4 PRUEBAS REALIZADAS A LA APLICACIÓN WEB GESTAR

Las pruebas se realizaron de manera constante e iterativa y fueron realizadas durante el desarrollo del proyecto. Durante la fase de diseño e implementación de realizaron pruebas sobre la aplicación con el fin de detectar los posibles errores y corregirlos a tiempo. De la misma manera se realizaron pruebas de usabilidad para verificar que el sistema sea entendible por el usuario, pruebas funcionales para comprobar que se cumplieron los requisitos planteados en los casos de uso y pruebas en diferentes plataformas para comprobar la portabilidad de la aplicación.

5.4.1 PRUEBAS DURANTE EL DESARROLLO

La metodología de desarrollo AUP (Agile Unified Process) requiere que se estén realizando pruebas durante la construcción para verificar el correcto funcionamiento del código programado.

5.4.2 PRUEBAS DE USABILIDAD

Las pruebas de usabilidad permiten conocer si la interfaz de usuario es intuitiva y amigable y si funciona correctamente. Para comprobar esto se dio la aplicación a 10 usuarios estudiantes de pregrado de la escuela de Ingeniería de Sistemas e Informática y se analizaron los siguientes aspectos:

- Número de actividades realizadas satisfactoriamente
- Tiempo tomado para realizar una actividad
- Número de errores detectados en la aplicación

Para analizar las pruebas realizadas se le pidió al usuario que iniciara sesión, registrara un equipo y enviara una tarea. El 80% de los usuarios logró hacer las tres actividades exitosamente y el 20% restante sólo lograron realizar algunas.

El tiempo para realizar una actividad fue óptimo para el 80% de los usuarios. El 20% restante tardó un poco más.

No se detectaron errores de la aplicación durante las actividades realizadas por los usuarios.

5.4.3 PRUEBAS FUNCIONALES

Las pruebas funcionales sirven para validar si el comportamiento de la aplicación cumple o no con las especificaciones planteadas. Estas se hicieron en base a los casos de uso y fueron aplicadas por los desarrolladores del presente proyecto. En todos los casos los resultados fueron favorables. La aplicación funciona de acuerdo a lo especificado.

6. GUIA DIDACTICA

Como objetivo específico de este proyecto se planteó el desarrollo de una guía didáctica, en la cual se encuentra la documentación necesaria para aprender los elementos básicos del Framework Vaadin. Esta guía está disponible para toda la comunidad interesada en esta tecnología.

La guía didáctica empieza con una breve introducción al Framework Vaadin, explica sus principales características, biblioteca y la arquitectura que este maneja, todo esto con el fin de dar al lector un marco teórico necesario para entender el funcionamiento del mismo.

Seguidamente se explica cómo realizar la instalación de NetBeans 6.9.1 con Apache Tomcat 6.0.26, la instalación del plugin de Vaadin y la actualización de la biblioteca.

Un siguiente capítulo explica la creación de una aplicación web, con los componentes básicos de interfaz de usuario, este contiene las líneas de código y una imagen en donde se muestra el resultado de ejecutar dicho código.

Para dar estilo y formato a los componentes de la interfaz de usuario, Vaadin integra el uso de CSS y Themes los cuales se explican en un capítulo posterior de esta guía didáctica.

Para finalizar se cuenta con un caso de ejemplo el cual explica e ilustra cómo crear un proyecto nuevo en Vaadin implementando algunos componentes de la interfaz de usuario.

La guía didáctica se encuentra en la aplicación web GESTAR.

7. CONCLUSIONES

Los objetivos planteados en el desarrollo del proyecto se cumplieron. El desarrollo de este proyecto de grado, permitió el aprendizaje de una tecnología de punta consolidándolo en la creación de una guía didáctica la cual refleja todos los conocimientos adquiridos e invita al lector a aprender esta tecnología facilitándole su experiencia.

Se realizó una investigación de los diferentes Frameworks web que utilizan Java como lenguaje de programación. A través de un estudio se realizó la elección del Framework Vaadin fundamentándolo como la herramienta de desarrollo para este proyecto. Se descubrió su funcionalidad, potencial y servicios y sobre todo se comprobó la facilidad a la hora de ponerlo en la práctica.

Se desarrolló una aplicación web que ayude a los estudiantes como soporte en el proceso de aprendizaje, estimulándolos a utilizar nuevas tecnologías y permitiendo al profesor tener al alcance la información de sus estudiantes. Esto fue base para el desarrollo de este proyecto.

Como resultado obtenido con la realización de este proyecto de grado, se obtuvo una aplicación web, que da soporte a las clases, prestando servicios para el estudiante y el profesor.

Cabe destacar que el desarrollo de la aplicación web ayudó a afianzar conocimientos y conceptos en la Ingeniería del software, el diseño y la educación, y permitió acceder a la experiencia del desarrollo de un proyecto de software.

La aplicación web desarrollada fue diseñada teniendo en cuenta la metodología del profesor de la asignatura de Ingeniería del Software I.

El desarrollo del proyecto según la metodología planteada permitió organizar el trabajo de manera correcta, siguiendo un orden que facilitó la documentación y organizar trabajos futuros.

Vaadin es un Framework para la creación de aplicaciones web enriquecidas, el uso de esta tecnología permitió desarrollar una aplicación robusta, completa, funcional y con una interfaz de usuario completamente amigable e intuitiva.

8. RECOMENDACIONES

Después de finalizar el desarrollo del proyecto, es posible realizar recomendaciones para futuros trabajos:

Que en la Universidad Industrial de Santander y en la escuela de Ingeniería de Sistemas e Informática se sigan promoviendo y realizando estudios en las nuevas tecnologías que el mercado ofrece, creando así estudiantes y profesionales más competitivos, con conocimientos de auge en la actualidad.

Incorporar nuevas funcionalidades a la aplicación, permitiendo así un mayor aprovechamiento de los componentes que Vaadin ofrece, tales como un foro de discusión, el servicio de e-mail, creación de perfil de usuario con una fotografía, el uso de multimedia (video, audio, imágenes), entre otros.

Es recomendable que para futuras versiones de GESTAR, se revise la guía didáctica de este presente documento, en la cual se encuentra toda la información relacionada con Vaadin y su funcionamiento.

Para futuros proyectos que den continuidad a la aplicación web GESTAR, es necesaria la actualización de la biblioteca de Vaadin, ya que esta presenta mejoras en los componentes, brechas en seguridad y arreglo en los errores reportados a los desarrolladores de Vaadin.

La persona interesada en aprender Vaadin, puede ayudar a la comunidad internacional de Vaadin en mejoras y reporte de errores relacionados con el funcionamiento del Framework, para su futura corrección en la actualización de la biblioteca en una nueva versión.

BIBLIOGRAFIA

- [1] Arquitectura del software. Model-View-Controller. Disponible en internet en: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/rivera_l_a/capitulo2.pdf.
Fecha de consulta: Septiembre de 2010.
- [2] Arquitectura Modelo/Vista/Controlador. Disponible en internet en: <http://abad.galeon.com/mvc.pdf>. Fecha de consulta: Septiembre de 2010.
- [3] Comparativa de Frameworks web. Disponible en internet en: http://www.javahispano.org/contenidos/archivo/100/frameworks_web.pdf. Fecha de consulta: Septiembre de 2010.
- [4] El Proceso Unificado Ágil. Disponible en internet en: <http://cgi.una.ac.cr/AUP/index.html>. Fecha de consulta: Octubre de 2010.
- [5] Grönroos, Marko. "Book of Vaadin". Vaadin Ltd. 2010.
- [6] Investigación y desarrollo de aplicaciones web enriquecidas. Disponible en internet en: <http://dspace.fica.ufro.cl:8080/jspui/bitstream/123456789/228/1/Hugo%20Pineda.pdf>. Fecha de consulta: Septiembre de 2010.
- [7] Learning Vaadin. Disponible en internet en: <http://java.dzone.com/articles/learning-vaadin>. Fecha de consulta: Octubre de 2010.
- [8] Model View Controler. Disponible en internet en: <http://blogdeaitor.wordpress.com/2008/10/20/model-view-controller/#more-156>.
Fecha de consulta: Septiembre de 2010.

- [9] Modelo Vista Controlador. Disponible en internet en:
http://es.wikipedia.org/wiki/Modelo_Vista_Controlador. Fecha de consulta:
Septiembre de 2010.
- [10] Patrón de diseño MVC (Modelo Vista Controlador). Disponible en internet
en: http://pis.unicauca.edu.co/moodle/file.php/291/Patron_Disenio_MVC.pdf Fecha
de consulta: Septiembre de 2010.
- [11] ¿Qué es un Framework?. Disponible en internet en:
<http://jordisan.net/blog/2006/que-es-un-framework> Fecha de consulta: Septiembre
de 2010.
- [12] ¿Qué es un Framework web?. Disponible en internet en:
http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf. Fecha de
consulta: Septiembre de 2010.
- [13] Rich Internet Applications. Disponible en internet en:
http://en.wikipedia.org/wiki/Rich_Internet_application. Fecha de consulta:
Septiembre de 2010.
- [14] The Agile Unified Process (AUP). Disponible en internet en:
<http://www.ambyssoft.com/unifiedprocess/agileUP.html>. Fecha de consulta: Octubre
de 2010.
- [15] Vaadin Fundamentals. Disponible en internet en:
<http://www.blackbeltfactory.com/ui#CoursePage/12648322/EN>. Fecha de consulta:
Octubre de 2010.
- [16] Vaadin – Learn. Disponible en internet en: <http://vaadin.com/learn> Fecha de
consulta: Octubre de 2010.

[17] Ventajas de una aplicación web. Disponible en internet en:
[http://www.esenciahumana.com.mx/Servicios/AplicacionesWeb/VentajasBeneficio
sAplicaciones.html](http://www.esenciahumana.com.mx/Servicios/AplicacionesWeb/VentajasBeneficiosAplicaciones.html). Fecha de consulta: Septiembre de 2010.

ANEXO A. DOCUMENTO DE ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

El presente documento recoge la información respecto a la Especificación de Requisitos de Software (ERS) de la aplicación web GESTAR, el cual ha sido el producto final como resultado del aprendizaje del Framework Vaadin.

Producto	APLICACIÓN WEB GESTAR PARA LA ASIGNATURA DE INGENIERÍA DEL SOFTWARE I
Etapa	Análisis y diseño
Actividad	ESPECIFICACIÓN DE REQUISITOS
Participantes	María Fernanda Lasprilla Suárez Sebastian Niño Peñaloza

PROPOSITO

Este documento de requerimientos tiene como fin analizar los alcances técnicos del proyecto y especificar los diferentes servicios que proveerá la aplicación en cada uno de los módulos que comprende, todo esto para tener claridad en el producto final.

ALCANCE DEL SISTEMA

La aplicación web GESTAR, es una herramienta que permite administrar y gestionar una asignatura específica durante un semestre.

La aplicación web GESTAR se ocupará de la gestión de usuarios con diferentes privilegios, el envío de tareas, de material de clase y la administración de notas. Lo anterior será reflejado en la aplicación web GESTAR con el fin de proporcionar estas utilidades al profesor y a los estudiantes, quienes finalmente serán los usuarios de esta aplicación.

Al finalizar el proyecto se esperan los siguientes beneficios:

- Una herramienta que permita a los estudiantes acceder en cualquier momento y lugar para estar al tanto de la asignatura.
- Un sistema para que el profesor pueda tener un seguimiento de los estudiantes y de la asignatura.

DESCRIPCIÓN GLOBAL DEL PRODUCTO A DESARROLLAR PERSPECTIVA DEL PRODUCTO

La aplicación web GESTAR es un producto software independiente, es decir, no depende de otros productos software, desarrollado con herramientas libres.

La información que provee el sistema es libre, es asequible a cualquier persona que se encuentre registrada.

La interfaz gráfica de usuario de la aplicación debe ser amigable e intuitiva, es decir, que permita al usuario familiarizarse con ella y que sea de fácil uso.

La aplicación web será desarrollada en lenguaje de programación Java sobre la plataforma Java Enterprise Edition 5 (JEE5), manejará como motor de base de datos MySQL versión 5.1.6. y como servidor web y de servlets Apache Tomcat versión 7.0.8.

FUNCIONES DEL PRODUCTO

- A través de una aplicación web, el profesor almacenará en la base de datos los grupos y los estudiantes que pertenecen a él, cargará las tareas y material de clase, editará las notas de los estudiante, visualizará

comentarios enviados por el estudiante, publicará noticias en la cartelera, visualizará el tutorial y cambiará la contraseña.

- La aplicación web GESTAR será para uso del profesor y de los estudiantes que estén inscritos en la asignatura.
- Los estudiantes podrán visualizar las noticias publicadas en la cartelera, registrar un equipo de trabajo, enviar tareas, descargar material de clase y formatos de tarea, visualizar sus respectivas notas, enviar comentarios al profesor respecto a las notas, visualizar el tutorial y cambiar la contraseña.

OBJETIVOS DEL SISTEMA

Diseñar e implementar una aplicación web para el profesor, que permita:

- Publicar noticias
- Registrar grupos y estudiantes
- Subir archivos
- Visualizar comentarios
- Descargar archivos
- Eliminar archivos
- Visualizar tareas
- Editar notas
- Cambiar contraseña

Diseñar e implementar una aplicación web para los estudiantes, que permita:

- Visualizar noticias
- Descargar archivos
- Ver notas
- Enviar comentarios

- Registrar equipo
- Enviar tareas
- Cambiar contraseña

CARACTERISTICAS DEL USUARIO

Los usuarios de la aplicación web son los estudiantes y el profesor que pertenecen a la asignatura de Ingeniería del Software I de la Universidad Industrial de Santander.

REQUISITOS FUNCIONALES

DIAGRAMA DE CASOS DE USO Y TABLAS ASOCIADAS A LOS MISMOS

Figura 39. Casos de uso GESTAR

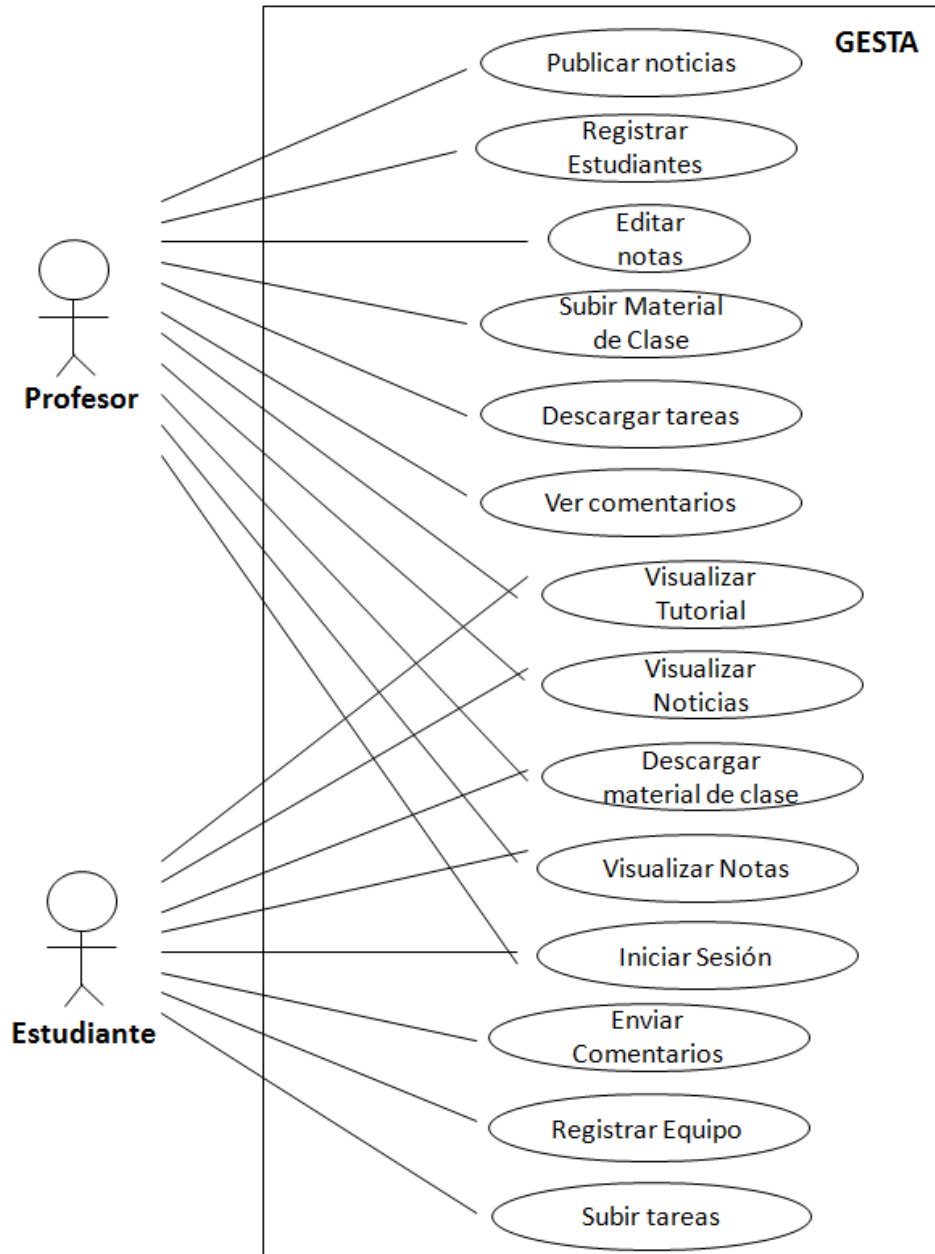


Tabla 6. Caso de uso GESTAR: Publicar Noticias

Nombre:	Publicar Noticias
Actor(es):	Profesor
Descripción:	El profesor edita noticias relacionadas con la materia publicándolas en la cartelera, permitiendo que tanto los estudiantes como el profesor mismo las puedan visualizar.
Precondición(es):	Se necesita validación correcta del usuario.
Flujo Principal:	<ol style="list-style-type: none">1. El profesor debe iniciar sesión2. El profesor debe llenar el campo de la noticia3. Pulsar el botón enviar para guardar la noticia
Pos-condición(es):	Las noticias quedan almacenadas.
Sub-flujos:	Si no escribe nada en el campo indicado no se enviará la noticia.
Excepciones:	Ninguno

Fuente: Autores

Tabla 7. Caso de uso GESTAR: Registrar Estudiantes

Nombre:	Registrar Estudiantes
Actor(es):	Profesor
Descripción:	El profesor registra estudiantes especificando el nombre, los apellidos, y el código en un grupo específico de la asignatura.
Precondición(es):	Se necesita validación correcta del usuario
Flujo Principal:	<ol style="list-style-type: none">1. El profesor debe iniciar sesión2. El profesor registra un grupo y estudiantes suministrando los nombres, apellidos y código de los estudiantes.3. Pulsa el botón enviar para insertar al nuevo

	estudiante.
Pos-condición(es):	Los estudiantes y el grupo han sido insertados satisfactoriamente.
Sub-flujos:	En caso de que el profesor deje alguno de los campos vacios no se almacenará la información.
Excepciones:	Ninguno

Fuente: Autores

Tabla 8. Caso de uso GESTAR: Editar Notas

Nombre:	Editar Notas
Actor(es):	Profesor
Descripción:	El profesor edita las notas de los estudiantes ya registrados anteriormente.
Precondición(es):	Se necesita validación correcta del usuario Los estudiantes deben estar registrados en el sistema
Flujo Principal:	<ol style="list-style-type: none"> 1. El profesor debe iniciar sesión 2. El sistema muestra una tabla con las notas de los estudiantes, seleccionando un grupo específico 3. El profesor edita las notas de los estudiantes. 4. Pulsa el botón enviar guardar las actualizaciones.
Pos-condición(es):	Las notas se han cambiado satisfactoriamente.
Sub-flujos:	En caso de que el profesor deje alguno de los campos vacios no se almacenará la información.
Excepciones:	Ninguno

Fuente: Autores

Tabla 9. Caso de uso GESTAR: Subir Material de Clase

Nombre:	Subir Material de Clase
Actor(es):	Profesor
Descripción:	El profesor sube el material de clase y los formatos de

	tarea.
Precondición(es):	Se necesita validación correcta del usuario
Flujo Principal:	<ol style="list-style-type: none"> 1. El profesor debe iniciar sesión 2. El profesor sube los archivos al servidor. 3. Pulsa el botón enviar guardar los archivos.
Pos-condición(es):	Los archivos serán subidos exitosamente.
Sub-flujos:	En caso de que el profesor no seleccione ningún archivo este no será subido.
Excepciones:	Ninguno

Fuente: Autores

Tabla 10. Caso de uso GESTAR: Descargar Tareas

Nombre:	Descargar Tareas
Actor(es):	Profesor
Descripción:	El profesor descarga las tareas que previamente fueron subidas por los equipos de los estudiantes. Sólo se listan los equipos que han subido la tarea seleccionada.
Precondición(es):	<p>Se necesita validación correcta del usuario.</p> <p>Los estudiantes deben estar registrados.</p> <p>Deben estar creados los equipos de trabajo.</p> <p>Las tareas deben estar subidas previamente.</p>
Flujo Principal:	<ol style="list-style-type: none"> 1. El profesor debe iniciar sesión 2. El profesor selecciona el grupo 3. El profesor selecciona el tipo de tarea que desea ver 4. El profesor selecciona el equipo 5. Pulsa el botón visualizar 6. Descarga la tarea
Pos-condición(es):	Las tareas serán visualizadas y descargadas

	exitosamente
Sub-flujos:	En caso de que el profesor no seleccione ninguna casilla no se visualizará ninguna tarea.
Excepciones:	Ninguno

Fuente: Autores

Tabla 11. Caso de uso GESTAR: Ver Comentarios

Nombre:	Ver comentarios
Actor(es):	Profesor
Descripción:	El profesor visualiza los comentarios enviados por los estudiantes respecto a las notas.
Precondición(es):	Se necesita validación correcta del usuario
Flujo Principal:	<ol style="list-style-type: none"> 1. El profesor debe iniciar sesión 2. El profesor visualiza en una tabla los comentarios enviados por los estudiantes.
Pos-condición(es):	Los comentarios se visualizan satisfactoriamente.
Sub-flujos:	En caso de que no se hayan enviado comentarios no se mostrarán.
Excepciones:	Ninguno

Fuente: Autores

Tabla 12. Caso de uso GESTAR: Visualizar Tutorial

Nombre:	Visualizar tutorial
Actor(es):	Profesor, Estudiante
Descripción:	El profesor y el estudiante visualizan el tutorial.
Precondición(es):	Se necesita validación correcta del usuario
Flujo Principal:	<ol style="list-style-type: none"> 1. El profesor y el estudiante deben iniciar sesión 2. El profesor y el estudiante visualizan el tutorial
Pos-condición(es):	El tutorial se visualiza satisfactoriamente.

Sub-flujos:	Ninguno
Excepciones:	Ninguno

Fuente: Autores

Tabla 13. Caso de uso GESTAR: Visualizar Noticias

Nombre:	Visualizar Noticias
Actor(es):	Profesor, Estudiante
Descripción:	El profesor y el estudiante visualizan las noticias publicadas anteriormente por el profesor.
Precondición(es):	Se necesita validación correcta del usuario.
Flujo Principal:	<ol style="list-style-type: none"> 1. El profesor y el estudiante deben iniciar sesión 2. El profesor y el estudiante visualizan en una tabla las noticias publicadas.
Pos-condición(es):	Las noticias se visualizan satisfactoriamente.
Sub-flujos:	En caso de que no se hayan enviado noticias no se mostrarán.
Excepciones:	Ninguno

Fuente: Autores

Tabla 14. Caso de uso GESTAR: Iniciar Sesión

Nombre:	Iniciar Sesión
Actor(es):	Profesor, Estudiante
Descripción:	El profesor y el estudiante digitan el código de usuario y la contraseña para acceder a los servicios de la aplicación.
Precondición(es):	El usuario debe pertenecer a la asignatura o ser el administrador.
Flujo Principal:	<ol style="list-style-type: none"> 1. Digita el código de usuario 2. Digita la contraseña

	3. Pulsa la tecla entrar para autenticar datos
Pos-condición(es):	Se mostrará la página de inicio de la aplicación
Sub-flujos:	Si el código de usuario y la contraseña no coinciden no podrá iniciar sesión.
Excepciones:	Ninguno

Fuente: Autores

Tabla 15. Caso de uso GESTAR: Enviar Comentarios

Nombre:	Enviar Comentarios
Actor(es):	Estudiante
Descripción:	El estudiante envía comentarios al profesor respecto a sus notas.
Precondición(es):	Se necesita validación correcta del usuario.
Flujo Principal:	<ol style="list-style-type: none"> 1. El usuario inicia sesión 2. El estudiante escribe el comentario 3. Pulsa la tecla enviar para guardar el comentario
Pos-condición(es):	Se almacenara el comentario en la base de datos
Sub-flujos:	Si el usuario no digita en los campos estos no se enviará.
Excepciones:	Ninguno

Fuente: Autores

Tabla 16. Caso de uso GESTAR: Registrar Equipo

Nombre:	Registrar Equipo
Actor(es):	Estudiante
Descripción:	El estudiante registra equipos conformados por 2 integrantes.
Precondición(es):	Se necesita validación correcta del usuario. Los estudiantes del equipo deben estar registrados en el sistema.
Flujo Principal:	

	<ol style="list-style-type: none"> 1. El usuario inicia sesión 2. El estudiante llena los campos para registrar el equipo, suministrando el nombre del equipo, el número y los códigos de los dos estudiantes. 3. Pulsa la tecla enviar para registrar el equipo.
Pos-condición(es):	Se almacenara el equipo en la base de datos
Sub-flujos:	Si el usuario deja algún campo vacío no se envía a la base de datos. Si algún código de estudiante no está registrado tampoco se almacenará.
Excepciones:	Ninguno

Fuente: Autores

Tabla 17. Caso de uso GESTAR: Subir Tareas

Nombre:	Subir Tareas
Actor(es):	Estudiante
Descripción:	El estudiante sube las tareas al sistema.
Precondición(es):	Se necesita validación correcta del usuario. El estudiante debe pertenecer a un equipo.
Flujo Principal:	<ol style="list-style-type: none"> 1. El usuario inicia sesión 2. El estudiante selecciona el tipo de tarea, ingresa el nombre de la tarea y el link de la tarea. 3. Pulsa la tecla enviar para guardar la tarea.
Pos-condición(es):	Se almacenarán los campos en la base de datos
Sub-flujos:	Si el usuario no digita en los campos estos no se enviarán.
Excepciones:	Ninguno

Fuente: Autores

REQUISITOS NO FUNCIONALES

A continuación se listan los requisitos no funcionales del sistema.

REQUISITOS DE LA INTERFAZ DE USUARIO

- **Fuente de entrada de datos:** A través de la aplicación web se puede acceder a las pestañas por medio del mouse y el teclado.
- **Destino de salida de datos:** El destino principal de salida de datos será la pantalla del computador.
- **Elementos de la interfaz grafica:** La aplicación web estará compuesta por elementos visuales como áreas de texto, imágenes, botones, iconos, tablas, etc.
- **Consistencia en la presentación:** La información y todos los elementos que componen el sistema deben estar siempre en la misma ubicación y guardando al mismo tiempo.
- **Consistencia en el comportamiento:** No deben existir inconsistencias para el usuario, la aplicación debe tener el mismo comportamiento siempre.
- **Comprensión del texto:** Los textos se deben presentar en lenguaje sencillo, no técnico, con el fin de que cualquier persona pueda hacer uso de la aplicación.
- **Efectos inmediatos:** Las acciones ejecutadas por el usuario se deben realizar en el menor tiempo posible.
- **Operaciones reversibles:** Las operaciones que el usuario realice en el sistema serán reversibles (modificar), con excepción de eliminación o edición de notas.
- **Validación de datos:** La aplicación validará las entradas de datos para evitar datos no deseados.

- **Alertas:** En la aplicación web existen alertas que advierten al usuario en caso de haber excepciones en envío o actualización de datos, todo esto con el fin de mantener al usuario informado de las acciones que realiza.
- **Perfil de usuario:** En la página de inicio se visualizará el nombre del usuario que ha iniciado sesión.

REQUISITOS DE HARDWARE

Para usar la aplicación web GESTAR no se requiere un computador con grandes capacidades de procesamiento y memoria, sólo se necesita una conexión a internet ya sea por banda ancha o acceso conmutado y contar con un explorador de internet.

REQUISITOS DE SOFTWARE

- Explorador de internet: Mozilla Firefox, Internet Explorer, Google Chrome, Opera o Safari.
- Cualquier sistema operativo (Linux, Windows, Mac OS, etc.)

REQUISITOS DE DISEÑO Y DESARROLLO

- **Lenguaje de diseño:** Los diferentes diagramas que especifiquen el diseño de la aplicación deben ser realizados con UML (Unified Modeling Language).
- **Herramientas libres:** La aplicación web será desarrollada con herramientas libres.
- **Documentación:** Se debe hacer la documentación de las funcionalidades del sistema con el fin de tener un soporte para futuros desarrollos sobre la misma aplicación y facilitar el mantenimiento de la misma.