

**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



Documentación Proyecto Final

ASIGNATURA: COMPARACIÓN DE LENGUAJES , INF-328

DOCENTE: LIC. SILVA CHOQUE MOISES MARTIN

TEMA: REPOSITORIO DE DOCUMENTOS

INTEGRANTES:

Univ.: Aguilar Lozano Erick

C.I.: 6893912 LP.

Univ.: Quino Jimenez Victor Daniel

C.I.: 6944974 LP

ÍNDICE

1. PLANTEAMIENTO DEL PROBLEMA.....	
2. REQUISITOS.....	
3. OBJETIVO GENERAL.....	
4. OBJETIVOS ESPECÍFICOS.....	
5. MÓDULOS.....	
6. REQUERIMIENTOS FUNCIONALES.....	
7. METODOLOGÍA EMPLEADA.....	
8. HERRAMIENTAS.....	
9. CASOS DE USO.....	
10. DIAGRAMA DE SECUENCIAS.....	
11. DIAGRAMA DE CLASES.....	
12. MODELO RELACIONAL.....	
13. CONCLUSIONES.....	
14. RECOMENDACIONES.....	

1. PLANTEAMIENTO DEL PROBLEMA.

En la actualidad existen buscadores especializados que permiten localizar documentos de carácter académico como artículos, tesis, libros, patentes, materiales de congresos y resúmenes de fuentes diversas como editoriales universitarias, asociaciones profesionales, repositorios de preprints, universidades y otras organizaciones académicas. Los resultados aparecen ordenados considerando el texto completo, el número de citas recibidas, el autor, la publicación fuente, etc. Por lo que en la carrera de INFORMÁTICA se considera necesario la elaboración de un websites que haga la tarea de buscador de contenidos de carácter académico.

2. REQUISITOS.

- R1. Se requiere de un websites de acceso al público en general, donde el usuario pueda buscar artículos, tesis, libros y otros.
- R2. Se requiere de un websites de acceso a usuarios registrados que tendrán el privilegio de ver y descargar artículos, tesis, libros y otros contenidos completos.
- R3. Se requiere de un websites de acceso al público en general, donde el usuario pueda registrarse para así poder acceder a información completa con la opción de descargar contenidos (artículos, tesis, libros y otros).
- R4. Se requiere de un websites donde usuarios de tipo autor con cierto nivel de privilegio puedan publicar y subir contenido (artículos, tesis, libros, etc.) al repositorio de documentos, previa consideración de normativas y reglamentos para la publicación de los mismos.
- R5. Se requiere de un websites donde un usuario administrador, registre usuarios del tipo autor, que tendrán el privilegio de publicar y subir contenido al repositorio de documentos; además este usuario administrador debe controlar y hacer seguimiento a las publicaciones considerando reglamentos y normativas.
- R6. Se requiere de un websites donde un usuario administrador, registre, documente y publique los reglamentos y normativas que deben cumplirse para la publicación de contenido (artículos, tesis, libros, etc.) en el websites.
- R7. Se requiere de un websites donde un usuario de tipo ADMINISTRADOR, controle y valide el contenido subido al servidor por los usuarios de tipo AUTOR, antes de que este contenido sea publicado en el websites para el público en general.

3. OBJETIVO GENERAL.

Elaborar un websites que sirva como REPOSITORIO DE DOCUMENTOS.

4. OBJETIVOS ESPECÍFICOS.

- (SOLUCIÓN R1).- El sistema tendrá una interfaz para el público en general, donde el usuario podrá hacer búsquedas y el sistema responderá a la misma con información resumida de artículos, tesis, libros y otros referentes al tema en sí.

- (SOLUCIÓN R2).- El sistema tendrá una interfaz para usuarios Nivel 1; donde el mismo tendrá acceso completo con la opción de descargar artículos, tesis, libros y otros.
- (SOLUCIÓN R3).- El sistema tendrá una interfaz en el inicio donde un usuario podrá registrarse mediante el llenado de un formulario, para así acceder a información completa del repositorio de documentos, con manejo de sesiones.
- (SOLUCIÓN R4).- El sistema tendrá una interfaz para los usuarios de tipo AUTORES, donde el usuario podrá subir contenido al servidor donde se aloja el websites mediante el llenado de un formulario, el cual además de contener datos necesarios del contenido (artículos, tesis, libros, etc.), debe hacer conocer al usuario de tipo AUTOR las normativas y reglamentos que deben ser cumplidas para la publicación de contenidos.
- (SOLUCIÓN R5).- El sistema tendrá una interfaz para los usuarios de tipo ADMINISTRADOR, donde el mismo tendrá las opciones de CREAR, ACTUALIZAR, ELIMINAR y VER a usuarios de tipo AUTOR y DOCUMENTOS.
- (SOLUCIÓN R6).- El sistema tendrá una interfaz para los usuarios de tipo ADMINISTRADOR, donde; haciendo uso de un formulario el usuario ADMINISTRADOR podrá hacer el CRUD con las normativas y reglamentos que deben ser consideradas por los usuarios de tipo AUTOR en el momento de subir contenido al servidor.
- (SOLUCIÓN R7).- El sistema tendrá una interfaz para los usuarios de tipo ADMINISTRADOR, donde; este usuario validará y verificará que el contenido subido al servidor por los usuarios de tipo AUTOR, cumplan con los reglamentos y normativas vigentes.

5. MÓDULOS:

- M1. Requisito 1 y Solución R1
- M2. Requisito 2 y Solución R2
- M3. Requisito 3 y Solución R3
- M4. Requisito 4 y Solución R4
- M5. Requisito 5 y Solución R5
- M6. Requisito 6 y Solución R6
- M7. Requisito 7 y Solución R7

6. REQUERIMIENTOS FUNCIONALES.

- El sistema mostrará en su página principal la últimas publicaciones de usuarios registrados.
- Se permitirá el registro de usuarios o autores nuevos que quieran compartir su contenido (artículos, tesis, libros, etc.). Antes de poder aprobarse el registro de los datos, los campos deben estar completos.
- El sistemas permitirá realizar una búsqueda de contenido en cualquier nivel de usuario.
- El sistema permitirá subir, eliminar, modificar todos los archivos manejados por un usuario siempre y cuando este tenga un registro.

- A cada usuario registrado se le asignará un identificador único, que será utilizado para identificarlo en todos los procesos al que este tenga acceso.
- Todos los campos de para el registro de usuarios nuevos están validados, los cuales solo aceptaran caracteres específicos relacionados con el campo.
- Cualquier intercambio de datos vía internet que realice el software se realizará por medio del protocolo encriptado https.
- La aplicación debe poder utilizarse sin necesidad de instalar ningún software adicional además de un navegador web.
- La aplicación debe poder utilizarse con los navegadores web Chrome, Firefox e Internet Explorer.

7. METODOLOGÍA EMPLEADA.

En el presente proyecto se utilizó la metodología Scrum, ya que esta es un marco de trabajo para la gestión y desarrollo de software basada en un proceso iterativo e incremental. Se orienta a la auto-organización de equipos y colaboración entre sus miembros para entregar un incremento del producto funcional con un grado de prioridad cada cierto periodo de tiempo y bajo unos roles que tienen que asumir las personas que conforman el equipo desarrollo.

En la siguiente tabla se tiene una lista de las personas implicadas e identificadas con sus roles Scrum para el desarrollo de proyecto:

ROL		NOMBRE
Propietario del producto		Univ. Aguilar Lozano Erick Univ. Quino Jimenez Victor Daniel
Scrum master		Univ. Aguilar Lozano Erick Univ. Quino Jimenez Victor Daniel
Equipo Scrum	Analista	Univ. Aguilar Lozano Erick Univ. Quino Jimenez Victor Daniel
	Diseñador	
	Desarrollador	
	Testeador	

8. HERRAMIENTAS

Para la implementación de este proyecto se hará uso de las siguientes herramientas:

- **Framework CodeIgniter v3.1:** Es un marco de trabajo para el lenguaje de programación del lado del servidor PHP, cuenta con un conjunto de herramientas para construir aplicaciones web de una forma más simplificada y organizada. Usa el patrón de desarrollo MVC (Modelo Vista Controlador), separando la lógica de la presentación.
- **Spring Boot v2.3:** Es una solución para crear aplicaciones basadas en Spring de una manera rápida, autónoma y con características deseables para producción. Entre sus características principales se tiene el uso de contenedores Java embebidos (Tomcat o Jetty), soporte para la automatización con Maven y Gradle y configuración sugerida para iniciar rápidamente con un proyecto (Starters) entre otras.
- **Base de Datos PostgreSQL (PgAdmin 4):** Para almacenar la información se usará PostgreSQL, un sistema gestor de base de datos relacional, multi plataforma, potente y fiable, siendo uno de los mejores y más utilizados motores de BD en la actualidad.
- **Framework Bootstrap:** Es un framework o conjunto de herramientas de software libre para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en los últimos estándares a la fecha de HTML5 y CSS3, así como, extensiones de JavaScript opcionales adicionales. Además que tiene un diseño responsivo donde sus componentes se adaptan al tamaño de la pantalla de cualquier dispositivo.
- **Visual Studio Code:** Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows , Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.
- **Atom:** Atom es un editor de código fuente de código abierto para macOS, Linux, y Windows, con soporte para múltiples plug-in escritos en Node.js y control de versiones Git integrado, desarrollado por GitHub. Atom es una aplicación de escritorio construida utilizando tecnologías web.
- **GitHub:** Es una forja (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador. El software que opera GitHub fue escrito en Ruby on Rails. Desde enero de 2010, GitHub opera bajo el nombre de GitHub, Inc. Anteriormente era conocida como Logical Awesome LLC. El código de los proyectos alojados en GitHub se almacena típicamente de forma pública.

9. CASOS DE USO

Se propone los siguientes diagramas de casos de uso.
En la Figura 9.1 se puede ver el diagrama de Caso de Uso de Alto Nivel del sistema a desarrollar:

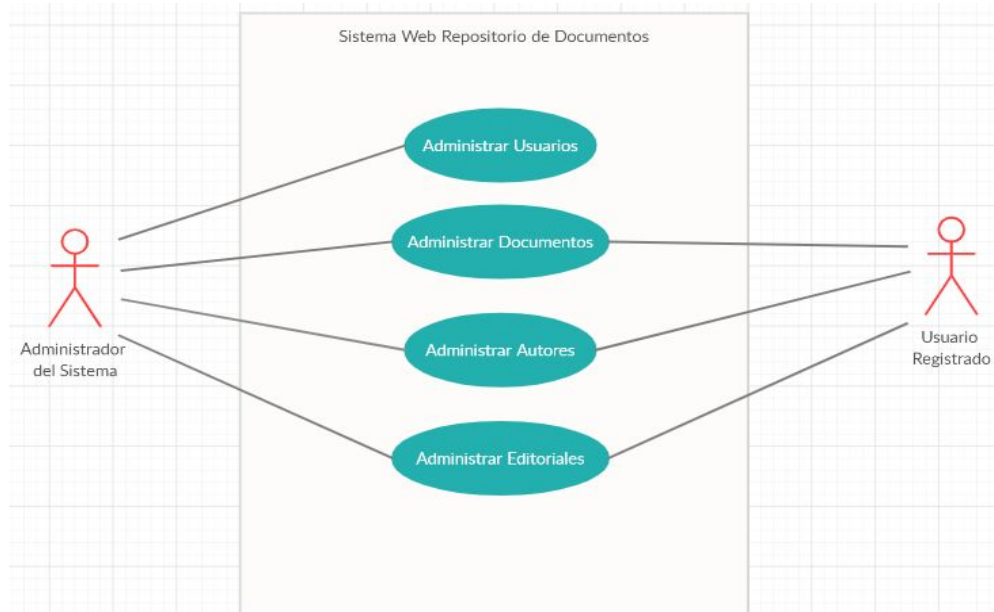


Figura 9.1

Se muestra el caso de uso para la Administración de Usuarios en la Figura 9.2 y en la Tabla 9.2 sus especificaciones:

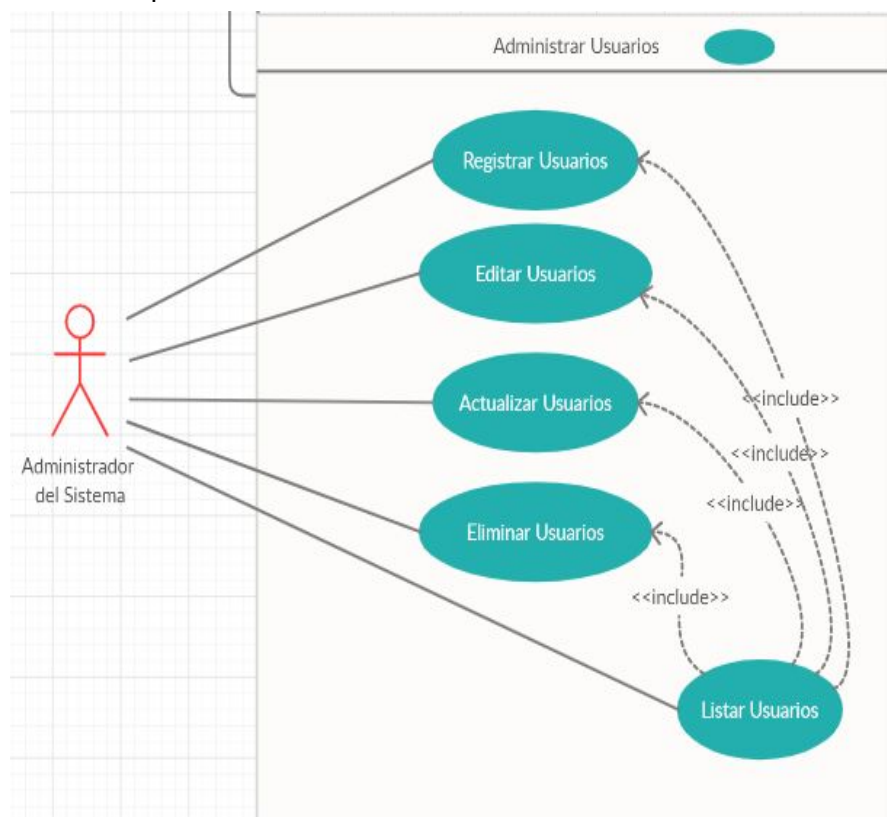


Figura 9.2

Caso de Uso: Administrar Usuarios
Actores: Administrador del Sistema
Descripción: Permite crear, editar, actualizar o eliminar usuarios, estos usuarios servirán para adicionar contenido al sistema.
Precondiciones: El actor debe ser un usuario registrado y tener el rol de Administrador del Sistema
Flujo Normal: 1. El actor ingresa a la opción usuarios. 2. El sistema despliega un listado de los usuarios registrados, junto con las opciones de crear, editar y eliminar. 3. El actor acciona una de las opciones.
Postcondiciones: El sistema ejecuta la opción elegida y muestra los cambios realizados.

Tabla 9.2

A continuación se muestra en la Figura 9.3 el diagrama de caso de uso para la Administración de Documentos junto con sus especificaciones en la Tabla 9.3:

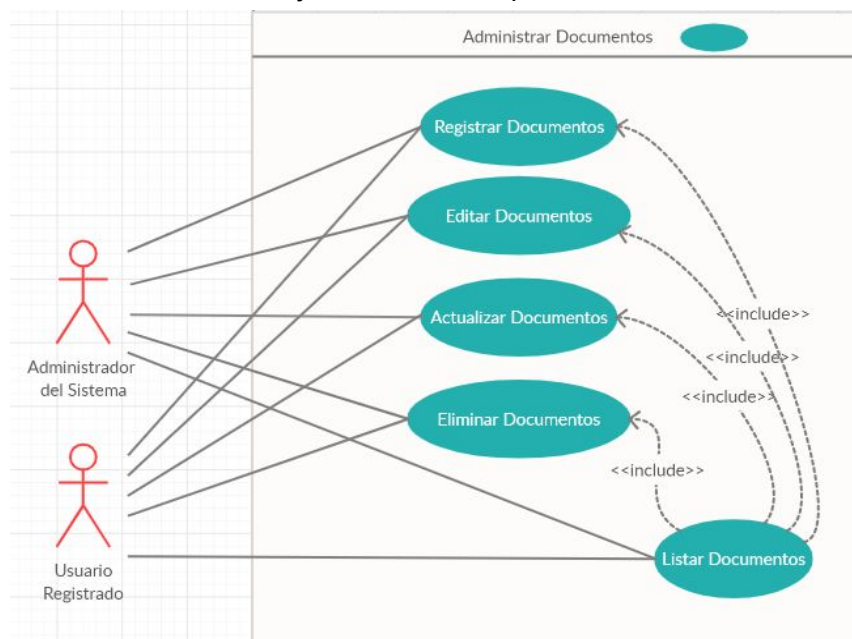


Figura 9.3

Caso de Uso: Administrar Documentos
Actores: Administrador del Sistema, Usuario Registrado
Descripción: Permite crear, editar, actualizar o eliminar documentos, estos pueden pertenecer al actor o a otros usuarios para el caso del administrador, para el usuario registrado se restringe a solo sus documentos.

Precondiciones: El actor debe ser un usuario registrado y tener el rol de Administrador del Sistema o Usuario Registrado, además de tener documentos registrados.

Flujo Normal: 1. El actor ingresa a la opción documentos.
2. El sistema despliega un listado de los documentos registrados, junto con las opciones de crear, editar y eliminar.
3. El actor acciona una de las opciones.

Postcondiciones: El sistema ejecuta la opción elegida y muestra los cambios realizados.

Tabla 9.3

En la Figura 9.4 se puede apreciar el diagrama de caso de uso para la Administración de autores y en la Tabla 9.4 están sus especificaciones:

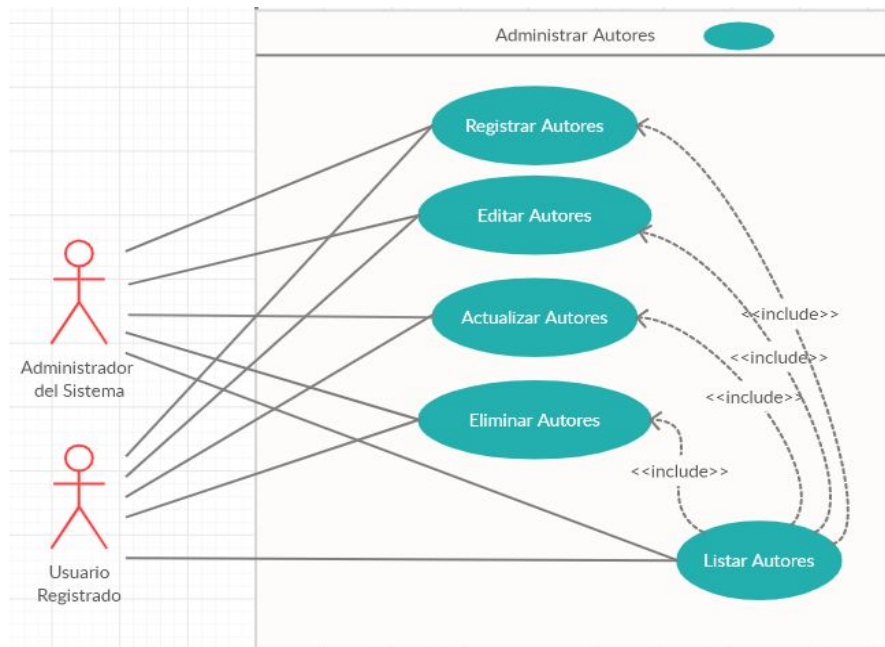


Figura 9.4

Caso de Uso: Administrar Autores

Actores: Administrador del Sistema, Usuario Registrado

Descripción: Permite crear, editar, actualizar o eliminar autores, estos pueden pertenecer al actor o a otros usuarios en el caso del administrador, para el usuario registrado se restringe a sólo sus autores.

Precondiciones: El actor debe ser un usuario registrado y tener el rol de Administrador del Sistema o Usuario Registrado, además de tener autores registrados.

Flujo Normal: 1. El actor ingresa a la opción autores.

2. El sistema despliega un listado de los autores registrados, junto con las opciones de crear, editar y eliminar.
3. El actor acciona una de las opciones.

Postcondiciones: El sistema ejecuta la opción elegida y muestra los cambios realizados.

Tabla 9.4

En la Figura 9.5 se muestra el diagrama de caso de uso para la Administración de Editoriales junto a sus especificaciones en la Tabla 9.5:

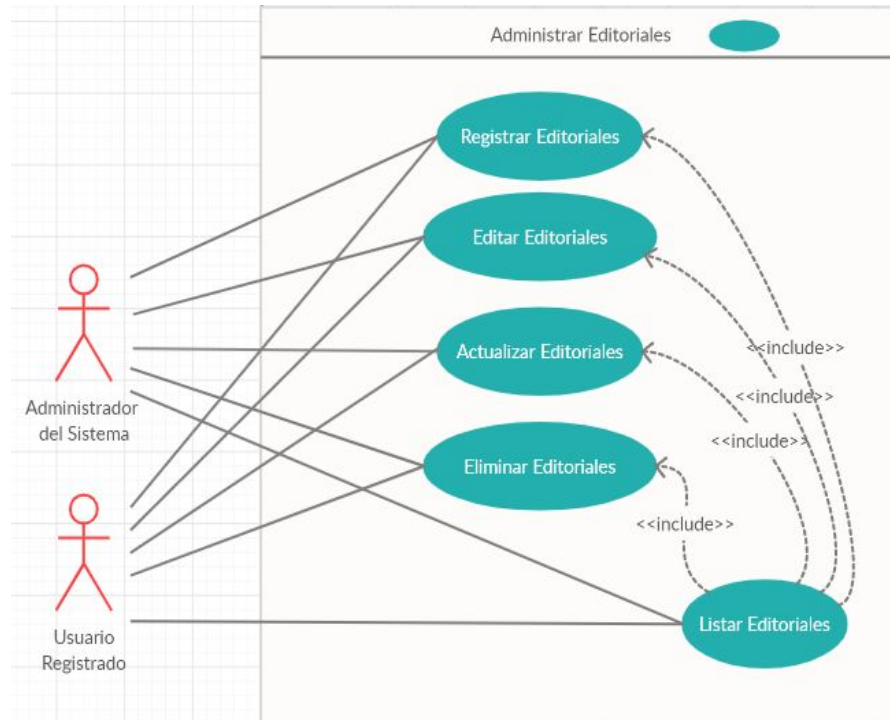


Figura 9.5

Caso de Uso: Administrar Editoriales

Actores: Administrador del Sistema, Usuario Registrado

Descripción: Permite crear, editar, actualizar o eliminar editoriales, estos pueden pertenecer al actor o a otros usuarios en el caso del administrador, para un usuario registrado se restringe solo a sus editoriales.

Precondiciones: El actor debe ser un usuario registrado y tener el rol de Administrador del Sistema o Usuario Registrado, además de tener editoriales registrados.

Flujo Normal: 1. El actor ingresa a la opción editoriales.
 2. El sistema despliega un listado de los editoriales registrados, junto con las opciones de crear, editar y eliminar.
 3. El actor acciona una de las opciones.

Postcondiciones: El sistema ejecuta la opción elegida y muestra los cambios realizados.

Tabla 9.5

10. DIAGRAMAS DE SECUENCIAS

Se tiene los siguientes diagramas de secuencias.

En la Figura 10.1 se puede ver el diagrama de secuencia publicar y subir documentos:

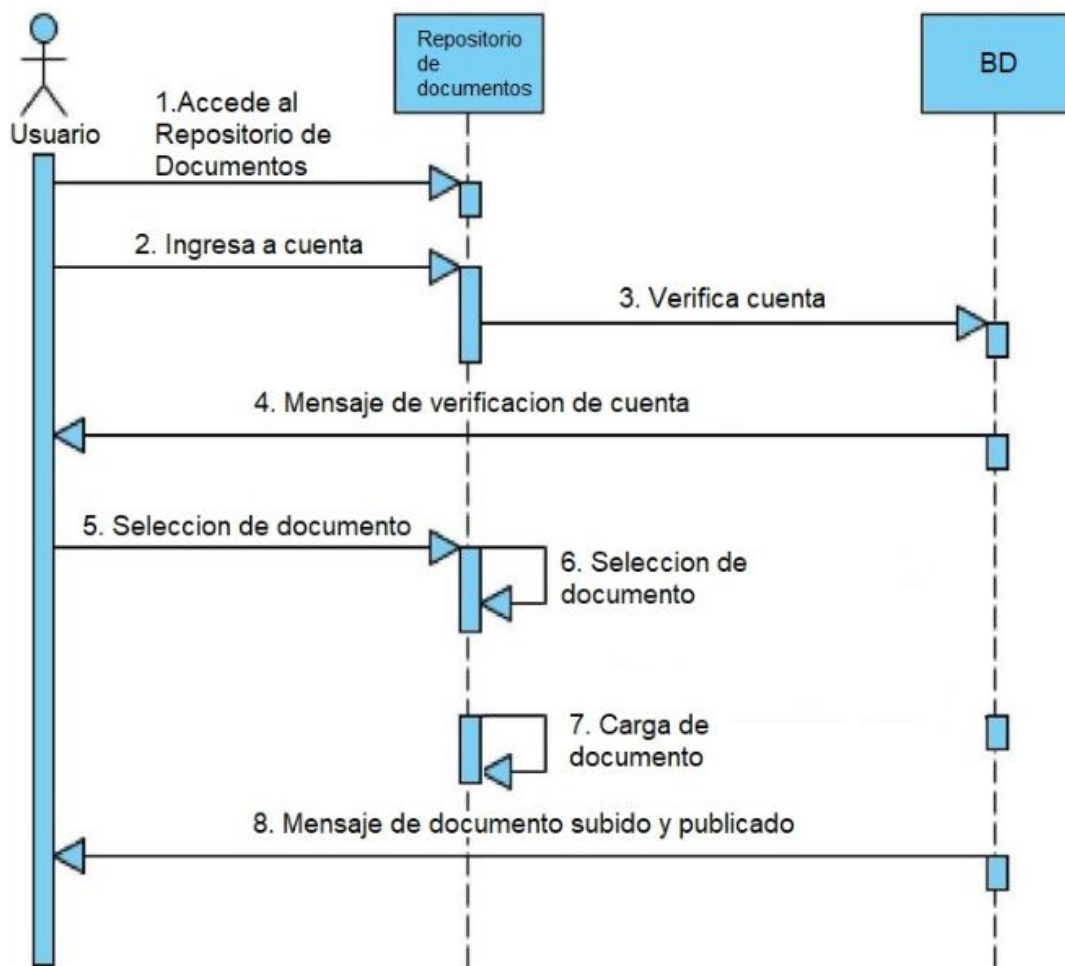


Figura 10.1

En la Figura 10.2 se puede ver el diagrama de secuencia de Creación de Cuenta:

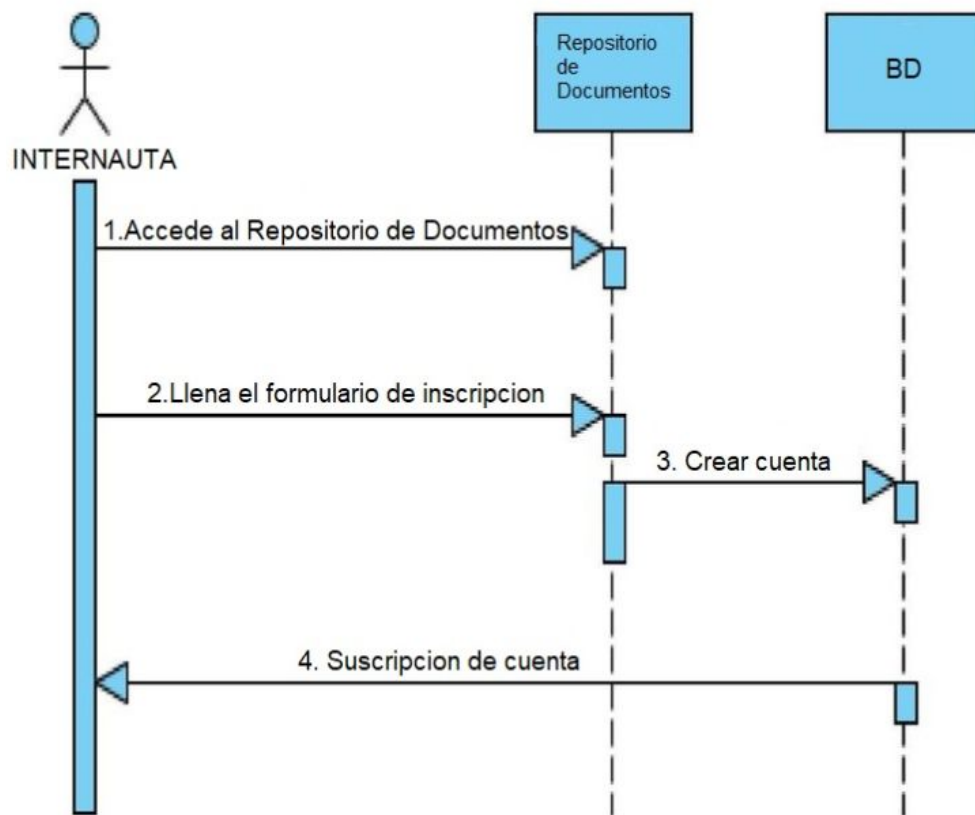
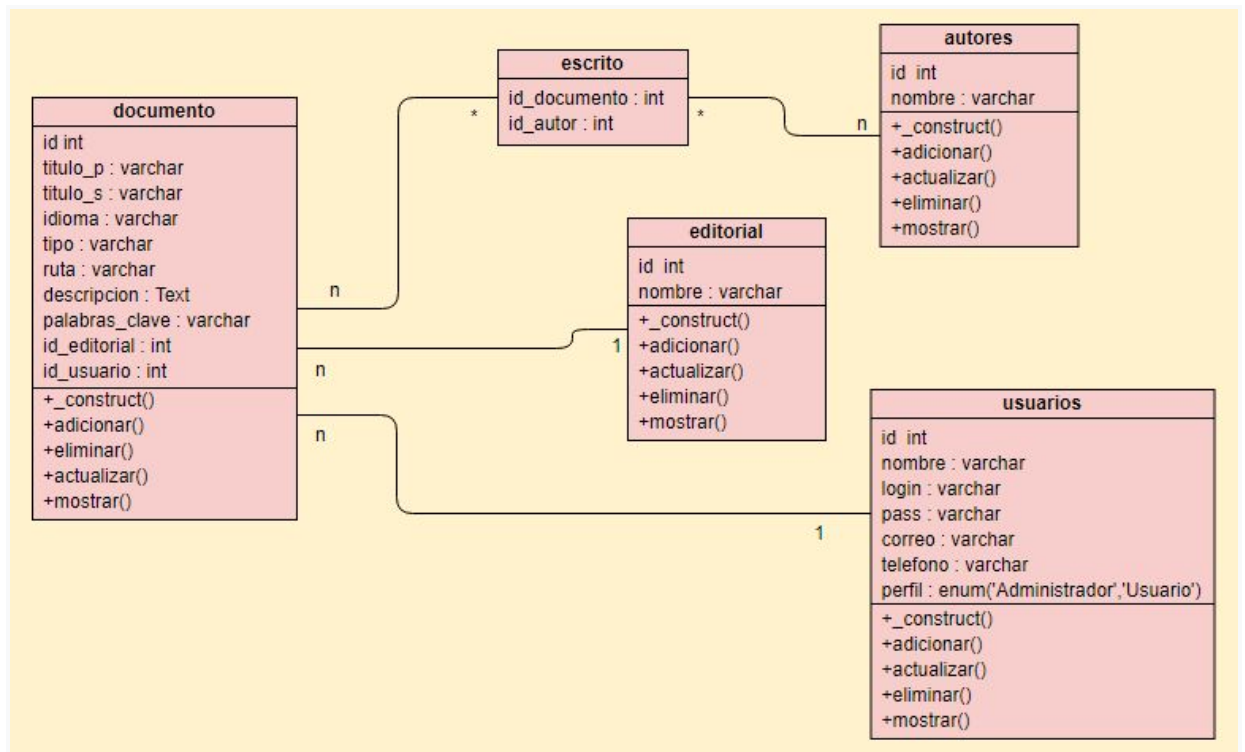


Figura 10.2

11. DIAGRAMA DE CLASES



12. MODELO RELACIONAL

En la Figura 12.1 se puede ver el modelo relacional de la base de datos del Sistema.

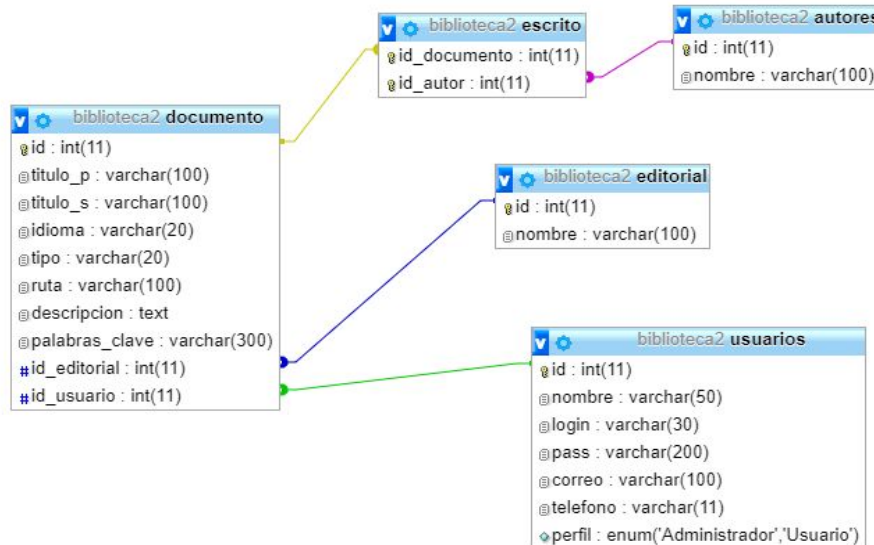


Figura 12.1

13. CONCLUSIONES

Luego de haber culminado con el desarrollo del proyecto implementado en dos lenguajes de programación, aplicando todas las metodologías de análisis y diseño de software en base a los problemas y objetivos planteados, se puede afirmar que se han cubierto las metas propuestas. Por lo que se concluye que:

- Se logró definir la funcionalidad principal del sistema, que es la de almacenar, mostrar y administrar documentos, lo que permite a los usuarios registrados tener el completo control de los documentos que estos subieron al sistema.
- Se logró implementar el control y restricción de accesos y acciones sobre la administración de los documentos de acuerdo a rol que cumple cada usuario en el sistema.
- Se logró implementar la funcionalidad para subir y eliminar documentos sin ninguna limitación.
- Se logró implementar un módulo de administración general donde los usuarios registrados pueden ser dados de alta o de baja.

14. RECOMENDACIONES

Las recomendaciones para la manipulación y buen funcionamiento del sistema son las siguientes:

- Realizar copias de seguridad de la base de datos periódicamente, por lo menos cada semana.
- Dar el mantenimiento del sistema de forma inmediata.
- Ver otras alternativas o tecnologías para mejorar las funcionalidades del sistema.