

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

**SISTEMA WEB ORIENTADO A LA RECOLECCIÓN DE INFORMACIÓN DE
MEDIOS DIGITALES MEDIANTE TÉCNICAS DE WEB SCRAPING**

CASO: AGETIC - HEMEROTECA

Para optar al título de Licenciatura en Ingeniería de Sistemas

Mención: INFORMÁTICA Y COMUNICACIONES

Postulante: *Univ. Roger Ivan Aroa Miranda*

Tutor Metodológico: *Ing. Enrique Flores Baltazar*

Tutor Especialista: *Lic. Freddy Salgueiro Trujillo*

Tutor Revisor: *Ing. Fanny Helen Perez Mamani*

**EL ALTO – BOLIVIA
2020**

DEDICATORIA:

El presente proyecto va dedicado en primer lugar a Dios que me acompaña y me guió en cada paso que doy. A mi amada esposa Carmen Rosa quien me ayudó a ser una mejor persona y esta siempre a mi lado apoyándome y dándome fuerzas para continuar, a mis padres Juan y Carmen que me recibieron como un hijo y hoy me brindan todo su apoyo.

AGRADECIMIENTO

Primero y antes que nada quiero agradecer a Dios por la oportunidad que me brinda para realizar este proyecto y aprender de él.

A mi esposa Carmen Rosa quien me brinda su apoyo incondicional y me inspira para seguir adelante.

A mis padres Juan y Carmen por todo el apoyo moral que me dan.

A la Agetic que es mi fuente de conocimientos, y me dieron la oportunidad de desarrollar este proyecto.

A mis tutores Ing. Fanny Helen Perez Mamani y Lic. Freddy Salgueiro Trujillo por toda la sabiduría y experiencia profesional, por sus consejos, colaboración y paciencia a lo largo del desarrollo de este proyecto de grado.

A mi tutor metodológico Ing. Enrique Flores por su tiempo y dedicación en el proceso del desarrollo de este proyecto.

A mis amigos Grosby Rodriguez, Luis Miguel Mamani y Maribel Callisaya quienes me acompañaron a lo largo de mi vida universitaria y me dieron siempre ánimos para seguir adelante.

RESUMEN

El presente proyecto surge con la idea de centralizar las noticias encontradas en medios de comunicación digitales (periódicos digitales) legalmente establecidos en nuestro país, con la implementación de este Sistema Web Orientado a la Recolección de Información de Medios Digitales Mediante Técnicas de Web Scraping podemos informar, almacenar y facilitar la búsqueda de noticias acontecidas en el día a día.

Para este fin se utilizó de forma general herramientas de software libre, entre las cuales podemos mencionar lenguajes de programación, base de datos, editor de texto y servidor web. Para el desarrollo del sistema se hace uso de la metodología UWE, así también para la evaluación de calidad de software se aplicó la ISO-9126, y finalmente se utilizó el modelo COCOMO para la estimación de costos del proyecto.

Palabras Clave: Periódico Digital, Web Scraping, UWE, Hemeroteca.

Abstract

The present project arises with the idea of centralizing the news found in digital media (digital newspapers) legally established in our country, with the implementation of this Web System Oriented to the Collection of Information from Digital Media Through Web Scraping Techniques we can inform , store and facilitate the search for news from day to day.

For this purpose, free software tools were used, among which we can mention programming languages, databases, text editor and web server. For the development of the system, the UWE methodology is used, as well as for the evaluation of software quality, ISO-9126 was applied, and finally the COCOMO model was used to estimate project costs.

Key Words: Digital Newspaper, Web Scraping, UWE, Hemeroteca.

ÍNDICE GENERAL

AGRADECIMIENTO.....	3
RESUMEN.....	4
CAPÍTULO I.....	15
1. MARCO PRELIMINAR.....	15
1.1 INTRODUCCIÓN.....	15
1.2 ANTECEDENTES.....	16
1.2.1 Antecedentes Académicos.....	16
1.3 PLANTEAMIENTO DEL PROBLEMA.....	20
1.3.1 Problema General.....	20
1.3.2 Problemas Específicos.....	21
1.3.3 Formulación del Problema.....	21
1.4 OBJETIVOS.....	22
1.4.1 Objetivo General.....	22
1.4.2 Objetivos Específicos.....	22
1.5 JUSTIFICACIÓN.....	23
1.5.1 Técnica.....	23
1.5.2 Económica.....	23
1.5.3 Social.....	23
1.6 METODOLOGÍA.....	24
1.6.1 UWE (UML-Based Web Engineering).....	24
1.6.2 Métricas de Calidad de Software.....	28
1.6.3 Métodos de estimación de costo de Software.....	29
1.7 HERRAMIENTAS.....	29
1.8 LÍMITES Y ALCANCES.....	33
1.8.1 Límites.....	33
1.8.2 Alcances.....	33
1.9 APORTES.....	34

CAPÍTULO II.....	37
2. MARCO TEÓRICO.....	37
2.1. DATO.....	37
2.2. SISTEMA.....	37
2.3 INFORMACIÓN.....	38
2.4. INTERNET.....	39
2.5. MEDIOS DE INFORMACIÓN.....	40
2.6 NOTICIAS.....	40
2.7 NOTICIAS FALSAS (FAKE NEWS).....	41
2.8 REDES SOCIALES.....	41
2.9 MEDIO DIGITAL.....	42
2.10 PERIÓDICO.....	43
2.11 PERIÓDICO DIGITAL.....	43
2.12 TÉCNICA.....	44
2.13 WEB SCRAPING.....	44
2.14 TÉCNICAS WEB SCRAPING.....	45
2.15 HEMEROTECA.....	46
2.15.1 ¿Que es una Hemeroteca?.....	47
2.15.2 Función.....	48
2.15.3 Tipos.....	48
2.15.4 Importancia de una Hemeroteca.....	49
2.15.5 Las ventajas que obtienes en una hemeroteca.....	51
2.16 HEMEROTECA DE MEDIOS.....	51
2.16.1 Que es una Hemeroteca de Medios.....	51
2.16.2 Tipos de hemerotecas de medios.....	52
2.17 INGENIERÍA DE REQUERIMIENTOS.....	53
2.17.1 Requerimientos Funcionales.....	54
2.17.2 Requerimientos No Funcionales.....	55
2.17.3 Características de un Requerimiento.....	57

2.17.4 Dificultades para definir un Requerimiento.....	57
2.18 INGENIERÍA DE SOFTWARE.....	58
2.18.1 Modelos de Desarrollo de Software UWE.....	59
2.18.1.1 Principales Aspectos.....	59
2.18.1.2 Fases del Desarrollo Web.....	60
2.18.1.3 Análisis de Casos de Uso.....	60
2.18.1.4 Representación UML del Modelado Conceptual.....	61
2.18.1.5 Elementos de Modelado.....	62
2.18.1.6 Diseño Navegacional.....	64
2.18.1.6.1 Estructura del modelo navegacional.....	65
2.18.1.6.2 Añadir menús.....	69
2.18.1.6.3 Elementos del modelados.....	70
2.18.1.7 Diseño de Presentación.....	73
2.18.1.7.1 Pasos para realizar el Storyboards.....	73
2.18.1.7.2 Elementos del Modelado.....	74
2.18.1.7.3 Modelado del Flujo de Presentación.....	75
2.18.2 Arquitectura del Software.....	77
2.18.2.1 Porque la Arquitectura de Software es Importante.....	78
2.18.2.2 Principios de la arquitectura del software.....	79
2.18.2.3 El Ciclo de Vida de la Arquitectura.....	81
2.18.2.4 Patrón Arquitectónico.....	83
2.18.3 Arquitectura Cliente – Servidor.....	84
2.18.3.1 Elementos que forman parte de una arquitectura Cliente Servidor	84
2.18.3.2 Esquema del Funcionamiento Cliente – Servidor.....	85
2.18.3.3 Componentes de la Arquitectura Cliente – Servidor.....	86
2.18.3.4 Elementos Principales.....	87
2.18.3.4.1 Cliente.....	87
2.18.3.4.2 Servidor.....	88

2.18.3.4.3 Middleware.....	89
2.18.4 Seguridad de Aplicaciones Web.....	90
2.18.4.1 Problemas Principales en la Programación de Sistemas Web....	91
2.18.4.2 Balancear Riesgo y Usabilidad.....	92
2.18.4.3 Rastrear el paso de los Datos.....	92
2.18.4.4 Filtrar Entradas.....	93
2.18.4.5 Escapado de Salidas.....	94
2.18.5 Calidad de Software.....	95
2.18.5.1 Pruebas de Software.....	95
2.18.5.2 Métodos de Pruebas de Software.....	96
2.18.5.2.1 Prueba de Especificación (Caja Negra).....	96
2.18.5.2.1.1 Limitaciones.....	97
2.18.5.2.2 Prueba de Código (Caja Blanca).....	97
2.18.5.2.2.1 Características de las pruebas de Caja Blanca.....	98
2.18.5.2.2.2 Tipos de pruebas de Caja Blanca.....	99
2.19 HERRAMIENTAS.....	100
2.19.1 Lenguaje de Programación Python.....	100
2.19.2 Flask.....	101
2.19.3 Base de Datos Relacional Potgresql.....	101
2.19.4 Celery.....	102
2.19.5 RabbitMQ.....	103
2.19.6 ElasticSearch.....	104
2.19.7 Lenguaje de Programación Javascript.....	105
2.19.8 VueJS.....	107
2.19.9 Bootstrap.....	108
2.19.10 Servidor Web Apache.....	109
2.20 MÉTRICAS DE CALIDAD.....	110
2.20.1 Que es una ISO.....	111
2.20.2 Métricas de Calidad establecido por el Estándar ISO 9126.....	111
2.20.2.1 Modelo de Calidad Interna y Externa.....	112

2.20.2.1 Modelo de Calidad de Uso.....	116
2.20.3 Punto de Función.....	118
2.21 MÉTODO DE ESTIMACIÓN DE COSTOS DE SOFTWARE.....	120
2.21.1 Modelo de Construcción de Costos (COCOMO).....	120
2.21.1.1 Modelos de Estimación.....	121
2.21.1.2 Modelos de COCOMO.....	122
2.21.1.2.1 Modelo Básico.....	122
2.21.1.2.2 Modelo Intermedio.....	123
2.21.1.2.3 Modelo Detallado.....	126
CAPÍTULO III.....	128
3. MARCO APLICATIVO.....	128
3.1 ESQUEMA DEL SISTEMA.....	128
3.2 ANÁLISIS DE REQUERIMIENTO.....	129
3.2.1 Análisis de Situación Actual.....	129
3.2.2 Ingeniería de Requerimientos.....	129
3.2.2.1 Requerimientos Funcionales.....	130
3.2.2.2 Requerimientos No Funcionales.....	132
3.2.3 Actores Identificados.....	132
3.3 MODELO DE CASOS DE USO.....	133
3.3.1 Diagrama de Caso de Uso General.....	133
3.3.2 Aplicación De La Metodología Uwe.....	134
3.3.3 Caso de Uso de Búsqueda de Noticias.....	134
3.3.4 Caso de Uso Registro y Publicación de Noticias.....	136
3.4 DISEÑO CONCEPTUAL.....	138
3.4.1 Modelo de Contenidos.....	138
3.5 DISEÑO NAVEGACIONAL.....	139
3.5.1 Modelo de Espacio de Navegación.....	139
3.5.2 Modelo de la Estructura de Navegación.....	140
3.6 DISEÑO DE PRESENTACIÓN.....	142

3.6.1 Modelo de Presentación.....	142
3.6.2 Modelo de Implementación.....	149
3.7 PRUEBAS DE SOFTWARE.....	162
3.7.1 Pruebas del Sistema.....	162
3.7.1.1 Pruebas de Caja Negra.....	163
3.7.1.2 Pruebas de Caja Blanca.....	165
CAPÍTULO IV.....	168
4 MÉTRICAS DE CALIDAD Y COSTO.....	168
4.1 MÉTRICAS DE CALIDAD.....	168
4.1.1 Funcionalidad.....	168
4.1.2 Usabilidad.....	171
4.1.3 Confiabilidad.....	172
4.1.4 Mantenibilidad.....	173
4.1.4.1 Mantenimiento Preventivo.....	174
4.1.5 Portabilidad.....	175
4.2 ANÁLISIS DE COSTO.....	176
4.2.1 COCOMO.....	176
CAPÍTULO V.....	182
5. CONCLUSIONES Y RECOMENDACIONES.....	182
5.1 CONCLUSIONES.....	182
5.2 RECOMENDACIONES.....	183
ANEXOS.....	193
ANEXO 1. ÁRBOL DE PROBLEMAS.....	193
ANEXO 2. ÁRBOL DE OBJETIVOS.....	194
ANEXO 3: MANUAL DE USUARIO ROL ADMINISTRADOR.....	196
ANEXOS. MANUAL DE USUARIO.....	206

ÍNDICE DE FIGURAS

<i>Figura 2.1:</i> Ciclo de Vida de la Ingeniería de Requerimientos.....	54
<i>Figura 2.2:</i> Ingeniería de Software.....	58
<i>Figura 2.3:</i> Casos de uso presentación de documentos.....	61
<i>Figura 2.4:</i> Clase con compartimiento adicional variants.....	62
<i>Figura 2.5:</i> Modelo Conceptual.....	63
<i>Figura 2.6:</i> Clase de Navegación.....	64
<i>Figura 2.7:</i> Clases de Navegación: Índices.....	68
<i>Figura 2.8:</i> Clases de Navegación: Vuelta Guiada.....	68
<i>Figura 2.9:</i> Clase con cadena de consulta como un atributo.....	69
<i>Figura 2.10:</i> Patrón para estructuras de acceso.....	70
<i>Figura 2.11:</i> Clase menú y taquigrafía de menú.....	72
<i>Figura 2.12:</i> Metamodelo para los elementos abstractos de la interfaz de usuario.....	75
<i>Figura 2.13:</i> Vista del flujo de presentación que puede hacer un usuario.....	77
<i>Figura 2.14:</i> Esquema de Modelo Cliente – Servidor.....	86
<i>Figura 2.15:</i> Entrada y salida del sistema.....	91
<i>Figura 2.16:</i> Login como medida de seguridad.....	92
<i>Figura 2.17:</i> Filtrado avanzado usando funciones PHP.....	94
<i>Figura 2.18:</i> Modelo de Calidad Interna y Externa.....	113
<i>Figura 2.19:</i> Modelo de Calidad de Uso.....	116
<i>Figura 2.20:</i> Cálculo de Punto de Función.....	119
<i>Figura 2.21:</i> Valores.....	120
<i>Figura 2.22:</i> Factores de Complejidad.....	120
<i>Figura 3.1:</i> Esquema del Sistema Hemeroteca.....	128
<i>Figura 3.2:</i> Diagrama de Caso de Uso.....	129
<i>Figura 3.3:</i> Diagrama de Caso de Uso General.....	133
<i>Figura 3.4:</i> Caso de Uso de Búsqueda de Noticias.....	134
<i>Figura 3.5:</i> Caso de Uso Registro y Publicación de Noticias.....	136

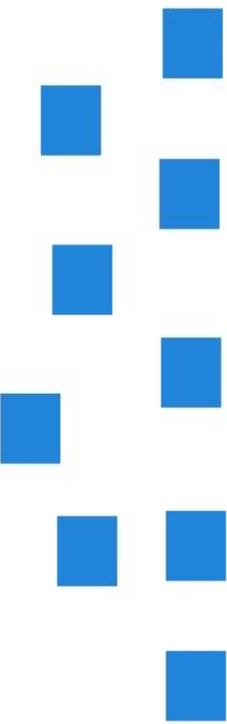
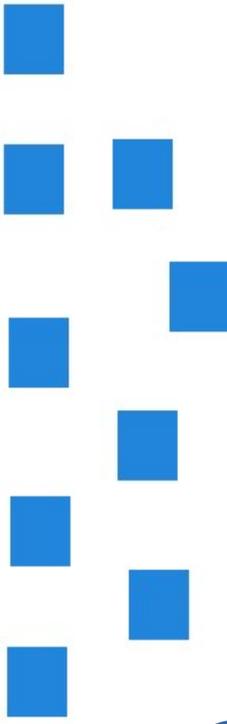
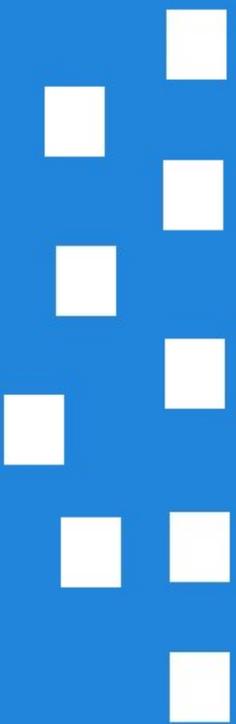
<i>Figura 3.6: Modelo de Contenido del Sistema.....</i>	<i>138</i>
<i>Figura 3.7: Modelo de Espacio de Navegación Directa (Usuario Común).....</i>	<i>139</i>
<i>Figura 3.8: Modelo de Estructura de Navegación.....</i>	<i>140</i>
<i>Figura 3.9: Modelo de Estructura de Navegación.....</i>	<i>141</i>
<i>Figura 3.10: Pantalla de Inicio de Sesión.....</i>	<i>142</i>
<i>Figura 3.11: Pantalla Usuario Ciudadanía Digital.....</i>	<i>143</i>
<i>Figura 3.12: Pantalla Contraseña Ciudadanía Digital.....</i>	<i>143</i>
<i>Figura 3.13: Pantalla Ingreso de Código Ciudadanía Digital.....</i>	<i>144</i>
<i>Figura 3.14: Pantalla Principal Administrador.....</i>	<i>145</i>
<i>Figura 3.15: Pantalla de Administración de Usuarios.....</i>	<i>146</i>
<i>Figura 3.16: Pantalla de Creación de Sitio Web.....</i>	<i>147</i>
<i>Figura 3.17: Pantalla de Noticias.....</i>	<i>148</i>
<i>Figura 3.18: Pantalla Principal de Autenticación "Administrador".....</i>	<i>149</i>
<i>Figura 3.19: Código para obtener ruta de Ciudadanía Digital.....</i>	<i>149</i>
<i>Figura 3.20: Autenticación Ciudadanía Digital: Usuario.....</i>	<i>150</i>
<i>Figura 3.21: Autenticación Ciudadanía Digital: Contraseña.....</i>	<i>151</i>
<i>Figura 3.22: Autenticación Ciudadanía Digital: Código.....</i>	<i>152</i>
<i>Figura 3.23: Código Verificar Usuario Registrado y Estado Activo.....</i>	<i>152</i>
<i>Figura 3.24: Pantalla Principal Administrador.....</i>	<i>153</i>
<i>Figura 3.25: Pantalla Crear y Editar Sitio Web.....</i>	<i>154</i>
<i>Figura 3.26: Código: Registro de Sitio Web.....</i>	<i>154</i>
<i>Figura 3.27: Página de Noticias "Los Tiempos".....</i>	<i>155</i>
<i>Figura 3.28: Xpath que recupera el Título de la Noticia.....</i>	<i>156</i>
<i>Figura 3.29: Ruta Xpath que Obtiene el Título de la Noticia.....</i>	<i>157</i>
<i>Figura 3.30: Pantalla Menú Principal Administrador.....</i>	<i>157</i>
<i>Figura 3.31: Pantalla Administrar Usuarios.....</i>	<i>158</i>
<i>Figura 3.32: Pantalla Crear, Editar Usuario.....</i>	<i>159</i>
<i>Figura 3.33: Registro de Usuario.....</i>	<i>159</i>
<i>Figura 3.34: Pantalla Principal Noticias Encontradas.....</i>	<i>160</i>

<i>Figura 3.35: Código: Búsqueda de Noticias</i>	161
<i>Figura 3.36: Pantalla Leer Noticia</i>	162
<i>Figura 4.37: Testing de Pruebas Unitarias</i>	165
<i>Figura 4.38: Testing de Pruebas de Integración</i>	166

ÍNDICE DE TABLAS

Tabla 1.1: Requerimientos de Hardware.....	31
Tabla 1.2: Software (BACKEND).....	31
<i>Tabla 1.3: Software (FRONTEND)</i>	32
Tabla 1.4: BASES DE DATOS.....	33
<i>Tabla 2.1: Características de la ISO-9126</i>	111
Tabla 2.2: Fórmulas ISO 9126.....	116
Tabla 2.3: Constantes COCOMO Básico.....	121
Tabla 2.4: Constantes COCOMO Intermedio.....	122
Tabla 2.5: Cuantificación del Proyecto.....	124
Tabla 3.1: Categoría de Funciones.....	129
Tabla 3.2: Requerimientos Funcionales.....	129
Tabla 3.3: Requerimientos No Funcionales.....	131
Tabla 3.4: Especificación del Caso de Uso: Búsqueda de Noticias.....	133
Tabla 3.5: Especificación de Caso de Uso: Registro y Publicación de Noticias... ..	135
Tabla 4.6: Prueba de Caja Negra Caso de Uso Búsqueda de Noticias.....	162
Tabla 4.7: Prueba de Caja Negra Caso de Uso Registro.....	163
Tabla 4.1: Factores de Ponderación.....	167
Tabla 4.2: Valores de ajuste de Complejidad.....	168
Tabla 4.3: Ajuste de preguntas.....	170
Tabla 4.4: Resultado Final de las Métricas De Calidad.....	175

Tabla 4.5: Coeficiente para medir el costo del software.....	176
Tabla 4.6: Calcula de los atributos FAE.....	176
Tabla 4.7: Resumen COCOMO.....	179



CAPÍTULO I

MARCO

PRELIMINAR

CAPÍTULO I

1. MARCO PRELIMINAR

1.1 INTRODUCCIÓN

El gran desarrollo tecnológico producido va evolucionando a pasos gigantescos y con esto a surgido la necesidad de manejar grandes cantidades de datos los cuales nos ayudan en todo ámbito y más aún en la toma de decisiones.

Los sistemas de información ayudan de gran manera en el manejo de la información y de esta manera se volvieron imprescindibles en todo ámbito, actualmente los sistemas de información se extendieron a todos los áreas.

Hoy en día y de forma cotidiana las grandes y pequeñas empresas manejan sistemas de información ya sean pequeños o grandes los cuales ayudan a obtener información, comunicarse, realizar negocios y otros.

La Agencia de Gobierno Electrónico y Tecnologías de Información y Comunicación (AGETIC) está acá para desarrollar tecnología, que permita modernizar el Estado, transformar la gestión pública y reducir la burocracia. Estas tareas son desarrolladas por bolivianas y bolivianos que trabajan investigando, innovando e implementando nuevas técnicas y tecnologías que permitan el desarrollo soberano de nuestra patria. Para esto, la AGETIC busca a los mejores profesionales, gente joven comprometida con su gente y el destino de su país.

Actualmente la AGETIC trabaja con muchos proyectos en diferentes ámbitos mayormente aplicados dentro de las entidades públicas, los cuales son

orientados al servicio e interacción con la ciudadanía, uno de estos proyectos es el de informar a la ciudadanía, ya que hoy en día no existe un centro de información, un lugar que pueda concentrar la información de los diferentes medios de comunicación en nuestro caso “EL PERIÓDICO DIGITAL¹”.

Por tal razón se ha tomado la decisión de realizar un sistema de información el cual pueda recolectar información de los diferentes periódicos digitales bolivianos, el cual se realizará en una plataforma de Software Libre cumpliendo con el decreto 1793².

Para la elaboración del proyecto de grado se utilizará la metodologías de desarrollo WEB que está conformado por las fases de diseño, desarrollo, implementación, implantación y pruebas los cuales nos permiten realizar un desarrollo confiable al utilizar herramientas como UWE que nos permite enfocarnos en el usuario final.

1.2 ANTECEDENTES

1.2.1 Antecedentes Académicos

Internacional

Según Hurtado & Hernández (2009) en su Sistema “Diseño e Implementación de la biblioteca virtual especializada sobre los derivados de la caña de azúcar” refieren lo siguiente:

-
- 1 Periódico Digital o Diario en Línea es la edición de un periódico que usa Internet como su principal medio de difusión. (Periódico en línea, párr. 1)
 - 2 Decreto 1793 en el cual el estado Plurinacional de Bolivia da prioridad al uso de Software Libre por parte de las entidades públicas y privadas dentro del Estado Plurinacional de Bolivia.

Se presentan los resultados del proyecto de diseño e implementación de la biblioteca virtual especializada sobre los derivados de la caña de azúcar. Los objetivos principales de este proyecto están orientados a la difusión del conocimiento científico como ayuda a las investigaciones y a posibilitar la integración entre las bibliotecas del sector azucarero.(Hurtado & Hernández, pág. 46)

Este sistema de biblioteca contiene una colección digital de artículos e informes de investigación, monografías y otros documentos a fin de posibilitar la investigación de usuarios de forma remota, se plantea una solución para poder tener toda la información con respecto a un tema en específico.

Avilés (2006) en su trabajo “Diseño y Planteamiento de un Estudio de Usuarios en un Hemeroteca Electrónica: El Servicio de Prensa Electrónica de la Biblioteca de la Facultad de Ciencias de la Información de la Universidad Complutense” realiza la siguiente descripción de su trabajo:

Describimos en este trabajo las principales características del estudio de usuarios de la hemeroteca electrónica de la Facultad de Ciencias de la Información, que se sustenta en la idea de llevar a cabo un análisis de las necesidades de información y del grado de utilización y satisfacción de este servicio, por parte de un segmento de la comunidad de usuarios de la Biblioteca Universitaria de la Facultad de Ciencias de la Información de la Universidad Complutense de Madrid.(Avilés, pág. 2)

En este proyecto se realiza un análisis a los usuarios y sus necesidades de información, demostrando de esta manera la importancia de una hemeroteca virtual en las Universidades.

Guallar (2012) en su tesis doctoral “Las hemerotecas de la prensa digital. Análisis de diarios españoles”, refiere lo siguiente:

Los sistemas de acceso y consulta a los contenidos retrospectivos de los diarios son conocidos indistintamente con las denominaciones “archivos de prensa” o “hemerotecas digitales” y han evolucionado desde meros depósitos de documentos sin sistema de búsqueda a disponer de una amplia variedad de formas de recuperar información, mayor cobertura temporal y diferentes posibilidades de visualizar los contenidos. (Guallar, pág. 11)

Guallar se enfoca en la memoria de un periódico, esta memoria es mas conocida como Hemeroteca que guarda la información publicada día a día en su caso de estudio diarios españoles.

Michael Newman (2009) en su trabajo “HighWire Press” refiere lo siguiente:

Con la reciente explosión en el campo de los libros electrónicos: diversidad de productos y modelos comerciales, aumento de la demanda del mercado y las opciones tecnológicas emergentes: HighWire Press se encontró en la posición de mirar hacia adelante y hacia atrás. Como otros en el industria, estamos mirando hacia el futuro tratando de entender y navegar esta evolución rápida ágil y responsable en nombre de nuestros clientes editores y sus propios constituyentes: lectores, investigadores y bibliotecarios. (Michael Newman, pág 2)

Highwire es una hemeroteca innovadora impulsada por la Universidad de Stanford, en la cual se hace énfasis en la digitalización de libros para sus usuarios.

Nacional

Jahaira (2007) en su trabajo “Hemeroteca virtual digital biblioteca central U.M.S.A.” plantea lo siguiente:

Desarrollar e implementar el sistema “Hemeroteca Virtual Digital”, que permita a los usuarios el acceso inmediato a la información de manera

eficiente desde cualquier punto mediante la Web y de esta manera mejorar los servicios prestados por la hemeroteca.

El fin de esta hemeroteca es poner a disposición de los usuarios finales el material digitalizado de libros, documentos, periódicos y otros que se tienen en la biblioteca de la Universidad.

ANTECEDENTES INSTITUCIONALES

La Agencia de Gobierno Electrónico y Tecnologías de Información y Comunicación (AGETIC) está acá para desarrollar tecnología, que permita modernizar el Estado, transformar la gestión pública y reducir la burocracia. Estas tareas son desarrolladas por bolivianas y bolivianos que trabajan investigando, innovando e implementando nuevas técnicas y tecnologías que permitan el desarrollo soberano de nuestra patria. Para esto, la AGETIC busca a los mejores profesionales, gente joven comprometida con su gente y el destino de su país.

Es una entidad pública bajo tuición del Ministerio de la Presidencia y fue creada en septiembre de 2015.

Liderar el proceso de desarrollo e implementación del Gobierno Electrónico y Tecnologías de la Información y Comunicación para la transformación de la gestión pública y aportar a la construcción de la Soberanía Científica y Tecnológica del Estado Plurinacional de Bolivia.

La AGETIC es la institución referente en desarrollo e implementación de Gobierno Electrónico y Tecnologías de la Información y Comunicación, generando conocimiento, mejorando el servicio público y facilitando la participación para el bienestar de la población.

Objetivos institucionales

Fortalecer institucionalmente a la AGETIC y posicionarla como líder en el desarrollo y aplicación de gobierno electrónico y tecnologías de la información y comunicación.

Aportar a la construcción de una gestión pública integrada, eficiente, transparente y participativa, orientada al servicio e interacción con la ciudadanía, a partir de la implementación de gobierno electrónico³ y el despliegue de infraestructura TIC.

Generar Políticas, herramientas y acciones referentes a seguridad de la información para proteger los datos e información de las instituciones del estado y la población.

Desarrollar políticas, programas, proyectos y acciones relacionadas a las tecnologías de información y comunicación para mejorar la calidad de vida de las ciudadanas y ciudadanos.

Impulsar la soberanía tecnológica a través de la generación de procesos de investigación, innovación y desarrollo para aportar a la producción de conocimiento, ciencia y tecnología.

Aportar a la construcción de una gestión pública integrada, eficiente, transparente y participativa, orientada al servicio e interacción con la ciudadanía, a partir de la implementación de gobierno electrónico y el despliegue de infraestructura TIC.

1.3 PLANTEAMIENTO DEL PROBLEMA

1.3.1 Problema General

Actualmente estamos viviendo una gran revolución de la información sin darnos cuenta y sin conocer totalmente la tecnología que nos rodea, todos

³ Gobierno Electrónico. Es una recolección sistematizada de experiencias en las cuales se utilizan tecnologías de información como apoyo de actividades Gubernamentales. (OEA, párr. 1)

conocemos las redes sociales, aplicaciones, páginas web, videos u otros que encontramos en internet y es precisamente esta red de redes llamada internet el medio por el cual tenemos acceso a la información, más en su búsqueda podemos encontrar gran cantidad de información que al final pueden llegar a ser tanto ciertas como inciertas, de tal forma que la población que busca informarse sobre el acontecer diario llega a confundirse más.

1.3.2 Problemas Específicos

- ▶ Las noticias falsas (en Inglés fake news) es la difusión de información a través de portales de noticias, prensa escrita, radio, televisión y redes sociales cuyo objetivo es la desinformación.
- ▶ No existen páginas en internet donde se encuentren las noticias del día a día recopilados de varios periódicos digitales los cuales informan a la población, específicamente de nuestro país.
- ▶ No se puede realizar una búsqueda de noticias acontecidas en una fecha en particular, ya que estas se van solapando una noticia con otra y al final solo se muestran las noticias más recientes.
- ▶ La publicidad dentro de una página web o detrás de una imagen o título de una noticia en internet muestra contenidos e información que confunden a la población.
- ▶ Se invierte mucho tiempo en ingresar a una o más páginas en búsqueda de noticias, información o sucesos relevantes que ocurren en el día a día.

1.3.3 Formulación del Problema

Por lo tanto, habiendo toma en cuenta las consideraciones y puntos de vista tratados, el problema identificado se enuncia:

¿Cómo lograr almacenar y publicar información confiable y del día a día con el desarrollo e implementación de un Sitio Web Orientado a la Recolección de Información de Medios Digitales Mediante Técnicas de Web Scraping?

1.4 OBJETIVOS

1.4.1 Objetivo General

Desarrollar un sistema web que pueda obtener de las noticias del día a día de fuentes digitales legalmente establecidas (periódicos digitales) y guardarlos en nuestro sistema, para luego poner la información a disponibilidad de la población.

1.4.2 Objetivos Específicos

- ▶ Recolectar noticias digitales de fuentes periodísticas legalmente establecidas en nuestro país.
- ▶ Recopilar noticias de varios periódicos y difundir la información en un único lugar.
- ▶ Mostrar las noticias mediante un buscador basado en fechas.
- ▶ Almacenar la información para un posterior análisis de datos.
- ▶ Aplicar pruebas de funcionamiento del sistema de información.
- ▶ Evaluar la calidad del sistema aplicando las métricas de calidad de software.

1.5 JUSTIFICACIÓN

1.5.1 Técnica

El desarrollo del sistema de información web para la recopilación de información será vía web, ya que la mejor forma de difundir la información hoy en día es mediante el uso de internet, los usuarios finales seremos todos aquellos que necesitamos informarnos día con día dentro del territorio boliviano y también fuera de él, el sistema será amigable para que pueda ser usado por personas que no necesariamente tengan conocimientos en el área de informática, simplemente bastará con que tenga conocimientos con respecto al uso del celular u ordenador a su alcance con un navegador y conexión a internet.

Con respecto al desarrollo del trabajo, se contempla el empleo de una metodología web como es UWE (Ingeniería Web Basado en UML) para el desarrollo del software.

1.5.2 Económica

El sistema estará disponible para todo aquel que tenga conexión a internet, sin tener ningún costo adicional tendrá acceso a la información centralizada de diferentes medios de información digitales.

Para evitar el pago de licencias Windows y otros y a su vez tomando en cuenta el Decreto 1793 se desarrollara el sistema en Tecnologías de Software Libre⁴, AGETIC cuenta con dominios oficiales bajo uno de los cuales estará alojado el sistema web.

1.5.3 Social

La implementación del Sistema Web Orientado a la Recopilación de Información de Medios Digitales Mediante Técnicas de Web Scraping de la AGETIC, permitirá recuperar y mostrar las noticias de diferentes periódicos

⁴ El Software Libre es todo software cuyo código fuente puede ser estudiado, modificado, y utilizado libremente con cualquier fin y redistribuido con cambios y/o mejoras o sin ellas. (Software Libre, párr. 1)

digitales en nuestro sistema.

El sistema beneficiará directamente a los empresarios, emprendedores, presentadores de noticias y población en general que necesite estar informado sobre el acontecer diario.

1.6 METODOLOGÍA

1.6.1 UWE (UML-Based Web Engineering)

Es una herramienta para modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en la sistematización y personalización (sistemas adaptativos).

UWE es una propuesta basada en UML y en el proceso unificado para modelar aplicaciones web. Esta propuesta está formada por una notación para especificar el dominio (basada en UML) y un modelo para llevar a cabo el desarrollo del proceso de modelado.

El principal objetivo del enfoque UWE es proporcionar: un lenguaje de modelado específico del dominio basado en UML; una metodología dirigida por modelos; herramientas de soporte para el diseño sistemático; y herramientas de soporte para la generación semi-automática de Aplicaciones Web.

La notación de UWE se define como una ligera extensión de UML, proporcionando un perfil UML para el dominio específico de la web.

Modelos de UWE

El método UWE consiste en la construcción de seis modelos de análisis y diseño. Dicha construcción se realiza dentro del marco de un proceso de diseño iterativo e incremental. Las actividades de modelado abarcan: el análisis de requerimientos, diseño conceptual, modelo de usuario, diseño de la navegación, de la presentación y diseño de la adaptación.

Los principales artefactos que produce el método de diseño de UWE son los siguientes:

- ▶ Un Modelo de Requerimientos que captura los requerimientos del sistema.
- ▶ Un Modelo Conceptual para el contenido (modelo de contenido).
- ▶ Un Modelo de Usuario.
- ▶ UN Modelo de Navegación que comprende la estructura de la navegación.
- ▶ Un Modelo de Presentación que abarca modelos estáticos y dinámicos (modelo de estructura de la presentación, modelo del flujo de la presentación, modelo de interface abstracta de usuario, y modelo de ciclo de vida del objeto).
- ▶ Un modelo de adaptación.

Modelo de Requerimientos

El primer paso para el desarrollo de un sistema web que se especificará con UWE, es realizar la identificación de los requerimientos y plasmarlos en un modelo de requerimientos.

Los requerimientos pueden ser documentados en diferentes niveles de detalle, para este caso, UWE propone dos niveles de granularidad. En primera instancia se deben describir detalladamente las funcionalidades del sistema, las cuales son modeladas con casos de uso UML. Como segundo paso, se debe elaborar una descripción de los casos de uso más detallada, por ejemplo, realizando diagramas de actividad UML donde se delimiten las responsabilidades y acciones de los actores involucrados.

Los casos de uso fueron propuestos por el Proceso de Desarrollo de Software Unificado (RUP) para capturar los requerimientos del sistema. Es una técnica

centrada en el usuario que obliga a definir quiénes son los usuarios (actores) de la aplicación y ofrece una forma intuitiva de representar la funcionalidad que una aplicación tiene que cumplir para cada actor.

UWE distingue tres tipos de casos de uso: navegación, proceso, y casos de uso personalizados.

Los casos de uso de navegación, se distinguen con el estereotipo <<navigation>> (0) y se utilizan para modelar el comportamiento típico del usuario cuando interactúa con una aplicación web, tal como navegar a través del contenido de la WebApp o buscar información por medio de palabras claves.

Los casos de uso de proceso, se utilizan para describir las tareas del negocio que los usuarios finales realizarán con el sistema, tales como acciones transaccionales sobre la Base de Datos. No se denota con ningún estereotipo específico, por lo tanto se utiliza en este caso la notación pura de UML.

Los casos de usos personalizados, implican la personalización de un sistema web, la cual es desencadenada por el comportamiento del usuario. Estos casos de uso se denotan con el estereotipo <<personalized>>

Modelo de Contenido (Conceptual) y Modelo de Usuario

El diseño conceptual se basa en el modelo de análisis e incluye los objetos involucrados en las actividades típicas que los usuarios realizan con la aplicación.

El propósito del modelo de contenido es proporcionar una especificación visual de la información relevante para el dominio del sistema web, que comprende principalmente el contenido de la aplicación Web.

Modelo de Navegación:

El modelo de estructura de navegación define la estructura de nodos y links de una WebApp mostrando cómo se puede realizar la navegación utilizando

elementos de acceso tales como índices, visitas guiadas, consultas y menús.

Los elementos de modelado son:

- ▶ **Clases de navegación**, que se denotan con (0), representan los nodos navegables de la estructura de hipertexto.
- ▶ **Links de navegación**, que muestran el vínculo directo entre las clases de navegación.
- ▶ **Caminos de navegación alternativos**, los cuales son visualizados con el estereotipo <<menu>> ().
- ▶ **Primitivas de acceso**, las cuales se utilizan ya sea para llegar a múltiples instancias de una clase de navegación(<<index>> o <<guided tour>>) o para seleccionar ítems (<<query>>).
- ▶ **Clases de procesos** (), las cuales modelan los puntos de entrada y de salida de los procesos de negocio. Cada clase de proceso está asociada a un caso de uso de proceso.
- ▶ **Links de procesos**, que representan el vínculo entre las clases de proceso y de navegación.

El modelo de estructura de navegación se representa mediante diagramas de clases UML estereotipados con las clases de navegación y procesos, menús y primitivas de acceso y así también los links de navegación y proceso.

Modelo de Presentación

El modelo de presentación proporciona una vista abstracta de la interfaz de usuario (UI) de la aplicación web. Se basa en el modelo de navegación y describe qué elementos (por ejemplo texto, elementos, links, formularios) se utilizarán para presentar los nodos de navegación.

Los elementos básicos del modelo de presentación son:

- ▶ **Clases de presentación**, las cuales se basan directamente en los nodos del modelo de navegación. Una clase de presentación () está compuesta por elementos de UI tales como, texto (<<text>>), vínculo (<<anchor>>), botón (<<button>>), imagen (<<image>>), formulario (<<form>>), y colección de vínculos (<<anchored collection>>)
- ▶ **Páginas web** (<<page>>), que se utilizan para modelar la información proveniente de varios nodos de navegación y que se presentan en una misma página web.
- ▶ **Grupo de presentación** (<<presentation group>>), el cual es un contenedor de clases de presentación, y a su vez de otros grupos de presentación

1.6.2 Métricas de Calidad de Software

Ardila, N. I. (2013) en el sitio web “Norma de Evaluación ISO/IEC 9126” refiere lo siguiente:

La métrica de calidad de software ISO/IEC 9126 para evaluar la calidad de software, este estándar identifica los siguientes atributos clave de calidad: Funcionalidad, Confiabilidad, Eficiencia, Facilidad de uso o mantenimiento y Portabilidad.

Puesto que la ISO 9126 se limita a la especificación de un modelo de calidad general, esta se debe aplicar en conjunto con la ISO 14598, según esta norma, los componentes fundamentales en la evaluación de la calidad del software son: Modelo de calidad, Método de evaluación, Medidas de software y Herramientas de soporte. (ISO 9126, párr. 2)

Motivo por el cual se utilizará la métrica de calidad de software ISO 9126 el cual nos ayudara a evaluar la calidad del sistema.

1.6.3 Métodos de estimación de costo de Software

Método de estimación de costos COCOMO para estimar el costo de software. El modelo Constructivo de Costos (COCOMO), por su acrónimo del inglés (CONstructive COst MOdel), es un modelo matemático de base empírica utilizada para estimación de costo de software (Pressman, 2002).

COCOMO II al ser un modelo basado en modelos matemáticos usa fórmulas y puede ser calculado a través de un software para tener el mínimo error. Además es orientado al modelo final no a fases intermedias.

1.7 HERRAMIENTAS

Las herramientas necesarias serán clasificadas en dos partes: hardware⁵ y software⁶, para el desarrollo de la aplicación no se necesitará un ordenador de última generación sino que con un ordenador básico será suficiente que por lo menos tenga un procesador Core i3 ya que actualmente en el mercado no existen a la venta ordenadores más básicos, en cuanto al software la AGETIC lidera el proceso de desarrollo e implementación de sistemas en tecnologías de software libre, siguiendo esta línea también el desarrollo será en dichas tecnologías, el entorno de desarrollo será Linux en su distribución Debian con estas referencias tenemos las siguientes tablas.

5 Hardware es el conjunto de componentes físicos de los que está hecho el equipo.

6 Software es el conjunto de programas o aplicaciones, instrucciones y reglas informáticas que hacen posible el funcionamiento del equipo. (¿Qué es hardware y software?, párr. 1)

HARDWARE

Tabla 1.1: Requerimientos de Hardware

Ordenador	Características
Requerimientos Básicos	Tarjeta Madre: Asrock Procesador: i3 de 2.4 GHz Memoria RAM: 4 GB HDD: 500 GB Sata Monitor: 19" Pulgadas Mouse Teclado

Fuente: (Elaboración Propia)

Fuente: MegaOfertas , Computer son tiendas de venta de computadoras en la Ciudad de El Alto.

Tabla 1.2: Software (BACKEND⁷)

Lenguaje de Programación	Ventajas	Desventajas
JSP	<ul style="list-style-type: none">- Multiplataforma- Permite el uso de servlets aprendizaje y por tal razón	<ul style="list-style-type: none">- Es de más complejo no es muy conocido en nuestro medio
PYTHON	<ul style="list-style-type: none">- Excelente integración con Servidores Linux.- Es más rápido y potente en aplicaciones Web- Código más comprensible	<ul style="list-style-type: none">- Es mas difícil su aprendizaje para la programación WEB por que no se cuenta con grandes cantidades de tutoriales y libros.
NODEJS	<ul style="list-style-type: none">- Orientado a Objetos- Separa el diseño del código.	<ul style="list-style-type: none">- Por ser Tecnología reciente no existe mucha documentación.

⁷ Backend es la capa de acceso a datos de un software o cualquier dispositivo, que no es directamente accesible por los usuarios, además contiene la lógica de la aplicación que maneja dichos datos.

PHP	-Es de fácil mantenimiento	
	- Buena seguridad	
	- Posee una comunidad muy grande.	- Todo el código es realizado en el servidor.
	- Conexión con diferentes manejadores de bases de datos	- Si no se configura bien se puede dejar muchas brechas de seguridad abiertas.

Fuente: (Elaboración Propia)

Tabla 1.3: Software (FRONTEND⁸)

FRAMEWORK	Ventajas	Desventajas
VUE JS	- Tiene una Documentación detallada.	- Riesgo de excesiva flexibilidad.
	- Excelente Integración para páginas web simples o complejas.	- No existe mucha documentación para nuestro idioma.
	- Puede ser escalado fácilmente.	
REACT JS	- Gran nivel de flexibilidad y máxima responsabilidad.	- Tiene una ausencia de documentación oficial.
	- Puede trabajar con alta carga.	- No existe un estándar de desarrollo.
ANGULAR JS	- Fácil de aprender.	
	- Tiene una documentación detallada .	- Sintaxis compleja.
	- Vínculo de datos en	- Presenta problemas en la migración entre versiones.

⁸ Frontend es la parte de un programa o dispositivo a la que un usuario puede acceder directamente. Son todas las tecnologías de diseño y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios.

ambos sentidos.
- Modularidad en general.

Fuente: (Elaboración Propia)

Tabla 1.4: BASES DE DATOS

Bases de Datos	Ventajas	Desventajas
Mysql	<ul style="list-style-type: none">- Es Multiplataforma- Conexión segura- Permite escoger entre múltiples motores de almacenamiento para cada tabla.- Es un gestor de base de datos relacional y orientado a objetos.	<ul style="list-style-type: none">- En Bolivia no se cuenta con soporte para Mysql.- Muchas de sus utilidades no presentan documentación.- Complejo aprendizaje.- Es lento en inserciones para bases de datos pequeñas.
PostgreSQL	<ul style="list-style-type: none">- Cuenta con soporte para Triggers y Procedimientos en el Servidor.- Puede trabajar con grandes cantidades de datos.	<ul style="list-style-type: none">- No cuenta con una línea de soporte oficial.
SQLite	<ul style="list-style-type: none">- Es Multiplataforma- Es Software Libre- Facilidad a la hora de usar Triggers y Procedimientos.- Ligero y rápido en respuestas en BD ligeras.	<ul style="list-style-type: none">- Cuando la Base de Datos va creciendo mucho el SGBD podría verse más lento que en otras herramientas de Bases de Datos.

Fuente: (Elaboración Propia)

De las tablas anteriores se optó por las siguientes herramientas:

- ▶ Python
- ▶ Vuejs
- ▶ Postgresql

Tales Herramientas nos proporcionan más ventajas al momento de desarrollar el sistema y a la vez cumplimos con el decreto 1793 para priorizar el Software Libre.

Se trabajará bajo la arquitectura Frontend y Backend (lado cliente y lado Servidor) se aprovechara esta estructura ya que separa la lógica del diseño.

1.8 LÍMITES Y ALCANCES

1.8.1 Límites

- ▶ El sistema recupera noticias en horas determinadas cada día lo cual hace que no tengamos las noticias en el mismo instante de su publicación.
- ▶ El sistema está diseñado para recopilar la información día con día, no podemos registrar un nuevo periódico digital y esperar que recupere las noticias hacia atrás.

1.8.2 Alcances

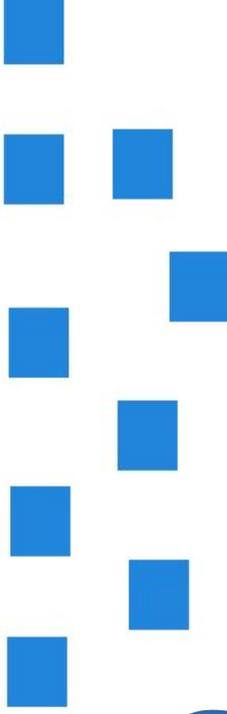
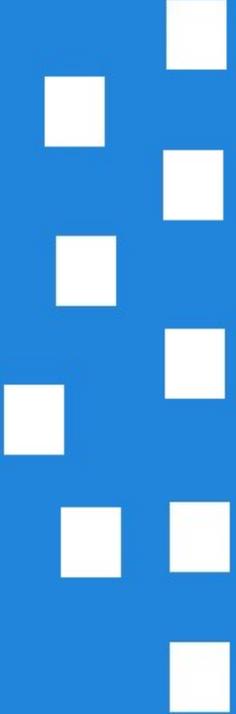
El presente proyecto de grado tiene el fin de crear un centro de recopilación de noticias obtenidas de los diferentes diarios digitales por lo que se pretende hacer una “**Hemeroteca**” donde se almacenara esta información y ponerla a disposición de la población en general.

Una vez implantado el sistema gozará de las siguientes características, con los siguientes módulos:

- ▶ Recuperar todos los links que se encuentren en el menú de la página principal.
- ▶ Descargar el contenido html de los links de páginas web encontrados.
- ▶ Extraer los datos encontrados códigos fuente de las páginas descargadas.
- ▶ Descargar las imágenes de las noticias.
- ▶ Crear un servicio que busque las noticias por periódico, fecha y contenido.
- ▶ Módulos:
- ▶ Administrador
- ▶ ABM (Altas bajas modificaciones) sitios web de periódicos digitales.
- ▶ Indexación de documentos (noticias).
- ▶ Sistema
- ▶ Recuperar de forma automática los enlaces.
- ▶ Recuperar el contenido de páginas web automáticamente.
- ▶ Extraer información automáticamente.
- ▶ Descargar las imágenes automáticamente.
- ▶ Mostrar la información recuperada.

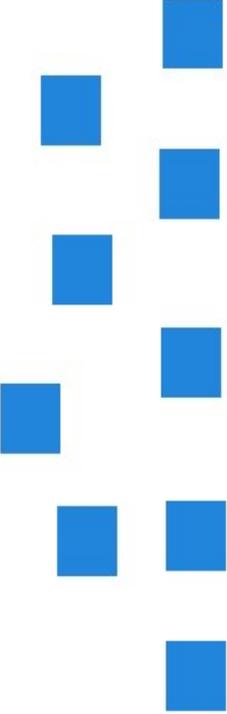
1.9 APORTES

El sistema beneficiará de gran manera a la población, el sistema no solo nos mostrará la información recuperada en el día también podrá mostrarnos las noticias de días anteriores, las noticias mostradas en el sistema serán veraces y ayudarán a no creer en falsas noticias encontradas comúnmente en redes sociales.



CAPÍTULO II

MARCO TEÓRICO



CAPÍTULO II

2. MARCO TEÓRICO

En el presente capítulo se realizará una descripción de un conjunto de los conceptos y definiciones importantes que sustentan teóricamente el presente proyecto, así también las distintas herramientas que se usarán para el lograr los objetivos propuestos en la misma.

2.1. DATO

Según Alegsa (2018) en su sitio web “Definición de Dato (informática)” refiere lo siguiente:

Un dato es una representación simbólica (numérica, alfabética, etc.) de un atributo de una entidad. Un dato no tiene valor semántico (sentido) en sí mismo, pero al ser procesado puede servir para realizar cálculos o tomar decisiones.(Definición de Dato, párr. 1)

Sobre Dato, Wikipedia (2008) en su página “Dato (informática)” refiere lo siguiente:

Los datos requieren una interpretación para convertirse en información. Para traducir datos a información, debe haberse considerado varios factores conocidos. Los factores involucrados están determinados por el creador de los datos y la información deseada.

El término metadatos se usa para hacer referencia a los datos sobre los datos. Los metadatos pueden estar implícitos, especificados o dados.(Dato. párr. 1)

2.2. SISTEMA

Significado (2020) en su pagina web “Significado de Sistema” refiere lo siguiente:

Un sistema es un conjunto de elementos relacionados entre sí que funciona

como un todo. Si bien cada uno de los elementos de un sistema puede funcionar de manera independiente, siempre formará parte de una estructura mayor. Del mismo modo, un sistema puede ser, a su vez, un componente de otro sistema.

La palabra sistema procede del latín *systema*, y este del griego σύστημα (*systema*), identificado en español como “unión de cosas de manera organizada”. De esta palabra se derivan otras como *antisistema* o *ecosistema*.

Cada uno de los componentes o elementos del sistema contribuye al logro de un objetivo o propósito. (Características del sistema. párr. 2)

2.3 INFORMACIÓN

Segun Wikipedia (2010) en su página web “Información” refiere lo siguiente:

La información es como se conoce a un conjunto organizado de datos procesados que constituyen un mensaje que cambia el estado de conocimiento del sujeto o sistema que recibe dicho mensaje.

En general la información tiene una estructura interna y puede ser calificada según varias características:

- ▶ **Significado (semántica):** Del significado extraído de una información, cada individuo (o sistema experto: sistema capaz de emular el pensamiento humano) evalúa las consecuencias posibles y adecua sus actitudes y acciones de manera acorde a esas consecuencias. La expectativas que tiene el ente se da de manera literal al significado de la información.
- ▶ **Importancia (relativa al receptor):** Es decir, si trata sobre alguna cuestión importante. La importancia de la información para un receptor se referirá a en qué grado cambia la actitud o la conducta de los individuos. En las modernas sociedades, los individuos obtienen de los

medios de comunicación masiva gran cantidad de información, una gran parte de la misma es poco importante para ellos, porque altera de manera muy poco significativa la conducta de los mismos. Esto se refiere a en qué grado cuantitativo deben alterarse las expectativas futuras. A veces se sabe que un hecho hace menos probables algunas cosas y más otras, la importancia tiene que ver con cuanto menos probables serán unas alternativas respecto a las otras.

- ▶ **Vigencia (en la dimensión espacio-tiempo):** Se refiere a si está actualizada o desfasada. En la práctica la vigencia de una información es difícil de evaluar, ya que en general acceder a una información no permite conocer de inmediato si dicha información tiene o no vigencia.
- ▶ **Validez (relativa al emisor):** Se evalúa si el emisor es fiable o puede proporcionar información no válida (falsa). Tiene que ver si los indicios deben ser considerados en la revaluación de expectativas o deben ser ignorados por no ser indicios fiables.
- ▶ **Valor (activo intangible volátil):** La utilidad que tiene dicha información para el destinatario.(Principales características de la información. párr. 1)

2.4. INTERNET

Raffino M. (2014) en el sitio web “¿Qué es Internet?” refiere lo siguiente:

Internet es una red de computadoras que se encuentran interconectadas a nivel mundial para compartir información. Se trata de una red de equipos de cálculo que se relacionan entre sí a través de la utilización de un lenguaje universal. El concepto Internet tiene sus raíces en el idioma inglés y se encuentra conformado por el vocablo inter (que significa entre) y net (proveniente de network que quiere decir red electrónica). Es un término que siempre debe ser escrito en mayúscula ya que, hace referencia a “La

Red” (que conecta a las computadoras mundialmente mediante el protocolo TCP/IP) y sin un artículo que lo acompañe (el/la) para hacerle referencia. ([¿Qué es Internet?. párr. 1](#))

2.5. MEDIOS DE INFORMACIÓN

Según Wikipedia (2016) en su página web “Medio de comunicación” refiere lo siguiente:

Un medio de comunicación es un instrumento o forma de contenido por el cual se realiza el proceso de comunicación. Usualmente se emplea el término para hacer referencia a los medios de comunicación de masas, sin embargo, otros medios de comunicación, como el teléfono, no son masivos sino interpersonales. Desde que los medios de comunicación nacieron y se desarrollaron, se han vuelto una gran fuente de poder e influencia social a nivel mundial.

Los medios de comunicación son instrumentos en permanente evolución ya que a través de los años su forma de transmitir información se ha hecho bastante masiva e instantánea. (Medios de Información, párr. 2)

2.6 NOTICIAS

Coelho, F (2019) en el sitio web “Significado de Noticia” refiere lo siguiente:

Una noticia es la información sobre un hecho o un conjunto de hechos que, dentro de una comunidad, sociedad o ámbito específico, resulta relevante, novedosa o inusual. La palabra, como tal, proviene del latín *notitia*. En este sentido, una noticia es el relato, la construcción y la elaboración de un hecho, evento o acontecimiento que se considera importante o pertinente divulgar para el conocimiento público.

Las noticias son los relatos de los eventos o acontecimientos más importantes en un día o una semana. Es lo que llena las páginas de los

periódicos o diarios, de los portales web noticiosos o de los programas informativos de radio y televisión. (Qué es Noticia, párr. 1)

2.7 NOTICIAS FALSAS (FAKE NEWS)

Según Wikipedia (2018) en su página web “Fake news” refiere lo siguiente:

Las noticias falsas, conocidas también con el anglicismo fake news, son un tipo de bulo que consiste en un contenido pseudo periodístico difundido a través de portales de noticias, prensa escrita, radio, televisión y redes sociales y cuyo objetivo es la desinformación.

Se diseñan y emiten con la intención deliberada de engañar, inducir a error, manipular decisiones personales, desprestigiar o enaltecer a una institución, entidad o persona u obtener ganancias económicas o rédito político. Al presentar hechos falsos como si fueran reales, son consideradas una amenaza a la credibilidad de los medios serios y los periodistas profesionales, a la vez que un desafío para el público receptor. (Fake news, párr. 1)

2.8 REDES SOCIALES

Según Wikipedia (2018) en su página web “Red Social” refiere lo siguiente:

Una red social es una estructura social compuesta por un conjunto de usuarios (tales como individuos u organizaciones) que están relacionados de acuerdo a algún criterio (relación profesional, amistad, parentesco, etc.).El tipo de conexión representable en una red social es una relación diádica o lazo interpersonal. Las redes sociales se han convertido, en pocos años, en un fenómeno global, se expanden como sistemas abiertos en constante construcción de sí mismos, al igual que las personas que las utilizan.

Las investigaciones han mostrado que las redes sociales constituyen

representaciones útiles en muchos niveles, desde las relaciones de parentesco hasta las relaciones de organizaciones a nivel estatal (se habla en este caso de redes políticas), desempeñando un papel crítico en la determinación de la agenda política y el grado en el cual los individuos o las organizaciones alcanzan sus objetivos o reciben influencias. La red social también puede ser utilizada para medir el capital social (es decir, el valor que un individuo obtiene de los recursos accesibles a través de su red social). (Red Social, párr. 1)

2.9 MEDIO DIGITAL

Wikipedia (2016) en su página web “Medios digitales” refiere lo siguiente:

Los medios digitales son cualquier medio codificado en un formato legible para una máquina. Los medios digitales se pueden crear, visualizar, distribuir, modificar y preservar en dispositivos electrónicos digitales. Ejemplos de medios digitales incluyen software, imágenes digitales, vídeo digital, videojuegos, páginas web y sitios web, redes sociales en Internet, archivos digitales y bases de datos, audio digital como MP3 y libros electrónicos. Los medios digitales a menudo contrastan con medios impresos, como libros, periódicos y revistas impresas, y otros medios tradicionales o analógicos, como película fotográfica, casetes o cintas de video. Implican la interacción y uso de formato de texto, imágenes, elementos gráficos, animación y sonido, para crear productos digitales que puedan ser usados con o sin conexión.

Los medios digitales contrastan frecuentemente con los medios impresos, tales como libros impresos, periódicos y revistas, y otros medios de comunicación tradicionales o análogos, tales como imágenes, películas o cintas de audio. En combinación con Internet y la informática personal, los medios digitales han causado una ruptura en la publicación, el periodismo, el entretenimiento, la educación, el comercio y la política. (Medio Digital,

párr. 1)

2.10 PERIÓDICO

Wikipedia (2012) en su página “Periódico” refiere lo siguiente:

Un periódico o diario es un documento que presenta en orden cronológico o temático de un cierto número de informaciones y comentarios sobre sucesos ocurridos o previsibles durante un período dado (generalmente una jornada o una sucesión de jornadas, de ahí precisamente el nombre de diario). Por extensión, un diario también designa una publicación (impresa) que agrupa y presenta cierto número de artículos sobre los sucesos del día. (Periódico párr. 1)

2.11 PERIÓDICO DIGITAL

Según Wikipedia (2006) en su página web “Periódico en línea” refiere lo siguiente:

Un periódico en línea, periódico digital, diario en línea o diario digital es la edición de un periódico que utiliza Internet como su principal medio de difusión. Aunque actualmente la mayoría de ellos son completamente gratuitos, todo parece indicar que en un futuro próximo sean de pago.

Se refiere así mismo a una publicación que a diferencia de su contraparte impresa se presenta en formato electrónico y su principal medio de difusión es Internet. La frecuencia con que publique puede variar, pero es común que se actualicen diariamente, aunque también hay ediciones semanales, quincenales, mensuales, semestrales o incluso anuales. El hecho de que haya repetición cíclica en sus ediciones los convierte en "periódicos". La publicación electrónica le da el carácter de "digital". (Periódico en Línea, párr. 1)

2.12 TÉCNICA

Wikipedia (2003) en su página “Técnica” refiere lo siguiente:

Técnica es el conjunto de procedimientos, reglas, normas o protocolos que tiene como objetivo obtener un resultado determinado y efectivo, ya sea en el campo de la informática, las ciencias, el arte, el deporte, la educación o en cualquier otra actividad.

La técnica primeramente requiere tanto destrezas manuales como intelectuales, frecuentemente el uso de herramientas y de varios conocimientos. En los animales las técnicas son características de cada especie. En el hombre, la técnica surge de su necesidad de modificar el medio y se caracteriza por ser transmisible, aunque no siempre es consciente o reflexiva. (Requerimientos de la técnica, párr. 1)

2.13 WEB SCRAPING

Según Wikipedia (2012) en su página web “Web scraping” refiere lo siguiente:

Web scraping es una técnica utilizada mediante programas de software para extraer información de sitios web. Usualmente, estos programas simulan la navegación de un humano en la World Wide Web ya sea utilizando el protocolo HTTP manualmente, o incrustando un navegador en una aplicación.

El web scraping se enfoca más en la transformación de datos sin estructura en la web (como el formato HTML) en datos estructurados que pueden ser almacenados y analizados en una base de datos central, en una hoja de cálculo o en alguna otra fuente de almacenamiento. Alguno de los usos del web scraping son la comparación de precios en tiendas, la monitorización de datos relacionados con el clima de cierta región, la detección de cambios en sitios webs y la integración de datos en sitios webs. También es

utilizado para obtener información relevante de un sitio. (Web scraping, párr. 2)

2.14 TÉCNICAS WEB SCRAPING

Wikipedia (2012) en su página “Web scraping” refiere lo siguiente:

Web scraping es el proceso de recopilar información de forma automática de la Web. Es un campo con desarrollos activos, compartiendo un propósito en común con la visión de la Web semántica. Utiliza soluciones prácticas basadas en tecnologías existentes que son comúnmente ad hoc. Existen distintos niveles de automatización que las existentes tecnologías de Web Scraping pueden brindar:

- ▶ **«Copiar y pegar» humano:** algunas veces incluso las mejores técnicas de web scraping no pueden reemplazar el examen manual de un humano, y a veces esta puede ser la única vía de solución cuando el sitio que tenemos en mente pone ciertas barreras para prevenir que se creen softwares para realizar tareas automáticas en este.
- ▶ **Uso de expresiones regulares:** una posible vía para extraer información de páginas webs pueden ser las expresiones regulares, aunque comúnmente no se recomienda utilizarlas para parsear el formato HTML.
- ▶ **Protocolo HTTP:** páginas webs estáticas y dinámicas pueden ser obtenidas haciendo peticiones HTTP al servidor remoto utilizando sockets, etc.
- ▶ **Algoritmos de minería de datos:** muchos sitios webs tienen grandes colecciones de páginas generadas dinámicamente a partir de una base de datos. Datos de la misma categoría aparecen usualmente en páginas similares mediante un script o una plantilla. En la minería de datos, un programa detecta estas plantillas en un contexto específico y

extrae su contenido.

- ▶ **Parsers de HTML:** Algunos lenguajes, como XQuery y HTQL pueden ser utilizados para parsear documentos, recuperar y transformar el contenido de documentos HTML.
- ▶ **Aplicaciones para web scraping:** existen muchas aplicaciones disponibles que pueden ser utilizadas para personalizar soluciones de Web Scraping. Estas aplicaciones podrían reconocer automáticamente la estructura de cierta página o brindar una interfaz al usuario donde este pudiera seleccionar los campos que son de interés dentro del documento. De esta forma no es necesario escribir manualmente código para realizar estas tareas.
- ▶ **Reconocimiento de información semántica:** las páginas que son analizadas podrían incluir metadatos o cierta información semántica como anotaciones o comentarios, los cuales pueden ser usados comúnmente. Si estas anotaciones están en las mismas páginas, como sucede con los microformatos, estas podrían ser de utilidad cuando parseamos el DOM del documento. En otro caso, las anotaciones, organizadas en una capa semántica, son almacenadas y manejadas de forma separada desde otras páginas, por lo que los scrapers pueden recuperar estos esquemas y las instrucciones desde esta capa antes de analizar los documentos. (Técnicas, párr. 1)

2.15 HEMEROTECA

Según Evelyn C. (2013) en su sitio web “Hemeroteca” refiere lo siguiente:

Proviene de dos palabras griegas Hemera (Día) y Theke (caja o depósito). Es un lugar físico para almacenar las publicaciones diarias o periódicas que llegan a un lugar específico, éste por lo general está ubicado en las bibliotecas. (Hemeroteca, párr. 1)

2.15.1 ¿Que es una Hemeroteca?

Boté J. (2013) en el sitio web “¿Qué es una hemeroteca y para que sirve?” refiere lo siguiente:

Una hemeroteca es un sitio para buscar información, que está editada por periódicos o revistas. Este tipo de información se llama publicaciones periódicas. Si lo quieres ver de otra forma una hemeroteca es un lugar donde puedes encontrar periódicos antiguos o revistas antiguas clasificadas. (Que es una hemeroteca, párr. 1)

Ucha F. (2019) en el sitio web “Definición de Hemeroteca” refiere lo siguiente:

Una Hemeroteca es aquella biblioteca que se especializa en la recolección, conservación y almacenaje de diarios, revistas u otras publicaciones periódicas afines a estos para su posterior consulta. La misma puede erigirse y funcionar en un edificio propio, en una sala específica o bien en un sector determinado pero dentro de una Biblioteca tradicional. (Definición de Hemeroteca, párr. 1)

Según Rossana A. (2019) en su sitio web “Hemeroteca” refiere lo siguiente:

La hemeroteca es el edificio, local o sala, generalmente ubicado en una biblioteca, donde se guardan, de forma organizada, colecciones de periódicos, diarios y revistas destinadas a servir al público. Para saber qué es una hemeroteca, se necesita conocer su origen, mismo que está asociado a la evolución de la biblioteca y la industria editorial. Esta edificación se desarrolla a medida que se advierte la necesidad de concentrar y organizar un número creciente de diarios, revistas y otras publicaciones periódicas, anuales e irregulares. En 1900 el término fue propuesto por Henri Martin en un Congreso Internacional de Bibliotecas, realizado en París. (Hemeroteca Definición, párr. 1)

2.15.2 Función

Evelyn C. (2013) en el sitio web “Hemeroteca” refiere lo siguiente:

La hemeroteca es aquella biblioteca que se especializa en la recolección, conservación y almacenaje de diarios, revistas u otras publicaciones periódicas afines a estos para su posterior consulta, así como la preparación y publicación de los catálogos para lograr una mejor difusión, facilitando, las operaciones de préstamo. La misma puede erigirse y funcionar en un edificio propio, en una sala específica o bien en un sector determinado pero dentro de una biblioteca tradicional. Generalmente, realizan la clasificación de sus contenidos a través de los siguientes criterios: tema, país, origen, fecha. (Función, párr. 1)

Según Rossana A. (2019) en el sitio web “Hemeroteca” refiere lo siguiente:

En principio, la función fundamental de estas bibliotecas diarias es la atención al público y de esta manera resolver las dudas que los usuarios puedan tener sobre diferentes temas, pero, esta también tiene otras funcionalidades. Las hemerotecas en papel tienen la misma función que las bibliotecas, la persona interesada busca una noticia en específico, es decir, con una fecha concreta y le solicita al bibliotecario la consulta de uno o varios números de la misma noticia. La única diferencia que existe entre una biblioteca y una hemeroteca, es que con esta última no se puede sacar el material de investigación del recinto. (Funciones de las hemerotecas, párr. 1)

2.15.3 Tipos

Rossana A. (2019) en el sitio web “Hemeroteca” refiere lo siguiente:

Dentro de las hemerotecas se puede encontrar diferentes clasificaciones, para lo cual cada texto incluido debe reunir una serie de características.

- ▶ **Periódicos:** Pueden clasificarse de acuerdo a su contenido o a su área

de implantación.

- ▶ **Por su contenido:** pueden ser de información general o especializada. (En ambos casos se requiere que sean textos de índole informativa pero en el segundo caso se exige también la precisión en el uso de un lenguaje científico determinado).
- ▶ **Público al que van dirigidas:** como técnicas (orientados a un público que maneja un determinado vocabulario y conocimientos específicos sobre un tema) o de divulgación (desarrolladas para todos los públicos, con pocas exigencias).
- ▶ **Por su área de implantación:** pueden ser locales, regionales, provinciales, nacionales o internacionales.
- ▶ **Revistas:** Podrían asemejarse a los periódicos e incluso a algunos libros, se distinguen por su apariencia externa.
- ▶ **Boletines, anuales y memorias:** Suelen salir con una periodicidad anual y son elaborados por instituciones que desean compartir con el público en general o con aquellos que forman parte de la misma, las actividades que allí realizan. Además, pueden hacerles llegar conocimientos en materia de estudios o investigaciones. (Tipos de Hemeroteca, párr. 1)

2.15.4 Importancia de una Hemeroteca

Rossana A. (2019) en el sitio web “Hemeroteca” refiere lo siguiente:

Se puede explicar su importancia a través de varias opciones, estas se basan en hacer historia a partir de su propia comunidad, cubrir necesidades intelectuales, comprender el presente sobre la base del conocimiento pasado, su relación con el pasado y el presente, establecer cimientos de los estudios de investigación y realizar especializaciones en diferentes

temas. Si explican de manera detallada, la información quedaría de la siguiente manera:

- ▶ **Hacer historia a partir de su propia comunidad:** esto se ha hecho desde tiempo inmemoriales porque cuentan con la información necesaria para mantener el legado de las diferentes regiones de un país.
- ▶ **Cubre necesidades intelectuales:** al facilitar información sobre diferentes temas de investigación, nutre de conocimientos a todos los usuarios que hacen uso de los servicios de la hemeroteca.
- ▶ **Comprende el presente sobre la base del conocimiento pasado:** este aspecto hace que sea más fácil imaginar lo que depara el futuro de la humanidad, de tal manera, brinda herramientas para construirlo y materializarlo en pocos lapsos de tiempo.
- ▶ **Relación pasado-presente:** al coleccionar información sobre diferentes temas de investigaciones, se pueden realizar comparaciones sobre sucesos del pasado y el cambio que se ha generado con el paso de los años hasta el presente.
- ▶ **Cimiento de todo estudio de investigación:** estas colecciones de revistas o periódicos poseen información que data de muchos años atrás, esto ha hecho que sean consideradas como cimientos investigativos en muchas partes del mundo.
- ▶ **Especialización en un tema específico:** con cada aporte realizado en las hemerotecas y la información que obtienen los investigadores, les facilita la especialización en uno o varios temas que sean de su agrado.

Con solo aprender a leer un periódico, se podían realizar diferentes trabajos de investigación, pero, hoy en día, se justifica la elaboración de instrumentos bibliográficos con la enseñanza de la búsqueda efectiva de

datos y esta, es otra gran importancia de estos depósitos de publicaciones en la actualidad. Ahora, al conocer qué es una hemeroteca, sus funciones, para qué sirve una hemeroteca y, por supuesto, el marco de importancia de hemeroteca, resulta imperativo explicar como hacer una hemeroteca en pocos pasos (tanto en físico como digital). (Importancia, párr. 1-3)

2.15.5 Las ventajas que obtienes en una hemeroteca

Boté J. (2013) en el sitio web “¿Qué es una hemeroteca y para que sirve?” refiere lo siguiente:

En una hemeroteca obtienes una ventaja clara. La información ha pasado un filtro editorial. Es información contrastada. Es decir, ningún periódico saca nada a la luz si no está contrastado. Por tanto la información es válida.

La información la buscas por fechas, por temas, por personalidades. Normalmente en todas las hemerotecas digitales que hay en internet, tienes un sistema de ayuda sobre cómo puedes emplear su buscador de noticias antiguas. Todo esto que te acabo de contar te da una idea de que es una hemeroteca y en concreto una hemeroteca virtual. (Las ventajas que obtienes en una hemeroteca, párr. 1)

2.16 HEMEROTECA DE MEDIOS

Según Auditmedia (2013) en el sitio web “¿Qué es una hemeroteca de medios?” refiere lo siguiente:

Hoy en día existe una gran cantidad de información en formato digital y muchos medios digitales que generan contenidos e información. Esto ha hecho que se puede hablar de hemeroteca de medios.[\(Hemeroteca de Medios, párr. 2\)](#)

2.16.1 Que es una Hemeroteca de Medios

Según Auditmedia (2013) en el sitio web “¿Qué es una hemeroteca de medios?” refiere lo siguiente:

Si sabes lo que es una hemeroteca debes saber que los diferentes medios y periódicos digitales disponen de hemerotecas propias donde recogen todas las noticias y publicaciones que han ido haciendo.

Con la evolución al mundo digital las hemerotecas tradicionales han digitalizado sus fondos facilitando así que su consulta se haga de forma mucho más sencilla. Ahora es más fácil acceder a esta información a través de Internet.

En una hemeroteca de medios puedes encontrar información de numerosas publicaciones digitales. Habitualmente la información de la biblioteca está organizada por fechas, temáticas, medios, autores, etc.

El acceso a la hemeroteca se realiza mediante unos formularios en los que puedes rellenar la información que estás buscando o que te interesa. Es un sistema muy sencillo de usar y te ofrecerá los resultados organizados y ordenados. [\(Qué ofrece una hemeroteca de medios, párr. 1-3\)](#)

2.16.2 Tipos de hemerotecas de medios

Según Auditmedia (2013) en el sitio web “¿Qué es una hemeroteca de medios?” refiere lo siguiente:

En cuanto a los tipos de hemerotecas de medios que puedes encontrar depende de la fuente en la que está publicada la información. Con carácter general podemos encontrar una clasificación como la siguiente:

- ▶ **Periódicos:** Se trata de la hemeroteca que cualquier periódico online tiene habilitada para la consulta de los fondos de noticias que han publicado desde su puesta en funcionamiento. Estos medios pueden también clasificarse en función de la información que ofrecen. Así puedes encontrar periódicos especializados en determinadas temáticas, puedes encontrar periódicos en función del área geográfica

a la que se refieren, tanto de área local como regional o nacional.

- ▶ **Revistas:** Otro tipo de hemeroteca que puedes encontrar es la correspondiente a las revistas. Las revistas son similares a los periódicos, pero se caracterizan por las temáticas que tratan y la forma de tratar la información. Estas revistas también podrás encontrarlas especializadas en temáticas varias o bien referidas a un determinado ámbito geográfico.
- ▶ **Boletines y memorias:** Estos documentos suelen tener un carácter más oficial y son elaborados por diferentes instituciones. La información que recogen suele estar referida a la administración y a estudios e investigaciones que se realizan en estos medios. Se trata de una información que se publica con una periodicidad más amplia en el tiempo. ([Tipos de hemerotecas de medios, párr. 1](#))

2.17 INGENIERÍA DE REQUERIMIENTOS

El proceso de recopilar, analizar y verificar las necesidades del cliente para un sistema de software es llamado Ingeniería de Requerimientos. La meta de la ingeniería de requerimientos es entregar una especificación de requerimientos de software correcta y completa.

La ingeniería de requerimientos apunta a mejorar la forma en que comprendemos y definimos sistemas de software complejos. Existen varias definiciones de requerimientos, de entre las cuales podemos citar las siguientes:

- ▶ Ingeniería de Requerimientos es la disciplina para desarrollar una especificación completa, consistente y no ambigua, la cual servirá como base para acuerdos comunes entre todas las partes involucradas y en dónde se describen las funciones que realizará el sistema (Boehm, 1988, pág 7)
- ▶ Es el proceso mediante el cual se intercambian diferentes puntos de

vista para recopilar y modelar lo que el sistema va a realizar. Este proceso utiliza una combinación de métodos, herramientas y actores, cuyo producto es un modelo del cual se genera un documento de requerimientos (Leite, 1987, pág 216)

Figura 2.1: Ciclo de Vida de la Ingeniería de Requerimientos



Fuente: (Boehm, 1988, pág 15)

2.17.1 Requerimientos Funcionales

Segun Pmoinformatica (2016) en el sitio web "Requerimientos no funcionales" refiere lo siguiente:

Los requerimientos funcionales son declaraciones de los servicios que proveerá el sistema, de la manera en que éste reaccionará a entradas particulares. En algunos casos, los requerimientos funcionales de los sistemas también declaran explícitamente lo que el sistema no debe hacer. (Requerimientos no funcionales: Ejemplo, párr. 2)

Muchos de los problemas de la ingeniería de software provienen de la imprecisión en la especificación de requerimientos. Para un desarrollador de sistemas es natural dar interpretaciones de un requerimiento ambiguo con el fin de simplificar su implementación.

Sin embargo, a menudo no es lo que el cliente desea. Se tienen que estipular nuevos requerimientos y se deben hacer cambios al sistema, retrasando la entrega de éste e incrementando el costo.

En la práctica, para sistemas grandes y complejos, es imposible cumplir los requerimientos de consistencia y compleción. La razón de esto se debe parcialmente a la complejidad inherente del sistema y parcialmente a que los diferentes puntos de vista tienen necesidades inconsistentes. Estas inconsistencias son obvias cuando los requerimientos se especifican por primera vez. Los problemas emergen después de un análisis profundo. Una vez que éstos se hayan descubierto en las diferentes revisiones o en las fases posteriores del ciclo de vida, se deben corregir en el documento de requerimientos.

2.17.2 Requerimientos No Funcionales

Google Sites (s.f.) en el sitio web “Metodología Gestión de Requerimientos” refiere lo siguiente:

Son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y la representación de datos que se utiliza en la interface del sistema.

(Identificación de Requerimientos no funcionales, párr. 1)

Los requerimientos no funcionales surgen de la necesidad del usuario, debido a las restricciones en el presupuesto, a las políticas de la organización, a la necesidad de interoperabilidad con otros sistemas de software o hardware o a factores externos como los reglamentos de seguridad, las políticas de privacidad, entre otros.

Estos diferentes tipos de requerimientos se clasifican de acuerdo con sus

implicaciones:

- ▶ **Requerimientos del producto:** Especifican el comportamiento del producto; como los requerimientos de desempeño en la rapidez de ejecución del sistema y cuánta memoria se requiere; los de fiabilidad que fijan la tasa de fallas para que el sistema sea aceptable; los de portabilidad y los de usabilidad.
- ▶ **Requerimientos organizacionales:** Se derivan de las políticas y procedimientos existentes en la organización del cliente y en la del desarrollador: estándares en los procesos que deben utilizarse; requerimientos de implementación como los lenguajes de programación o el método de diseño a utilizar, y los requerimientos de entrega que especifican cuándo se entregará el producto y su documentación.
- ▶ **Requerimientos externos:** Se derivan de los factores externos al sistema y de su proceso de desarrollo. Incluyen los requerimientos de interoperabilidad que definen la manera en que el sistema interactúa con los otros sistemas de la organización; los requerimientos legales que deben seguirse para asegurar que el sistema opere dentro de la ley, y los requerimientos éticos. Estos últimos son impuestos al sistema para asegurar que será aceptado por el usuario.

En la práctica, la especificación cuantitativa de requerimientos es difícil. A los clientes no les es posible traducir sus metas en requerimientos cuantitativos; para algunas de éstas, como las de mantenimiento, no existen métricas que se puedan utilizar; el costo de verificar de forma objetiva los requerimientos no funcionales cuantitativos es muy alto.

2.17.3 Características de un Requerimiento

Un requerimiento debe cumplir ciertos criterios y características:

- ▶ **Único:** El requerimiento debe poder ser interpretado inequívocamente de una sola manera.
- ▶ **Verificable:** Su implementación debe poder ser comprobada. El test debe dar como resultado CORRECTO o INCORRECTO.
- ▶ **Claro:** Los requerimientos no deben contener terminología innecesaria. Deben ser establecidos de forma clara y simple.
- ▶ **Viable (realista y posible):** El requerimiento debe ser factible según las restricciones actuales de tiempo, dinero y recursos disponibles.
- ▶ **Necesario:** Un requerimiento no es necesario si ninguno de los interesados necesita el requerimiento o bien si la retirada de dicho requerimiento no tiene ningún efecto.

2.17.4 Dificultades para definir un Requerimiento

Según Blogspot (2012) en el sitio web “Recopilación de información dentro del entorno del Desarrollo de Software.” refiere lo siguiente:

- ▶ Los requerimientos no son obvios y vienen de muchas fuentes.
- ▶ Son difíciles de expresar en palabras (el lenguaje es ambiguo).
- ▶ Existen muchos tipos de requerimientos y diferentes niveles de detalle.
- ▶ La cantidad de requerimientos en un proyecto puede ser difícil de manejar.
- ▶ Nunca son iguales. Algunos son más difíciles, más riesgosos, más importantes o más estables que otros.
- ▶ Los requerimientos están relacionados unos con otros, y a su vez se relacionan con otras partes del proceso.
- ▶ Un requerimiento puede cambiar a lo largo del ciclo de desarrollo.
- ▶ Cada requerimiento tiene propiedades únicas y abarcan áreas

funcionales específicas.

- ▶ Son difíciles de cuantificar, ya que cada conjunto de requerimientos es particular para cada proyecto. (Dificultades para Definir los Requerimientos, párr. 1)

2.18 INGENIERÍA DE SOFTWARE

“La ingeniería de software es el establecimiento y uso de principios fundamentales de la ingeniería con objeto de desarrollar en forma económica software que sea confiable y que trabaje con eficiencia en máquinas reales”. (Bauer, 1969, pág 9)

“La ingeniería de software es una disciplina que integra métodos, herramientas y procedimientos para el desarrollo de software de computadora” (Pressman, 1994)

Figura 2.2: Ingeniería de Software



La ingeniería de software es una tecnología con varias capas. El fundamento para la ingeniería de software es la capa proceso. El proceso de ingeniería de software es el aglutinante que une las capas de la tecnología y permite el desarrollo racional y oportuno del software de cómputo. (Pressman, 2010, pág. 11)

La ingeniería de software es una disciplina de la ingeniería que se interesa por todos los aspectos de la producción del software.(Sommerville, 2011, pág. 6)

La ingeniería de software es un conjunto de métodos, técnicas, procesos y procedimientos aplicados al desarrollo de software cuyo objetivo es tener un desarrollo de software con buena calidad al menor tiempo posible.

2.18.1 Modelos de Desarrollo de Software UWE

La propuesta de Ingeniería Web basada en UML (UWE) es una metodología detallada para el proceso de autoría de aplicaciones con una definición exhaustiva del proceso de diseño que debe ser utilizado. Este proceso, iterativo e incremental, incluye flujos de trabajo y puntos de control, y sus fases coinciden con las propuestas en el Proceso Unificado de Modelado.

UWE está especializada en la especificación de aplicaciones adaptativas, y por tanto hace especial hincapié en características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimiento o tareas de usuario.

Otras características relevantes del proceso y método de autoría de UWE son el uso del paradigma orientado a objetos, su orientación al usuario, la definición de un meta-modelo (modelo de referencia) que da soporte al método y el grado de formalismo que alcanza debido al soporte que proporciona para la definición de restricciones sobre los modelos.

2.18.1.1 Principales Aspectos

Los principales aspectos en los que se fundamenta UWE son los siguientes:

- ▶ **Uso de una Notación Estándar:** Para todos los modelos (UML: Lenguaje de Modelado Unificado).
- ▶ **Definición de Métodos:** Definición de los pasos para la construcción de los diferentes modelos.
- ▶ **Especificación de Restricciones:** Se recomienda el uso de restricciones escritas (OCL: Lenguaje de restricciones de objetos) para

aumentar la exactitud de los modelos.

2.18.1.2 Fases del Desarrollo Web

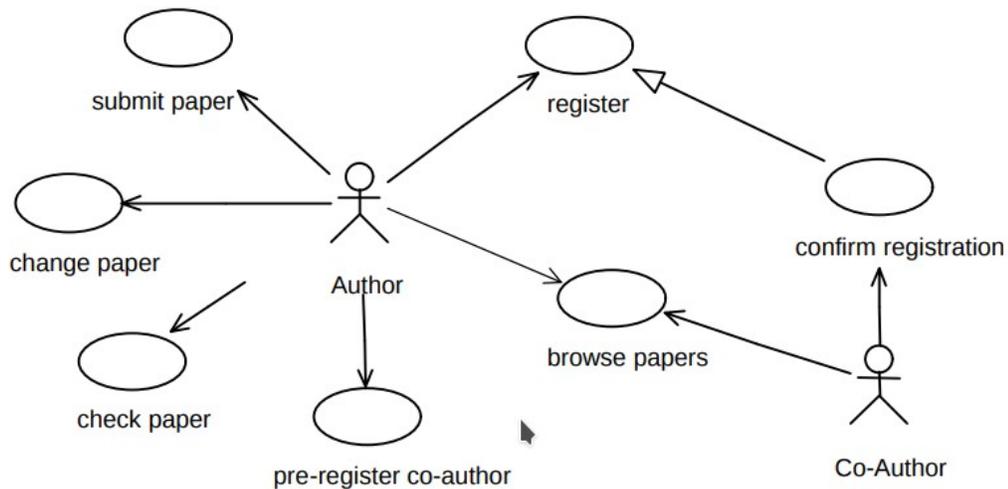
Por lo que respecta al proceso de autoría de la aplicación UWE hace un uso exclusivo de estándares reconocidos como UML y el lenguaje de especificación de restricciones asociado OCL. Para simplificar la captura de las necesidades de las aplicaciones web, UWE propone una extensión que se utiliza a lo largo del proceso de autoría. Este proceso de autoría está dividido en cuatro pasos o actividades:

- ▶ **Análisis de Requerimientos:** Fija los requisitos funcionales de la aplicación web para reflejarlos en un modelo de casos de uso.
- ▶ **Diseño Conceptual:** Materializado en un modelo de dominio considerando los requisitos reflejados en los casos de uso.
- ▶ **Diseño Navegacional:** Lo podemos subdividir en:
 - ◆ Modelo del Espacio Navegacional.
 - ◆ Modelo de la Estructura de Navegación.
- ▶ **Diseño de Presentación:** Representa las vistas del interfaz del usuario mediante modelos estándares de interacción UML.

2.18.1.3 Análisis de Casos de Uso

Siguiendo el Proceso de Desarrollo de Software Unificado aplicado a la Ingeniería Web, se propone los casos de uso necesarios para capturar los requisitos del sistema. Ésta es una técnica que fuerza a definir quienes son los actores (usuarios) de la aplicación y ofrece una manera intuitiva de representar la funcionalidad y aplicación en cada uno de los actores.

Figura 2.3: Casos de uso presentación de documentos.



Fuente: (Nora Koch, 2001, pág 5)

Los principales elementos de modelado utilizados para el modelado de casos de uso son: actores y casos de uso. Ellos pueden ser relacionados por herencia, incluyen o extienden relaciones. Todos estos elementos de modelado, así como el los mecanismos de paquete y vista se utilizan con la semántica definida en UML (1999) y representada gráficamente con la notación UML.

2.18.1.4 Representación UML del Modelado Conceptual

El diseño conceptual se basa en el análisis del paso anterior. Incluye los objetos involucrados en la interacción entre el usuario y la aplicación, es decir, especificados en los casos de uso.

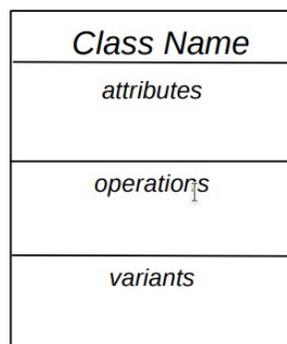
El diseño conceptual tiene como objetivo construir un modelo de clase con estos objetos, que intenta ignorar como muchos de los caminos de navegación, presentación y aspectos de interacción como sea posible. Estos aspectos son pospuesto a los pasos de navegación y presentación del proceso.

2.18.1.5 Elementos de Modelado

Los principales elementos de modelado UML utilizados en el modelo conceptual son: clase, asociación y paquete. Estos se representan gráficamente utilizando la notación UML (1999). Si lo conceptual el modelo consta de muchas clases, se recomienda que se agrupen con el paquete UML.

- ▶ **Clase:** Una clase se describe por un nombre, atributos, operaciones y variantes El compartimento opcional llamado variantes se puede agregar a clases (Koch, 2000, pág 7). Contiene información adicional utilizada para funcionalidad de contenido adaptativo, es decir, para presentar diferentes o contenido adicional para los usuarios según su perfil de usuario. La representación gráfica de una clase con variantes se representa en la Figura 2.4.
- ▶ **Asociación y paquete** Las asociaciones y paquetes se usan como en la clase UML estándar.

Figura 2.4: Clase con compartimiento adicional variants



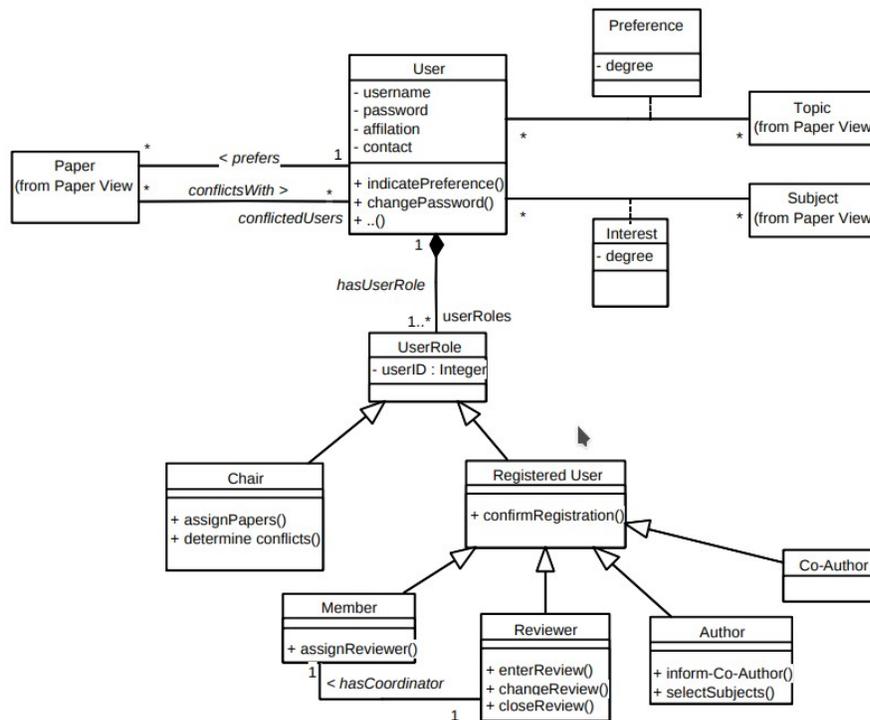
Fuente: (Nora Koch, 2001, pág 7)

Los pasos a seguir para obtener el modelado son los siguientes:

- ▶ Distinguir las clases.
- ▶ Especificar los atributos más importantes.
- ▶ Determinar las asociaciones entre clases.
- ▶ Agregar las clases e identificar la composición de estas.
- ▶ Definir las jerarquías de herencia.
- ▶ Definir las restricciones de los métodos.

En la **Figura 2.4** podemos apreciar el modelo conceptual de un usuario y sus preferencias en los documentos que revisa en un caso práctico.

Figura 2.5: Modelo Conceptual



Fuente: (Nora Koch, 2001, pág 9)

2.18.1.6 Diseño Navegacional

El diseño navegacional no es solo útil para la generación de la documentación de la estructura de la aplicación sino que también permite mejorar la estructura de navegabilidad. El modelo de la navegación comprende de:

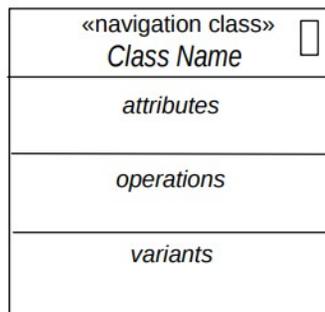
- ▶ El modelo de espacio de navegación que especifica qué objetos pueden ser visitados a través de la aplicación Web.
- ▶ El modelo de estructura de navegación que define cómo se alcanzan estos objetos a través de la Web.

El modelo del espacio navegacional se construye con las clases de navegación y asociaciones entre las mismas y son representadas por un diagrama de clase en UML.

Se usan dos elementos de modelado para la construcción del modelo del espacio navegacional:

- ▶ **Las clases de navegación:** Es una clase cuyas instancias son visitadas por los usuarios durante la navegación. Se les dan el mismo nombre que a sus correspondientes clases conceptuales. Se representan a través de los estereotipos UML de la figura.

Figura 2.6: Clase de Navegación



Fuente: (Nora Koch, 2001, pág. 11)

- ▶ **La navegabilidad directa:** Las asociaciones entre el espacio navegacional representan la navegabilidad directa entre la clase de navegación inicial y la clase de navegación final para determinar las direcciones de la navegación se usan flechas, en la que se indica el rol y su multiplicidad.

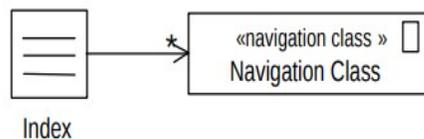
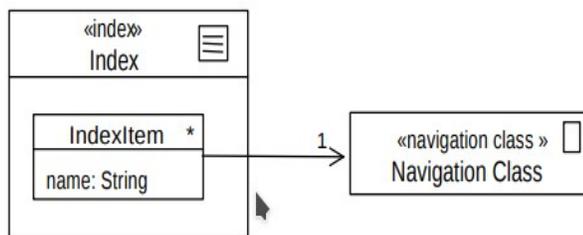
2.18.1.6.1 Estructura del modelo navegacional

En el modelo de estructura navegacional los elementos de acceso son los índices, consultas, y visitas guiadas. Los caminos de la navegación junto con los elementos de acceso se representan por un modelo de clases que puede construirse en dos pasos: El primer paso consiste en reforzar el espacio de navegación mediante los índices, tutoriales y consultas. El segundo paso consiste en derivar los menús directamente del modelo.

Los elementos de modelado que usamos son los índices, consultas y tutoriales.

▶ Índices

Figura 2.7: Clases de Navegación: Índices



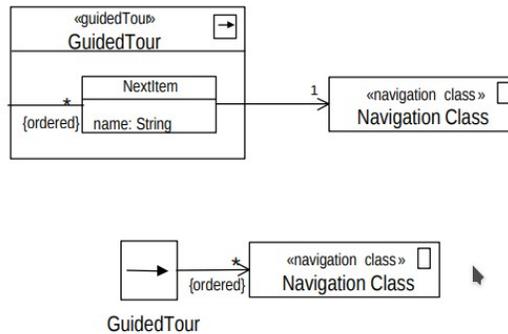
Un índice permite el acceso directo a las instancias de las clases navegacionales.

Está compuesto por un objeto que tiene un nombre que identifica a cada instancia y posee un enlace a una instancia de una clase de navegación.

► **Vuelta guiada**

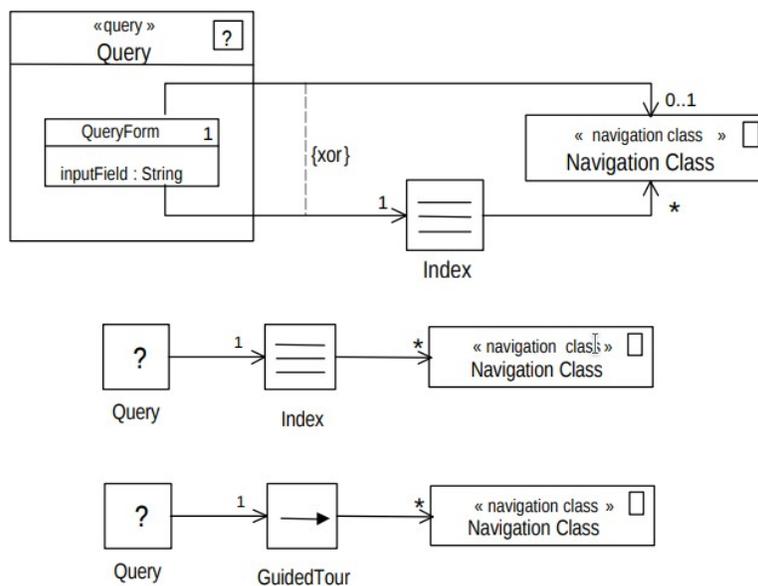
Proporciona el acceso secuencial a las instancias a las instancias de una clase de navegación. Estas deben ser controladas por el usuario del sistema.

Figura 2.8: Clases de Navegación: Vuelta Guiada



► **Consultas**

Figura 2.9: Clase con cadena de consulta como un atributo.



Fuente: (Nora Koch, 2001, pág. 16)

2.18.1.6.2 Añadir menús

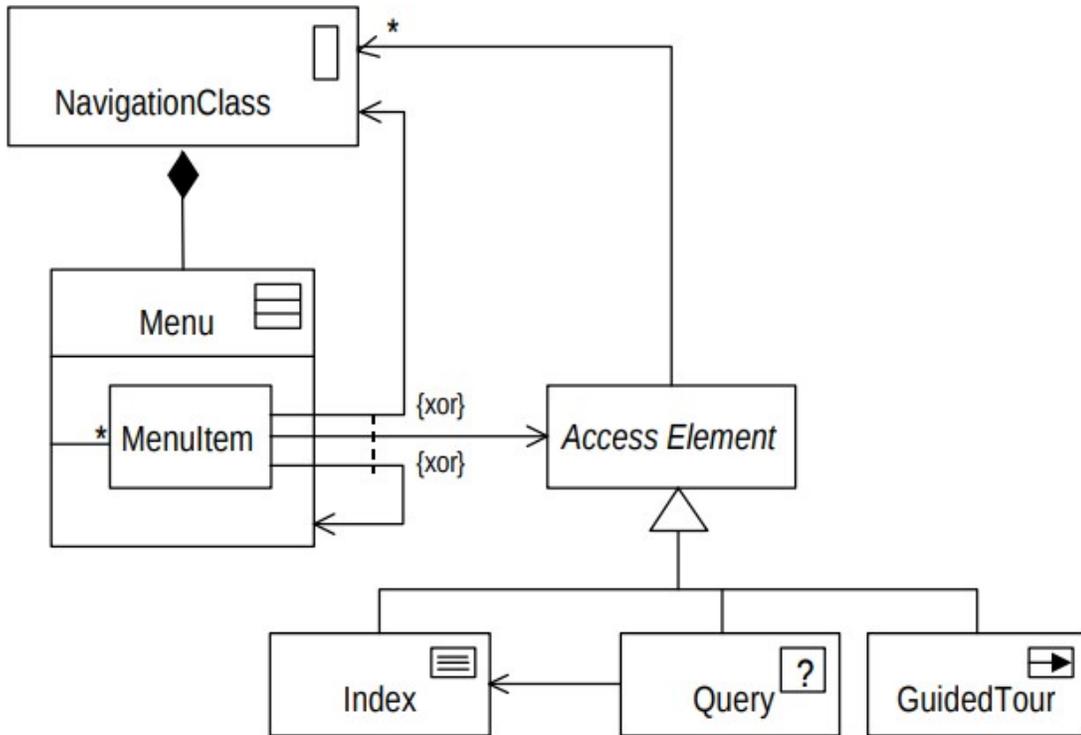
El modelo del espacio navegacional es enriquecido por los elementos de acceso de menú de las siguientes reglas.

Considerar que las asociaciones tienen como fuente una clase de la navegación.

- ▶ Asoció a cada clase de navegación del modelo anterior que tiene al menos una asociación saliente de una clase menú.
- ▶ Reorganizar un menú en submenús.
- ▶ Introduzca para cada rol del extremo de una asociación dirigida del modelo anterior, su correspondiente elemento del menú.
- ▶ Cualquier asociación del modelo anterior que tiene ahora como su fuente una clase de la navegación.
- ▶ Se vuelve una asociación el rol del menú correspondiente.

El resultado que obtenemos es un modelo de estructura de navegación de la aplicación.

Figura 2.10: Patrón para estructuras de acceso.



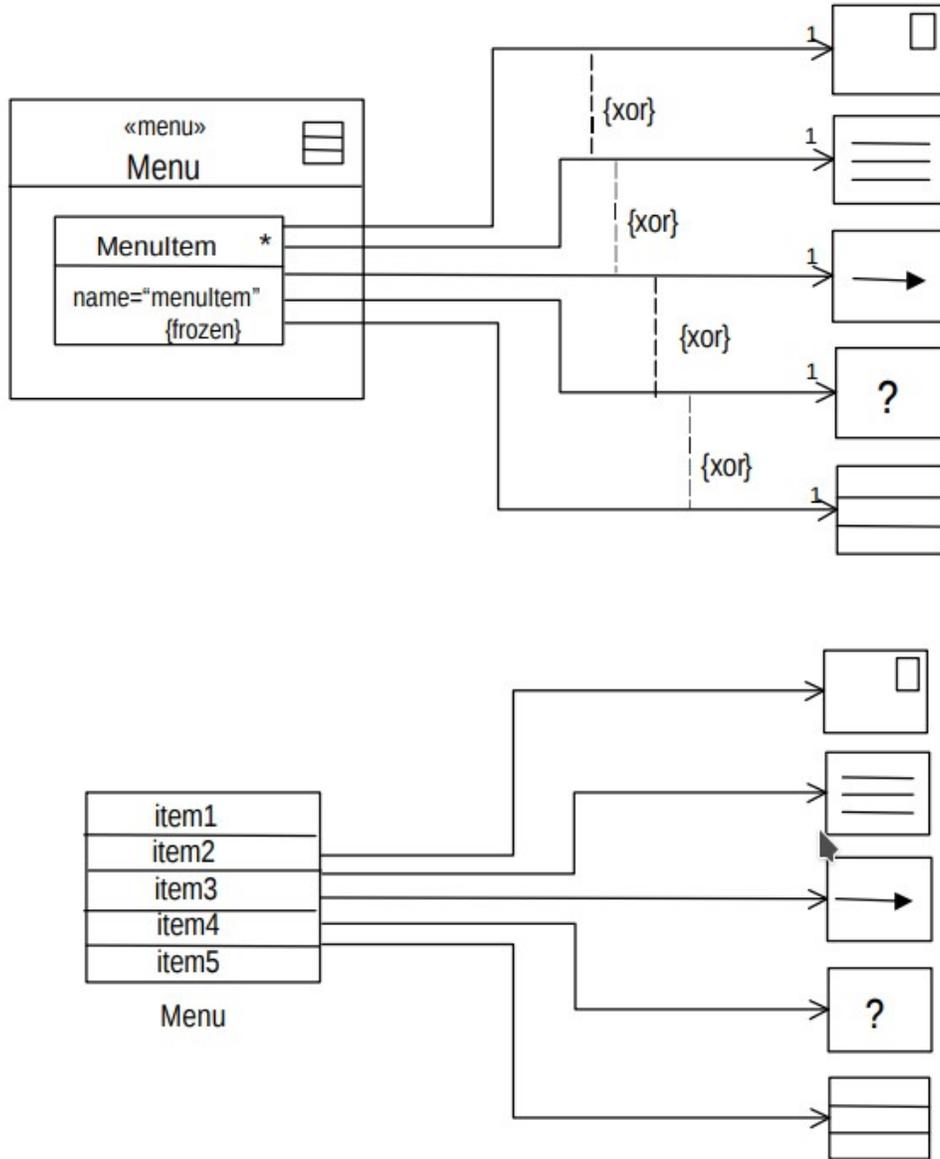
Fuente: (Nora Koch, 2001, pág 20)

2.18.1.6.3 Elementos del modelados

El elemento de modelado menú es acceso primitivo adicional que puede ser añadido a la lista presentada anteriormente (consultas, índices y vueltas guiadas).

- ▶ **Menú:** Está compuesto por un número fijo de elementos de menú. Cada elemento del menú tiene un nombre constante y posee un enlace ya sea a una instancia de una clase de navegación o un elemento de acceso. Cualquier menú es un instancia de alguna clase con un icono correspondiente.

Figura 2.11: Clase menú y taquigrafía de menú.



Fuente: (Nora Koch, 2001, pág. 18)

2.18.1.7 Diseño de Presentación

El diseño de presentación soporta la construcción de un modelo de presentación basado en el modelo de estructura de navegación y la información adicional recogida durante el análisis de requisitos. El modelo de la presentación consiste en un conjunto que muestran el contenido y la estructura de nodos.

Se propone la construcción de bocetos, storyboards y un modelo de flujo de presentación.

Primero el diseñador web propone un boceto de cada vista de interfaz del usuario principal, es decir el plan de interfaces abstractas de usuario.

En el segundo paso está basado en el modelo del storyboard en el que el diseñador puede decidir si el quiere usar la técnica de ventanas múltiples o si el quiere usar frames.

El objetivo del modelo de flujo de presentación es mostrar las vistas de interfaz de usuario del modelo storyboard que se le presentarán al usuario. También muestra cómo los contenidos son reemplazados cuando el usuario interactúa con el sistema.

2.18.1.7.1 Pasos para realizar el Storyboards

Para el diseño de los storyboards (diagramas de cajas) empezamos con un primer modelo de navegación de la aplicación web. Las reglas que nos van a guiar son:

- ▶ Construir una clase presentación para cada clase de navegación que aparece en el modelo de estructura de navegación
- ▶ Construir una clase de presentación para cada menú e índice que aparece en el modelo de estructura de navegación.
- ▶ Construir una clase de la presentación para cada consulta y vuelta

guiada.

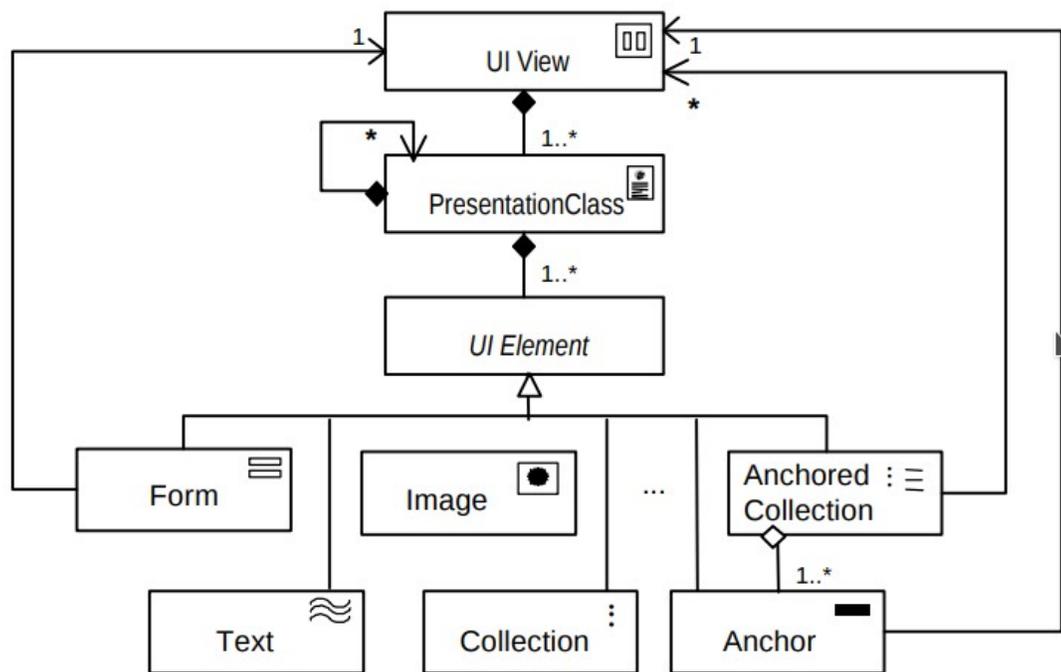
- ▶ Construir una clase presentación para apoyo de la navegación.
- ▶ Agregar los enlaces a las clases de la presentación para permitir creación, destrucción y ejecución de operaciones sobre el modelo conceptual.
- ▶ Determinar que elementos de la presentación deben presentarse juntos al usuario en la ventana.
- ▶ Agregar las restricciones OCL, si son necesarias.
- ▶ Construir el escenario de storyboarding representados por las sucesiones de vistas de interfaz de usuario.

2.18.1.7.2 Elementos del Modelado

Para la construcción de los bocetos nosotros proponemos un conjunto de elementos modelados algunos de los cuales son los siguientes:

- ▶ **Vista de interfaz de usuario:** Especifica que cada instancia de esta clase es un contenedor de todos los elementos de interfaz de usuario abstractos que se presentan simultáneamente al usuario.
- ▶ **La clase presentación:** Es una estructura única la cual permite la división en vistas de interfaz de usuario dentro de grupos de elementos de interfaz.
- ▶ **Elementos de interfaz de usuario:** Es una clase abstracta la cual tiene elemento de interfaz de usuario describiendo los elementos particulares de la interfaz.

Figura 2.12: Metamodelo para los elementos abstractos de la interfaz de usuario



Fuente: (Nora Koch, 2001, pág. 21)

2.18.1.7.3 Modelado del Flujo de Presentación

Se modela el flujo de presentación mostrando donde se presentarán los objetos de navegación y elementos de acceso al usuario, es decir en qué frame o ventana se desplegará el contenido y qué enlaces se activarán.

El diseñador tiene que especificar si usará una sola o múltiples ventanas, si se usarán los frames y en ese caso, en cuántos framesets se dividirán.

La construcción de la estructura de presentación no puede ser automatizada totalmente, pero hay ciertas pautas que podemos seguir:

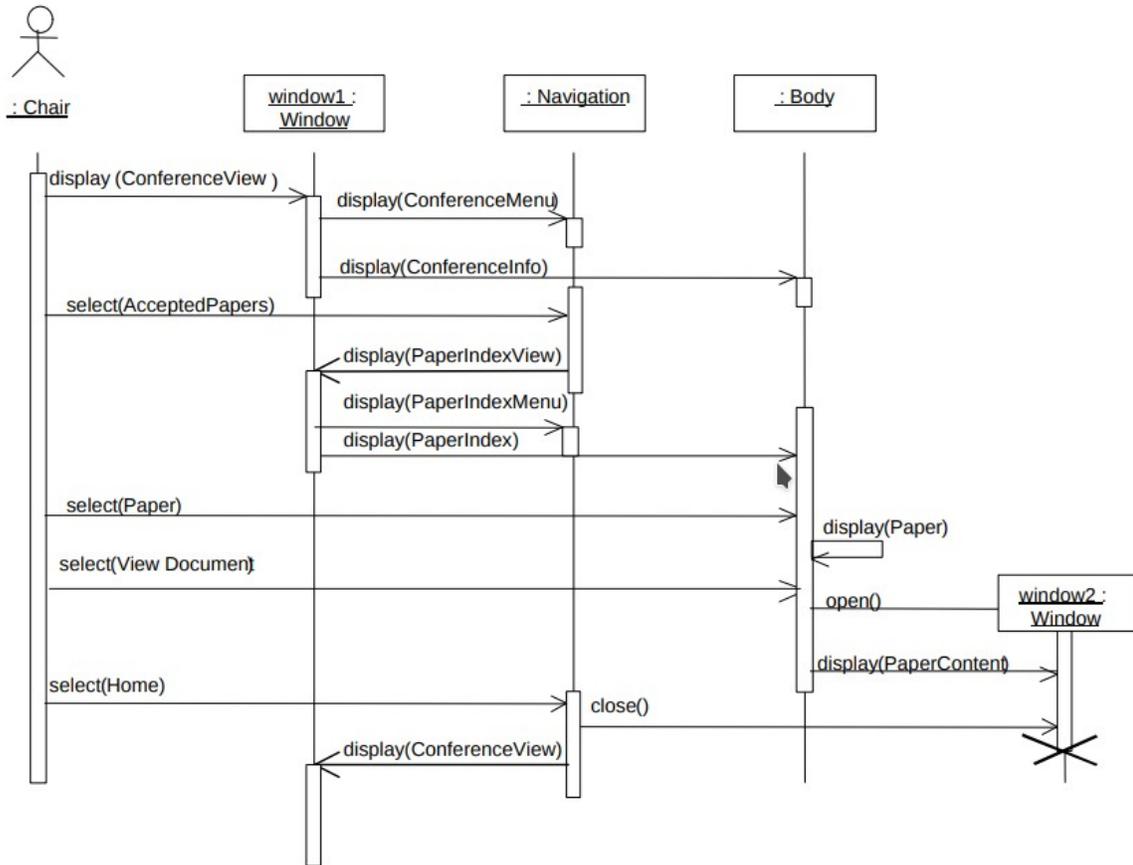
- ▶ Escoger si usar una única ventana o múltiples ventanas.
- ▶ Escoger el estilo del marco.
- ▶ Representar la estructura de presentación con un diagrama de clases

UML.

- ▶ Definir qué modelo navegacional del diagrama de estructura de navegación es empleado.
- ▶ Representar al usuario, ventanas y objetos del marco en la dimensión horizontal.
- ▶ Especificar un mensaje display para cada objeto de la presentación que va mostrando al usuario.
- ▶ Incluir un mensaje select para cada acción del usuario que selecciona un botón.
- ▶ Especificar un mensaje fill y submit para cada acción del usuario en que consiste en suministrar datos en formato de consulta.
- ▶ Incluir un mensaje cada vez que se abra o cierre una ventana.
- ▶ Especificar el periodo de tiempo que una ventana o frame es activo.

A continuación se mostrará un flujo de presentación que representa un escenario para una secuencia de posible de navegación actividades que puede realizar el presidente de una conferencia.(Koch, Kraus y Hennicker, 2002)

Figura 2.13: Vista del flujo de presentación que puede hacer un usuario.



Fuente: (Nora Koch, 2001, pág. 29)

2.18.2 Arquitectura del Software

Una arquitectura de software se entiende como la estructura de un sistema, que contiene componentes, sus propiedades visibles a otros componentes y las relaciones entre ellos

Una arquitectura de software define la estructura básica del sistema (p. Ej., Los módulos de nivel que comprenden las funciones principales del sistema, la gestión y distribución de datos, el tipo y estilo de su interfaz de usuario, qué plataforma(s) ejecutará y así sucesivamente).(Hohmann, 2001, pág. 72)

La base de una solución ganadora y sólida reside en la arquitectura que la crea y la sostiene.

La arquitectura de software es la suma de los módulos, procesos y datos no triviales de un sistema, con estructura y relaciones exactas entre sí, cómo pueden ser y cómo son se espera que se extienda y modifique, como también de qué tecnologías dependen de esta, desde cuál puede deducir las capacidades y flexibilidades exactas del sistema, y de cuál puede formar un plan para la implementación o modificación del sistema .

2.18.2.1 Porque la Arquitectura de Software es Importante

La arquitectura de software es importante porque una buena arquitectura de software es un elemento clave de su largo éxito. Para ilustrar, estas son algunas de las formas en que la arquitectura influye en su éxito. No todos los siguientes son igualmente importantes. Lo más importante depende de su contexto. Todo ellos están relacionados con su arquitectura.

- ▶ **Longevidad:** La mayoría de las arquitecturas viven mucho más que los equipos que las crearon. Estimaciones de el sistema o la longevidad arquitectónica van de 12 a 30+ años, mientras que la longevidad del desarrollador varía de 2 a 4 años.
- ▶ **Estabilidad:** La estabilidad arquitectónica ayuda a garantizar que haya una cantidad mínima de re trabajo fundamental ya que la funcionalidad del sistema se extiende a lo largo de múltiples ciclos de lanzamiento, reduciendo los costos totales de implementación. La estabilidad arquitectónica proporciona una base importante para el equipo de desarrollo en lugar de trabajar en algo que cambia constantemente, el equipo puede concentrar sus esfuerzos en hacer los cambios que brinden mayor valor.
- ▶ **Grado y naturaleza del cambio:** La arquitectura determina la naturaleza del cambio dentro del sistema. Algunos cambios se perciben

como fáciles mientras que otros se perciben como difíciles cuando es “fácil” se correccional fuertemente con el conjunto de cambios “deseados” que mejoran la satisfacción del cliente o permiten para agregar características que atraen a nuevos clientes, generalmente nos referimos a la arquitectura como “buena” arquitectura.

- ▶ **Rentabilidad:** Una buena arquitectura es una arquitectura rentable. Por “rentable”, quiere decir que la compañía que ha creado la arquitectura puede sostenerla con una estructura de costos aceptables, si los costos asociados al mantenimiento de la arquitectura se vuelven demasiado altos, se abandonara.
- ▶ **Estructura social:** Una buena arquitectura funcionara para el equipo que la creó. Aprovecha sus fortalezas y puede a veces minimizar sus debilidades.
- ▶ **Los límites se definen a través de la arquitectura:** Durante el proceso de diseño arquitectónico el equipo toma numerosas decisiones sobre lo que está dentro o fuera de su sistema estos límites y la forma en que se crean y gestionan es vital para el éxito final de la arquitectura.
- ▶ **Ventaja justa y sostenible:** Una gran arquitectura proporciona una técnica sofisticada, difícil de duplicar siendo una ventaja competitiva sostenible. Una buena arquitectura ayudará a mantener o extender fácilmente y rápidamente un sistema.

2.18.2.2 Principios de la arquitectura del software

La creación, maduración, evolución y mantenimiento de una arquitectura se guía por el conjunto de opciones de diseño realizado por arquitectos y desarrolladores. Todas las opciones de diseño no son iguales. Algunos son francamente triviales y no tienen relación con la arquitectura. Otros son bastante profundos todos, todos los desarrolladores de software y los

arquitectos en particular deben perfeccionar su capacidad para reconocer mejores alternativas de diseño esto significa que por supuesto debemos tener un conjunto de criterios para evaluar alternativas de diseño y que debemos aplicar lo mejor que podamos. Los siguientes principios han resistido las pruebas de un buen diseño arquitectónico:

- ▶ **El principio de encapsulación:** La arquitectura se organiza en forma separada y relativamente en piezas independientes que ocultan detalles de implementación internos entre si.
- ▶ **El principio de interfaces:** Las formas en que los subsistemas dentro de un diseño más grande interactúan son claramente definidos idealmente, estas interacciones se especifican de tal manera que puedan permanecer relativamente estable durante la vida útil del sistema. Una forma de lograr esto es crear abstracciones en programación, esta permite una mayor variabilidad a medida que la implementación necesita cambios.
- ▶ **El principio de acoplamiento:** El acoplamiento se refiere al grado de interconexión entre diferentes piezas en un sistema. En general las piezas sueltas son más fáciles de entender, probar, reutilizar y mantener, porque estas operaciones se pueden hacer de forma aislada de otras partes del sistema.
- ▶ **El principio de granularidad apropiada:** Se refiere al nivel de trabajo realizado por un componente. Si bien los componentes sueltos pueden ser fáciles de entender, probar, reutilizar y mantener de forma aislada, cuando se crean con una granularidad demasiado fina creando soluciones utilizar estos componentes pueden ser más difíciles porque hay que unir muchos de ellos para lograr un trabajo significativo.
- ▶ **El principio de alta cohesión:** La cohesión describe que tan cerca están las actividades dentro de una sola pieza (componente) o entre un

grupo de piezas están relacionadas entre si. Un altamente cohesivo componente significa que sus elementos se relacionan fuertemente entre sí. Damos las mejores notas para cohesión a aquellos componentes cuyos elementos contribuyen a una sola tarea.

- ▶ **El principio de parametrización:** Aunque los componentes están encapsulados, esto no significa que realizan su trabajo sin algún tipo de parametrización o instrumentación. Los componentes efectivos realizan una cantidad adecuada de trabajo con el número y tipo de parámetros que permiten a su usuario ajustar su funcionamiento.
- ▶ **El principio de aplazamiento:** Muchas veces el equipo de desarrollo se enfrenta a una decisión difícil que no se puede hacer con certeza. A veces la decisión no se puede tomar fácilmente debido a razones técnicas como cuando el equipo está tratando de elegir una biblioteca para realizar una función específica y no están seguros de que proveedor puede proporcionar el mejor rendimiento.

2.18.2.3 El Ciclo de Vida de la Arquitectura

Según Wikipedia (2005) en su página web “Arquitectura de software” refiere lo siguiente:

Dentro de un proyecto de desarrollo e independientemente de la tecnología que se utilice, se puede hablar de desarrollo de la arquitectura de software. Este desarrollo que precede a la construcción del sistema está dividido en las siguientes etapas: requerimientos, diseño, documentación y evaluación. Cabe señalar que las actividades relacionadas con el desarrollo de la arquitectura de software generalmente forman parte de las actividades definidas dentro de las metodologías de desarrollo.

A continuación se describen dichas etapas.

- ▶ **Requerimientos:** La etapa de requerimientos se enfoca en la captura, documentación y priorización de requerimientos que influyen a la arquitectura. Como se mencionó anteriormente los atributos de calidad juegan un papel preponderante dentro de estos requerimientos, así que esta etapa hace énfasis en ellos. Sin embargo también son relevantes para la arquitectura los requerimientos funcionales primarios y las restricciones.
- ▶ **Diseño:** La etapa de diseño es la etapa central en la relación con la arquitectura y probablemente la más compleja. Durante esta etapa se definen las estructuras que componen la arquitectura. La creación de estas estructuras se hace en base a patrones de diseño, tácticas y requerimientos que influyen a la arquitectura y no simplemente incorporar diversas tecnologías que están de moda.
- ▶ **Documentación:** Una vez creado el diseño de la arquitectura es necesario poder comunicarlo a otros involucrados dentro del desarrollo. La comunicación exitosa del diseño muchas veces depende de que dicho diseño sea documentado de forma apropiada. La documentación de una arquitectura involucra la representación de varias de sus estructuras que son representadas a través de distintas vistas. Una vista generalmente contiene un diagrama, además de información adicional que apoya en la comprensión de dicho diagrama.
- ▶ **Evaluación:** Dado que la arquitectura de software juega un papel crítico en el desarrollo es conveniente evaluar el diseño una vez que este ha sido documentado, con el fin de identificar posibles problemas y riesgos. La ventaja de evaluar el diseño es que es una actividad que se puede realizar de manera temprana (aún antes de codificar) y que el costo de corrección de los defectos identificados a través de la evaluación es mucho menor al costo que tendría el corregir estos

defectos una vez que el sistema ha sido construido. ([Arquitectura de software, párr. 1](#))

2.18.2.4 Patrón Arquitectónico

Según Cori W. (2018) en el sitio web “Los 10 patrones comunes de arquitectura de software” refiere lo siguiente:

Un patrón arquitectónico es una solución general y reutilizable a un problema común en la arquitectura de software dentro de un contexto dado. Los patrones arquitectónicos son similares al patrón de diseño de software pero tienen un alcance más amplio. (¿Qué es un patrón arquitectónico?, párr. 1)

Wikipedia (2005) en su página web “Arquitectura de software” refiere lo siguiente:

Generalmente no es necesario inventar una nueva arquitectura de software para cada sistema de información. Lo habitual es adoptar una arquitectura conocida en función de sus ventajas e inconvenientes para cada caso en concreto.

Así las arquitecturas más universales son:

- ▶ **Descomposición Modular:** Donde el software se estructura en grupos funcionales muy acoplados.
- ▶ **Cliente – Servidor:** Donde el software reparte su carga de cómputo en dos partes independientes pero sin reparto claro de funciones.
- ▶ **Arquitectura de tres niveles:** Especialización de la arquitectura cliente-servidor donde la carga se divide en tres partes o capas con un reparto claro de funciones: una capa para la presentación (interfaz de usuario), otra para el cálculo (donde se encuentra modelado el negocio) y otra para el almacenamiento (persistencia). Una capa solamente tiene relación con la siguiente. (Arquitecturas más comunes, párr. 1)

2.18.3 Arquitectura Cliente – Servidor

La estructura cliente/servidor es una arquitectura de computación en la que se consigue un procesamiento cooperativo de la información por medio de un conjunto de procesadores, de tal forma que uno o varios clientes, distribuidos geográficamente o no, solicitan servicios de computación a uno o más servidores.

De esta forma y gracia a esta arquitectura, la totalidad de los procesadores, clientes y servidores, trabajan de forma cooperativa para realizar un determinado tratamiento de la información.

Atendiendo a esta división descentralizada, la arquitectura cliente/servidor consiste en una arquitectura distribuida de computación, en la que las tareas de cómputo se reparten entre distintos procesadores, obteniendo los usuarios finales el resultado final de forma transparente, con independencia del número de equipos (servidores) que han intervenido en el tratamiento. Se puede decir por tanto que la arquitectura cliente/servidor es un tipo de arquitectura distribuida, posiblemente la más extendida.

2.18.3.1 Elementos que forman parte de una arquitectura Cliente Servidor

Un sistema cliente/servidor es un sistema de información distribuido basado en las siguientes características:

- ▶ **Servicio:** Unidad básica de diseño. El servidor los proporciona y el cliente los utiliza.
- ▶ **Recursos Compartidos:** Muchos clientes utilizan los mismos servidores y a través de ellos comparten tanto recursos lógicos como físicos.
- ▶ **Protocolos Asimétricos:** Los clientes inician “conversaciones”. Los servidores esperan su establecimiento pasivamente.
- ▶ **Transparencia de localización física de los servidores y clientes:** El

cliente no tiene porque saber donde se encuentra situado el recurso que desea utilizar.

- ▶ **Independencia:** Se tiene independencia de la plataforma hardware y software que se emplee.
- ▶ **Sistemas débilmente acoplados:** Interacción basada en envío de mensajes.
- ▶ **Encapsulamiento de servicios:** Los detalles de la implementación de un servicio son transparentes al cliente.
- ▶ **Escalabilidad horizontal:** Ampliación de la cantidad de servidores.
- ▶ **Integridad:** Datos y programas centralizados en servidores facilitan su integridad y mantenimiento.

En el modelo usual cliente/servidor, un servidor (daemon en la terminología sajona basada en sistemas UNIX/LINUX, traducido como “demonio”) se activa y espera las solicitudes de los clientes.

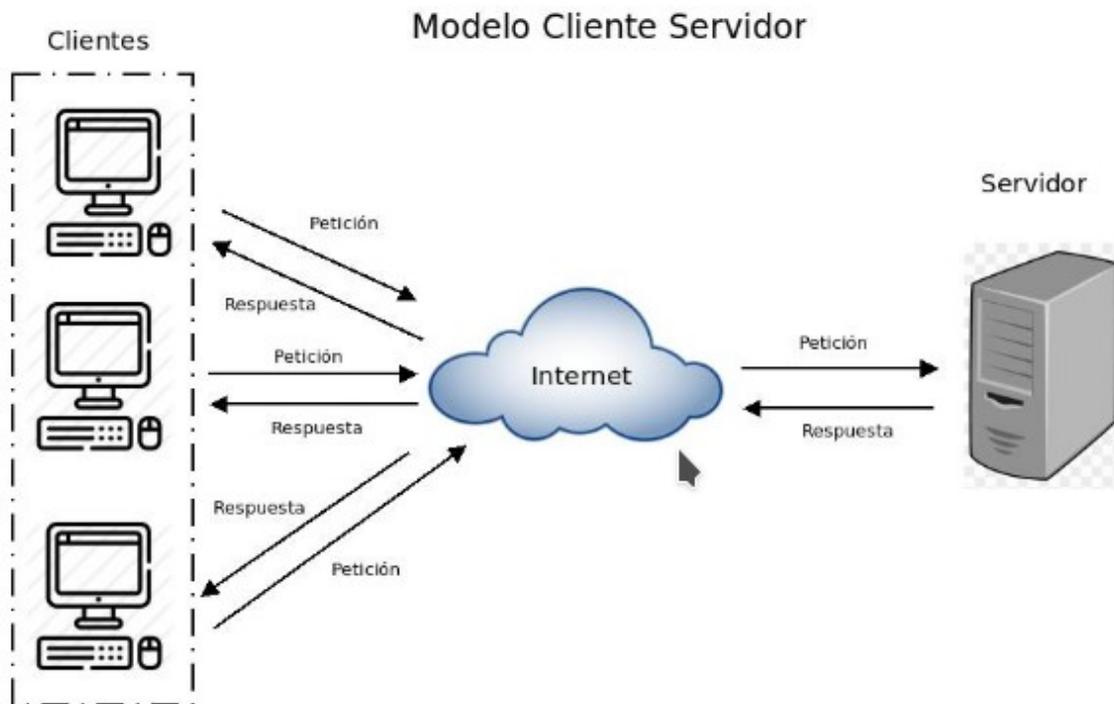
Lo normal es que los servicios de un mismo servidor puedan ser utilizados por múltiples clientes distintos. Tanto los programas cliente como los servidores son con frecuencia parte de un programa o aplicación.

2.18.3.2 Esquema del Funcionamiento Cliente – Servidor

El esquema de funcionamiento del un sistema cliente/servidor sería el siguiente:

1. El cliente solicita una información al servidor.
2. El servidor recibe la petición del cliente.
3. El servidor procesa dicha solicitud.
4. El servidor envía un resultado obtenido al cliente.
5. El cliente recibe el resultado y lo procesa.

Figura 2.14: Esquema de Modelo Cliente – Servidor



Fuente: (oposicionestic, 2016, fig. 1)

2.18.3.3 Componentes de la Arquitectura Cliente – Servidor

El modelo cliente/servidor es un modelo basado en la idea del servicio, en el que el cliente es un proceso consumidor de servicios y el servidor es un proceso proveedor de servicios. Además esta relación está establecida en función del intercambio de mensajes que es el único elemento de acoplamiento entre ambos.

De estas líneas se deducen los tres elementos fundamentales sobre los cuales se desarrollan e implantan los sistemas Cliente/Servidor: el proceso cliente que es quien inicia el diálogo, el proceso servidor que pasivamente espera a que lleguen peticiones de servicio y el middleware que corresponde a la interfaz que provee la conectividad entre cliente y el servidor para poder

intercambiar mensajes.

Para entender en forma más ordenada y clara los conceptos y elementos involucrados en esta tecnología se puede aplicar una descomposición o arquitectura de niveles. Esta descomposición principalmente consiste en separar los elementos estructurales de esta tecnología en función de aspectos más funcionales de la misma:

- ▶ **Nivel de Presentación:** Agrupa a todos los elementos asociados al componente Cliente.
- ▶ **Nivel de Aplicación:** Agrupa a todos los elementos asociados al componente Servidor.
- ▶ **Nivel de Comunicación:** Agrupa a todos los elementos que hacen posible la comunicación entre los componentes Cliente/Servidor.
- ▶ **Nivel de Base de Datos:** Agrupa a todas las actividades asociadas al acceso de los datos.

2.18.3.4 Elementos Principales

2.18.3.4.1 Cliente

Un cliente es todo proceso que reclama servicios de otro. Una definición un poco más elaborada podría ser la siguiente: cliente es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor. Se lo conoce con el término de front-end.

Este normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario (GUI), además de acceder a los servicios distribuidos en cualquier parte de la red. Las funciones que lleva a cabo el proceso cliente se resumen en los siguientes pasos:

- ▶ Administrar la interfaz de usuario.
- ▶ Interactuar con el usuario.
- ▶ Procesar la lógica de la aplicación y hacer validaciones locales.
- ▶ Generar requerimientos de bases de datos.
- ▶ Recibir resultados del servidor.
- ▶ Formatear resultados.

La funcionalidad del proceso cliente marca la operativa de la aplicación (flujo de información o lógica de negocio). De este modo el cliente se puede clasificar en:

- ▶ **Cliente basado en aplicación de usuario:** Si los datos son de baja interacción y están fuertemente relacionados con la actividad de los usuarios de esos clientes.
- ▶ **Cliente basado en lógica de negocio:** Toma datos suministrados por el usuario y/o la base de datos y efectúa los cálculos necesarios según los requerimientos del usuario.

2.18.3.4.2 Servidor

Un servidor es todo proceso que proporciona un servicio a otros. Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Al proceso servidor se lo conoce con el término back-end. El servidor normalmente maneja todas las funciones relacionadas con la mayoría de las reglas de negocio y los recursos de datos. Las principales funciones que lleva a cabo el proceso servidor se muestran a continuación:

- ▶ Aceptar los requerimientos de bases de datos que hacen los clientes.
- ▶ Procesar requerimientos de bases de datos.
- ▶ Formatear datos para transmitirlos a los clientes.

- ▶ Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.

Puede darse el caso que un servidor actúe a su vez como cliente de otro servidor. Existen numerosos tipos de servidores, cada uno de los cuales da lugar a un tipo de arquitectura Cliente/Servidor diferente.

El término "servidor" se suele utilizar también para designar el hardware, de gran potencia, capacidad y prestaciones, utilizado para albergar servicios que atienden a un gran número de usuarios concurrentes. Desde el punto de vista de la arquitectura cliente/servidor y del procesamiento cooperativo un servidor es un servicio software que atiende las peticiones de procesos software clientes.

2.18.3.4.3 Middleware

OposicionesTic (2011) en el sitio web "Arquitectura Cliente/Servidor" refiere lo siguiente:

El middleware es un módulo intermedio que actúa como conductor entre sistemas permitiendo a cualquier usuario de sistemas de información comunicarse con varias fuentes de información que se encuentran conectadas por una red.

En el caso que nos concierne, es el intermediario entre el cliente y el servidor

y se ejecuta en ambas partes.

La utilización del middleware permite desarrollar aplicaciones en arquitectura Cliente/Servidor independizando los servidores y clientes, facilitando la interrelación entre ellos y evitando dependencias de tecnologías propietarias. El concepto de middleware no es un concepto nuevo. Los primeros * monitores de teleproceso* de los grandes sistemas basados en tecnología Cliente/Servidor ya se basaban en él, pero es con el

nacimiento de la tecnología basada en sistemas abiertos cuando el concepto de middleware toma su máxima importancia. El middleware se estructura en tres niveles:

- ▶ Protocolo de transporte.
- ▶ Network Operating System (NOS).
- ▶ Protocolo específico del servicio.
- ▶ Las principales características de un middleware son:
 - ▶ Simplifica el proceso de desarrollo de aplicaciones al independizar los entornos propietarios.
 - ▶ Permite la interconectividad de los Sistemas de Información del Organismo.
 - ▶ Proporciona mayor control del negocio al poder contar con información procedente de distintas plataformas sobre el mismo soporte.
 - ▶ Facilita el desarrollo de sistemas complejos con diferentes tecnologías y arquitecturas.

Dentro de los inconvenientes más importantes destacan la mayor carga de máquina necesaria para que puedan funcionar. [\(Elementos que forman parte de la arquitectura cliente-servidor, párr. 1\)](#)

2.18.4 Seguridad de Aplicaciones Web

En la actualidad el crecimiento de internet ha impactado directamente en la seguridad de la información manejada cotidianamente. Sitios de comercio electrónico, servicios, bancos e incluso redes sociales contienen información sensible que en la mayoría de los casos resulta ser muy importante.

Se puede decir que uno de los puntos más críticos de la seguridad en Internet son las herramientas que interactúan de forma directa con los usuarios, en este caso los servidores web. Es común escuchar sobre fallas en los sistemas

de protección de los servidores de aplicaciones. Sin embargo la mayoría de los problemas detectados en servicios web no son provocados por fallas de ninguna de estas partes, si no que los problemas se generan por malas prácticas de parte de los desarrolladores.

Debemos entender que programar aplicaciones web seguras no es una tarea fácil ya que requiere por parte del programador, no solo cumplir con el objetivo funcional básico de la aplicación sino una concepción general de los riesgos que puede correr la información procesada por el sistema.

2.18.4.1 Problemas Principales en la Programación de Sistemas Web

Gran parte de los problemas de seguridad en las aplicaciones web son causados por la falta de seguimiento en dos rubros muy importantes de los que depende cualquier aplicación, las entradas y salidas del sistema.

Figura 2.15: Entrada y salida del sistema.



Fuente: (Seguridad UNAN, 2016)

Además de verificar estos dos rubros, es importante considerar la exposición accidental de datos que pueden ser empleados en un posible ataque sobre el sistema. Los mensajes de error enviados por el servidor, que suelen ser de gran utilidad durante el proceso de desarrollo de la aplicación, pueden ser empleados maliciosamente cuando siguen apareciendo en un entorno de producción, por lo que es necesario deshabilitar todos estos mensajes y editar algunos otros (como los que se envían cuando el servidor no encuentra algún

archivo en particular) los cuales también pueden ser utilizados por los atacantes para obtener información sobre nuestro sistema.

2.18.4.2 Balancear Riesgo y Usabilidad

Si bien la usabilidad y la seguridad en una aplicación web no son excluyentes una de la otra, alguna medida tomada para incrementar la seguridad con frecuencia afecta la usabilidad. Normalmente siempre se debe pensar en las maneras en que usuarios ilegítimos nos pueden atacar y la facilidad de uso para los usuarios legítimos.

Figura 2.16: Login como medida de seguridad



Fuente: (Seguridad UNAN, 2016)

2.18.4.3 Rastrear el paso de los Datos

Es muy importante mantener conocimiento de los pasos que ha recorrido la información en todo momento. Conocer de dónde vienen los datos y hacia dónde van. En muchas ocasiones lograr esto puede ser complicado, especialmente sin un conocimiento profundo de cómo funcionan las aplicaciones web.

En las aplicaciones web, existen maneras de distinguir los orígenes de los datos y poder reconocer cuando los datos pueden ser dignos de confianza y

cuando no.

Normalmente existen arreglos globales en la aplicación (por ejemplo en PHP los arreglos `$_GET`, `$_POST`, `$_COOKIE` y `$_SESSION` entre otros) que sirven para identificar de forma clara las entradas proporcionadas por el usuario si esto lo combinamos con una convención estricta para el nombrado de las variables podemos tener un control sobre el origen de los datos usados en el código.

Además de entender los orígenes de la información se debe dar la misma importancia a entender cuales son las salidas que tienen la aplicación y hacia a donde se devuelven los resultados.

2.18.4.4 Filtrar Entradas

El filtrado es una de las piedras angulares de la seguridad en aplicaciones web. Es el proceso por el cual se aprueba la validez de los datos, Si nos aseguramos que los datos son filtrados apropiadamente al entrar, podemos eliminar el riesgo de que datos contaminados sean usados para provocar funcionamientos no deseados en la aplicación.

Existen muchos puntos de vista diferentes sobre cómo realizar el filtrado o proceso de limpieza. Lo que usualmente se recomienda es ver al filtrado como un proceso de inspección, no debemos tratar de corregir los datos, es mejor forzar a los usuarios a jugar con las reglas válidas.

Al usar listan blancas asumimos que los datos son inválidos a menos que prueben ser válidos al encontrarse patrones coincidentes. Una limitante de usar este punto de vista es considerar inválidos datos que debieron considerarse válidos pero que no fueron tomados en cuenta patrones similares al construir la lista blanca.

Si llegamos a utilizar algún framework se debe tener especial cuidado, ya que estos brindan tantas comodidades que muchos desarrolladores inexpertos los

utilizan sin preocuparse en entender el código que están observando y por lo tanto implementan medidas de validación en entradas, variables, entre otros, sin entender exactamente el funcionamiento de la solución empleada.

Es importante notar que en los lenguajes de programación existen una buena cantidad de filtros pero evidentemente estos no llegan a cubrir todas las necesidades que puede tener un desarrollador. En este caso, se llegan a utilizar funciones creadas y adaptadas a nuestras necesidades a modo de filtro especial, en la mayoría de estos casos es donde se puede encontrar el uso de expresiones regulares.

En la figura tenemos un ejemplo en PHP de variables GET sanitizadas mediante una función la cual valida que únicamente sean números enteros (FILTER_SANITIZE_NUMBER_INT) y no contengan etiquetas HTML (strip_tags).

Figura 2.17: Filtrado avanzado usando funciones PHP.

```
$idsol=filter_var(strip_tags(isset($_GET['idsol'])?$_GET['idsol']:0),FILTER_SANITIZE_NUMBER_INT);
```

Fuente: (Seguridad UNAN, 2016)

Una vez concluido el paso del filtrado solo resta usar convenciones apropiadas en el nombramiento de las variables para poder distinguir las que ya han sido filtradas.

2.18.4.5 Escapado de Salidas

Otro punto importante de la seguridad es el proceso de escapado y su contraparte para codificar o decodificar caracteres especiales de tal forma que su significado original sea preservado. Si llegamos a utilizar una codificación en particular es necesario conocer los caracteres reservados los cuales serán necesarios escapar.

El proceso de escapado debe estar compuesto a su vez por los siguientes

pasos:

- ▶ Identificar las salidas.
- ▶ Escapar la salidas.
- ▶ Distinguir entre datos escapados y no escapados.

El proceso de escapado debe adecuarse al tipo de salida de que se trate (si es al cliente, a la base de datos, etc). Para la mayoría de los destinatarios, existen funciones nativas en los lenguajes de programación para esta finalidad. En alguno de los casos como podría ser el de base de datos es importante incluso observar la codificación en la que son enviados.

2.18.5 Calidad de Software

La calidad de software es un problema actual que afecta tanto a los productores de software como a los clientes. Con el aumento de la informatización a escala mundial la demanda de software crece exponencialmente y los desarrolladores le han brindado poco interés a la calidad de sus productos. Sucede que muchas veces los clientes reciben el software cuando se han violado las etapas de pruebas.

Obteniendo la calidad requerida en el software, se logra reducir su número de errores o eliminarlos completamente, se alcanza una mayor fiabilidad para las funciones que debe realizar el mismo, mayor eficiencia e integridad de los datos así como flexibilidad y reusabilidad.

2.18.5.1 Pruebas de Software

La prueba es una actividad fundamental en muchos procesos de desarrollo, incluyendo el del software. Estas permiten detectar la presencia de errores que pudieran generar las entradas o salidas de datos y comportamientos inapropiados durante su ejecución. Un concepto más específico puede ser:

“La prueba del software es un elemento crítico para la garantía de calidad del

software y representa una revisión de las especificaciones, del diseño y de la codificación” (Pressman, 1998)

Teniendo en cuenta las definiciones anteriores se puede concluir que la prueba de software es una actividad en la cual el sistema es ejecutado bajo condiciones específicas para demostrar que no tiene la madurez necesaria para ser implantado. Dentro de las actividades que se practican para obtener un software con la madurez necesaria están:

- ▶ Revisiones: consiste en que cada integrante del equipo de desarrollo revisa el producto que va generando.
- ▶ Inspecciones: revisión de cada producto por parte de colegas.
- ▶ Validaciones: es el cliente quien revisa el producto para decir si cumple con sus necesidades.

Esta definición implica que se considera una prueba exitosa si se demuestran deficiencias en el software. Las fallas pueden ser en el código o en el modelado, en dependencia del tipo de pruebas que se le apliquen al software.

2.18.5.2 Métodos de Pruebas de Software

Existen diversos métodos para realizar las pruebas de software, entre las más importantes se encuentran la prueba de Caja Blanca, prueba de Caja Negra y prueba de la Estructura de Control.

El uso de la prueba de Caja Blanca es mejor para verificar que se recorran todos los caminos y detectar un mayor número de errores. La Caja Negra brinda la posibilidad de cubrir la mayor parte de las combinaciones de entradas y lograr así un juego de pruebas más eficaz.

Las pruebas mencionadas permiten probar cada una de las condiciones existentes en el programa, identificar claramente las entradas, salidas y estudiar las relaciones que existen entre ellas, permitiendo así maximizar la calidad de las pruebas y en dependencia del resultado se contará con un

sistema más estable y confiable.

2.18.5.2.1 Prueba de Especificación (Caja Negra)

También suelen ser llamadas funcionales y basadas en especificaciones. En ellas se pretende examinar el programa en busca de que cuente con las funcionalidades que debe tener y cómo lleva a cabo las mismas, analizando siempre los resultados que devuelve y probando todas las entradas en sus valores válidos e inválidos.

Al ejecutar las pruebas de Caja Negra se desarrollan casos de prueba reales para cada condición o combinación de condiciones y se analizan los resultados que arroja el sistema para cada uno de los casos. En esta estrategia se verifica el programa considerándolo una caja negra. Las pruebas no se hacen en base al código, sino a la interfaz. No importa que se cubran todas las rutas dentro del programa, lo importante es probar todas las entradas en sus valores válidos e inválidos y lograr que el sistema tenga una interfaz amigable.

2.18.5.2.1.1 Limitaciones

Lograr una buena cobertura con pruebas de caja negra es un objetivo deseable; pero no suficiente a todos los efectos. Un programa puede pasar con holgura millones de pruebas de especificación y sin embargo tener defectos internos que surgen en el momento más inoportuno.

Por ejemplo, una computadora que contenga el virus Viernes-13 puede estar pasando pruebas de caja negra durante años y años. Sólo falla si es viernes y es día 13; pero ¿a quién se le iba a ocurrir hacer esa prueba?

Las pruebas de caja negra nos convencen de que un programa realizar bien sus funcionalidades programadas, pero no de que haga (además) otras cosas menos aceptables.

2.18.5.2.2 Prueba de Código (Caja Blanca)

También suelen ser llamadas estructurales o de cobertura lógica. En ellas se pretende investigar sobre la estructura interna del código, exceptuando detalles referidos a datos de entrada o salida, para probar la lógica del programa desde el punto de vista algorítmico. Realizan un seguimiento del código fuente según se va ejecutando los casos de prueba, determinando de manera concreta las instrucciones, bloques, etc. que han sido ejecutados por los casos de prueba.

En las pruebas de Caja Blanca se desarrollan casos de prueba que produzcan la ejecución de cada posible ruta del programa o módulo, considerándose una ruta como una combinación específica de condiciones manejadas por un programa.

Hay que señalar que no todos los errores de software se pueden descubrir verificando todas las rutas de un programa, hay errores que se descubren al integrar unidades del sistema y pueden existir errores que no tengan relación con el código específicamente.

2.18.5.2.2.1 Características de las pruebas de Caja Blanca

En las pruebas de Caja Blanca, se pretende indagar sobre la estructura interna del código, omitiendo detalles referidos a datos de entrada o salida. Su objetivo principal es probar la lógica del programa desde el punto de vista algorítmico.

Estas se basan en el diseño de Casos de Prueba que usa la estructura de control del diseño procedimental para derivarlos. Mediante las pruebas de Caja Blanca el ingeniero de software puede obtener Casos de Prueba que:

- ▶ Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
- ▶ Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.

- ▶ Ejecuten todos los bucles en sus límites operacionales.
- ▶ Ejerciten las estructuras internas de datos para asegurar su validez.

Las pruebas de Caja Blanca son consideradas entre las más importantes que se aplican a los sistemas, con la que se obtienen como resultados la disminución en un gran por ciento el número de errores existentes en el software y por ende una mayor calidad y confiabilidad en la codificación.

2.18.5.2.2.2 Tipos de pruebas de Caja Blanca

Salazar E. (2015) en el sitio web “Procedimiento para realizar pruebas de Caja Blanca” refiere lo siguiente:

De estructura de datos locales:

Se centran en el estudio de las variables del programa. Busca que toda variable este declarada y que no existan con el mismo nombre, ni declaradas local y globalmente, que haya referencias a todas las variables y para cada variable, analiza su comportamiento en comparaciones.

De cobertura lógica:

- ▶ **De Cobertura de Sentencias:** Comprueba que todas las sentencias se ejecuten al menos una vez.
- ▶ **De Cobertura de Decisión:** Ejecuta casos de prueba de modo que cada decisión se pruebe al menos una vez a Verdadero (True) y otra a Falso (False).
- ▶ **De Cobertura de Condición:** Ejecuta un caso de prueba a True y otro a False por cada condición, teniendo en cuenta que una decisión puede estar formada por varias condiciones.
- ▶ **De Cobertura de Condición/Decisión:** Se realizan las pruebas de cobertura de condición y las de decisión a la vez.
- ▶ **De Condición Múltiple:** Cada decisión multi condición se traduce a

una condición simple, aplicando posteriormente la cobertura de decisión.

- ▶ **De Cobertura de Caminos:** Se escriben casos de prueba suficientes para que se ejecuten todos los caminos de un programa. Entendiéndose camino como una secuencia de sentencias encadenadas desde la entrada del programa hasta su salida. (Tipos de pruebas de Caja Blanca, párr. 1)

2.19 HERRAMIENTAS

Para el desarrollo del sistema se utilizarán las herramientas descritas a continuación:

2.19.1 Lenguaje de Programación Python

Según Alvarez M. (2003) en el sitio web “Qué es Python” refiere lo siguiente:

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad.

En los últimos años el lenguaje se ha hecho muy popular, gracias a varias razones como:

- ▶ La cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero.
- ▶ La sencillez y velocidad con la que se crean los programas. Un programa en Python puede tener de 3 a 5 líneas de código menos que su equivalente en Java o C.
- ▶ La cantidad de plataformas en las que podemos desarrollar, como Unix,

Windows, OS/2, Mac, AmigaOS y otros.

- ▶ Además, Python es gratuito, incluso para propósitos empresariales. ([Qué es Python, párr. 3](#))

2.19.2 Flask

ManualWeb (s. f.) en su sitio web “Introducción a Flask” refiere lo siguiente:

Flask es un Microframework de Python que está basado en Werkzeug, Jinja 2 y buenas intenciones. Mediante Flask podemos construir aplicaciones web y servicios Restful con Python de una forma extraordinariamente sencilla. Con pocas líneas podemos llegar a tener un servicio restful funcionando. (Introducción a Flask, párr. 1)

Según Andrearrs (2014) en el sitio web “Flask: minimalismo para el desarrollo web en Python” refiere lo siguiente:

Flask, a diferencia de Django y Pyramid, no trae cientos de módulos para abordar las tareas más comunes en el desarrollo web, más bien se enfoca en proporcionar lo mínimo necesario para que puedas poner a funcionar una aplicación básica en cuestión de minutos. Es perfecto, por ejemplo, para el prototipado rápido de proyectos.

Incluye un servidor web de desarrollo para que puedas probar tus aplicaciones sin tener que instalar algo como Nginx o Apache. También trae un depurador y soporte integrado para pruebas unitarias. Tiene un excelente soporte para Unicode y es compatible 100% con WSGI 1.0.

Entre las cosas más hermosas que tiene este microframework están el esquema de rutas y la documentación. Con el decorador de rutas puedes hacer que tu aplicación responda a peticiones totalmente restful. (¿Qué tiene de bueno flask?, párr. 1-3)

2.19.3 Base de Datos Relacional Potgresql

HostingPedia (2019) en el sitio web "PostgreSQL" refiere lo siguiente:

PostgreSQL es un gestor de bases de datos relacional y orientado a objetos. Su licencia y desarrollo es de código abierto, siendo mantenida por una comunidad de desarrolladores, colaboradores y organizaciones comerciales de forma libre y desinteresadamente. Esta comunidad es denominada PDGD (*PostgreSQL Global Development Group*, por sus siglas en inglés).

Es reconocido actualmente como uno de los sistemas gestores de bases de datos relacionales más potentes del mercado. Presenta fácil accesibilidad, es multiplataforma y está disponible para su utilización en casi todos los sistemas operativos utilizados en la actualidad sin disminuir su rendimiento.

Inicialmente, el desarrollo del proyecto comenzó a mediados de los años 80's en la Universidad de Berkeley en California. El proyecto estuvo liderado por Michael Stonebraker, junto a un grupo de trabajo de la universidad, luego de su experiencia con un proyecto anterior llamado "Ingres", uno de los primeros desarrollos de motor de bases de datos relacional. PGSQL fue basado con este desarrollo y fue lo que le generó el nombre, por el juego de palabras "*post-Ingres*", adaptándose inicialmente como "Postgres". ([¿Qué es PostgreSQL?, párr. 1-3](#))

2.19.4 Celery

Aaloy (2011) en el sitio web "Introducción a Celery" refiere lo siguiente:

Celery es una aplicación que nos permite crear tareas de trabajo asíncronas gestionadas por un gestor de colas que está basada en el envío de mensajes de manera distribuida. Se focaliza en operaciones en tiempo real pero también soporta la calendarización de tareas, es decir, puede lanzar tareas que se tengan que ejecutar en un momento determinado o de

manera periódica. Las unidades de ejecución, llamadas tareas, se ejecutan de manera concurrente en uno o más nodos de trabajo. Estas tareas pueden ejecutarse de manera asíncrona bien de manera síncrona (esperando hasta que la tarea está lista). El sistema de mensajería recomendado por Celery es RabbitMQ aunque es bastante agnóstico en el tema y puede usar como sustituto Redis, MongoDB o una base de datos sql. Aunque el programa surgió ligado a Django, actualmente es una librería que se puede utilizar de manera independiente. Se han creado dos proyectos a partir del original. Celery, que mantiene la estructura básica de la librería y se puede utilizar desde cualquier aplicación Python y django-celery que se encarga de la integración con Django. (Introducción a Celery, párr. 1)

2.19.5 RabbitMQ

Mesa L. (2019) en el sitio web “Conozcamos sobre RabbitMQ, sus componentes y beneficios” refiere lo siguiente:

RabbitMQ es un broker de mensajería de código abierto, distribuido y escalable, que sirve como intermediario para la comunicación eficiente entre productores y consumidores.

RabbitMQ implementa el protocolo mensajería de capa de aplicación AMQP (Advanced Message Queueing Protocol), el cual está enfocado en la comunicación de mensajes asíncronos con garantía de entrega, a través de confirmaciones de recepción de mensajes desde el broker al productor y desde los consumidores al broker.

En una forma simplificada, en RabbitMQ se definen colas que van a almacenar los mensajes que envían los productores hasta que las aplicaciones consumidoras obtienen el mensaje y lo procesan. Esto nos permite diseñar e implementar sistemas distribuidos, en los cuales un sistema se divide en módulos independientes que se comunican entre sí a

través de mensajes.

Podemos decir entonces que RabbitMQ en su rol de intermediario, tiene como labor asegurarse de que los mensajes que un productor envía, se enrutan al consumidor correcto. (¿Qué es RabbitMQ?, párr. 1-4)

2.19.6 ElasticSearch

Según David Ochobits (2018) en el sitio web “¿Qué es y cómo funciona Elasticsearch?” refiere lo siguiente:

Se trata de un motor de búsqueda y análisis. Es distribuible y fácilmente escalable, enfocado sobre todo al mundo empresarial y científico. Es accesible a través de una extensa y elaborada API. Con esta herramienta podemos impulsar búsquedas extremadamente rápidas que respalden nuestras aplicaciones de descubrimientos de datos.

Está escrito en Java, utilizando la filosofía de desarrollo de código abierto y utiliza una licencia Apache, similar a la de software libre.

Trabaja conjuntamente con un motor de recopilación de datos llamado Logstash, y una plataforma de análisis y visualización llamada Kibana. Los tres productos están diseñados para trabajar de manera conjunta, una solución llamada “Elastic Stack” (anteriormente llamada ELK)

Sus características:

- ▶ Es orientado a documentos: Utiliza JSON.
- ▶ No utiliza esquemas, aunque si son necesarios se pueden llegar a definir.
- ▶ Distribuido: Realiza escalado de manera dinámica e implementa alta disponibilidad (HA).
- ▶ Utiliza una potente API: expone prácticamente todas sus funcionalidades utiliza una API REST.

- ▶ Permite búsquedas tanto estructuras como no estructuradas.

Principales ventajas:

- ▶ **Rápido:** Mediante el uso de índices invertidos distribuidos, Elasticsearch encuentra rápidamente las mejores coincidencias para nuestras búsquedas de texto completo, incluso de conjuntos de datos muy grandes.
- ▶ **API fácil de usar:** Elasticsearch ofrece una API potente, una interfaz HTTP simple además de utilizar documentos JSON sin esquemas, lo que facilita su indexar, buscar y consultar datos.
- ▶ **El poder de tres:** Elasticsearch viene integrado con Kibana, una de las herramientas más populares para la visualización de informes. Se integra con Logstash para transformar los datos utilizando plantillas predefinidas y así cargarlos en un índice.
- ▶ **Actualizaciones de índice en tiempo real:** Las actualizaciones de índice de Elasticsearch, como por ejemplo agregar un nuevo documento al índice, generalmente puede demorar un segundo o incluso menos; este le permite a Elasticsearch utilizarse para fines que necesiten un tiempo muy rápido de respuesta.
- ▶ **Soporte para nuestros lenguajes de desarrollo favoritos:** existen una larga lista de clientes de fuente abierta, para los desarrolladores que deseen utilizar Elasticsearch, como por ejemplo para Java, PHP, Python, JavaScript, Node.js u otros. ([¿Qué es Elasticsearch?, párr. 1-3](#))

2.19.7 Lenguaje de Programación Javascript

Lucas Pelhom (2019) en el sitio web “Qué es JavaScript: primeros pasos” refiere lo siguiente:

JavaScript es uno de los lenguajes de programación más populares para el desarrollo web, ejecutado del lado del cliente (esta expresión significa que

los scripts o programas son ejecutados en el navegador del usuario), y funciona como una extensión del código HTML.

Por ser un lenguaje interpretado, no necesita de compilación y, por lo tanto, no utiliza los recursos del servidor. Javascript es un lenguaje que soporta la programación orientada a objetos. Aunque no posee varias características de esta orientación nativamente, es posible realizar algunas emulaciones.

Es muy importante recordar que JavaScript no es lo mismo que Java, su sintaxis es diferente y necesita un compilador para que el navegador pueda interpretarlo. En cambio, JavaScript tiene su propio *script* interpretado por él mismo.

El *script* del desarrollador será enviado junto al HTML para el navegador, que, al leerlo, hará la diferenciación del código HTML por medio de la tag `<script>`.(Qué es JavaScript: primeros pasos, párr. 2)

Giampiere J. (2016) en el sitio web “¿Qué es JavaScript?” refiere lo siguiente:

El poder de JavaScript está disponible principalmente en lado frontend, agregando mayor interactividad a la web, también puedes usar las librerías y framework como: jquery, angular, backbone, react y demás, escritas sobre JavaScript, y que te ayudan a crear una mejor experiencia de usuario en nuestros sitios web. De igual manera JavaScript se puede utilizar en los servidores web. Node.JS es tu mejor opción para usar este lenguaje del lado del servidor.

Conozcamos ahora las características de JavaScript que hace de este lenguaje, uno de los más populares en la actualidad.

- ▶ Es Liviano.
- ▶ Multiplataforma, ya que se puede utilizar en Windows, Linux o Mac o en el navegador de tu preferencia.
- ▶ Es Imperativo y estructurado, mediante un conjunto de instrucciones

indica al computador qué tarea debe realizar.

- ▶ Prototipado, debido a que usa prototipos en vez de clases para el uso de herencia.
- ▶ Orientado a objetos y eventos.
- ▶ Es Interpretado, no se compila para poder ejecutarse. (¿Qué es JavaScript?, párr. 1-3)

2.19.8 VueJS

García E. (2019) en el sitio web “¿Qué Es Vue.JS?” refiere lo siguiente:

Vue es un framework open source de JavaScript, el cual nos permite construir interfaces de usuarios de una forma muy sencilla. La curva de aprendizaje, desde mi punto de vista, es relativamente baja, claro, debes conocer muy bien JavaScript, saber trabajar con callbacks, promesas, objetos, entre otros temas más.

Si queremos hacer uso de Vue no será necesario instalar absolutamente nada en nuestra computadora, si así lo deseamos podemos utilizar este framework mediante un CDN.

Vue fue creado por Evan You ex trabajador de Google, quien, es importante mencionar, fue desarrollador Angular. Vue fue lanzado en el año 2014. Aunque inicialmente fue pensado para ser un biblioteca personal, la comunidad hizo que el proyecto creciera a un ritmo impresionante, posicionándolo hoy en día como uno de los Frameworks web más populares, junto con Angular y React.

Una de las características más importantes de Vue es el trabajo con componentes. Un componente Vue, en términos simples, es un elemento el cual se encapsula código reutilizable. Dentro de un componente podremos encontrar etiquetas HTML, estilos de CSS y código JavaScript. Los componentes nos permiten desarrollar proyectos modularizados y fáciles de

escalar, si nosotros así lo deseamos podemos reemplazar un componente por otro de una forma muy sencilla, como si de piezas de lego se trataran. (Vue, párr. 1-4)

Según Rosa J. (2017) en el sitio web “¿Qué es Vue.js?” refiere lo siguiente:

Características:

- ▶ **Accesible.**
- ▶ **Versátil:** Su núcleo es bastante pequeño y se escala a través de plugins.
- ▶ **Escalable** por el mismo tema de la versatilidad.
- ▶ **Reactivo.**
- ▶ **Optimizado:** Su core ocupa 74KB, como ves es bastante liviano.
- ▶ **Comunidad:** Va creciendo a un ritmo importante con más 66500 estrellas en GitHub y 130 personas contribuyendo al core cada día.
- ▶ **Licencia MIT:** se publicó bajo el amparo de esta licencia. ([Características, párr. 1](#))

2.19.9 Bootstrap

Casas V. (2019) en el sitio web “¿Qué es Bootstrap y para qué sirve?” refiere lo siguiente:

Bootstrap es uno de los frameworks CSS de código abierto más conocidos y más utilizados en el mundo del desarrollo web.

Este framework facilita enormemente la maquetación de páginas web, ya que te permite crear una interfaz muy limpia y completamente responsive, es decir, adaptable a cualquier tamaño de pantalla. Además, ofrece tantas herramientas y funcionalidades que te permite crear una web desde cero muy fácilmente.

Estas son las seis principales razones por las que Bootstrap es uno de los frameworks más utilizados:

- ▶ **Soporte:** Es una de las características más destacadas desde framework. Existe muchísima documentación en la red sobre su manejo y muchos blogs especializados en ello.
- ▶ **Framework de código abierto:** Esto quiere decir que todo su código está disponible y accesible a través de GitHub.
- ▶ **Compatibilidad:** No vas a tener problemas de incompatibilidad con los principales navegadores (Firefox, Google Chrome, Internet Explorer, Opera o Internet Explorer).
- ▶ **Integración con librerías JavaScript:** Tipografías, botones, cuadros...
- ▶ **Facilidad de uso:** Tanto a la hora de crear contenido como en el momento de subir archivos como imágenes o vídeos.
- ▶ **Utiliza un sistema grid:** Es decir, un sistema de cuadrículas (rejillas) que te permite crear el diseño de una web insertando el contenido en bloques o columnas. (Ventajas de Bootstrap, párr. 1)

2.19.10 Servidor Web Apache

Según Fumas E. (2014) en el sitio web “Apache HTTP Server: ¿Qué es, cómo funciona y para qué sirve?” refiere lo siguiente:

El servidor Apache HTTP , también llamado Apache, es un servidor web HTTP de código abierto para la creación de páginas y servicios web. Es un servidor multiplataforma, gratuito, muy robusto y que destaca por su seguridad y rendimiento.

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Para entender mejor lo que es Apache, primeramente definiremos lo que es

un servidor web. La definición más sencilla de servidor web, que es un programa especialmente diseñado para transferir datos de hipertexto, es decir, páginas web con todos sus elementos (textos, widgets, banners, etc). Estos servidores web utilizan el protocolo http.

Ventajas

- ▶ **Instalación/Configuración:** Software de código abierto.
- ▶ **Coste:** El servidor web Apache es completamente gratuito.
- ▶ **Funcional y Soporte:** Alta aceptación en la red y muy popular, esto hace que muchos programadores de todo el mundo contribuyen constantemente con mejoras, que están disponibles para cualquier persona que use el servidor web y que Apache se actualice constantemente.
- ▶ **Multiplataforma:** Se puede instalar en muchos sistemas operativos, es compatible con Windows, Linux y MacOS.
- ▶ **Rendimiento:** Capacidad de manejar más de un millón de visitas/día.
- ▶ **Soporte de seguridad SSL y TLS.**

Inconvenientes

- ▶ Falta de integración
- ▶ Posee formatos de configuración no estándar.
- ▶ No posee un buen panel de configuración. (¿Qué es Apache HTTP Server?, párr. 1-4)

2.20 MÉTRICAS DE CALIDAD

Barrientos D. (2018) en el sitio web “Métricas de Calidad de Software” refiere lo siguiente:

La calidad es un concepto relativo y multidimensional, referido a las expectativas y cualidades solicitados por el cliente, a su vez, está ligada a restricciones y compromisos (presupuesto y tiempo de desarrollo, entre otros). Sin embargo, existe algo que nadie puede negar, cuando algo es de calidad suele pasar desapercibido, pero, por el contrario, la mala calidad es algo que destaca negativamente. (Métricas de Calidad de Software, párr. 1)

Uno de los problemas que se afrontan en la computación es la calidad de software. Desde la década de los setenta, este tema ha sido motivo de preocupación para especialistas, ingenieros, investigadores y comercializadores de software. La calidad del software es una compleja mezcla de factores que varían a través de diferentes aplicaciones y según el cliente que las pida (Pressman, 2007).

2.20.1 Que es una ISO

BlogSpot (s.f) en el sitio web “¿Qué es la ISO?” refiere lo siguiente:

La Organización Internacional de Normalización o ISO (del griego ἴσος, «isos», que significa «igual»), nacida tras la Segunda Guerra Mundial (23 de febrero de 1947), es el organismo encargado de promover el desarrollo de normas internacionales de fabricación (tanto de productos como de servicios), comercio y comunicación para todas las ramas industriales. Su función principal es la de buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones (públicas o privadas) a nivel internacional. (Que es la ISO, párr. 1)

2.20.2 Métricas de Calidad establecido por el Estándar ISO 9126

BlogSpot (s.f) en el sitio web “¿Qué es la ISO?” refiere lo siguiente:

La ISO 9126 es un estándar internacional para evaluar la calidad del software en base a un conjunto de características y sub-características de la calidad. Cada sub-característica consta de un conjunto de atributos que

son medidos por una serie de métricas.

Estas métricas miden artefactos obtenidos en etapas tardías del desarrollo de software, aumentando el costo de detección y corrección de errores. Por esta razón, en la literatura ha surgido un mayor interés por la definición de métricas que pretenden evaluar una o varias de las características de calidad definidas en el estándar ISO 9126, en etapas tempranas del desarrollo de software.

Esta norma Internacional fue publicado en 1992, la cual es usada para la evaluación de la calidad de software, llamado (Information technology Software product evaluation Quality characteristics and guidelines for their use); o también conocido como ISO 9126 (o ISO/IEC 9126).

El modelo ISO 9126 fue desarrollado por la ISO (organización de estandarización internacional por sus siglas en Ingles ISO) y este es uno de los grandes grupos reconocidos por los estándares aplicados internacionalmente a través de un amplio rango de solicitudes.

A continuación mostramos las características que la ISO-9126 establece para un software.(ISO 9126, párr. 1-4)

Tabla 2.1: Características de la ISO-9126

Características	Preguntas
Funcionalidad	0
Confiabilidad	0
Usabilidad	0
Mantenibilidad	0

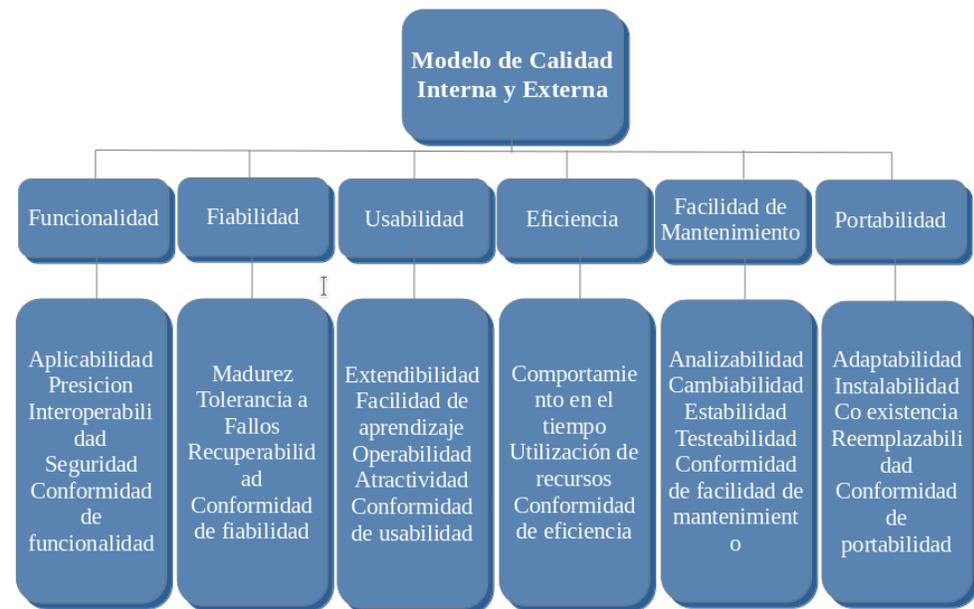
Portabilidad

0

1. **Fuente:** (Abud Figueros, 2005, pág. 1)

2.20.2.1 Modelo de Calidad Interna y Externa

Figura 2.18: Modelo de Calidad Interna y Externa



Fuente: (Abud Figueros, 2005, pág. 12)

BlogSpot (s.f) en el sitio web “¿Qué es la ISO?” refiere lo siguiente:

Específica seis características para calidad interna y externa, las cuales, están subdivididas. Estas divisiones se manifiestan externamente cuando el software es usado como parte de un sistema Informático, y son el resultado de atributos internos de software.

- **Funcionabilidad:** conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen lo indicado o implica

necesidades.

- ♦ **Idoneidad:** Se enfoca a evaluar si el SW cuenta con un conjunto de funciones apropiadas para efectuar las tareas que fueron especificadas en su definición.
 - ♦ **Exactitud:** Permite evaluar si el SW presenta resultados o efectos acordes a las necesidades para las cuales fue creado.
 - ♦ **Interoperabilidad:** Permite evaluar la habilidad del SW de interactuar con otros sistemas previamente especificados.
 - ♦ **Seguridad:** Se refiere a la habilidad de prevenir el acceso no autorizado, ya sea accidental o promediado, a los programas y datos.
 - ♦ **Conformidad:** Evalúa si el SW se adhiere a estándares, convenciones o regulaciones en leyes y prescripciones similares.
- **Confiabilidad:** conjunto de atributos relacionados con la capacidad de mantener un nivel de presentación bajo condiciones establecidas durante un periodo de tiempo establecido.
- ♦ **Madurez:** Permite medir la frecuencia de falla por errores en el SW.
 - ♦ **Recuperación:** Se refiere a la capacidad de restablecer el nivel de operación y recobrar los datos que hayan sido afectados directamente por una falla, así como al tiempo y el esfuerzo necesario para lograrlo.
 - ♦ **Tolerancia de fallos:** Se refiere a la habilidad de mantener un nivel específico de funcionamiento en caso de fallas del SW o de cometer infracciones de su interfaz específica.

- ▶ **Usabilidad:** Conjunto de atributos relacionados con el esfuerzo necesitado para el uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.
 - ◆ **Comprensión:** Se refiere al esfuerzo requerido por los usuarios para reconocer la estructura lógica del sistema y los conceptos relativos a la aplicación del SW.
 - ◆ **Facilidad de Aprender:** Establece atributos del SW relativos al esfuerzo que los usuarios deben hacer para aprender a usar la aplicación.
 - ◆ **Operatividad:** Agrupa los conceptos que evalúan la operación y el control del sistema.

- ▶ **Eficiencia:** Conjunto de atributos relacionados con la relación entre el nivel de desempeño del SW y la cantidad de recursos necesitados bajo condiciones establecidas.
 - ◆ Comportamiento en el tiempo: Atributos del SW relativos a los tiempos de respuesta y de procedimiento de los datos.
 - ◆ Comportamiento de recursos: Atributos de SW relativos a la cantidad de recursos usados y la duración de su uso en la realización de sus funciones.

- ▶ **Mantenibilidad:** Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema SW.
 - ◆ **Estabilidad:** Capacidad del SW de tener un desempeño normal a pesar de hacerse modificaciones.
 - ◆ **Facilidad de análisis:** Relativo al esfuerzo necesario para

diagnosticar las deficiencias o causas de fallas, o para identificar las partes que deberán ser modificadas.

- ◆ **Facilidad de cambios:** Capacidad del que tiene el SW para que la modificación pueda ser válida.
 - ◆ **Facilidad de pruebas:** Capacidad del que tiene el SW para que la modificación pueda ser válida.
- ▶ **Portabilidad:** Conjunto de atributos relacionados con la capacidad de un sistema SW para ser transferido desde una plataforma a otra.
- ◆ **Adaptabilidad:** Evalúa la oportunidad para adaptar el SW a diferentes ambientes sin necesidad de aplicarle modificaciones.
 - ◆ **Facilidad de instalación:** Es el esfuerzo necesario para instalar el SW en un ambiente determinado.
 - ◆ **Cumplimiento:** Permite evaluar si el SW de adhiere a estándares o convenciones relativas a portabilidad.
 - ◆ **Capacidad de reemplazo:** Se refiere a la oportunidad y el esfuerzo usando en sustituir el SW por otro producto con funciones similares.

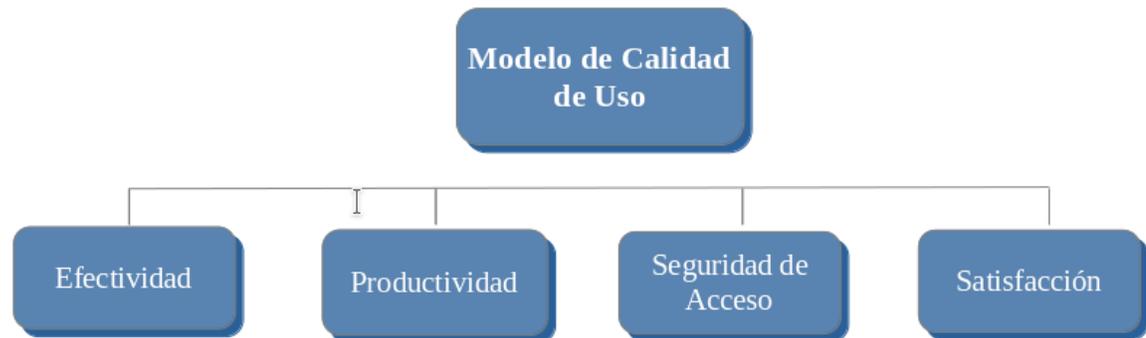
La ISO/IEC 9126 permite especificar y evaluar la calidad del software desde diferentes criterios asociados con adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoría de software. (Características, párr. 1-7)

2.20.2.1 Modelo de Calidad de Uso

BlogSpot (s.f) en el sitio web “¿Qué es la ISO?” refiere lo siguiente:

Calidad en uso es el efecto combinado para el usuario final de las 6 características de la calidad interna y externa del software. Especifica 4 características para la calidad en uso.

Figura 2.19: Modelo de Calidad de Uso



Fuente: (Abud Figueros, 2005, pág. 18)

- ▶ **Efectividad:** Permitir a los usuarios alcanzar objetivos especificados con exactitud y completitud, en un contexto de uso especificado.
- ▶ **Productividad:** Permitir a los usuarios gastar una cantidad adecuada de recursos con relación a la efectividad alcanzada, en un contexto de uso especificado.
- ▶ **Seguridad física:** Alcanzar niveles aceptables del riesgo de hacer daño a personas, al negocio, al software, a las propiedades o al medio ambiente en un contexto de uso especificado.
- ▶ **Satisfacción:** Satisfacer a los usuarios en un contexto de uso especificado.

El estándar 9126 permite a cada organización establecer su propio modelo de calidad en función de las características del software que se quiera evaluar.(Calidad en Uso, párr. 1-2)

Tabla 2.2: Fórmulas ISO 9126

Característica	Fórmulas
Funcionalidad	$Funcionalidad = \left(\frac{PF}{PF(max)} \right) * 100$
Confiabilidad	$F(t) = f * e^{(-u*t)}$
Usabilidad	$USABILIDAD = \left[\left(\frac{\sum (Xi)}{n} \right) * 100 \right]$
Mantenibilidad	$IMS = \frac{[Mt - (Fa + Fb + Fd)]}{Mt}$
Portabilidad	$Portabilidad = 1 - \frac{A}{B}$

Fuente: (Elaboración Propia)

2.20.3 Punto de Función

Es una métrica que permite traducir en un número el tamaño de la funcionalidad que brinda un producto de software desde el punto de vista del usuario, a través de una suma ponderada de las características del producto.

Se determinan 5 características del ámbito de la información y los cálculos aparecen en la posición apropiada de la tabla. Los valores del ámbito de información están definidos de la siguiente manera:

- ▶ **Números de entradas de usuario:** se cuenta cada entrada del usuario que proporcione al software diferentes datos orientados a la aplicación. Las entradas deben ser distinguidas de las peticiones que se contabilizan por separado. Ej. Pantallas de entrada de datos, lector de

códigos de barras, lector de tarjetas magnéticas y electrónicas, captura de imágenes, voz, etc.

- ▶ **Número de salidas del usuario:** se encuentra cada salida que proporciona al usuario información orientada a la aplicación. En este contexto las salidas se refieren a informes, pantallas de salida de datos, grabación de bandas magnéticas, listados, mensajes de error, Transferencia de datos a otras aplicaciones, ya sea mediante archivos o transmisión de datos. Son todos aquellos procesos que hacen llegar datos desde la aplicación hacia el exterior, a un usuario o a otra aplicación.
- ▶ **Números de consultas al usuario:** una petición o consulta está definida como una entrada interactiva que resulta de la generación de algún tipo de respuesta en forma de salida interactiva (en línea). Son todos aquellos procesos que están formados por una combinación de entradas y salidas, produciendo una consulta a los datos.
- ▶ **Número de archivos lógicos internos:** Es un grupo de datos relacionados, tal como los percibe el usuario y que son mantenidos por la aplicación. Se cuenta cada archivo maestro lógico, o sea una agrupación lógica de datos que puede ser una parte en una gran base de datos o un archivo independiente. Los archivos se cuentan una sola vez, independientemente del número de procesos que los acceden.
- ▶ **Número de interfaces externas:** agrupamiento lógico de datos que reside fuera de la aplicación, pero que proporciona información que puede usar la aplicación. Se cuentan todas las interfaces legibles por la máquina por ejemplo: archivos de datos, en cinta o discos que son utilizados para transmitir información a otro sistema. Son archivos internos de otra aplicación.

En base a las tablas anteriores, se coloca el número de elementos de las características de nuestro sistema en el peso correspondiente a la tabla de valores de pesos y puntos de función sin ajustar.

Figura 2.20: Cálculo de Punto de Función

Parámetros de medición	Cuenta	Factor de ponderación			=	[]
		Simple	Medio	Complejo		
Número de entradas de usuario	[] ×	3	4	6	=	[]
Número de salidas de usuario	[] ×	4	5	7	=	[]
Número de peticiones de usuario	[] ×	3	4	6	=	[]
Número de archivos	[] ×	7	10	15	=	[]
Número de interfaces externas	[] ×	5	7	10	=	[]
Cuenta total	→					[]

Fuente: (Pressman, 2005, pág. 532)

Figura 2.21: Valores

0	1	2	3	4	5
Sin influencia	Incidental	Moderado	Medio	Significativo	Esencial

Fuente: (Pressman, 2005)

Figura 2.22: Factores de Complejidad

#	Factor de Complejidad	Valor
1	Comunicación de datos	
2	Proceso distribuido	
3	Objetivos de rendimiento	
4	Configuración de explotación usada por otros sistemas	
5	Tasa de transacciones	
6	Entrada de datos en línea	
7	Eficiencia con el usuario final	
8	Actualizaciones en línea	
9	Lógica del Proceso Interno Compleja	
10	Reusabilidad del código	
11	Contempla la conversión e instalación	
12	Facilidad de operación	
13	Instalaciones múltiples	
14	Facilidad de cambios	
Factor de Complejidad Total (GTI)		\sum <i>valor de los factores</i>

Fuente: (Pressman, 2005)

Fórmula de Punto de Función:

$$PF = CuentaTotal * [0.65 + 0.01 * \sum (Fi)]$$

Fuente: (Pressman, 2005)

2.21 MÉTODO DE ESTIMACIÓN DE COSTOS DE SOFTWARE

2.21.1 Modelo de Construcción de Costos (COCOMO)

Vecino & Miranda & Obregón & Barrios & Pava D en el sitio web “Modelo Cocomo (Ingeniera de Software)” refieren lo siguiente:

El Modelo Constructivo de Costes COCOMO Constructive Cost Model es utilizado en proyectos de software para estimar los costes del mismo, este modelo fue diseñado por Barry W. Boehm para dar una estimación del número de meses hombre que tomará para desarrollar un producto de

software.

Es un modelo matemático de base empírica utilizado para estimación de costos de software. Incluye tres submodelos, cada uno ofrece un nivel de detalle y aproximación cada vez mayor, a medida que avanza el proceso de desarrollo del software: Básico, Intermedio y Detallado. (Modelo Cocomo, párr. 1)

2.21.1.1 Modelos de Estimación

Las ecuaciones que se utilizan en los tres modelos son:

$$E = a(KI)^b * m(X) \text{ personas mes}$$

$$Tdev = c(E)^d \text{ meses}$$

$$P = \frac{E}{Tdev} \text{ personas}$$

Donde:

E es el esfuerzo requerido por el proyecto, en persona-mes

Tdev es el tiempo requerido por el proyecto, en meses

P es el número de personas requerido por el proyecto

a, b, c y d son constantes con valores definidos en una tabla, según cada submodelo

KI es la cantidad de líneas de código, en miles.

m(X) Es un multiplicador que depende de 15 atributos.

A la vez, cada submodelo también se divide en modos que representan el tipo de proyecto, y puede ser:

- ▶ **Modo Orgánico:** un pequeño grupo de programadores experimentados desarrollan software en un entorno familiar. El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas

decenas de miles (medio).

- ▶ **Modo Semilibre O Semiencajado:** corresponde a un esquema intermedio entre el orgánico y el rígido; el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.
- ▶ **Modo Rígido O Empotrado:** el proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único y es difícil basarse en la experiencia, puesto que puede no haberla.

2.21.1.2 Modelos de COCOMO

2.21.1.2.1 Modelo Básico

Se utiliza para obtener una primera aproximación rápida del esfuerzo, y hace uso de la siguiente tabla de constantes para calcular distintos aspectos de costes:

Tabla 2.3: Constantes COCOMO Básico

MODO	a	b	c	d
Orgánico	2.40	1.05	2.50	0.38
Semi Orgánico	3.00	1.12	2.50	0.35
Empotrado	3.60	1.20	2.50	0.32

Fuente: (Wikipedia Cocomo, 2005, pág. 1)

Estos valores son para las fórmulas:

Personas necesarias por mes para llevar adelante el proyecto (**MM**) = **a*(Klb)**

Tiempo de desarrollo del proyecto (**TDEV**) = **c*(MMd)**

Personas necesarias para realizar el proyecto (**CosteH**) = **MM/TDEV**

Costo total del proyecto (**CosteM**) = **CosteH * Salario medio** entre los programadores y analistas.

2.21.1.2.2 Modelo Intermedio

Este añade al modelo básico quince modificadores opcionales para tener en cuenta en el entorno de trabajo, incrementando así la precisión de la estimación.

Para este ajuste, al resultado de la fórmula general se lo multiplica por el coeficiente surgido de aplicar los atributos que se decidan utilizar.

Los valores de las constantes a reemplazar en la fórmula son:

Tabla 2.4: Constantes COCOMO Intermedio

MODO	a	b
Orgánico	3.20	1.05
Semi Orgánico	3.00	1.12
Empotrado	2.80	1.20

Fuente: (Wikipedia Cocomo, 2005, pág. 1)

Se puede observar que los exponentes son los mismos que los del modelo básico, confirmando el papel que representa el tamaño; mientras que los coeficientes de los modos orgánico y rígido han cambiado, para mantener el equilibrio alrededor del semilibre con respecto al efecto multiplicador de los atributos de coste.

Cada atributo se cuantifica para un entorno de proyecto. La escala es muy bajo - bajo - nominal - alto - muy alto - extremadamente alto. Dependiendo de la calificación de cada atributo, se asigna un valor para usar de multiplicador en la fórmula (por ejemplo, si para un proyecto el atributo DATA es calificado como muy alto, el resultado de la fórmula debe ser multiplicado por 1000).

El significado de los atributos es el siguiente, según su tipo:

De Software

- ▶ **RELY:** garantía de funcionamiento requerida al software. Indica las posibles consecuencias para el usuario en el caso que existan defectos en el producto. Va desde la sola inconveniencia de corregir un fallo (muy bajo) hasta la posible pérdida de vidas humanas (extremadamente alto, software de alta criticidad).
- ▶ **DATA:** tamaño de la base de datos en relación con el tamaño del programa. El valor del modificador se define por la relación: D/K , donde D corresponde al tamaño de la base de datos en bytes y K es el tamaño del programa en cantidad de líneas de código.
- ▶ **CPLX:** representa la complejidad del producto.

De hardware

- ▶ **TIME:** limitaciones en el porcentaje del uso de la CPU.
- ▶ **STOR:** limitaciones en el porcentaje del uso de la memoria.
- ▶ **VIRT:** volatilidad de la máquina virtual.
- ▶ **TURN:** tiempo de respuesta requerido.
- ▶ De personal
- ▶ **ACAP:** calificación de los analistas.
- ▶ **AEXP:** experiencia del personal en aplicaciones similares.
- ▶ **PCAP:** calificación de los programadores.
- ▶ **VEXP:** experiencia del personal en la máquina virtual.
- ▶ **LEXP:** experiencia en el lenguaje de programación a usar.
- ▶ De proyecto
- ▶ **MODP:** uso de prácticas modernas de programación.
- ▶ **TOOL:** uso de herramientas de desarrollo de software.
- ▶ **SCED:** limitaciones en el cumplimiento de la planificación.

El valor de cada atributo, de acuerdo a su calificación, se muestra en la siguiente tabla:

Tabla 2.5: Cuantificación del Proyecto

ATRIBUTOS	VALOR					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
Atributos de software						
Fiabilidad	0,75	0,88	1,00	1,15	1,40	
Tamaño de Base de datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
Atributos de hardware						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual		0,87	1,00	1,15	1,30	
Tiempo de respuesta		0,87	1,00	1,07	1,15	
Atributos de personal						
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90		
Experiencia en el lenguaje	1,14	1,07	1,00	0,95		
Atributos del proyecto						
Técnicas actualizadas de programación	1,24	1,10	1,00	0,91	0,82	
Utilización de herramientas de software	1,24	1,10	1,00	0,91	0,83	
Restricciones de tiempo de desarrollo	1,22	1,08	1,00	1,04	1,10	

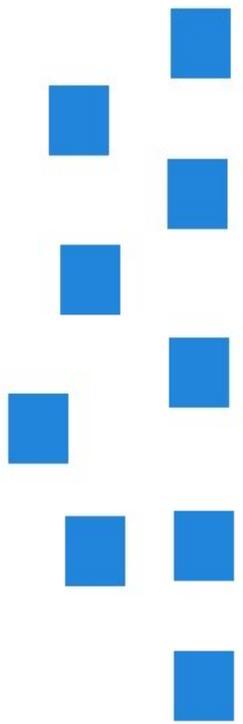
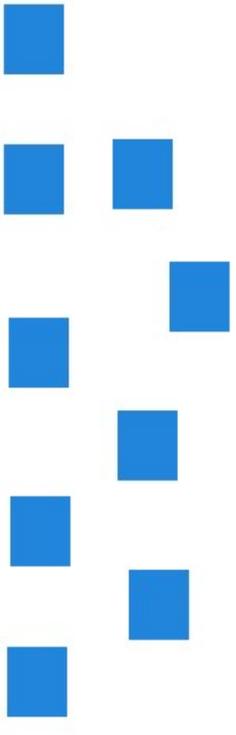
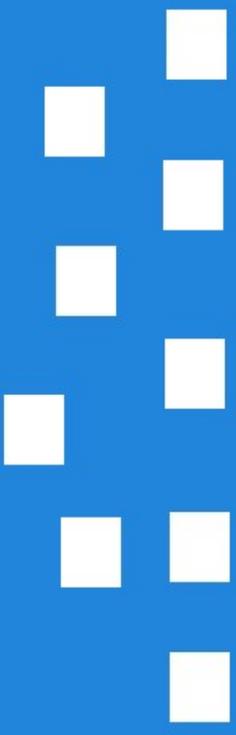
Fuente: (Wikipedia Cocomo, 2005, pág. 1)

2.21.1.2.3 Modelo Detallado

Según Wikipedia (2005) en su sitio web "" refiere lo siguiente:

Presenta principalmente dos mejoras respecto al modelo intermedio.

- ▶ Los factores correspondientes a los atributos son sensibles o dependientes de la fase sobre la que se realizan las estimaciones. Aspectos tales como la experiencia en la aplicación, utilización de herramientas de software, etc., tienen mayor influencia en unas fases que en otras, y además van variando de una etapa a otra.
- ▶ Establece una jerarquía de tres niveles de productos, de forma que los aspectos que representan gran variación a bajo nivel, se consideran a nivel módulo, los que representan pocas variaciones, a nivel de subsistema; y los restantes son considerados a nivel sistema. (Modelo Detallado, párr. 1)



CAPÍTULO III

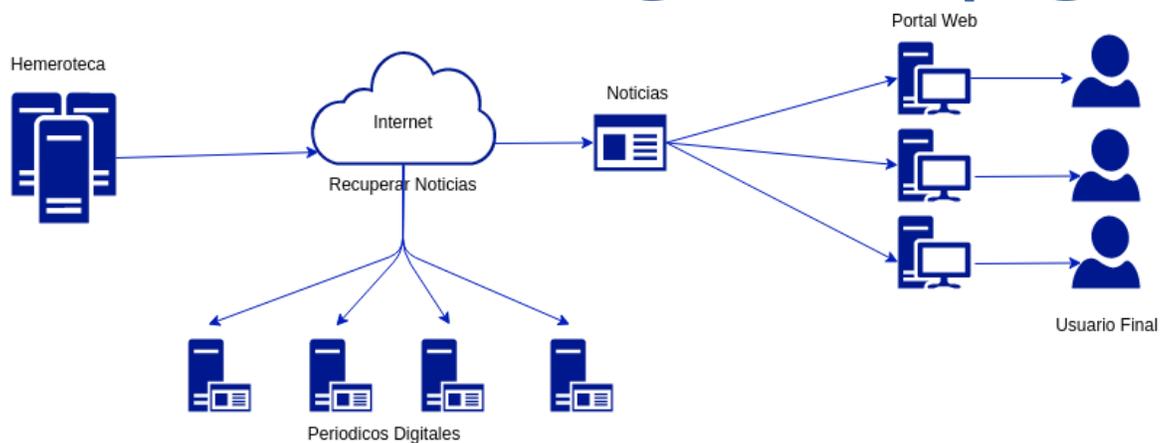
3. MARCO APLICATIVO

En el presente capítulo se aplicará la ingeniería de requerimientos, para luego continuar con las diferentes fases correspondientes a la conceptualización, análisis y diseño siguiente el proceso de desarrollo UWE detallado en el marco teórico.

3.1 ESQUEMA DEL SISTEMA

En la Figura 3.1 podemos apreciar el esquema de funcionamiento de el sistema, el cual ingresa a internet y va a buscar las noticias recientes de los periódicos digitales registrados, una vez encontradas las noticias se procede a descargar el código fuente dentro de nuestro sistema y de ese modo tiene un registro histórico de las noticias, luego procedemos con el web scraping, una vez obtenida la información se procede a publicarla para que los usuarios puedan informarse sobre el acontecer diario encontrado en las múltiples páginas de noticias.

Figura 3.1 Esquema del Sistema Hemeroteca



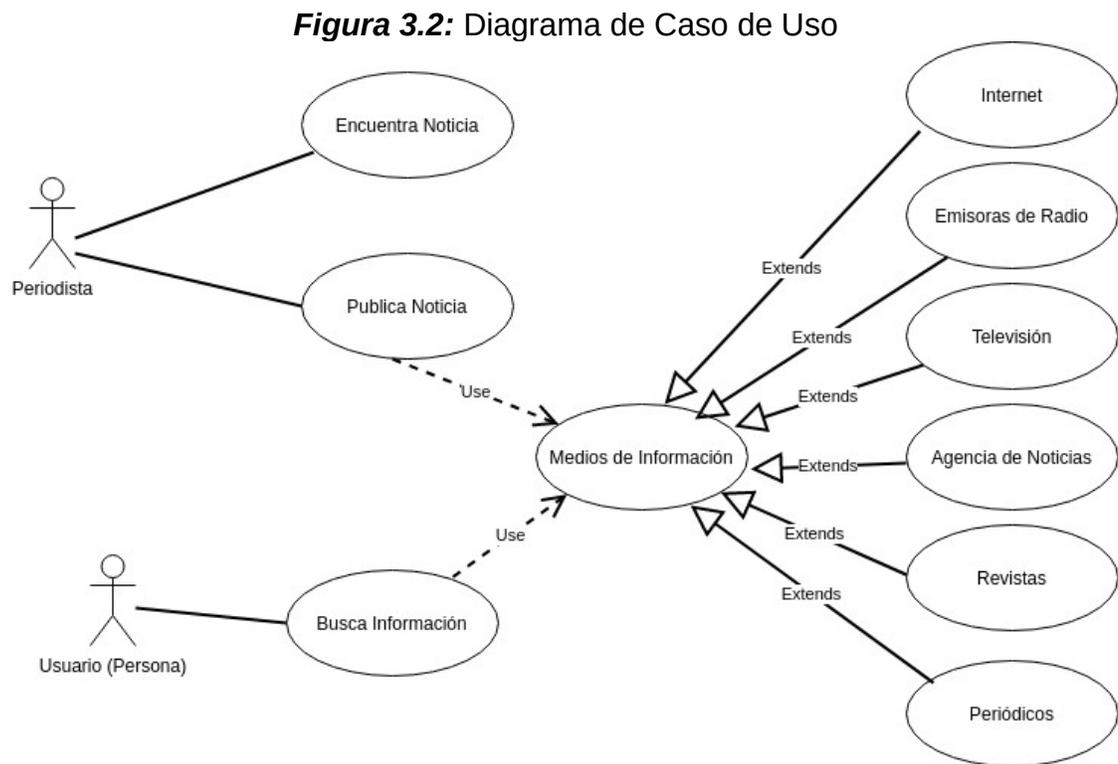
Fuente: (Elaboración Propia)

3.2 ANÁLISIS DE REQUERIMIENTO

3.2.1 Análisis de Situación Actual

La necesidad de informarse es vital para nuestro día a día, desde cómo será el clima el día de hoy hasta saber si el pan subirá de precio. En este sentido hay varios sectores para los cuales la información es muy importante, tales son los empresarios, profesionales, estudiantes y personas en general.

En la siguiente Figura 3.2 se describe el actual funcionamiento de cómo las personas acceden a la información.



Fuente: (Elaboración Propia)

3.2.2 Ingeniería de Requerimientos

Ingeniería de requerimientos es una descripción de una condición o capacidad que debe cumplir un sistema, ya sea derivada de una necesidad de usuario

identificada, o bien, estipulada en un contrato, especificación u otro documento formalmente impuesto al inicio del proceso. (Arias Chávez, 2006)

En la **Tabla 3.1** se describen las categorías de funciones que se utilizan para poder clasificar los requerimientos que se realizan en las **Tablas 3.2 y 3.3**.

Tabla 3.1: Categoría de Funciones

CATEGORÍA DE LA FUNCIÓN	SIGNIFICADO
Evidente	Debe realizarse y los usuarios están conscientes de que se ha realizado.
Oculto	Debe realizarse aunque no es visible para los usuarios. Esto se aplica a muchos servicios técnicos subyacentes, por ejemplo, guardar información en un mecanismo persistente de almacenamiento.
Superflua	Opcionales, su inclusión no repercute de forma significativa en costo ni otras funciones.

Fuente: (Larman, 1999)

3.2.2.1 Requerimientos Funcionales

Los requerimientos funcionales para el modelado del sistema se detallan y describen en la Tabla 3.2.

Tabla 3.2: Requerimientos Funcionales

REFERENCIA	FUNCIÓN	CATEGORÍA
R.1.1	El sistema debe tener seguridad en el acceso a la información y manipulación del sistema.	Evidente

R.1.2	Acceder al sistema por un rol único (Administrador)	Evidente
R.1.3	Registrar periódicos digitales.	Evidente
R.1.4	Recuperar de forma automática los enlaces de noticias de cada periódico.	Oculto
R.1.5	Recuperar de forma automática el código fuente de cada enlace de noticia.	Oculto
R.1.6	Recuperar de forma automática el título, autor, subtítulo, contenido, enlace de la imagen y fecha de una noticia.	Oculto
R.1.7	Recuperar de forma automática la imagen de la noticia.	Oculto
R.1.8	Inicio y Cierre del sistema.	Evidente
R.1.9	Acceso a las noticias proporcionadas por el sistema sin autenticación.	Evidente
R.1.10	Mostrar las cantidad de noticias encontradas por periódico.	Evidente
R.1.11	Crear filtros de búsqueda de noticias por fecha, periódico y contenido.	Evidente
R.1.12	Mostrar la noticia	Evidente

Fuente (Elaboración Propia)

3.2.2.2 Requerimientos No Funcionales

Tabla 3.3: Requerimientos No Funcionales

REFERENCIA	FUNCIÓN	CATEGORÍA
R.1.1	El sistema debe visualizarse y funcionar correctamente en cualquier navegador Mozilla, Chrome u Opera.	Evidente
R.1.2	Debe crearse un backup cada cierto tiempo.	Evidente
R.1.3	Respaldo energético del servidor para asegurar la disponibilidad del sistema.	Evidente
R.1.4	Soporte y mantenimiento periódico para asegurar el buen rendimiento del sistema.	Evidente
R.1.5	Escalar el servidor según sea necesario.	Evidente

Fuente: (Elaboración Propia)

3.2.3 Actores Identificados

Administrador, persona encargada de registrar y configurar los periódicos digitales de donde se extraerá la información dentro de el sistema.

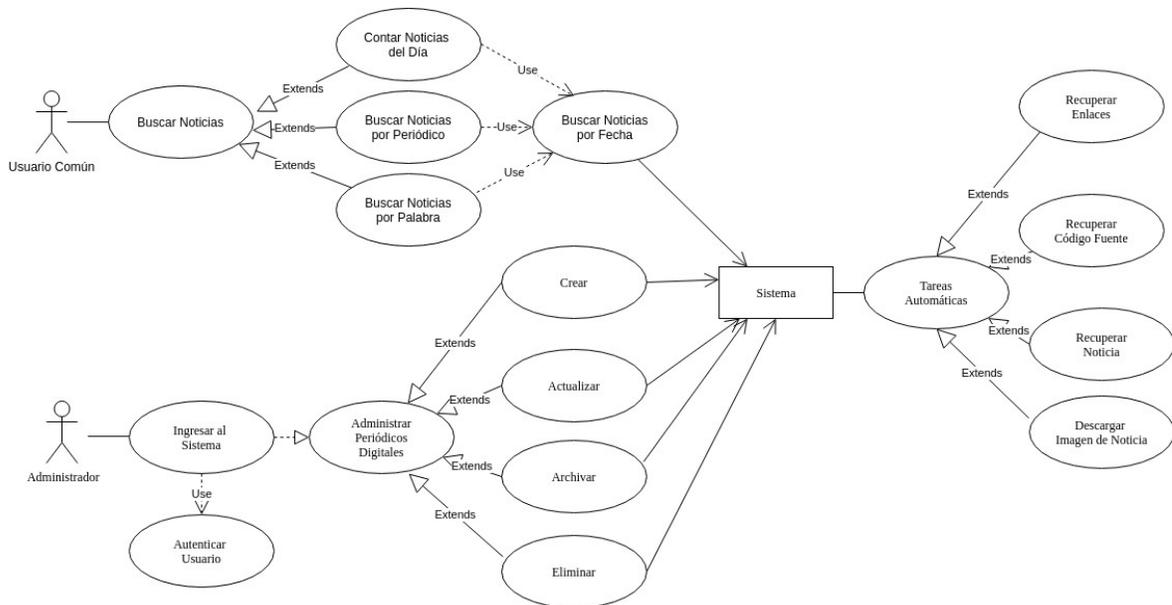
Persona Común (Usuario Final), persona que necesita informarse sobre el acontecer diario dentro del territorio boliviano.

3.3 MODELO DE CASOS DE USO

A continuación plasmamos el análisis de requerimientos del sistema mediante el diseño de casos de uso expresados en el comportamiento del sistema frente a las acciones de los actores del mismo, funcionalidades del sistema y demás elementos que permitan la contemplación del problema.

3.3.1 Diagrama de Caso de Uso General

Figura 3.3: Diagrama de Caso de Uso General

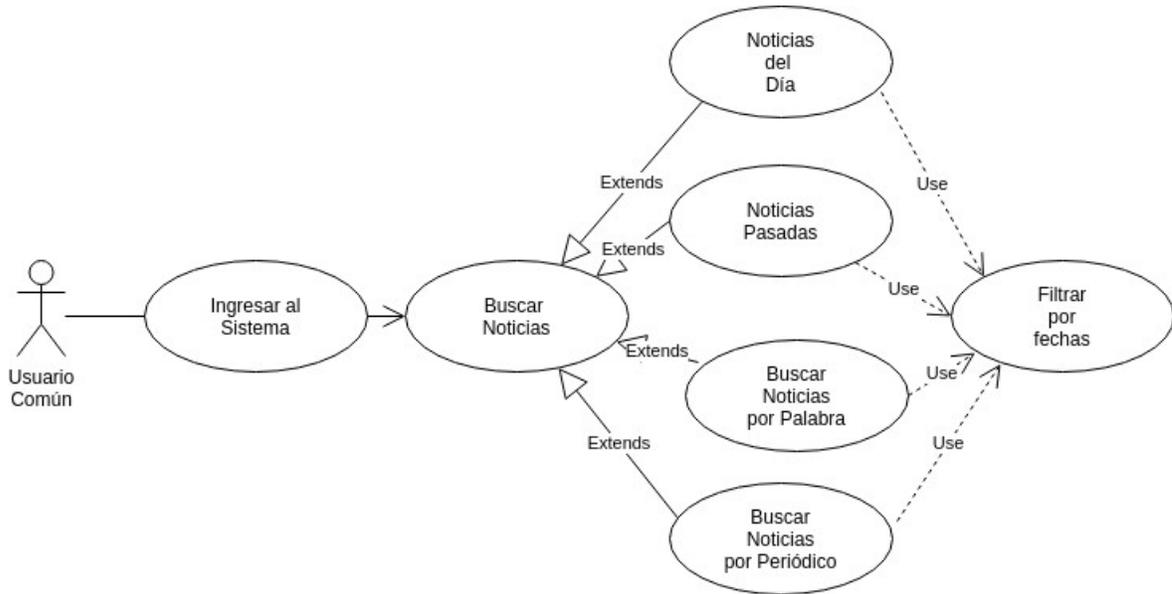


Fuente (Elaboración Propia)

3.3.2 Aplicación De La Metodología Uwe

3.3.3 Caso de Uso de Búsqueda de Noticias

Figura 3.4: Caso de Uso de Búsqueda de Noticias



Fuente (Elaboración Propia)

Tabla 3.4: Especificación del Caso de Uso: Búsqueda de Noticias

DESCRIPCIÓN DEL CASO DE USO			
Nombre	Búsqueda de Noticias		
Actor	Usuario (Población en General)		
Descripción	El usuario común del sistema puede ingresar sin necesidad de tener una cuenta en el sistema, puede realizar búsquedas de noticias recientes o pasadas.		
Flujo Principal	<table border="0"> <thead> <tr> <th>Evento Actor</th> <th>Evento Sistema</th> </tr> </thead> </table>	Evento Actor	Evento Sistema
Evento Actor	Evento Sistema		

Verificar la fecha Actual.

Ingresa a la página principal del sistema.

Desplegar una pantalla con los diferentes periódicos registrados y la cantidad de noticias recuperadas en el día.

Elige un rango de fechas anterior al día actual y presiona buscar.

Buscar las noticias acontecidas en la fecha introducida.

En el menú del buscador introduce una palabra y presiona buscar.

Desplegar una pantalla con todas las noticias encontradas.

Buscar las noticias que contengan la palabra introducida dentro del título o contenido.

Desplegar una pantalla con todas las noticias coincidentes.

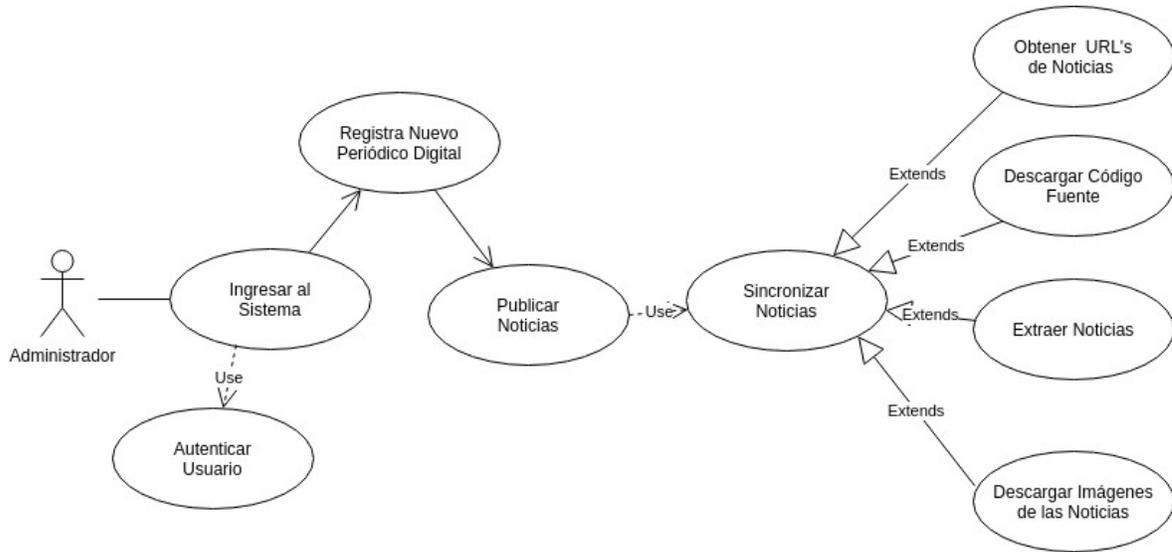
En el menú selecciona el nombre de un periódico de su preferencia.

Desplegar en una pantalla las noticias encontradas de el periódico seleccionado.

Fuente (Elaboración Propia)

3.3.4 Caso de Uso Registro y Publicación de Noticias

Figura 3.5: Caso de Uso Registro y Publicación de Noticias



Fuente (Elaboración Propia)

Tabla 3.5: Especificación de Caso de Uso: Registro y Publicación de Noticias

DESCRIPCIÓN DEL CASO DE USO		
Nombre	Registro y Publicación de Noticias	
Actor	Administrador	
Descripción	El administrador del sistema puede crear, actualizar, eliminar y archivar un periódico digital.	
Flujo Principal	Evento Actor	Evento Sistema
	Ingresar al Sistema	Verificar Usuario. Si el usuario es correcto despliega la página de principal con las respectivas

opciones del administrador.

Se despliega un modal, el cual contiene los campos necesarios para registrar un nuevo periódico digital.

Se registran los datos del periódico digital.

Iniciar sincronización de enlaces de noticias.

En la página principal hace click en el botón de crear y Iniciar sincronización para descargar el publicar un nuevo Periódico código fuente. Digital.

Iniciar sincronización para obtener la noticia de los códigos fuente descargados.

Iniciar sincronización de imágenes correspondientes a las noticias descargadas.

Publicar las noticias encontradas.

3.4 DISEÑO CONCEPTUAL

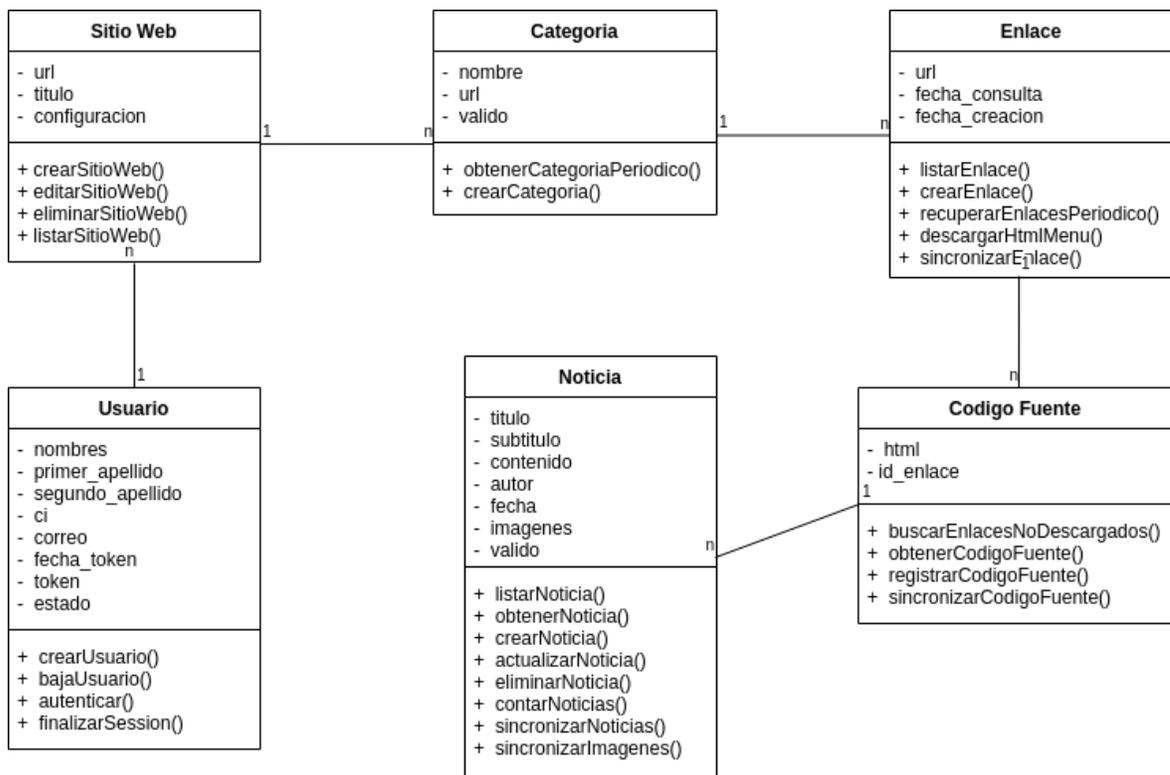
El diseño conceptual se basa en el análisis de requerimientos vistos en los casos de uso del punto anterior. Incluye los objetos involucrados en la interacción entre el usuario y la aplicación web.

3.4.1 Modelo de Contenidos

El diagrama de contenido tiene por propósito mostrar las relaciones entre las entidades y la estructura de los datos que se encuentran alojados en el sistema (UWE-Engineering).

Este modelo se representa mediante un diagrama de clases UML, como se muestra en la siguiente Figura.

Figura 3.6: Modelo de Contenido del Sistema



Fuente: (Elaboración Propia)

3.5 DISEÑO NAVEGACIONAL

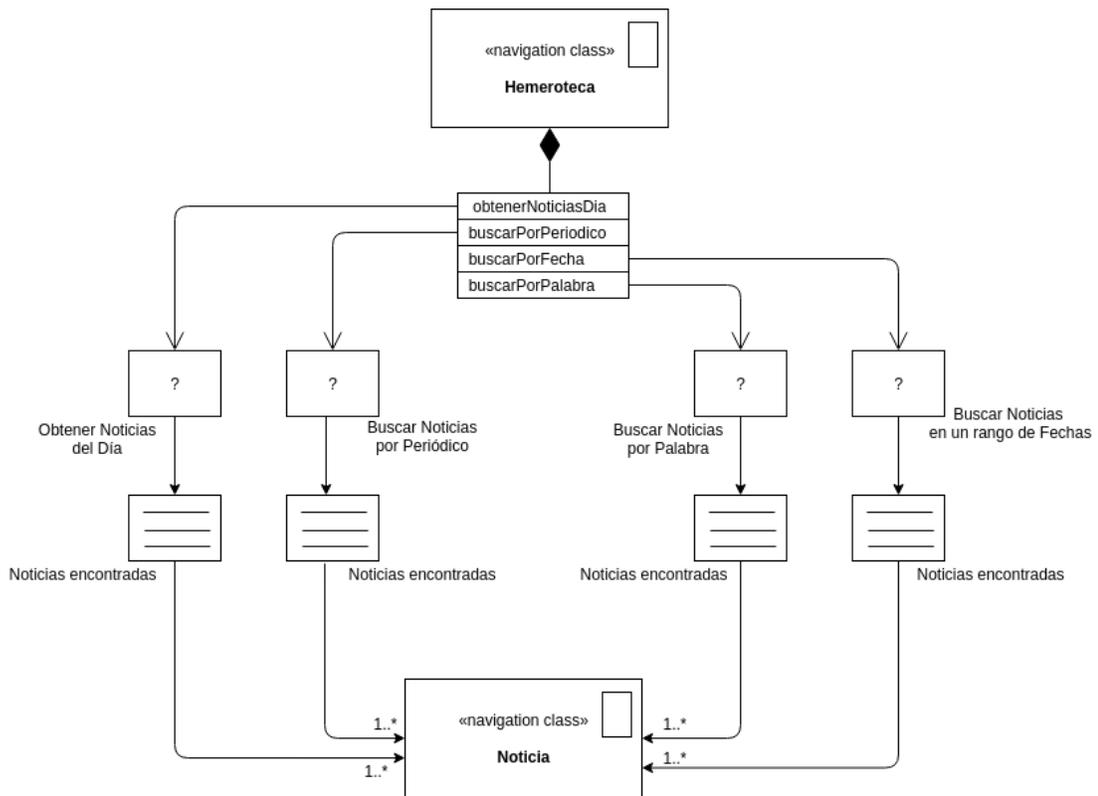
El diseño navegación nos da una percepción mejor de los vínculos lógicos y de navegación entre clases.

Por tratarse de una aplicación orientado a la web es necesario modelar la navegación de la aplicación, el cual presenta dos modelos de navegación.

3.5.1 Modelo de Espacio de Navegación

La Figura 3.7 muestra el modelo de clase navegación y las asociaciones de navegación general del proyecto, que expresa la navegación directa aplicando la metodología UWE.

Figura 3.7: Modelo de Espacio de Navegación Directa (Usuario Común)

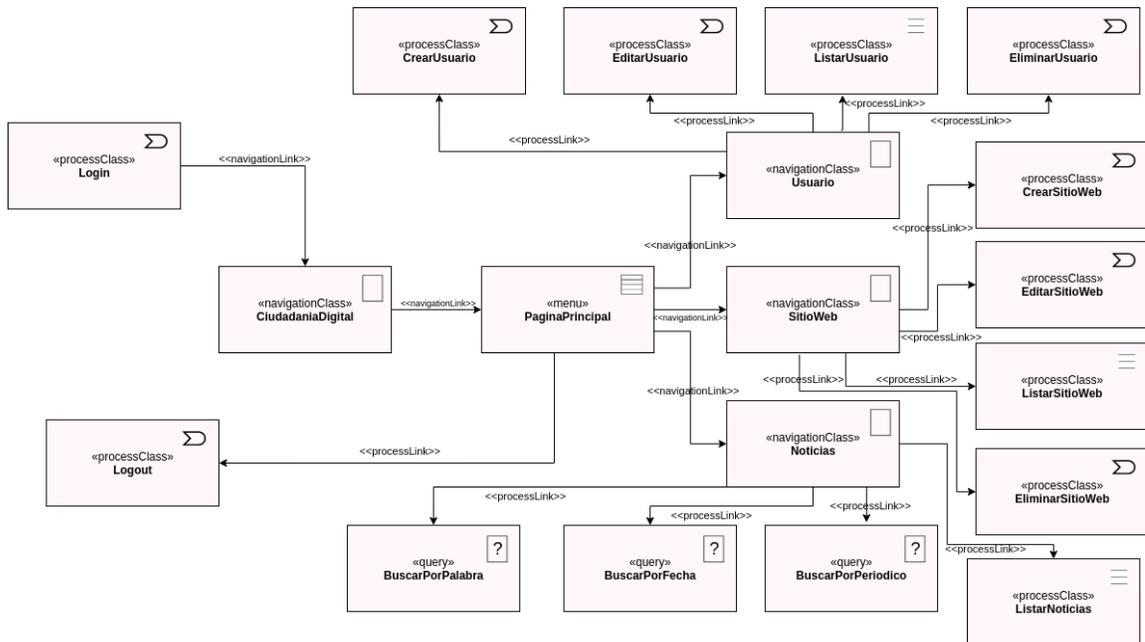


Fuente: (Elaboración Propia)

3.5.2 Modelo de la Estructura de Navegación

Este modelo de estructura de navegación permite ilustrar los vínculos lógicos y de navegación entre clases, menús, índices y clases de proceso para tener una mejor comprensión del sistema se muestra en la siguiente figura:

Figura 3.8: Modelo de Estructura de Navegación
Página Principal

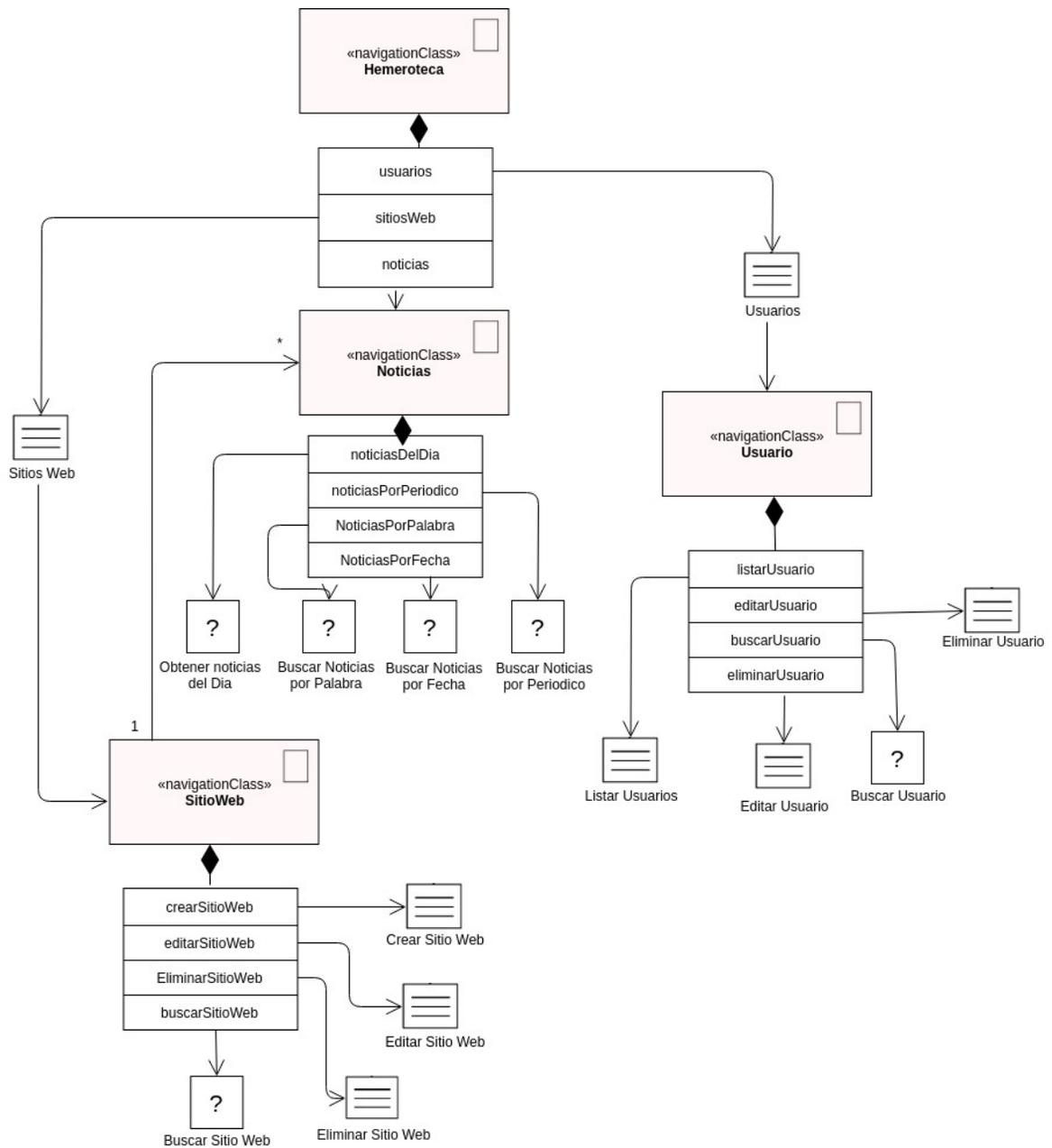


Fuente: (Elaboración Propia)

La **Figura 3.8** Muestra los vínculos lógicos y de navegación sobre las operaciones que se realizan en el Página Principal.

El administrador tiene acceso a la información sobre las noticias, sitios web y gestión de usuarios.

Figura 3.9: Modelo de Estructura de Navegación Menú Principal



Fuente: (Elaboración Propia)

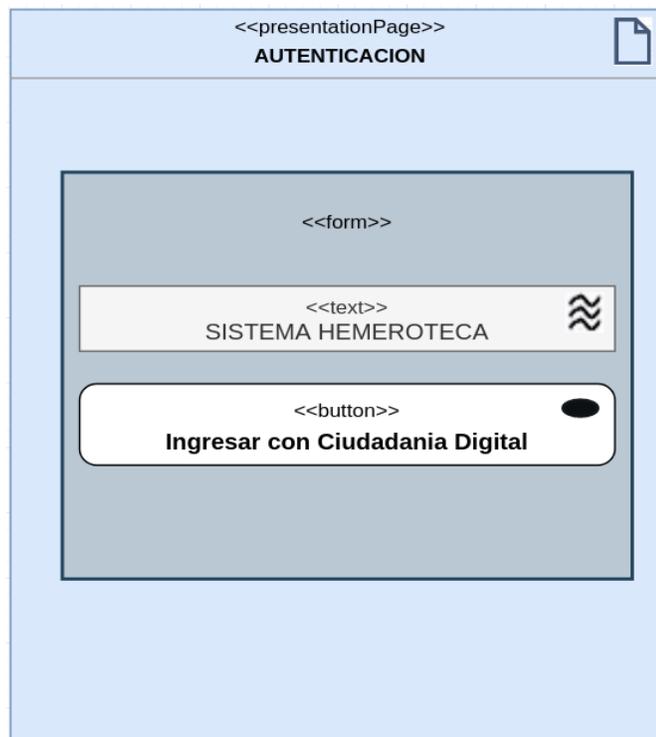
La **Figura 3.9** nos muestra el modelo de navegación en base a los menús del Sistema.

3.6 DISEÑO DE PRESENTACIÓN

3.6.1 Modelo de Presentación

A continuación mostraremos los modelos de presentación según UWE, el cual nos plantea la construcción de páginas en forma de bosquejos derivadas de la **Figura 3.7** y la **Figura 3.8** donde se muestra cómo los usuarios acceden al sistema mostrando los menús correspondientes.

Figura 3.10: Pantalla de Inicio de Sesión



Fuente: (Elaboración Propia)

En la **Figura 3.10** se aprecia el diseño para el inicio de autenticación del usuario, podemos ver que solo vemos un botón que dice “Ingresar con Ciudadanía Digital” esto se debe a que el sistema utiliza OAuth2.0 como método de autenticación, tal como Google lo hace en muchas aplicaciones.

Figura 3.11: Pantalla Usuario Ciudadanía Digital

The screenshot shows a mobile application interface for 'CIUDADANIA DIGITAL'. At the top, there is a header with the text '<<presentationPage>>' and 'CIUDADANIA DIGITAL' next to a document icon. Below the header is a central form area with a light blue background. The form contains three main components: a text field labeled 'CARNET DE IDENTIDAD' with a wavy icon on the right; a text input field containing the number '123456' with an 'abi' icon on the right; and a button labeled 'Siguiete' with a black circle icon on the right.

Figura 3.12: Pantalla Contraseña Ciudadanía Digital

The screenshot shows a mobile application interface for 'CIUDADANIA DIGITAL'. At the top, there is a header with the text '<<presentationPage>>' and 'CIUDADANIA DIGITAL' next to a document icon. Below the header is a central form area with a light blue background. The form contains three main components: a text field labeled 'CONTRASEÑA' with a wavy icon on the right; a text input field containing a series of asterisks '*****' with an 'abi' icon on the right; and a button labeled 'Siguiete' with a black circle icon on the right.

Fuente: (Elaboración Propia)

En la **Figura 3.11** es proporcionada por “Ciudadanía Digital” en la cual tenemos la opción de ingresar nuestro Usuario el cual corresponde a nuestro carnet de identidad.

Posteriormente en la **Figura 3.12** podremos ingresar nuestra contraseña de “Ciudadanía Digital”.

Nota.- Solo las personas registradas en “Ciudadanía Digital” tiene la posibilidad de ser registrado como administrador de nuestro sistema.

Figura 3.13: Pantalla Ingreso de Código Ciudadanía Digital

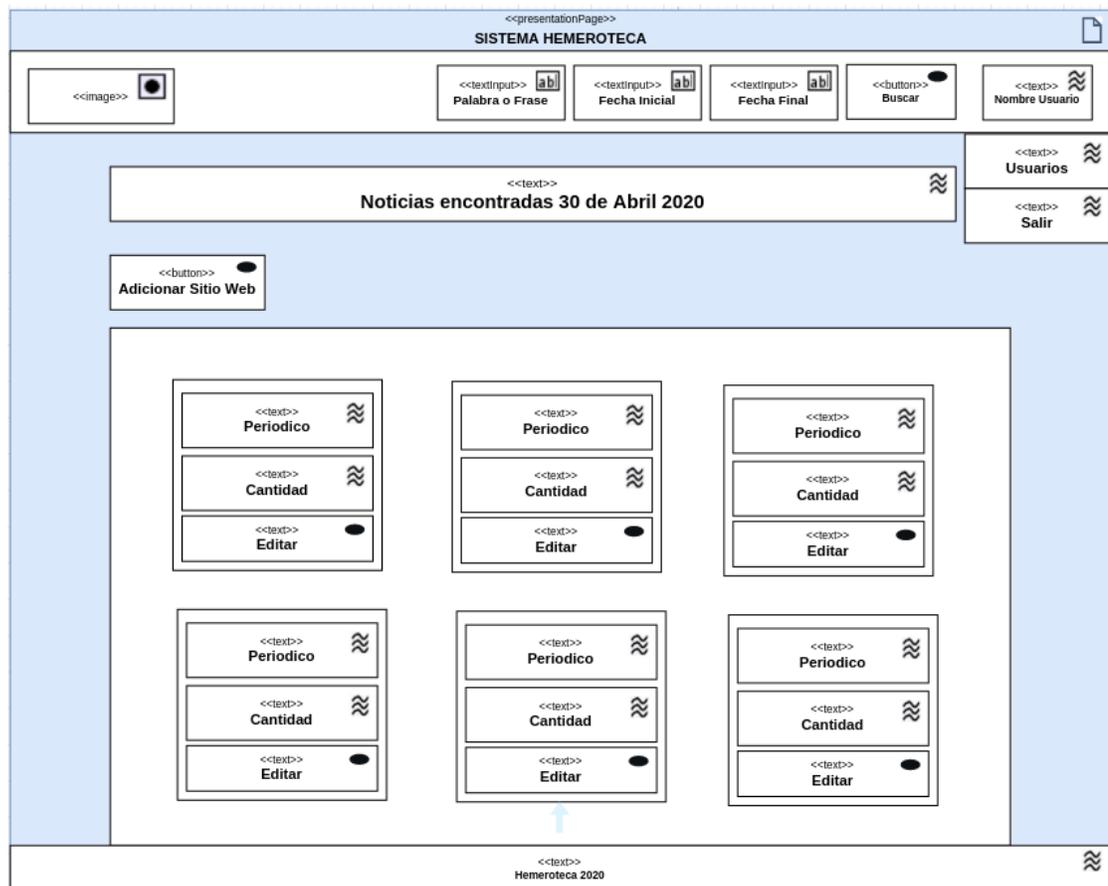
Diagrama de la interfaz de usuario para el ingreso de código de Ciudadanía Digital. El formulario está contenido en un contenedor etiquetado como <<form>>. Dentro del formulario, hay un campo de texto etiquetado como <<text>> CODIGO con un ícono de ondas a la derecha. Debajo de este campo hay un campo de texto de entrada etiquetado como <<textInput>> 2133 con un ícono 'a|b|' a la derecha. En la parte inferior del formulario hay un botón etiquetado como <<button>> Ingresar con un ícono de círculo negro a la derecha. El formulario está dentro de una pantalla etiquetada como <<presentationPage>> CIUDADANIA DIGITAL con un ícono de documento en la esquina superior derecha.

Fuente: (Elaboración Propia)

En la **Figura 3.13** debemos ingresar un código, este código es enviado por el sistema de Ciudadanía Digital a nuestro celular por sms o correo dependiendo lo que elegimos al momento de registrarnos en Ciudadanía Digital.

A este código se lo llama segundo factor el cual valida que la persona es quien dice ser. Es un punto de seguridad extra que se le da a la cuenta personal.

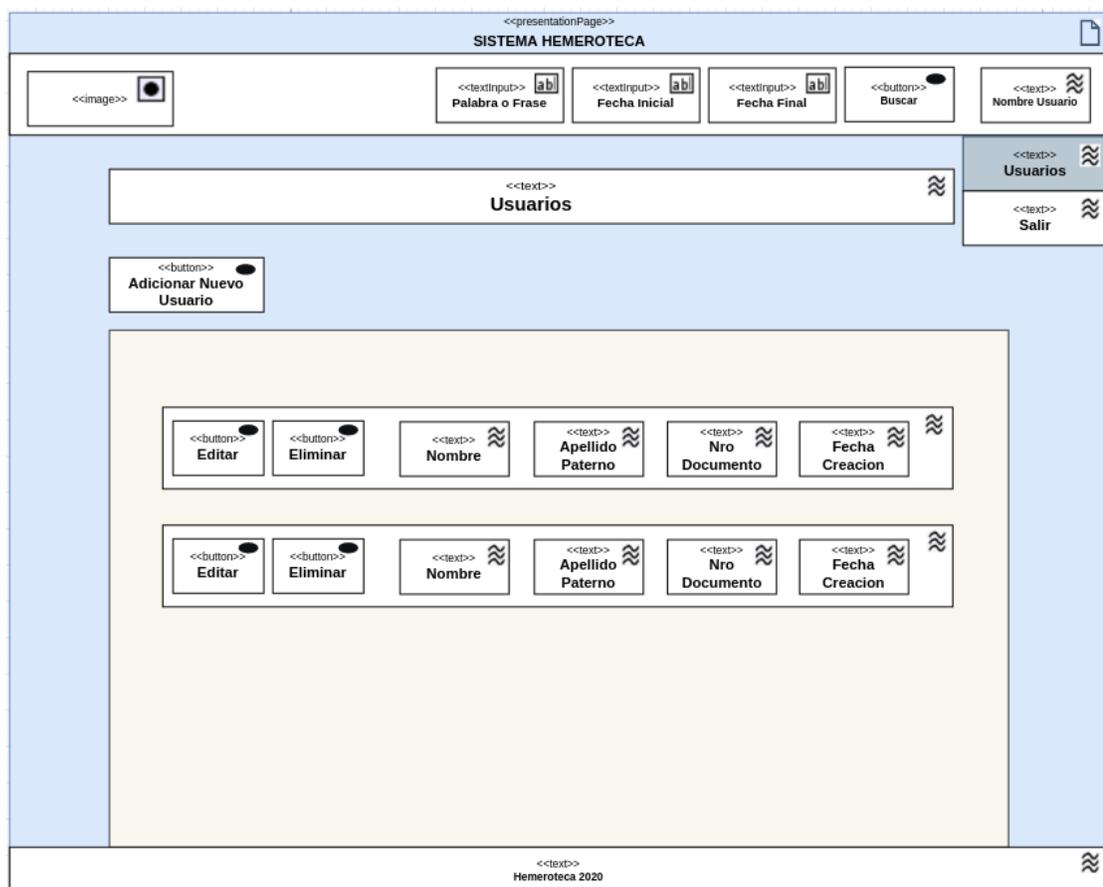
Figura 3.14: Pantalla Principal Administrador



Fuente: (Elaboración Propia)

En la **Figura 3.14** se muestra la página principal una vez el usuario ingrese al sistema, en el cual puede realizar la navegación por usuarios, noticias y sitios web.

Figura 3.15: Pantalla de Administración de Usuarios



Fuente: (Elaboración Propia)

En la **Figura 3.15** podemos apreciar la pantalla donde podemos crear, editar, actualizar, eliminar un Usuario.

Nota.- Para crear un usuario tendremos que preguntar primero al sistema de Ciudadanía Digital si la persona que queremos registrar existe, si existe recuperaremos los datos de Ciudadanía Digital para poder crearlo, de otro modo no podremos registrar ningún usuario en nuestro sistema.

Figura 3.16: Pantalla de Creación de Sitio Web

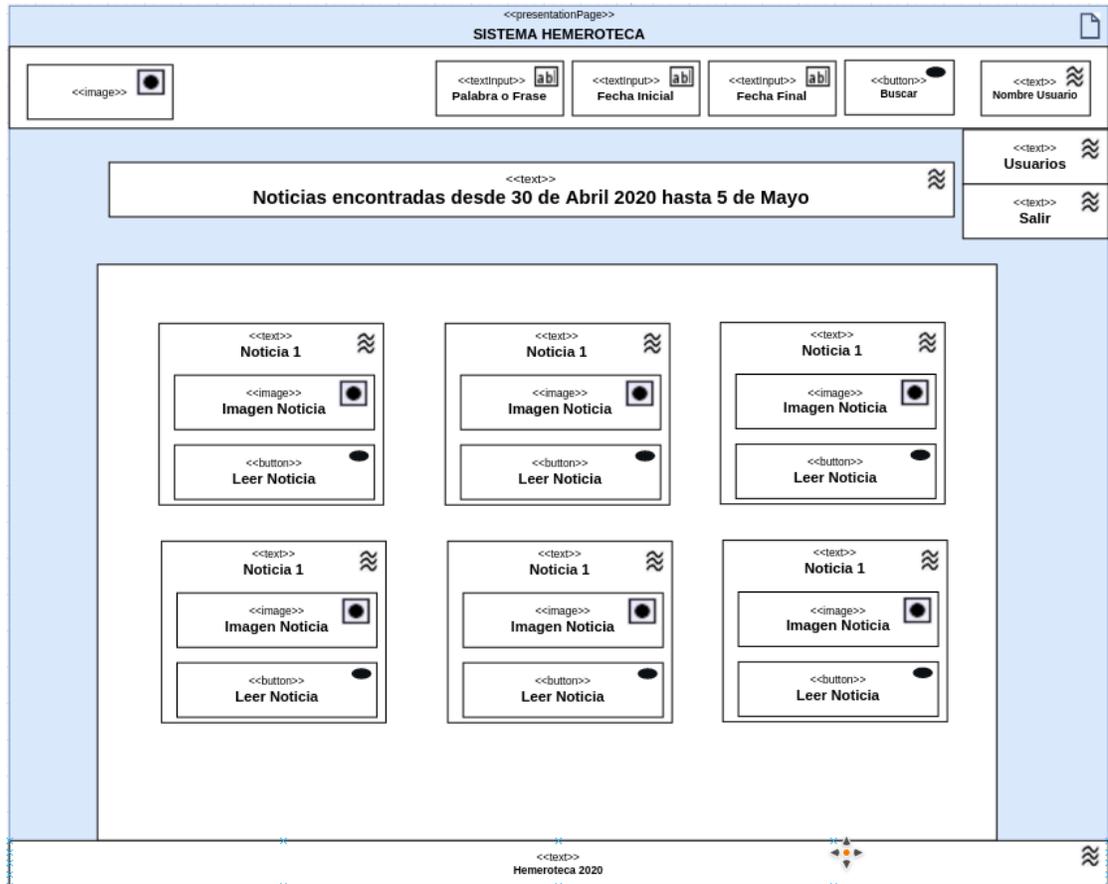
The screenshot displays the 'SISTEMA HEMEROTECA' interface. At the top, there is a header with the system name and a search bar. The search bar contains the text 'Noticias encontradas desde 30 de Abril 2020 hasta 5 de Mayo'. Below the search bar, there is a form titled 'Adicionar Nuevo Sitio Web'. The form has three input fields: 'Nombre', 'Url', and 'Configuracion'. There are also two buttons: 'Cancelar' and 'Guardar'. A mouse cursor is pointing at a plus sign next to the 'Nombre' field.

Fuente: (Elaboración Propia)

En la **Figura 3.16** se puede apreciar el formulario de creación de un sitio web, en este caso el sitio web sería la página principal de un periódico digital.

En el formulario tenemos un campo en particular llamado configuración en este campo escribiremos un JSON el cual contiene las rutas llamadas xpath que nos dan al acceso a la información de un periódico digital, en nuestro caso nos sirven para diferenciar un título del contenido, fecha, imagen, autor y otros que necesitamos recuperar.

Figura 3.17: Pantalla de Noticias



Fuente: (Elaboración Propia)

En la **Figura 3.17** se muestra la pantalla que visualizará tanto el usuario común como el administrador, esta pantalla lista las noticias mostrando la imagen y un botón para poder ingresar a leer cada noticia.

La imagen nos muestra una búsqueda de noticias por fecha, también podemos hacerlo mediante una palabra o por algún periódico en particular.

Las noticias se deberán descargar de nueve en nueve para no sobrecargar el navegador de el usuario ya que se pueden mostrar muchos resultados en una simple consulta.

3.6.2 Modelo de Implementación

Esta fase de implementación consiste en mostrar el desarrollo del sistema a partir de los modelo de presentación de interfaces del sistema y sus elementos constructivos UWE que se realizaron anteriormente. A continuación veremos las pantallas principales del sistema.

En la **Figura 3.18** podemos apreciar la pantalla de autenticación de nuestro Sistema de Hemeroteca, esta interfaz contiene un solo botón el cual nos redireccionará a la autenticación con “Ciudadanía Digital”.

Figura 3.18: Pantalla Principal de Autenticación "Administrador"



Fuente: (Elaboración Propia)

Figura 3.19: Código para obtener ruta de Ciudadanía Digital

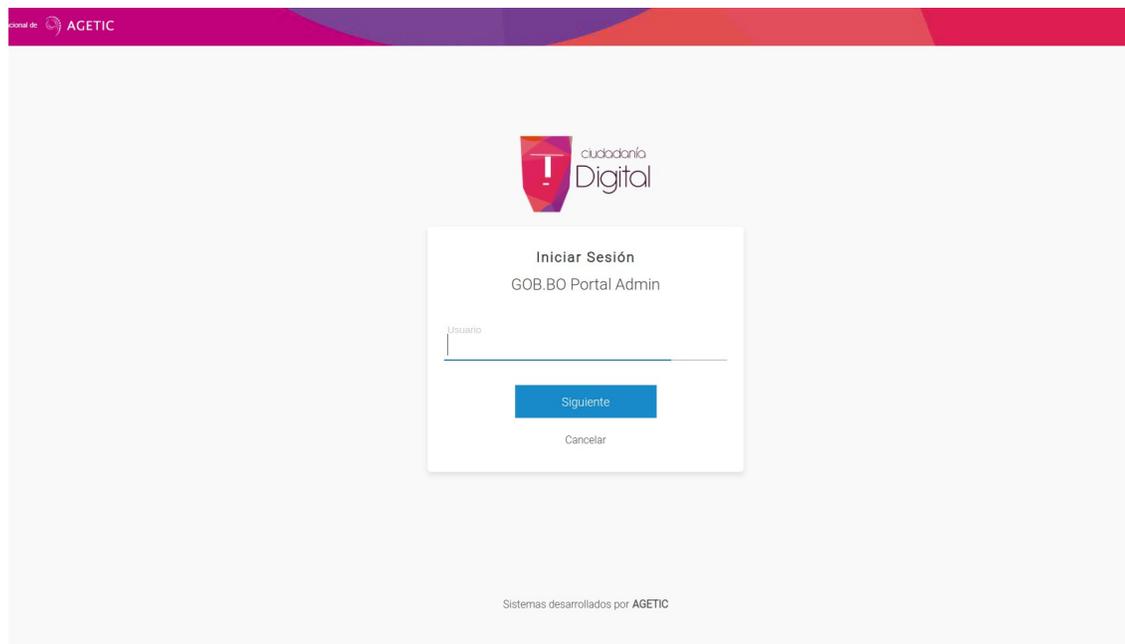
```
@api_ciudadania.route('/getCode')
def obtener_ruta_ciudadania():
    try:
        url_ciudadania = OpenIdAuth().obtenerUrl(client, datastore, session["state"], session["nonce"])
        return respuesta.exito(url_ciudadania)
    except Exception as e:
        return respuesta.error(e)
```

Fuente: (Elaboración Propia)

Para usar Ciudadanía Digital primero hay que registrar nuestra aplicación enviando los parámetros necesarios, Ciudadanía Digital nos da la posibilidad de autenticarnos al igual que lo haríamos con google, github y otros usando el estándar de autorización oauth2.0.

En la Figura 3.19 se aprecia el código que se ejecuta una vez presionado el botón principal, el cual toma los datos necesarios recogidos de la configuración del sistema como también de la librería oic de python.

Figura 3.20: Autenticación Ciudadanía Digital: Usuario

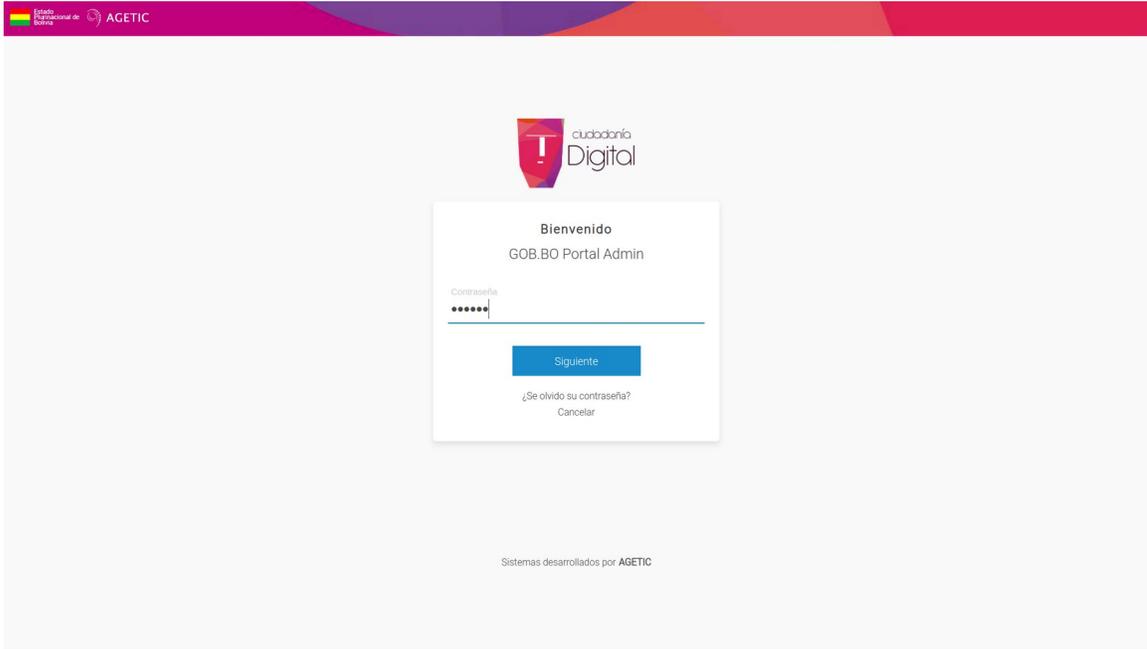


The image shows a login interface for 'ciudadanía Digital'. At the top left, there is a logo for 'AGETIC'. The main content area features the 'ciudadanía Digital' logo and the text 'Iniciar Sesión' and 'GOB.B0 Portal Admin'. Below this, there is a text input field labeled 'Usuario' with a blue underline. Underneath the input field are two buttons: a blue 'Siguiete' button and a 'Cancelar' button. At the bottom of the page, it says 'Sistemas desarrollados por AGETIC'.

Fuente: (AGETIC 2020)

En la **Figura 3.20** tenemos la pantalla en donde podemos ingresar nuestro usuario, en este caso al estar registrados en Ciudadanía Digital nuestro usuario sera el número de carnet de identidad.

Figura 3.21: Autenticación Ciudadanía Digital: Contraseña



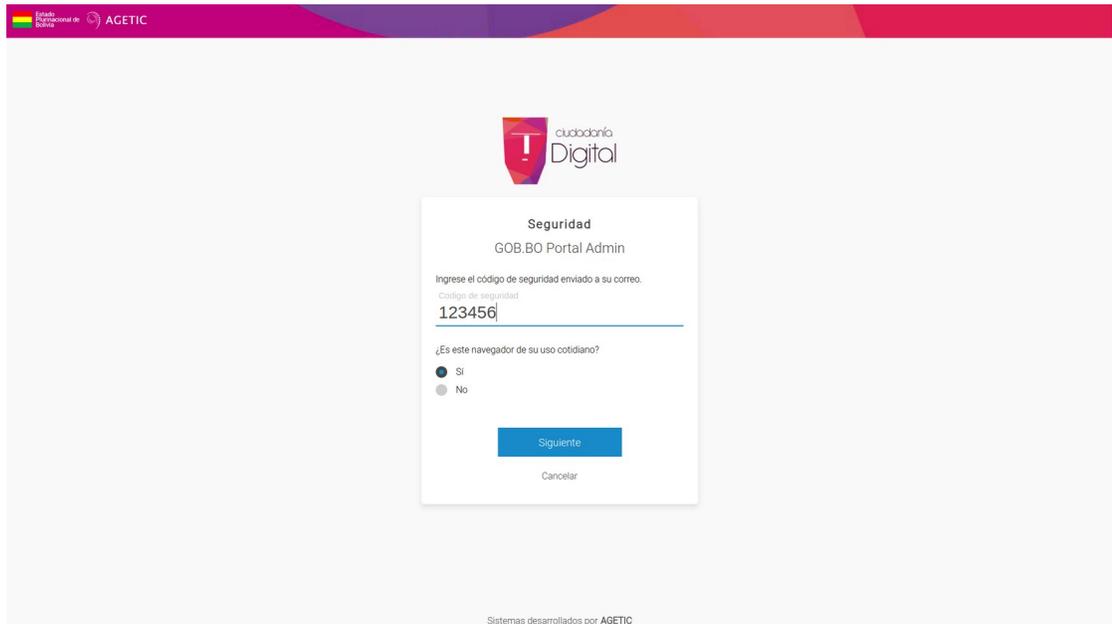
The image shows a web interface for digital citizenship authentication. At the top left, there is a header with the Spanish flag, the text 'Estado Boliviano de Bolivia', and the logo 'AGETIC'. The main content area features the 'ciudadanía Digital' logo, which consists of a stylized 'I' inside a shield. Below the logo, the text reads 'Bienvenido' and 'GOB.BO Portal Admin'. There is a password input field labeled 'Contraseña' with a masked password '*****'. A blue button labeled 'Siguiente' is positioned below the input field. Underneath the button, there is a link that says '¿Se olvidó su contraseña?' and a 'Cancelar' option. At the bottom of the page, it says 'Sistemas desarrollados por AGETIC'.

Fuente (AGETIC 2020)

La **Figura 3.21** nos muestra la pantalla siguiente a la introducción del usuario en la cual pondremos nuestra respectiva contraseña de Ciudadanía Digital.

Puede darse el caso de habernos olvidado la contraseña por lo cual contiene la opción “¿Se olvidó su contraseña?” y así poder restablecer su contraseña.

Figura 3.22: Autenticación Ciudadanía Digital: Código



Fuente: (AGETIC)

Figura 3.23: Código Verificar Usuario Registrado y Estado Activo

```
try:
    usuario, token = OpenIdAuth().autenticarOpenId(request, client, session["state"], datos)
    if usuario:
        datos["storage"] = {}
        datos["storage"]["token"] = token
        datos["storage"]["user"] = usuario
        datos["redirect_url"] = URL_FRONTEND_HOME
        return respuesta.exito(datos)
    datos["redirect_url"] = URL_FRONTEND
    return respuesta.exito_con_errores(datos, "Usuario no registrado en el sistema.")
except Exception as e:
    return respuesta.error(e)
```

Fuente: (Elaboración Propia)

En la **Figura 3.22** nos encontramos con la pantalla de código, este código es enviado a nuestro correo o a nuestro celular usando la aplicación de Ciudadanía Digital.

El código funciona como segundo factor el cual válida que la persona sea quien dice ser, este segundo factor lo podemos ver mas a menudo cuando pagamos algún servicio por Internet (bancos).

La **Figura 3.23** muestra el código donde se recibe los datos encontrados del usuario en ciudadanía digital, los cuales verificamos en nuestro sistema para ver si este figura o no, también verificamos que el usuario tenga un estado activo.

En la **Figura 3.24** visualizamos la pantalla principal del sistema de hemeroteca, la cual nos muestra en la esquina superior derecha un botón en forma de usuario con el cual tendrá acceso al menú principal.

Esta pantalla nos muestra todos los periódicos digitales registrados y la cantidad de noticias encontradas por cada uno en el día. También podemos apreciar que cada periódico digital tiene un botón mediante el cual podremos actualizar los datos de un determinado sitio web(períodico digital).

Figura 3.24: Pantalla Principal Administrador



Fuente: (Elaboración Propia)

Podemos registrar nuevos periódicos con el botón Adicionar Sitio Web, se llama sitio web ya que siendo un sistema que usa el web scraping podríamos sacar los datos de cualquier página no solamente de periódicos digitales, también podemos usar revistas, artículos y otros siempre y cuando tengamos la dirección web y lo registremos con los parámetros correctos.

Figura 3.25: Pantalla Crear y Editar Sitio Web

Editar Sitio Web

Titulo: Los Tiempos

Url: http://www.lostiempos.com

Configuración:

```
1- |
2- | "options": null,
3- | "properties": {
4- |   "autor": [
5- |     "//div[@class=\"node-content\"]/div[@class=\"autor\"]"
6- |   ],
7- |   "contenido": [
8- |     "//div[@property=\"content:encoded\"]",
9- |     "//div[@class=\"body\"]"
10- |   ]
11- | }
```

Cerrar Aceptar

Fuente: (Elaboración Propia)

Figura 3.26: Código: Registro de Sitio Web

```
@login_required
def post(self):
    try:
        data = json.loads(request.data.decode('utf-8'))
        resultado = SitioWebLogic.crear_actualizar_registro(data) # noqa
        return respuesta.exito(resultado, 'Operación realizada exitosamente',
    except Exception as e:
        return respuesta.error(e)
```

Fuente: (Elaboración Propia)

Figura 3.25 En esta figura podemos ver el formulario de creación y edición de un sitio web. Este es un modal abierto al presionar el botón editar un sitio web visto en la **Figura 3.24**.

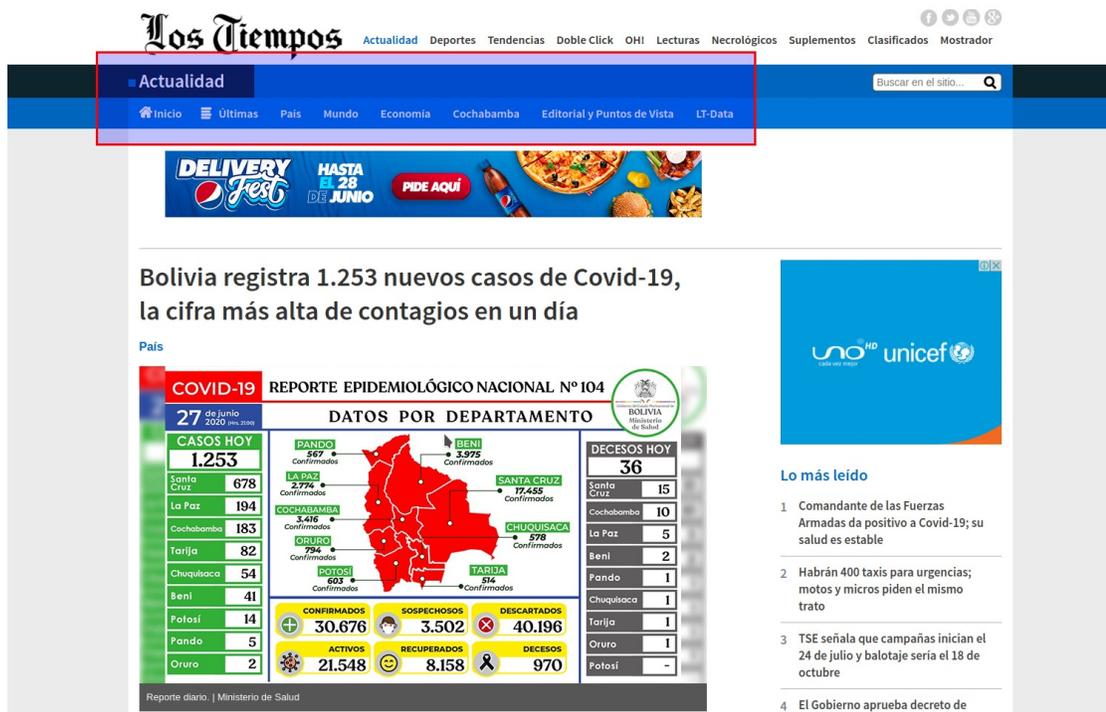
Título: Este campo es solo un texto.

Url: Se refiere a la dirección web en donde se publican las noticias.

Configuración: La configuración consta de un JSON el cual tiene los XPATHS para ubicar el contenido dentro del código fuente. Este contenido recuperado con los xpaths nos ayudan a recuperar el autor, contenido, fecha, imagen, título, subtítulo y menú principal del sitio web.

La **Figura 3.26** nos muestra el código que se ejecuta al crear un sitio web primeramente activándose el middleware llamando a `@login_required` y posteriormente enviando los datos a la capa de lógica de negocio.

Figura 3.27: Página de Noticias "Los Tiempos"



Fuente: (Los Tiempos)

Fuente: (Elaboración Propia)

En la **Figura 3. 27** podemos apreciar el periódico digital “Los Tiempos” dentro de el cual remarcamos su menú, esto se debe a que es de donde nosotros obtendremos los enlaces a las secciones del periódico, una vez obtenidos todos los enlaces ingresamos a cada una de ellos y obtendremos las noticias publicadas.

Figura 3.28 en esta figura podemos apreciar el modo desarrollador del navegador Google Chrome, también podemos usar otros navegadores.

Figura 3.28: Xpath que recupera el Título de la Noticia

The image shows a screenshot of the Los Tiempos website. The main headline reads: "Bolivia registra 1.253 nuevos casos de Covid-19, la cifra más alta de contagios en un día". Below this is a section titled "REPORTE EPIDEMIOLÓGICO NACIONAL N° 104" with a date of "27 de Junio 2020". It includes a map of Bolivia and a table of data by department. The table shows the following data:

Departamento	Casos Hoy
Santa Cruz	678
La Paz	194
Cochabamba	183
Tarija	82
Chuquisaca	54
Beni	41
Potosí	14
Fando	5
Oruro	2

Summary statistics: CASOS HOY: 1.253; CONFIRMADOS: 30.676; SUSPECHOSOS: 3.502; DESCARTADOS: 40.196; ACTIVOS: 21.548; RECUPERADOS: 8.158; DECESOS: 970.

On the right side of the page, there is an advertisement for "ENTRE BOLIVIANOS" and a section titled "Lo más leído" with three items:

- Comandante de las Fuerzas Armadas da positivo a Covid-19; su salud es estable
- Habrán 400 taxis para urgencias; motos y micros piden el mismo trato
- TSE señala que campañas inician el 24 de julio y balotaje sería el 18 de octubre

The developer tool window on the right shows the following XPath expression: `//*[@header/h1]`. The selected element is an `h1` tag with the text "Bolivia registra 1.253".

Fuente: (Elaboración Propia)

Los Xpath's nos permite construir expresiones que recorren y procesan un documento en nuestro caso la página web de el diario “Los Tiempos”.

En la figura se puede apreciar el remarcado del cursor apuntando al buscador del modo desarrollador de Google Chrome el cual contiene lo siguiente.

Figura 3.29: Ruta Xpath que Obtiene el Título de la Noticia

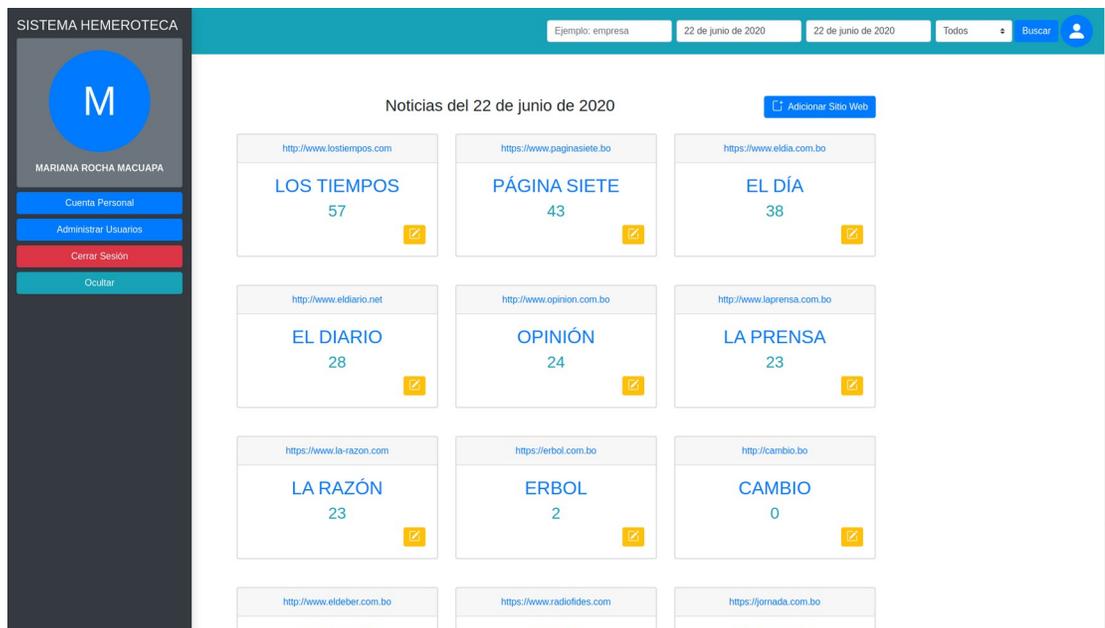


Fuente: (Elaboración Propia)

La **Figura 3.29** nos muestra `//*/header/h1` el cual corresponde al xpath de el título de la noticia actual.

De la misma manera podemos obtener diferentes partes del documento, en nuestro caso nos ayudará a obtener los datos de una noticia como ser el título, subtítulo, autor, contenido, fecha, imagen de una noticia.

Figura 3.30: Pantalla Menú Principal Administrador



Fuente: (Elaboración Propia)

En la **Figura 3.30** podemos ver el menú de administración, el mismo podemos abrirlo al presionar el icono de usuario que se encuentra en la esquina superior derecha del sistema.

Figura 3.31: Pantalla Administrar Usuarios

SISTEMA HEMEROTECA

Ejemplo: empresa 22 de junio de 2020 22 de junio de 2020 Todos + Buscar

MARIANA ROCHA MACUAPA

Cuenta Personal

Administrar Usuarios

Cerrar Sesión

Ocultar

Usuarios

Adicionar Usuario

Acciones	Carnet	Nombres	Apellido Paterno	Correo	Estado
	7024739	Roy Reportes	Aroa	roivan24@gmail.com	ACTIVO
	4206088	MARIANA	ROCHA	roivan324@gmail.com	ACTIVO

Hemeroteca 2020

Fuente: (Elaboración Propia)

La **Figura 3.31** se muestra la pantalla de administración de usuarios, mediante esta pantalla podemos crear, actualizar y dar de baja a un usuario.

En la **Figura 3.32** visualizamos el modal de crear o editar un usuario, este modal podemos activarlo al presionar el botón de crear usuario o el icono de editar que se ve en la **Figura 3.31**.

Para crear un usuario este mismo tiene que estar registrado a Ciudadanía Digital.

Figura 3.32: Pantalla Crear, Editar Usuario

Crear Usuario

Nro Documento: 4206088 Fecha de Nacimiento: 19/09/1966 Buscar

Nombres: MARIANA Apellido Paterno: ROCHA

Apellido Materno: MACUAPA Nro Celular:

Correo: correo@gmail.com Estado:

Guardar Cerrar

Fuente: (Elaboración Propia)

Figura 3.33: Registro de Usuario

```
@login_required
def post(self):
    try:
        data = json.loads(request.data)
        UsuarioLogic().contrastar_ciudadania(data['ci'], data['fecha_nacimiento'])
        usuario = UsuarioLogic().crear_usuario(data)
        return respuesta.exito(utils.object_as_dict(usuario), 'Usuario creado con ex
    except Exception as e:
        return respuesta.error(e)
```

Fuente: (Elaboración Propia)

Nro Documento: Este campo se refiere al número de carnet del usuario.

Fecha de Nacimiento: Campo que contiene la fecha de nacimiento.



Este botón activará un servicio de contrastación con Ciudadanía Digital el cual nos retornara los datos básicos del usuario como ser: nombres, apellido paterno, apellido materno.

Nombres, Apellido Paterno, Apellido Materno: Estos campos son llenados automáticamente si se encontró el usuario en Ciudadanía Digital.

Estado: Este es el campo proveerá información sobre si el usuario es activo y puede entrar al sistema o si está inactivo y lo bloqueamos.

La **Figura 3.33** nos muestra el código que se ejecuta para crear un usuario, el cual primeramente verifica la sesión y posteriormente contrasta su existencia con Ciudadanía Digital.

Figura 3.34: Pantalla Principal Noticias Encontradas

SISTEMA HEMEROTECA Ejemplo: empresa 22 de junio de 2020 22 de junio de 2020 Todos Buscar

Noticias del 22 de junio de 2020

Total: 238 noticias

1 / 238 Opinión Sirmes exige garantías laborales para profesionales de salud 22 de junio de 2020

2 / 238 Los Tiempos Gobierno reforzará presencia militar y policial en El Alto tras ataque a funcionarios 22 de junio de 2020

3 / 238 ERBOL Mesa: "En el momento en que la Presidenta promulga la ley se hace plenamente responsable" 22 de junio de 2020

El PSG reanuda su actividad con la vista puesta en la Champions 22 de junio de 2020

«Cobarde», «Incongruente», los calificativos de Camacho y Mesa a Añez por promulgación de ley de comicios 22 de junio de 2020

Ortiz informa que Presidente del TSE insistió en la fecha de elecciones pese a las proyecciones sobre la pandemia 22 de junio de 2020

Fuente: (Elaboración Propia)

La **Figura 3.34** nos muestra la pantalla principal de noticias las cuales se encontraron el 22 de junio de 2020, se logró acceder a esta pantalla al

presionar el botón buscar encontrado en la parte superior del sistema.

En la pantalla podemos apreciar dos campos que contienen la fecha actual por defecto, es por eso que al presionar el botón buscar nos muestra las noticias encontradas en el día.

Figura 3.35: Código: Búsqueda de Noticias

```
page = 1
if request.args.get('page'):
    page = int(request.args.get('page'))

resultado = NoticiaLogic.obtener_noticias(fields, query, limit, page) # noqa

return respuesta.exito(resultado['datos'], 'Noticias obtenidas exitosamente', 200, resultado['metadatos'])
except Exception as e:
    logging.error('Error al obtener la noticia', e)
return respuesta.error(e)
```

Fuente: (Elaboración Propia)

En la **Figura 3.35** se puede apreciar el código ejecutado al presionar “buscar” en la pantalla principal vista en la **Figura 3.34**, lo primero que verifica es la página en la se encuentra, posteriormente enviamos los datos a la Lógica de Negocio el cual devolverá las noticias encontradas.

El sistema trajo consigo un total de 238 noticias encontradas en el transcurso del día los cuales podemos ver en el cuerpo del mismo, también podemos apreciar que nos trajo consigo noticias de todos los periódicos registrados gracias a que el botón de selección situado en la parte superior del sistema tiene el dato “Todos”.

La **Figura 3.36** nos muestra el contenido de la noticia mostrando los datos obtenidos en el web scraping, esta página es accesible al hacer clic en cualquier noticia encontrada en la **Figura 3.34**.

Figura 3.36: Pantalla Leer Noticia



Fuente: (Elaboración Propia)

Administrador: Todas las figuras mostradas en este punto son accesibles por el administrador del sistema.

Usuario Común: Para ver las noticias no es necesario tener una cuenta en el sistema, simplemente bastará con ingresar a la url del sistema de hemeroteca el cual nos mostrará las pantallas vistas en la **Figura 3.30**, **Figura 3.34** y **Figura 3.36** un usuario común no puede ver los botones de editar, solo podrá ver y leer las noticias como también buscarlas dentro de un rango de fechas.

Todas las figuras encontradas en esta fase son de interacción con el usuario, por detrás se automatizaron las sincronizaciones respectivas para lograr obtener noticias cada día.

3.7 PRUEBAS DE SOFTWARE

3.7.1 Pruebas del Sistema

En esta fase se realizará las pruebas de los procesos críticos del sistema, para este caso emplearemos las pruebas de caja blanca y caja negra.

3.7.1.1 Pruebas de Caja Negra

Las pruebas de caja negra, también denominadas pruebas de comportamiento, se centran en los requisitos funcionales del software.

Para realizar la prueba de caja negra se toman los casos de Búsqueda de Noticias, Registro y Publicación de Noticias los cuales se muestran en las siguientes tablas.

La **Tabla 4.6** nos muestra la prueba de caja negra para el caso de búsqueda de noticias, como es un buscador este no debería enviar mensaje de error tal como el buscador de google.

Tabla 4.6: Prueba de Caja Negra Caso de Uso Búsqueda de Noticias

PRUEBA 1	CASO DE USO BÚSQUEDA DE NOTICIAS - BÚSQUEDA DE NOTICIAS
Descripción	El usuario administrador o persona común realiza una búsqueda mediante el formulario.
Condiciones de Ejecución	1.- Ninguna Llenar las entradas de búsqueda.
Entradas y Acciones del Usuario	1.- En el formulario introducir rango de fecha. 2.- En la búsqueda introducir una palabra o frase. 3.- En el formulario de búsqueda seleccionar el periódico.

	Luego de haber realizado las operaciones descritas en las entradas y acciones del Usuario deberá mostrarse lo siguiente respectivamente:
Resultado Esperado	<ol style="list-style-type: none"> 1.- El sistema muestra noticias encontradas en el rango de fechas. 2.- El sistema muestra noticias que contengan la palabra. 3.- El sistema muestra las noticias del periódico respectivo.

Fuente: (Elaboración Propia)

Para realizar la prueba de caja negra se toman los casos de registro de sitio web los cuales se muestran y describen en la siguiente tabla.

Tabla 4.7: Prueba de Caja Negra Caso de Uso Registro

PRUEBA 2	CASO DE USO REGISTRO DE SITIO WEB – REGISTRO DE SITIO WEB
Descripción	El usuario administrador efectúa el registro de un nuevo sitio web.
Condiciones de Ejecución	<ol style="list-style-type: none"> 1.- Usuario debe estar autenticado. Llenar el formulario de registro de Sitio Web.
Entradas y Acciones del Usuario	<ol style="list-style-type: none"> 1.- Si no se registran los campos obligatorios. 2.- Si se duplica la URL del periódico digital. 3.- Incluir un JSON en configuración.
Resultado Esperado	Luego de haber realizado las operaciones descritas en las entradas y acciones del Usuario deberá mostrarse lo siguiente respectivamente:

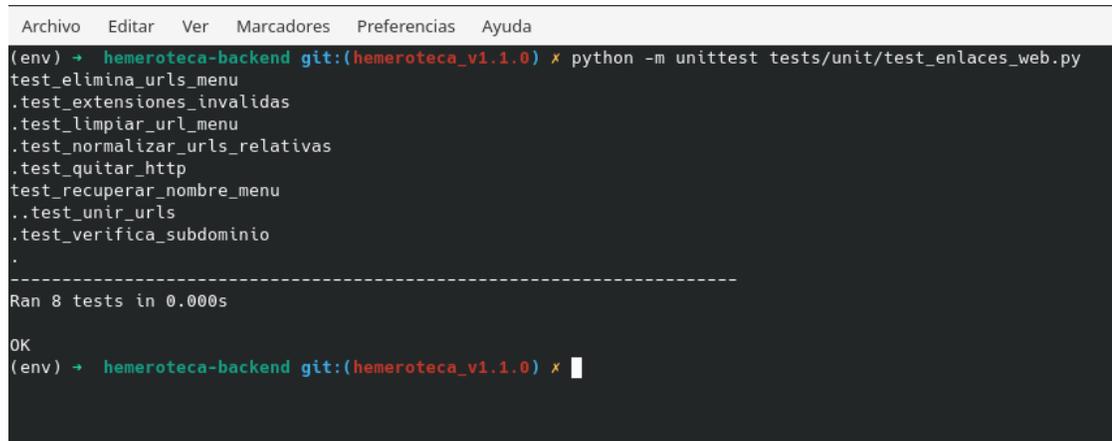
-
- 1.- Muestra mensaje de Error: Ingrese campos requeridos.
 - 2.- Muestra mensaje de Error: Url registrada.
 - 3.- Muestra mensaje de Error: Por favor ingrese configuración en formato JSON.
-

Fuente: (Elaboración Propia)

3.7.1.2 Pruebas de Caja Blanca

Para realizar esta prueba se utilizó librerías de test en nuestro caso usaremos Unittest con el cual realizaremos pruebas unitarias y de integración, los cuales se muestran en las siguientes figuras:

Figura 4.37: Testing de Pruebas Unitarias



```
Archivo  Editar  Ver  Marcadores  Preferencias  Ayuda
(env) → hemeroteca-backend git:(hemeroteca_v1.1.0) x python -m unittest tests/unit/test_enlaces_web.py
test_elimina_urls_menu
.test_extensiones_invalidas
.test_limpiar_url_menu
.test_normalizar_urls_relativas
.test_quitar_http
test_recuperar_nombre_menu
..test_unir_urls
.test_verifica_subdominio
.
-----
Ran 8 tests in 0.000s

OK
(env) → hemeroteca-backend git:(hemeroteca_v1.1.0) x
```

Fuente: (Elaboración Propia)

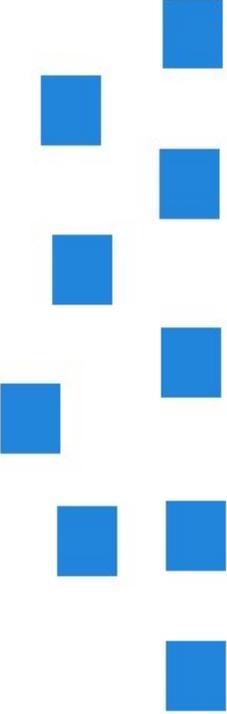
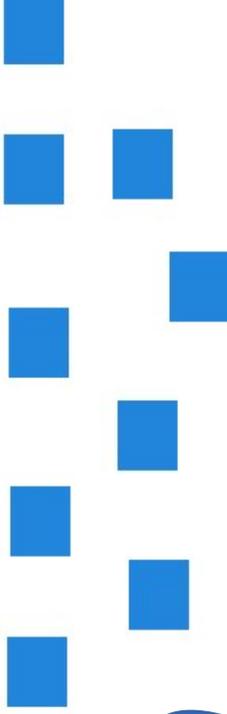
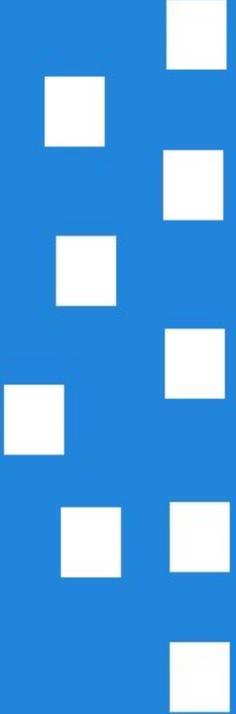
Las pruebas unitarias de la **Figura 4.37** nos dicen que el sistema no presenta fallas y se cubren todos los casos al momento de recuperar una url de noticia correctamente.

Figura 4.38: Testing de Pruebas de Integración

```
Archivo  Editar  Ver  Marcadores  Preferencias  Ayuda
(env) → hemeroteca-backend git:(hemeroteca_v1.1.0) x python -m unittest tests/integration/test_crear_noticia_por_url
.py
...
Obtener datos url 1
Obtener datos url 2
Obtener datos url 3
Obtener datos url 4
Obtener datos url 5
Obtener datos url 6
Obtener datos url 7
Obtener datos url 8
Obtener datos url 9
Obtener datos url 10
Obtener datos url 11
Obtener datos url 12
Obtener datos url 13
Obtener datos url 14
-----
Ran 14 tests in 9.673s
OK
```

Fuente: (Elaboración Propia)

Las pruebas de Integración de la **Figura 4.38** afirman que el sistema no presenta errores al momento de extraer la noticia, título, imagen, autor, fecha, con lo cual se afirma que el sistema cuenta con un funcionamiento óptimo y no presenta fallas.



CAPÍTULO IV

MÉTRICAS

DE CALIDAD

Y COSTO

CAPÍTULO IV

4 MÉTRICAS DE CALIDAD Y COSTO

4.1 MÉTRICAS DE CALIDAD

Se realizará la medición de la calidad de software usando la métrica ISO-9126, que establece que cualquier componente de la calidad de software puede ser descrito en término de una o mas de seis características básicas las cuales son funcionalidad, confiabilidad, eficiencia, mantenibilidad y portabilidad (Figueroa, 2005)

4.1.1 Funcionalidad

La funcionalidad del sistema se mide a través de la complejidad, a través del punto de Función (PF), que permite un resultado medible y cuantificable a partir de la siguiente fórmula:

$$PF = \text{conteo total} \times [0.65 + 0.01 \times \sum (Fi)]$$

Para realizar el cálculo de la cuenta total de los factores de ponderación se debe realizar la siguiente tabla:

Tabla 4.1: Factores de Ponderación

PARÁMETROS DE MEDICIÓN	CUENTA	FACTOR(MEDIA)	TOTAL
Entrada de usuario	16	4	64
Salidas de Usuario	31	5	155
Peticiones de Usuario	12	4	48
Número de archivos	36	10	360
Número de interfaces externas	2	7	14

Cuenta Total	641
--------------	-----

Fuente: Elaboración en base a criterios (Pressman, 2007)

Los valores de ajuste de complejidad se deben indicar según corresponda a las siguientes preguntas en la escala de 0 a 5:

Tabla 4.2: Valores de ajuste de Complejidad

ESCALA	Sin	incidental	Moderado	Medio	Significativo	Esencial	Valor
	importancia						
FACTOR	0	1	2	3	4	5	
1. ¿Requiere el sistema copias de seguridad y de recuperación fiables?						X	5
2. ¿Se requiere comunicación de datos?						X	5
3. ¿Existen funciones de procesamiento distribuido?						X	5
4. ¿Es crítico el rendimiento?					X		4
5. ¿Será ejecutado el sistema en un entorno operativo existente y fuertemente utilizado?						X	5
6. ¿Requiere el sistema entrada de datos interactiva?						X	5
7. ¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?				X			3
8. ¿Se actualizan los archivos maestros de forma interactiva?				X			3
9. ¿Son complejas las entradas, las salidas, los archivos o las peticiones?				X			3
10. ¿Es complejo el procesamiento interno?					X		4
11. ¿Se ha diseñado la conversación y la instalación?						X	5
12. ¿Están incluidas en el diseño la conversación y la instalación?						X	5
13. ¿Se ha diseñado el sistema para soportar						X	5

múltiples instalaciones en diferentes organizaciones?		
14. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	X	5
TOTAL		62

Fuente: Elaboración en base a criterios (Pressman, 2007)

Para calcular los puntos de función se utiliza la siguiente ecuación:

$$PF = \text{conteo total} \times [0.65 + 0.01 \times \sum (Fi)]$$

Donde:

Punto Función: Medida de funcionalidad entregada a la aplicación como valor de normalización.

Conteo Total: Es el nivel de complejidad del sistema respecto al usuario.

[0.65 + 0.01 * ΣFi]: Ajuste de la complejidad según el dominio de la información de software, en donde ΣFi son los valores de complejidad.

0.01: Factor de conversión, es decir, un error del 1%.

0.65: Valor mínimo del ajuste de complejidad.

La expresión ΣFi se determina a partir de los datos que se muestran en la **Tabla 4.4** de valores de ajuste de complejidad.

Reemplazando en la ecuación:

$$PF = 641 \times [0.65 + 0.01 \times 62]$$

$$PF = 814.17 \text{ (Valor de Punto de Función)}$$

Si calculamos al 100% el nivel de confianza consideramos la sumatoria de **Fi=70** como el máximo valor de ajuste de complejidad se tiene:

$$PF_{max} = 641 \times [0.65 + 0.01 \times \sum (Fi)]$$

$$PF_{max} = 641 \times [0.65 + 0.01 \times 70]$$

$$PF_{max} = 865.35$$

Con los máximos valores de ajuste de complejidad se tiene la funcionalidad real:

$$Funcionalidad = \frac{814.17}{865.35}$$

$$Funcionalidad = 0.94$$

$$Funcionalidad = 0.94 \times 100 = 94\%$$

El resultado de 94% quiere decir que el sistema tiene un porcentaje alto de que funcione sin riesgo y un 6% de que pueda suceder un colapso en algún momento.

4.1.2 Usabilidad

La usabilidad es la facilidad de uso, esta métrica nos muestra el costo de aprender el producto, para lo cual se tiene la siguiente fórmula:

$$Usabilidad = \left[\left(\frac{\sum (Xi)}{n} \right) \times 100 \right]$$

En la **Tabla 4.3** se debe responder a las preguntas, obteniendo el resultado en base a la ecuación de usabilidad.

Tabla 4.3: Ajuste de preguntas

Factor	Respuesta		Ponderación
	SI	NO	% (x ₁)
¿Puede utilizar con facilidad el sistema?	4	1	80%
¿Puede controlar operaciones que el sistema solicite?	4	1	80%
¿Las respuestas del sistema no son complicadas?	5	0	100%
¿El sistema permitió la retroalimentación de información?	5	0	100%

¿El sistema cuenta con interfaz agradable a la vista?	5	0	100%
¿La respuesta del sistema es satisfactoria?	5	0	100%
¿Le parece complicadas las funciones del sistema?	4	1	80%
Se hace difícil o dificultoso aprender a manejar el sistema	3	2	60%
¿Los resultados que proporciona el sistema facilitan el trabajo?	5	0	100%
¿Durante el uso del sistema se produjo errores?	4	1	80%
USABILIDAD			88%

Fuente: (Elaboración Propia)

El resultado nos dice que hay un 88% de comprensión o entendimiento de los usuarios respecto al sistema.

4.1.3 Confiabilidad

La confiabilidad del sistema se define como la probabilidad de operación libre de fallos.

Para calcular la confiabilidad del sistema se toma en cuenta el periodo de tiempo en el cual se ejecuta y se obtiene muestras.

$$F(t) = f \times e^{(-u \times t)}$$

En el inicio de ejecución $t_0=0$ lo que significa el tiempo inicial en el cual dará inicio el funcionamiento del sistema.

El tiempo en el cual el sistema trabaja sin falla está dado por la ecuación (3) y tiempo en el cual no falla el sistema está dado por la ecuación (4).

$$P(T \leq t) = F(t)$$

$$P(T \leq t) = F(t)$$

En un periodo de 20 días como tiempo de prueba de cada 20 ejecuciones una falla.

Conociendo la funcionalidad del 94% del sistema calculamos para el periodo establecido.

$$P(T \leq t) = 1 - F(t)$$

$$F(t) = 1 - 94 \times e^{(1/20 + 20)}$$

$$F(t) = 0.99$$

$$F(t) = 0,99 \times 100 = 99 \%$$

La confiabilidad del sistema es del 99% en un periodo de 20 días como tiempo de prueba.

4.1.4 Mantenibilidad

El mantenimiento se desarrolla para mejorar el sistema en respuesta a los nuevos requerimientos.

El estándar IEE94 sugiere un índice de madurez del software (IMS) que proporciona un indicador en la estabilidad de un producto, se lo determina con la siguiente fórmula.

$$IMS = \frac{[Mt - (Fa + Fb + Fd)]}{Mt}$$

Donde:

Mt = Número de módulos de la versión actual

Fa = Número de módulos en la versión que se han añadido

Fc = Número de módulos en la versión actual que se han cambiado

Fd = Número de módulos en la versión anterior que se han borrado en la versión actual.

Calculando el IMS:

$$IMS = \frac{[8 - (1 + 2 + 0)]}{8}$$

$$IMS = 0.625$$

$$IMS = 0.625 \times 100 = 62.5 \%$$

Siendo el IMS del 62,5% nos dice que el sistema es moderadamente estable y el restante 37,5% es el margen de error correspondiente a los cambios y modificaciones.

El mantenimiento del software está dado por cuatro clases:

- ▶ **Correctivo:** Tiene como objetivo localizar y eliminar los posibles defectos de los programas. Un defecto es una característica del sistema con el potencial de causar una falla.
- ▶ **Adaptativo:** Consiste en la modificación de un programa debido a cambios en el entorno (hardware o software) en el cual se ejecuta. Estos cambios pueden afectar al sistema operativo, a la arquitectura física del sistema o al entorno de desarrollo del software.
- ▶ **Perfectivo:** Es el conjunto de actividades para mejorar o añadir nuevas funcionalidades requeridas del software.
- ▶ **Preventivo:** Consiste en la modificación del software para mejorar sus propiedades (calidad y/o mantenibilidad) sin alterar sus especificaciones funcionales.

4.1.4.1 Mantenimiento Preventivo

Para la prevención de pérdidas de datos y de información se deberán tomar en cuenta las siguientes recomendaciones.

- ▶ **Respaldos:** Se deberán obtener un Backup de la base de datos mensualmente, este backup deberá contener en el nombre, fecha y año

en el cual fue creado.

- ▶ **Caídas de sistema:** Es importante que el computador donde se implante la aplicación cuente con generadores de energía eléctrica para preservar la información de probables caídas de tensión.
- ▶ **Conectividad a Internet:** El sistema necesita contar con una conexión a internet constante.

Por los puntos descritos anteriormente, se concluye que el manteniendo es la etapa más costosa del ciclo de vida de un producto de software, la mayor porción del mantenimiento, por lo general, consiste en mejorar el sistema.

4.1.5 Portabilidad

Un sistema es portable, si el costo de transportar y adaptar la aplicación a un nuevo entorno es menor al costo de re diseñar el sistema para dicho ambiente.

La relación que permite determinar el grado de portabilidad es el siguiente:

A nivel de backend y frontend es portable bajo los sistemas operativos Windows, MacOs y Distribuciones Linux.

$$Portabilidad = 1 - \frac{A}{B}$$

A = Número de días para portar el sistema.

B = Número de días para implementar el sistema.

Entonces:

$$Portabilidad = 1 - \frac{0.5}{3 \text{ días}}$$

$$Portabilidad = 0.83$$

$$Portabilidad = 0.83 \times 100 = 83\%$$

La **Tabla 4.4** muestra el total obtenido de la sumatoria de los parámetros de medición.

Tabla 4.4: Resultado Final de las Métricas De Calidad

CUALIDAD	RESULTADO
Funcionalidad	94
Usabilidad	88
Confiabilidad	99
Mantenibilidad	62,5
Portabilidad	83
TOTAL	85,3

Fuente: (Elaboración Propia)

Pressman nos dice que si el valor de calidad encontrado es mayor a 60 el software es aceptable.

Podemos concluir que nuestro sistema tiene un nivel de calidad aceptable y cumple con la ISO 9126.

4.2 ANÁLISIS DE COSTO

4.2.1 COCOMO

Para calcular el esfuerzo, necesitamos hallar la variable KLDC (Kilo-líneas de código).

El proyecto se implementa en:

$$\text{Líneas de Código} = \frac{6020 \text{ Backend}}{3027 \text{ Frontend}}$$

$$\text{Líneas de Código} = 9047 \text{ Líneas}$$

Aplicando conversiones tenemos:

$$KLCD = \frac{(LDC)}{1000}$$

$$KLCD = \frac{(9047)}{1000}$$

$$KLCD = 9.047$$

En este caso el tipo orgánico es el más apropiado ya que el proyecto no supera los 50 KLCD por consiguiente los coeficientes que se usarán serán los siguientes: **Tabla 4.5**

Tabla 4.5: Coeficiente para medir el costo del software

PROYECTOS DE SOFTWARE	a	b	c	d
Orgánico	3.2	1.05	2.50	0.38

Fuente: (Pressman, 2007)

Para hallar los valores de de FAE, se utilizara la **Tabla 4.6** para el cálculo de los atributos FAE.

Tabla 4.6: Calcula de los atributos FAE

ATRIBUTOS	VALOR					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
Atributos de software						
Fiabilidad	0,75	0,88	1,00	1,15	1,40	

Tamaño de Base de datos	0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,65
Atributos de hardware					
Restricciones de tiempo de ejecución		1,00	1,11	1,30	1,66
Restricciones de memoria virtual		1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual	0,87	1,00	1,15	1,30	
Tiempo de respuesta	0,87	1,00	1,07	1,15	
Atributos de personal					
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90	
Experiencia en el lenguaje	1,14	1,07	1,00	0,95	
Atributos del proyecto					
Técnicas actualizadas de programación	1,24	1,10	1,00	0,91	0,82
Utilización de herramientas de software	1,24	1,10	1,00	0,91	0,83
Restricciones de tiempo de desarrollo	1,22	1,08	1,00	1,04	1,10
Total	0,975				

Fuente: (Elaboración Propia)

Por tanto nuestro factor de ajuste sera el resultado de la multiplicación de los valores evaluados de la **Tabla 4.6**

Por lo tanto el ajuste sera:

$$FAE = 0.975$$

Aplicando la fórmula de Esfuerzo E tenemos:

$$E = a \times KLCD^{(e)} \times FAE$$

$$E = a \times 9.047^{(1.05)} \times 0.975$$

$$E = 31.51 \text{ Personas/mes}$$

Ahora calculamos el tiempo de desarrollo T:

$$T = c \times E^d$$

$$T = 2.50 \times 31.51^{0.38}$$

$$T = 9.27$$

Después procedemos al Cálculo de Productividad PR:

$$PR = \frac{LCD}{E}$$

$$PR = \frac{9047}{31.51}$$

$$PR = 287.115 \text{ LCD/Personames}$$

Ahora calculamos el personal Promedio P:

$$PR = 287.115 \text{ LCD/Personames}$$

$$P = \frac{31.51}{9.27}$$

$$P = 3.39$$

$$P = 4 \text{ Personas Promedio}$$

Por último calculamos el Costo Mes del proyecto CM:

Sueldo Promedio Programador: 3500 Bs.

$$CM = P \times \text{Sueldo Promedio}$$

$$CM = 4 \times 3500$$

$$CM = 14000$$

$$\text{Costo Total} = T \times CM$$

$$\text{Costo Total} = 9.27 \times 14000$$

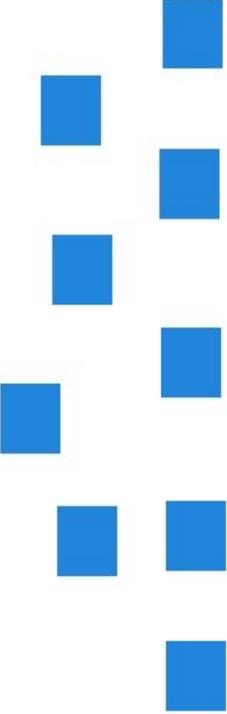
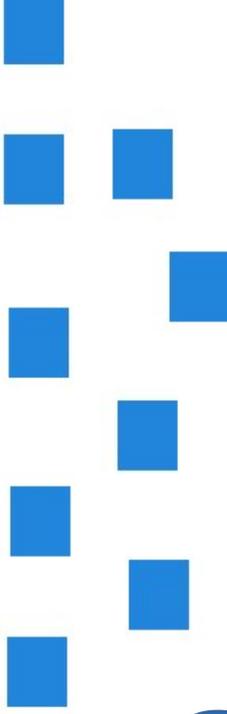
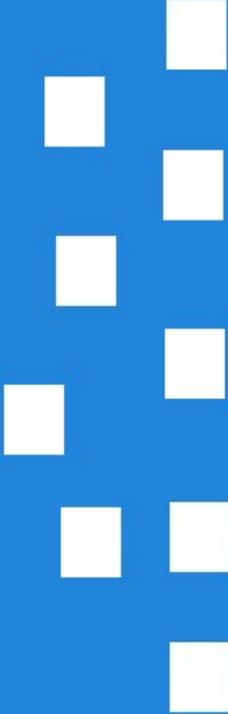
$$\text{Costo Total} = 129.780 \text{ Bs.}$$

Tabla 4.7: Resumen COCOMO

Símbolo	Descripción	Valor	Unidad
FAE	Factor de Ajuste	0,975	Número
E	Esfuerzo	31,51	Personas/Mes
T	Tiempo	9,27	Meses
PR	Productividad	287,115	LCD/Persona Mes
P	Persona	4	Persona
CM	Costo Mes	14000	Bolivianos.
CT	Costo Total	129780	Bolivianos

Fuente: (Elaboración Propia)

En resumen se requiere 4 personas por un lapso de 10 meses el cual nos dará un costo total de 129.780 Bs.



CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

El proyecto se ha desarrollado e implementado satisfactoriamente y el resultado final cumple de forma general con los objetivos inicialmente planteados en los capítulos anteriores, puesto que se ha conseguido desarrollar e implementar el sistema que permite obtener las noticias del día a día de los diferentes periódicos registrados en el sistema, tales serán puestas a disposición de la población.

El objetivo general descrito en el CAPÍTULO I:

“Desarrollar un sistema web que pueda obtener de las noticias del día a día de fuentes digitales legalmente establecidas (periódicos digitales) y guardarlos en nuestro sistema, para luego poner la información a disponibilidad de la población en general.”

Se puede concluir que el objetivo general fue alcanzado en un 84% dado que el sistema aún no fue publicado oficialmente pero si funciona internamente, en el CAPÍTULO III se logró diseñar, desarrollar e implementar el Sistema Web Orientado a la Recolección de Información de Medios Digitales Mediante Técnicas de Web Scraping, logrando centralizar la información del día a día.

A continuación se realiza la conclusión de objetivos específicos:

- ✓ *“Recolectar noticias digitales de fuentes periodísticas legalmente establecidas en nuestro país”*. Se cumplió al registrar en nuestro sistema solamente a periódicos digitales legalmente establecidos en el Estado Plurinacional de Bolivia.
- ✓ *“Recopilar noticias de varios periódicos y difundir la información en un*

único lugar". Se cumplió registrando varios periódicos de los cuales se obtienen las noticias del día a día. No se cumplió con la difusión de información porque el sistema no se publicó oficialmente.

- ✓ *"Mostrar las noticias mediante un buscador basado en fechas"*. Se cumplió dado que las noticias se registran con una fecha de publicación, por la cual podemos filtrar.
- ✓ *"Almacenar la información para un posterior análisis de datos"*. Se cumplió porque se guarda el histórico de noticias y no se los elimina en ningún momento.
- ✓ *"Aplicar pruebas de funcionamiento del sistema de información"*. Se cumplió realizando las pruebas de caja negra, caja blanca.
- ✓ *"Evaluar la calidad del sistema aplicando las métricas de calidad de software"*. Se cumplió realizando cálculos bajo las métricas de calidad de la norma ISO-9126.

5.2 RECOMENDACIONES

Debido al funcionamiento del sistema se recomienda tener una conexión estable a internet, este no ser interrumpido porque si así fuera puede ocurrir que este no puede navegar por las diferentes páginas de periódicos digitales y descargar el contenido.

Se debe realizar un mantenimiento de por lo menos una vez al mes, dado que el sistema operativo puede tender a ponerse lento cuando la Memoria RAM se llena.

El sistema cuenta con una base de datos cercana a los 50 GB, las imágenes de los periódicos ascienden a 200 GB por lo que se recomienda un disco duro de 1Tb y mínimamente 4 procesadores para un funcionamiento a largo plazo.

Para mejorar la velocidad de búsquedas se recomienda poner el servidor en modo cluster y de esta manera poder escalar cuando sea necesario.

Entre las mejoras que se le puede hacer al sistema se puede mencionar:

- ▶ Crear una app móvil la cual consuma la información de nuestro servidor y de esta manera sea mas fácil el acceso al sistema para el usuario final.
- ▶ Posteriormente podría añadirse la funcionalidad de poder compartir una noticia mediante las redes sociales.
- ▶ Crear reacciones al igual que lo hace facebook para saber qué noticias agradan a la población y cuáles no, con las cuales podríamos tener diferentes gráficas sobre las noticias más relevantes del día.

BIBLIOGRAFÍA

- Hurtado y Hernandez (2009), Diseño e Implementación de la biblioteca virtual especializada sobre los derivados de la caña de azucar.
- Jahaira. (2007). Hemeroteca Virtual Digital Biblioteca Central U.M.S.A.
- Naur, P. y B. Randall (1969), Software Engineering: A Report on a Conference Sponsored by the NATO Science Committee, NATO.
- Pressman, S. R. (1993) Ingeniería de Software: Un enfoque práctico. 3ra edición
- Pressman, S. R. (2002). Ingeniería de Software 5ta Edición. Mc Graw-Hill/Interamericana de España, S.A.U: Concepción Fernández Madrid.
- Pressman, S. R. (2005). Ingeniería del Software: Un Enfoque Práctico, 6ta Edición. Mc Graw-Hill/Interamericana de España, S.A.U: Concepción Fernández Madrid.
- Pressman, S. R. (2010). Ingeniería del Software Un enfoque Práctico: Un Enfoque Práctico, 6ta Edición. Mc Graw-Hill/Interamericana de España, S.A.U: Concepción Fernández Madrid.
- SAROKA, R. H. (2002). Sistemas de Información en la Era Digital. Argentina: Fundación OSDE.
- Boehm, B. (2000). Software cost Estimation with Cocomo II. Prentice Hall: Edición 1.
- Kendal, K. &. (2005). Analisis y Diseño: Sexta Edición .
- Sommerville, I. (2005). Ingeniería de Software. Séptima Edición.
- Jacobson, &. Booch .G. (2000). Proceso Unificado de Desarrollo de Software.
- Nora Koch & Andreas Kraus & Rolf Hennicker (2001). The Authoring Process of the UML-based Web Engineering Approach

DISPONIBLE EN:

Periódico en línea, párr. 1: Wikipedia (2017). Periódico en Línea. Obtenido de https://es.wikipedia.org/wiki/Periódico_en_línea.

OEA, párr. 1: OEA (s. f.). Gobierno Electrónico. Obtenido de https://www.oas.org/es/sap/dgpe/guia_egov.asp.

Software Libre, párr. 1: Wikipedia (Octubre, 2016). Software Libre. Obtenido de https://es.wikipedia.org/wiki/Software_libre.

ISO 9126, párr. 2: Ardila, N. I. (Marzo, 2013). Evaluación de software. Obtenido de <http://actividadreconocimiento-301569-8.blogspot.com/2013/03/norma-de-evaluacion-isoiec-9126.html>.

¿Qué es hardware y software?, párr. 1: GCFGlobal (s. f.). ¿Qué es hardware y software?. Obtenido de <https://edu.gcfglobal.org/es/informatica-basica/que-es-hardware-y-software/1/>.

Definición de Dato, párr. 1: Alegsa L. (2018). Dato. Obtenido de <http://www.alegsa.com.ar/Dic/dato.php>.

Dato. párr. 1: Wikipedia (2008). Dato Informatica. Obtenido de [https://es.wikipedia.org/wiki/Dato_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Dato_(inform%C3%A1tica)).

Características del sistema. párr. 2: Significados (2020). Qué es Sistema. Obtenido de <https://www.significados.com/sistema/>.

Principales características de la información. párr. 1: Wikipedia (2010). Información. Obtenido de <https://es.wikipedia.org/wiki/Información>.

¿Qué es Internet?. párr. 1: Raffino M. (2014). ¿Qué es Internet?. Obtenido de <https://concepto.de/internet/>.

Medios de Información, párr. 2: Pmoinformatica (Mayo, 2016). Requerimiento no funcional. Obtenido de <http://www.pmoinformatica.com/2015/05/requerimientos-no-funcionales-ejemplos.html>.

Qué es Noticia, párr. 1: Coelho, F (Mayo, 2019). Qué es Noticia. Obtenido de

<https://www.significados.com/noticia/>.

Fake news, párr. 1: Wikipedia (2018). Fake news. Obtenido de https://es.wikipedia.org/wiki/Fake_news.

Red Social, párr. 1: Wikipedia (2004). Red social. Obtenido de https://es.wikipedia.org/wiki/Red_social.

Medio Digital, párr. 1: Wikipedia (Abril, 2016). Medios digitales. Obtenido de https://es.wikipedia.org/wiki/Medios_digitales.

Periódico párr. 1: Wikipedia (Julio, 2012). Periódico. Obtenido de <https://es.wikipedia.org/wiki/Peri%C3%B3dico>.

Periódico en Linea, párr. 1: Wikipedia (2006). Periódico en Linea. Obtenido de https://es.wikipedia.org/wiki/Peri%C3%B3dico_en_l%C3%ADnea.

Requerimientos de la técnica, párr. 1: Wikipedia (Agosto, 2003). Técnica. Obtenido de <https://es.wikipedia.org/wiki/T%C3%A9cnica>.

Web scraping, párr. 2: Wikipedia (Diciembre, 2012). Web scraping. Obtenido de https://es.wikipedia.org/wiki/Web_scraping.

Técnicas, párr. 1: Wikipedia (Diciembre, 2012). Web scraping. Obtenido de https://es.wikipedia.org/wiki/Web_scraping.

Hemeroteca, párr. 1: Evelyn C. (Diciembre, 2013). Hemeroteca. Obtenido de <https://www.ecured.cu/Hemeroteca>.

Que es una hemeroteca, párr. 1: Boté J. (Abril, 2013). ¿Qué es una hemeroteca y para que sirve?. Obtenido de <https://juanjobote.com/que-es-una-hemeroteca-y-para-que-sirve/>.

Definición de Hemeroteca, párr. 1: Ucha F. (Noviembre, 2009). Definición de Hemeroteca. Obtenido de <https://www.definicionabc.com/comunicacion/hemeroteca.php>.

Hemeroteca Definición, párr. 1: Rossana A. (Agosto, 2019). Hemeroteca. Obtenido de <https://conceptodefinicion.de/hemeroteca/>.

Función, párr. 1: Evelyn C. (Diciembre, 2013). Hemeroteca. Obtenido de <https://www.ecured.cu/Hemeroteca>.

Funciones de las hemerotecas, párr. 1: Rossana A. (Agosto, 2019). Hemeroteca. Obtenido de <https://conceptodefinicion.de/hemeroteca/>.

Tipos de Hemeroteca, párr. 1: Rosana (2019). Hemeroteca. Obtenido de <https://conceptodefinicion.de/hemeroteca/>.

Importancia, párr. 1-3: Rossana A. (Agosto, 2019). Hemeroteca. Obtenido de <https://conceptodefinicion.de/hemeroteca/>.

Las ventajas que obtienes en una hemeroteca, párr. 1: Boté J. (Abril, 2013). ¿Qué es una hemeroteca y para que sirve?. Obtenido de <https://juanjobote.com/que-es-una-hemeroteca-y-para-que-sirve/>.

Hemeroteca de Medios, párr. 2: Auditmedia (Noviembre, 2015). ¿Qué es una hemeroteca de medios?. Obtenido de <http://www.auditmedia.es/Post/que-es-una-hemeroteca/>.

Qué ofrece una hemeroteca de medios, párr. 1-3: Auditmedia (Noviembre, 2015). ¿Qué es una hemeroteca de medios?. Obtenido de <http://www.auditmedia.es/Post/que-es-una-hemeroteca/>.

Tipos de hemerotecas de medios, párr. 1: Auditmedia (Noviembre, 2015). ¿Qué es una hemeroteca de medios?. Obtenido de <http://www.auditmedia.es/Post/que-es-una-hemeroteca/>.

Requerimientos no funcionales: Ejemplo, párr. 2: Pmoinformatica (Mayo, 2016). Requerimiento no funcional. Obtenido de <http://www.pmoinformatica.com/2015/05/requerimientos-no-funcionales-ejemplos.html>.

Identificación de Requerimientos no funcionales, párr. 1: Google Sites (s.f.). Metodología Gestión de Requerimientos. Obtenido de <https://sites.google.com/site/metodologiareq/capitulo-ii/tecnicas-para-identificar-requisitos-funcionales-y-no-funcionales>.

Dificultades para Definir los Requerimientos, párr. 1: Blogspot (Marzo, 2012). Memorias dentro del Desarrollo de Software. Obtenido de <http://phigux.blogspot.com/2012/03/dificultades-para-definir-los.html>.

Arquitectura de software, párr. 1: Wikipedia (Mayo, 2005). Arquitectura de software. Obtenido de https://es.wikipedia.org/wiki/Arquitectura_de_software.

¿Qué es un patrón arquitectónico?, párr. 1: Cori W. (Septiembre, 2018). Los 10 patrones comunes de arquitectura de software. Obtenido de <https://medium.com/@maniakhitoccori/los-10-patrones-comunes-de-arquitectura-de-software-d8b9047edf0b>.

Arquitecturas más comunes, párr. 1: Wikipedia (Mayo, 2005). Arquitectura de software. Obtenido de https://es.wikipedia.org/wiki/Arquitectura_de_software.

Elementos que forman parte de la arquitectura cliente-servidor, párr. 1: OposicionesTic (2011). Arquitectura cliente servidor. Obtenido de <https://oposicionestic.blogspot.com/2011/06/arquitectura-cliente-servidor.html>.

Tipos de pruebas de Caja Blanca, párr. 1: Salazar E. (Enero, 2015). Procedimiento para realizar pruebas de Caja Blanca. Obtenido de <http://www.informatica-juridica.com/trabajos/procedimiento-realizar-pruebas-caja-blanca/>.

Qué es Python, párr. 3: Alvarez M. (Noviembre, 2003). Qué es Python. Obtenido de <https://desarrolloweb.com/articulos/1325.php>.

Introducción a Flask, párr. 1: ManualWeb (s. f.). Introducción a Flask. Obtenido de <http://www.manualweb.net/flask/introduccion-flask/>.

¿Qué tiene de bueno flask?, párr. 1-3: Andrearrs (Agosto, 2014). Flask: minimalismo para el desarrollo web en Python. Obtenido de <https://hipertextual.com/archivo/2014/08/flask-python/>.

¿Qué es PostgreSQL?, párr. 1-3: HostingPedia (Febrero, 2019). PostgreSQL. Obtenido de <https://hostingpedia.net/postgresql.html>.

Introducción a Celery, párr. 1: Aaloy (Enero, 2011). INTRODUCCIÓN A CELERY. Obtenido de <https://www.apsl.net/blog/2011/01/14/introduccion-a-celery/>.

¿Qué es RabbitMQ?, párr. 1-4: Mesa L. (Enero, 2019). Conozcamos sobre

RabbitMQ, sus componentes y beneficios. Obtenido de <https://www.pragma.com.co/academia/lecciones/conozcamos-sobre-rabbitmq-sus-componentes-y-beneficios>.

¿Qué es Elasticsearch?, párr. 1-3: David Ochobits (Agosto, 2018). ¿Qué es y cómo funciona Elasticsearch?. Obtenido de <https://www.ochobitshacenunbyte.com/2018/08/28/que-es-y-como-funciona-elasticsearch/>.

Qué es JavaScript: primeros pasos, párr. 2: Lucas Pelhom (2019). Qué es JavaScript: primeros pasos. Obtenido de <https://www.hostgator.mx/blog/que-es-javascript/>.

¿Qué es JavaScript?, párr. 1-3: Giampiere J. (Mayo, 2016). ¿Qué es JavaScript?. Obtenido de <https://devcode.la/blog/que-es-javascript/>.

Vue, párr. 1-4: Garcia E. (Abril, 2019). ¿Qué Es Vue.JS?. Obtenido de <https://codigofacilito.com/articulos/que-es-vue>.

Características, párr. 1: Rosa J. (Octubre, 2017). ¿Qué es Vue.js?. Obtenido de <https://openwebinars.net/blog/que-es-vuejs/>.

Ventajas de Bootstrap, párr. 1: Casas V. (Octubre, 2019). ¿Qué es Bootstrap y para qué sirve?. Obtenido de <https://www.lucushost.com/blog/que-es-bootstrap/>.

¿Qué es Apache HTTP Server?, párr. 1-4: Fumas E. (Junio, 2014). ¿Qué es Apache HTTP Server?. Obtenido de <https://www.ibrugor.com/blog/apache-http-server-que-es-como-funciona-y-para-que-sirve/>.

Métricas de Calidad de Software, párr. 1: Barrientos D. (Octubre, 2018). MÉTRICAS DE CALIDAD DE SOFTWARE. Obtenido de <http://blog.desafiolatam.com/metricas-de-calidad-de-software/>.

Que es la ISO, párr. 1: BlogSpot (s. f.). ¿Qué es la ISO?. Obtenido de http://iso9126uts.blogspot.com/p/i_8.html.

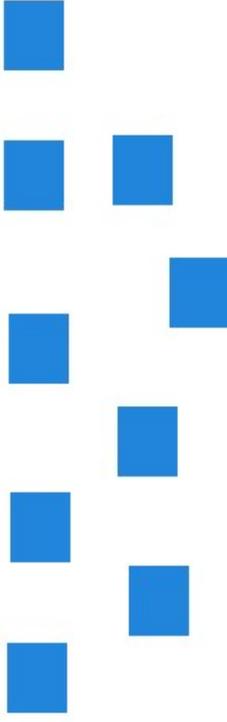
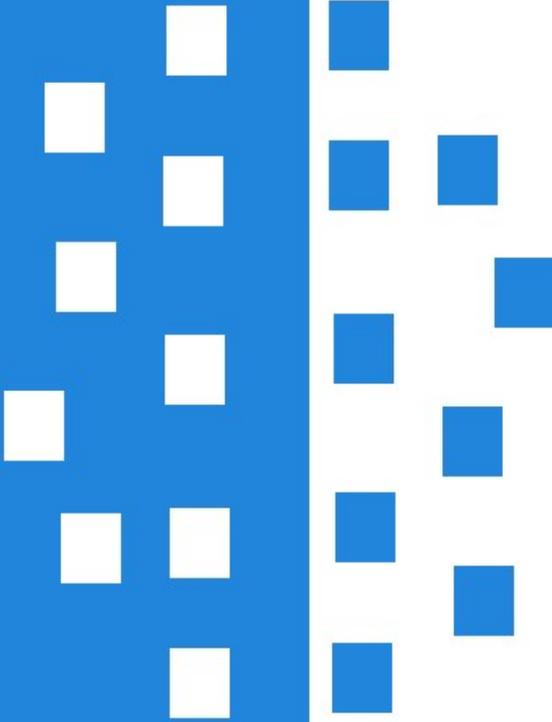
ISO 9126, párr. 1-4: BlogSpot (s. f.). ¿Qué es la ISO?. Obtenido de http://iso9126uts.blogspot.com/p/i_8.html.

Características, párr. 1-7: BlogSpot (s. f.). ¿Qué es la ISO?. Obtenido de http://iso9126uts.blogspot.com/p/i_8.html.

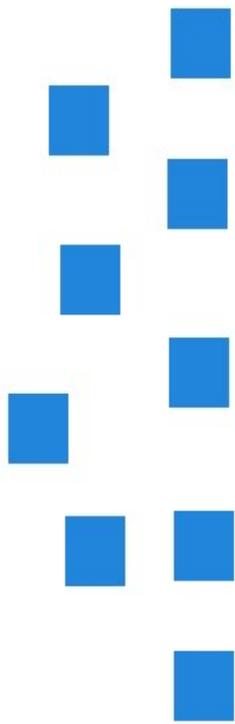
Calidad en Uso, párr. 1-2: BlogSpot (s. f.). ¿Qué es la ISO?. Obtenido de http://iso9126uts.blogspot.com/p/i_8.html.

Modelo Cocomo, párr. 1: Vecino A. & Miranda Y. Obregón V. & Barrios A. & Pava D. (Noviembre, 2015). MODELO COCOMO (INGENIERA DE SOFTWARE). Obtenido de <https://es.slideshare.net/yadithmiranda/modelo-cocomo-ingeniera-de-software>.

Modelo Detallado, párr. 1: Wikipedia (Enero, 2005). COCOMO. Obtenido de <https://es.wikipedia.org/wiki/COCOMO>.

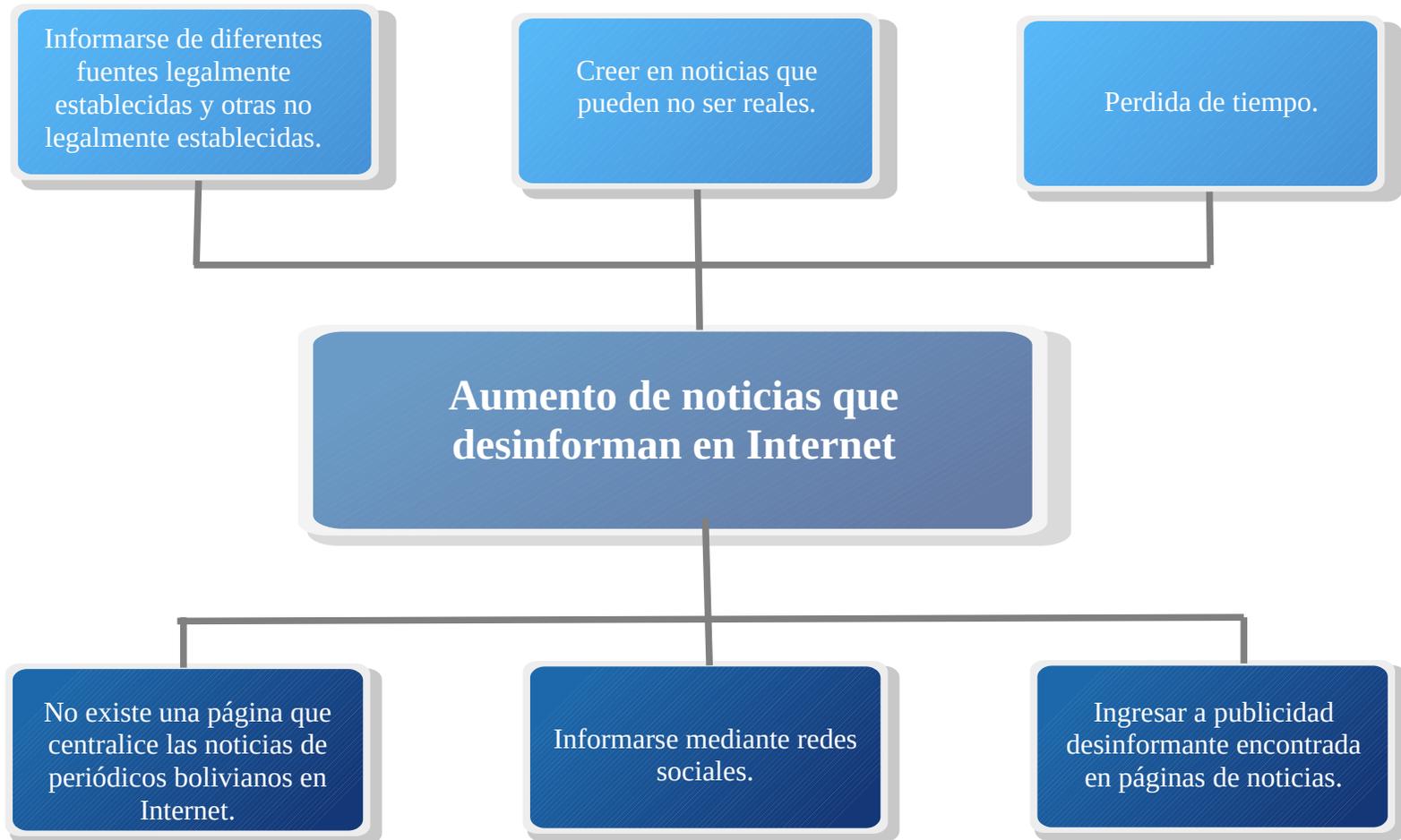


ANEXOS

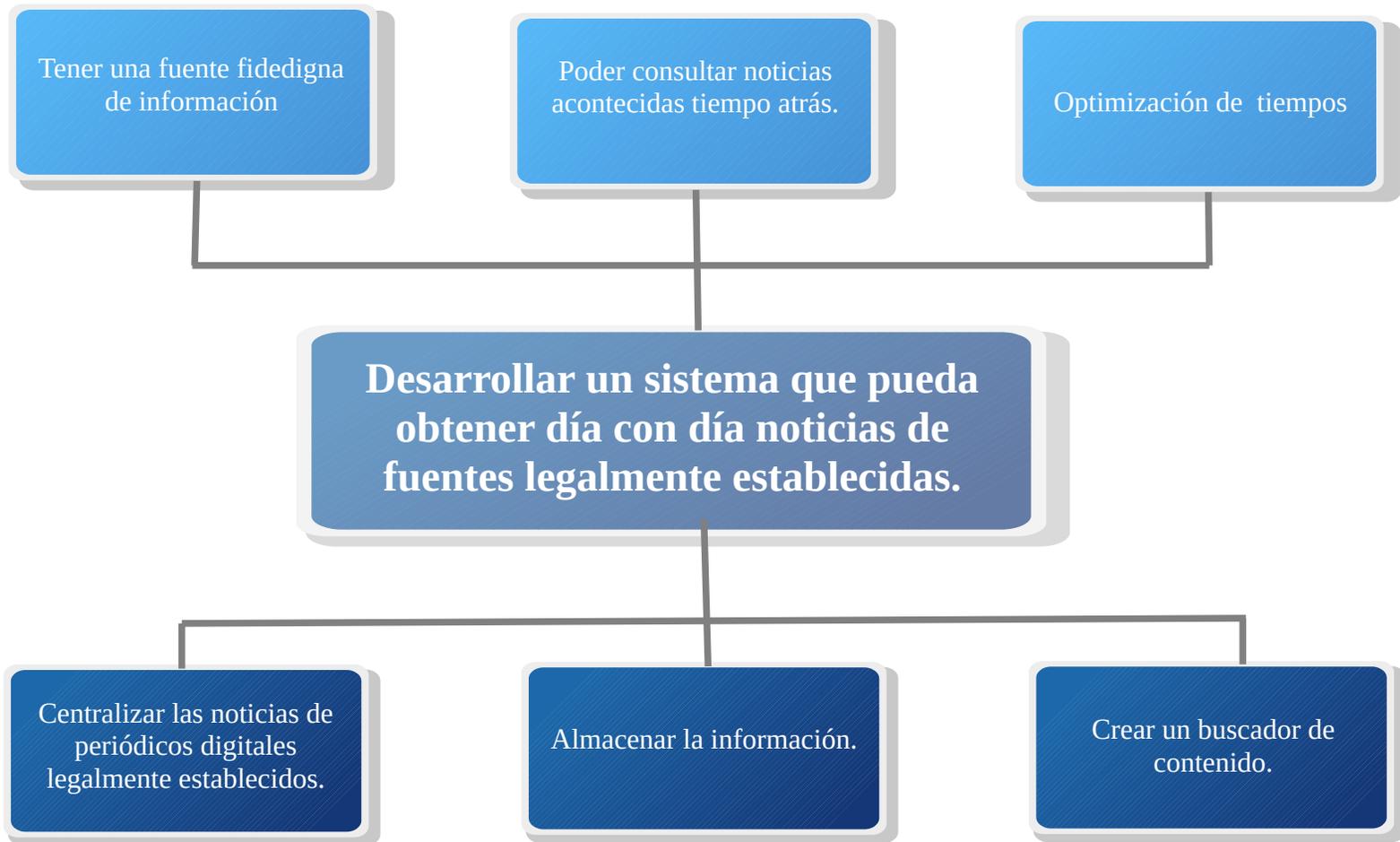


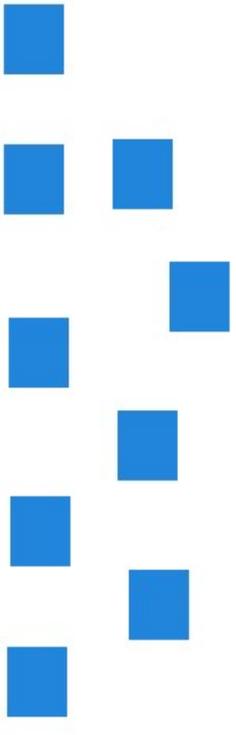
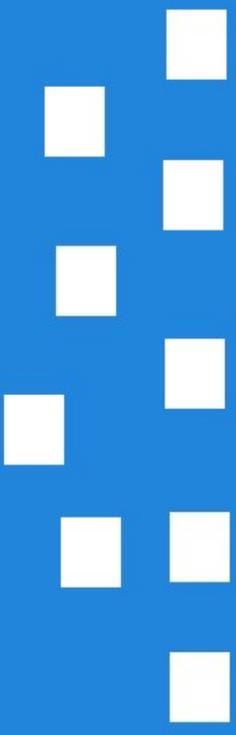
ANEXOS

ANEXO 1. ÁRBOL DE PROBLEMAS

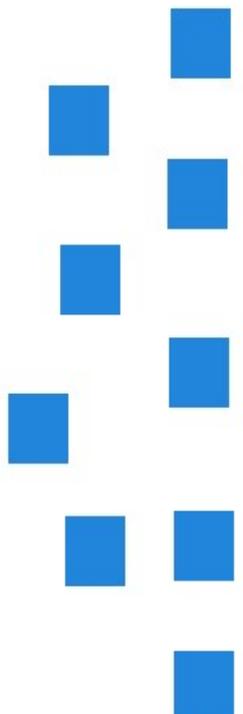


ANEXO 2. ÁRBOL DE OBJETIVOS





MANUAL DE USUARIO



ANEXO 3: MANUAL DE USUARIO ROL ADMINISTRADOR

En el presente manual se dará una explicación sobre el manejo del Sistema Web Orientado A La Recolección De Información De Medios Digitales Mediante Técnicas De Web Scraping (Hemeroteca).

Ingreso al Sistema

Para el acceso al sistema se recomienda utilizar uno de estos navegadores de internet: Google Chrome o Firefox en sus versiones más recientes.

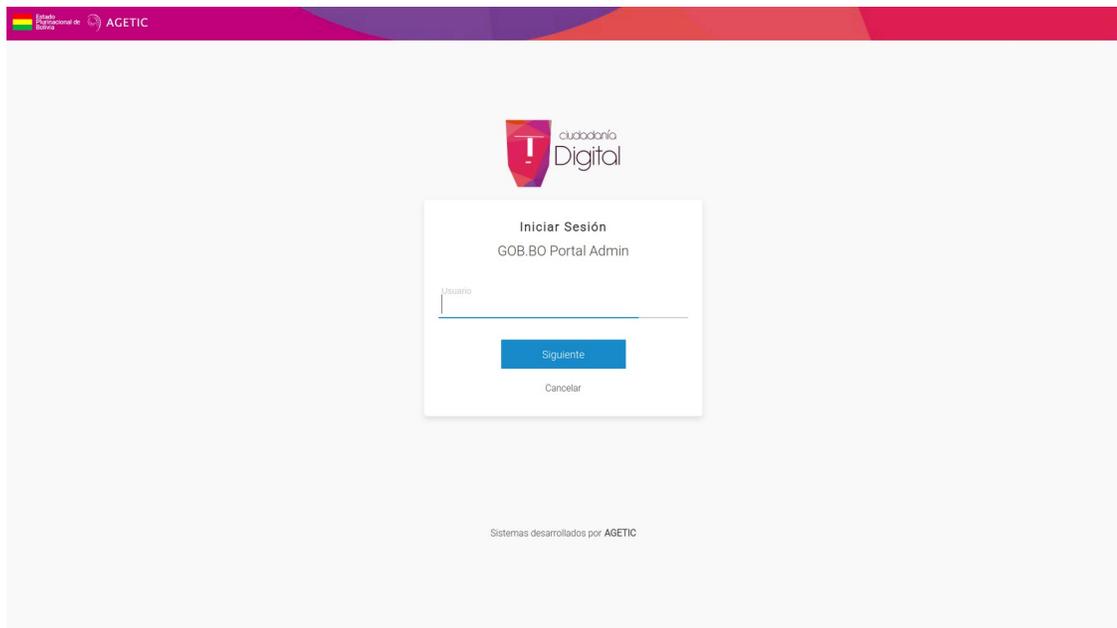


Para el acceso a esta pantalla necesitaremos ir a la URL donde está instalado el sistema y agregarle al final de la misma /login.

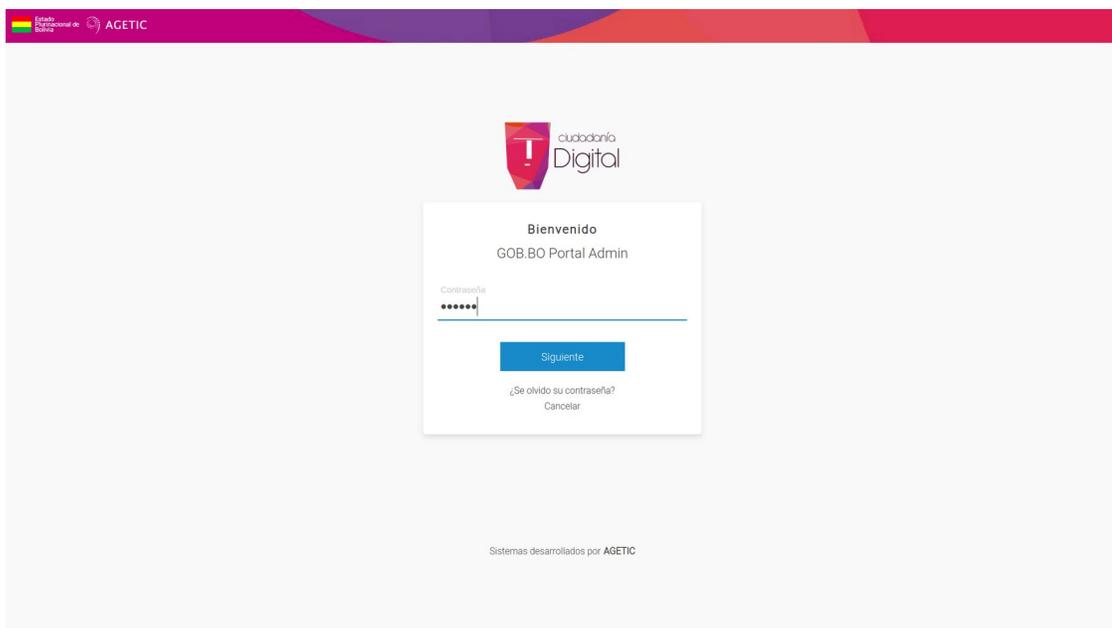
<https://test.agetec.gob.bo/hemeroteca/login>

Dentro de esta pantalla se puede apreciar solamente una opción la cual es ingresar mediante “Ciudadanía Digital”, para lo cual el usuario administrador deberá estar registrando antes en dicho sistema.

Una vez presionado el botón “Ciudadanía Digital ->”, este nos redirigirá a la siguiente pantalla.

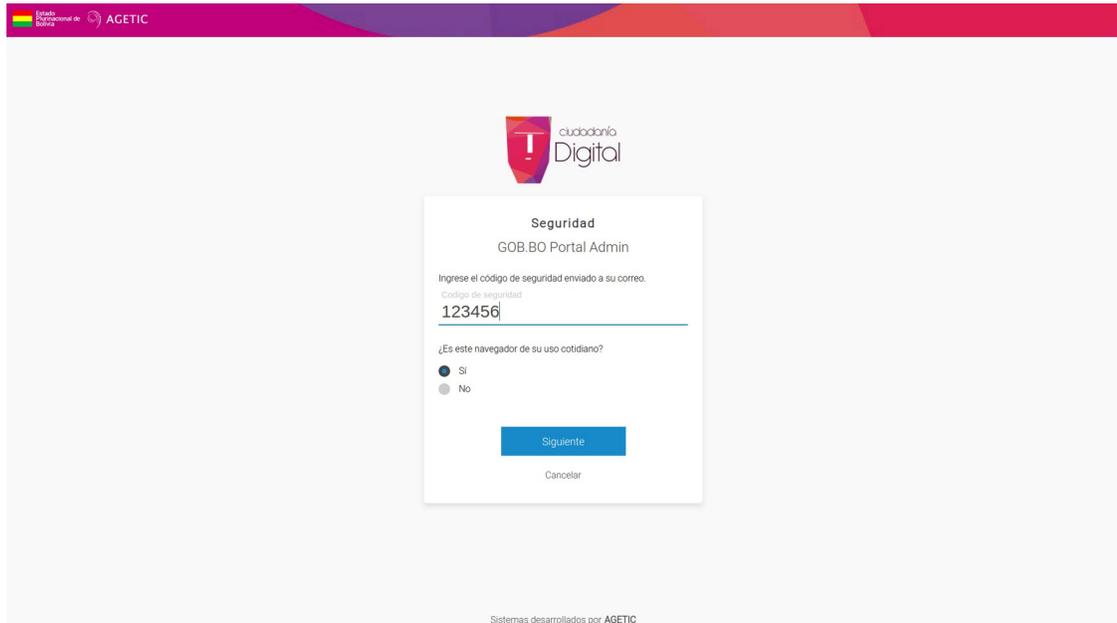


Dentro de esta pantalla podemos ingresar nuestro usuario que gracias al registro en ciudadanía digital sera nuestro carnet de identidad, luego procedemos a presionar el botón siguiente.



En esta pantalla podemos ingresar nuestra contraseña de “Ciudadanía Digital” o si se nos recordamos la contraseña ingresamos a la opción ¿Se olvidó su contraseña?, para nuestro caso pondremos nuestra contraseña y

presionaremos el botón “Siguiete”.



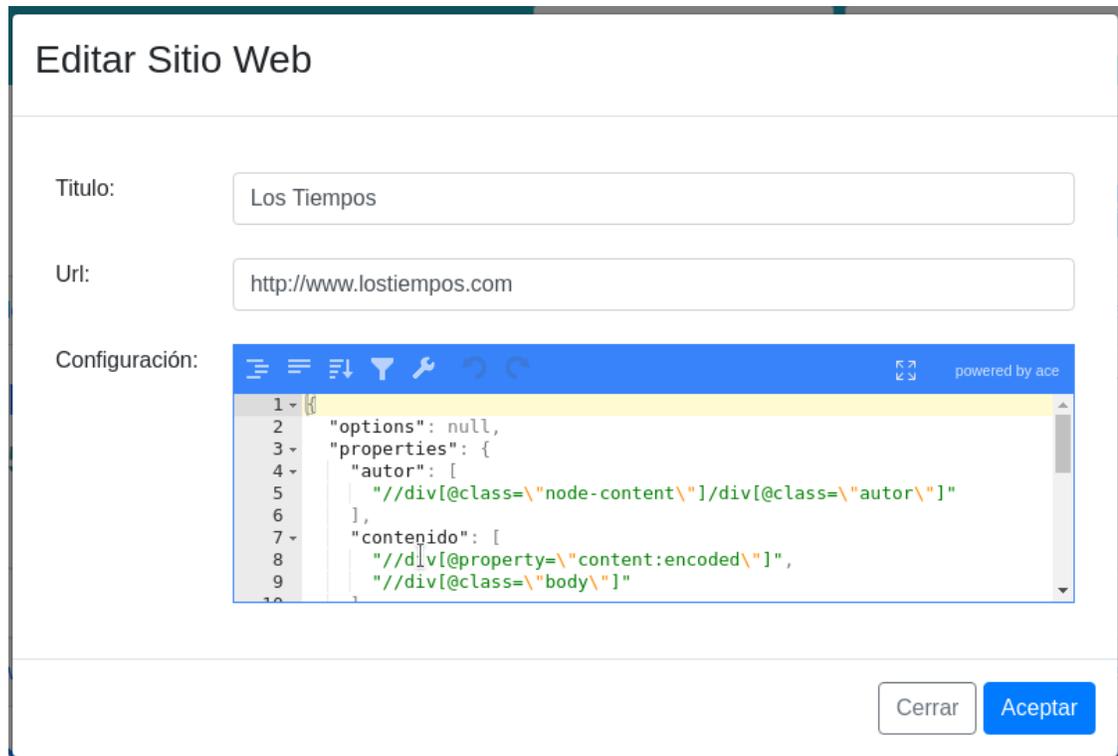
En esta pantalla podemos apreciar que nos pide un código, este código es enviado a nuestro correo o si tenemos la aplicación de ciudadanía digital a nuestro celular.

Una vez ingresado el código presionamos el botón “Siguiete”, y podremos ingresar al sistema.



Una vez dentro el sistema podremos ver todos los periódicos digitales registrados, los cuales podemos editar al hacer clic en el botón del lápiz situado en cada uno de ellos, como también podemos crear un nuevo registro al presionar el botón “Adicionar Sitio Web”.

Una vez presionado el botón de editar podremos ver la siguiente pantalla:



Editar Sitio Web

Titulo:

Url:

Configuración:

```
1 {
2   "options": null,
3   "properties": {
4     "autor": [
5       "//div[@class=\"node-content\"]/div[@class=\"autor\"]"
6     ],
7     "contenido": [
8       "//div[@property=\"content:encoded\"]",
9       "//div[@class=\"body\"]"
10    ]
11  }
12 }
```

En esta pantalla podemos ingresar los siguientes campos:

- ▶ Título: Nos referencia al título, nombre que lleva el periódico digital.
- ▶ Url: Dirección web donde se encuentra alojada la página del periódico.
- ▶ Configuración: Es un json en el cual añadimos las rutas Xpath para poder localizar el autor, título, subtítulo, contenido, imagen, fecha de una noticia, la página de un periódico digital puede tener diferentes formas de mostrar un título u otro dato de la noticia, por lo cual cada atributo dentro del json está formado por un array el cual se va iterando en busca de la información requerida por el sistema.

Ejemplo de Xpath de una noticia:

Para este ejemplo utilizaremos el periódico digital “Los Tiempos” del cual obtendremos el menú que cuenta con todas las secciones de dicho periódico.

Para obtener el Xpath del título realizamos lo mismo que se hizo para obtener

Economía

Economista dice que la pospandemia será un reto para los jóvenes emprendedores

Jóvenes en un emprendimiento. | Jose Rocha

Redacción Central
Publicado el 04/07/2020 a las 0h:38

Python Online Training
Certification Training

Lo más leído

1. Barral pide a la población cuidarse: "El rondar entre la vida y la muerte es muy complicado"
2. Murillo asegura que la Presidenta realizará propuestas a Mesa: "Habrá sorpresas"
3. Gobierno resguarda salud de Añez y opta que trabaje vía Zoom, informa Viceministro
4. Esposa de Policía que murió por Covid-19 recibe sus restos
5. Anuncian nuevo censo contra la...

```
//*[@header/h1]
```

Styles Computed Event Listeners DOM Breakpoints

Filter: :hov .cls

```
element.style {  
  }  
.node .node-title {  
  margin: 0;  
  }  
h1.node-title {  
  font-size: 3.6rem;  
  line-height: 1.3;  
  font-weight: normal;  
  }
```

el menú, en la imagen se puede ver el xpath del título que corresponde a **//*[@header/h1]** con el cual obtenemos el título de la noticia.

De esta manera podemos obtener todos los atributos de una noticia al navegar dentro de este periódico digital.

Verificar la obtención de datos

Volvemos a la página principal del sistema

Noticias del 4 de julio de 2020 [Adicionar Sitio Web](#)

https://www.paginasiete.bo PÁGINA SIETE 40	http://www.lostiempos.com LOS TIEMPOS 25	http://www.eldiario.net EL DIARIO 22
http://www.laprensa.com.bo LA PRENSA 9	https://www.la-razon.com LA RAZÓN 10	http://cambio.bo CAMBIO 0
https://www.eldia.com.bo EL DÍA 22	http://www.eldeber.com.bo EL DEBER 13	https://erbol.com.bo ERBOL 3



Botón de pruebas de configuración.

Al presionarlo nos redirigirá a la siguiente pantalla.

Pruebas de Configuración

URL:

Periódico: **Página Siete**

Fecha:	sábado, 4 de julio de 2020 · 00:04
Autor:	EFE / Lima
Título:	Perú acumula siete días con más pacientes sanados
Subtítulo:	S/N
Contenido:	EFE / Lima Perú acumuló ayer siete días consecutivos donde el número de pacientes recuperados de Covid-19 supera los nuevos contagios, una tendencia que ya hizo que los casos activos de coronavirus estén por debajo de los 100.000 infectados. Desde el viernes pasado un total de 26.046 enfermos han recibido el alta epidemiológica tras vencer al virus SARS-CoV-2, por encima de los 23.235 nuevos contagios que se han detectado en el mismo periodo. En el último día se registraron 3.595 casos positivos, la segunda cifra más alta de los últimos siete días, tras haber dado positivo el 17 % de las más de 20.000 pruebas que se procesaron el jueves. Esto mantiene a Perú como el sexto país del mundo y el segundo Latinoamérica con más casos acumulados de Covid-19 al sumar 295.599 infectados. El número de fallecidos confirmados por coronavirus subió a los 10.226 tras registrarse 181 nuevos decesos en el último día, una tendencia que se permanece en cifras similares desde hace ya varias semanas. Asimismo, un total de 3.755 pacientes recibieron el alta en el último día para situar el número de enfermos recuperados en 185.852, lo que hace que el número de casos activos siga descendiendo y ahora sea de 99.521. Después de que la cuarentena se levantase en dieciocho de las veinticinco regiones del país, el Gobierno fijó el 15 de julio para reanudar el transporte interprovincial de pasajeros por tierra y aire en aquellos departamentos donde ya se terminó el confinamiento por la pandemia. El ministro de Transportes y Comunicaciones, Carlos Lozada, estimó que la demanda de viajes internos será de un 30 % durante el primer mes de actividades.
Imagen:	u/fotografias/m/2020/7/3/1800x450-316493_367939_4926.jpg

Si se necesita saber si los xpaths que configuramos están correctos o si algún periódico está obteniendo bien los datos entonces entramos a sus pruebas de configuración.

Dentro de la URL copiamos la dirección de una noticia, posteriormente presionamos “Probar”, este traerá consigo los datos de la noticia como se Fecha, Autor, Título, Subtítulo, Contenido e Imagen.

Si algún dato no se obtuvo correctamente procedemos a volver a configurar en la página principal y volvemos a probar.

Menú Principal

SISTEMA HEMEROTECA

Ejemplo: empresa 22 de junio de 2020 22 de junio de 2020 Todos Buscar

Noticias del 22 de junio de 2020

Adicionar Sitio Web

URL	Nombre	Cantidad
http://www.lostiempos.com	LOS TIEMPOS	57
https://www.paginasiete.bo	PÁGINA SIETE	43
https://www.eldia.com.bo	EL DÍA	38
http://www.eldiario.net	EL DIARIO	28
http://www.opinion.com.bo	OPINIÓN	24
http://www.laprensa.com.bo	LA PRENSA	23
https://www.la-razon.com	LA RAZÓN	23
https://erbol.com.bo	ERBOL	2
http://cambio.bo	CAMBIO	0
http://www.eldeber.com.bo		
https://www.radiofides.com		
https://jornada.com.bo		

Para abrir el menú principal del sistema tenemos que presionar en el icono de usuario.

Dentro del cual tenemos las opciones de: Administrar Usuarios, Sincronización Manual, Cerrar Sesión y Ocultar el Menú.

Administrar Usuarios

SISTEMA HEMEROTECA

Ejemplo: empresa 22 de junio de 2020 22 de junio de 2020 Todos Buscar

Usuarios

Adicionar Usuario

Acciones	Carnet	Nombres	Apellido Paterno	Correo	Estado
<input checked="" type="checkbox"/>	7024739	Roy Reportes	Aroa	roivan24@gmail.com	ACTIVO
<input checked="" type="checkbox"/>	4206088	MARIANA	ROCHA	roivan324@gmail.com	ACTIVO

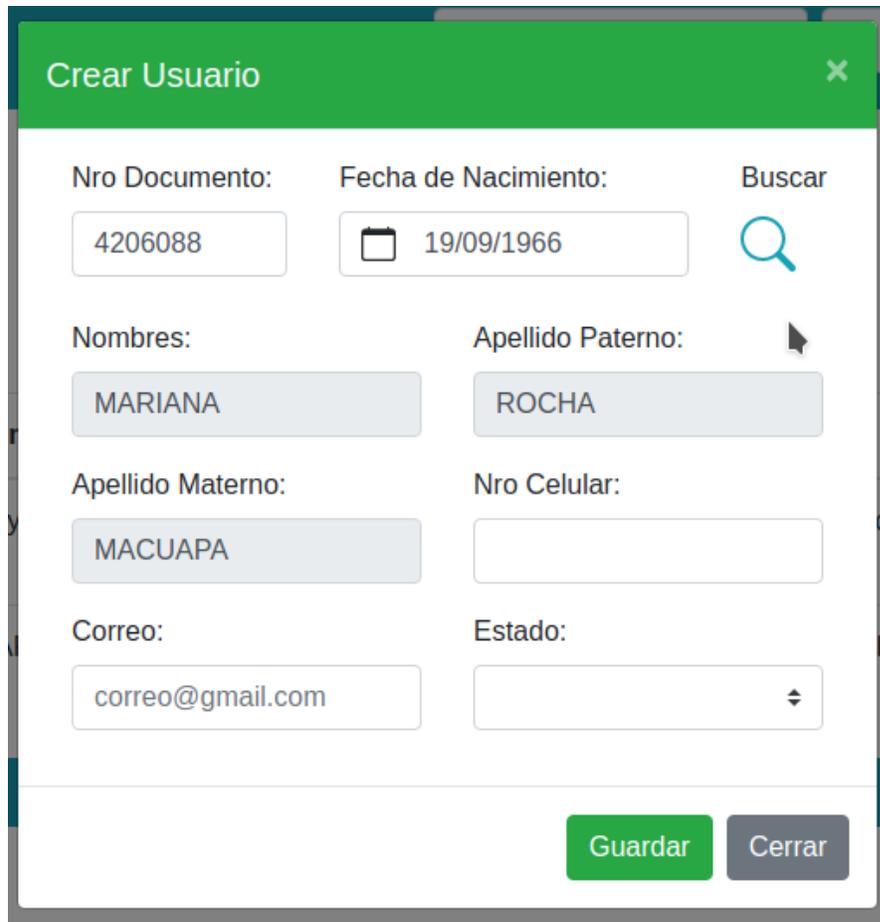
Hemeroteca 2020

Dentro de la pantalla de Administrar Usuarios tenemos una tabla mediante la

cual podemos:

Adicionar Usuario: Despliega un modal en el cual podemos crear un usuario administrador en el sistema.

Editar Usuario: Despliega un modal en el cual podemos modificar el estado del usuario, correo y celular.



The image shows a modal window titled "Crear Usuario" with a green header and a close button (X). The form contains the following fields:

- Nro Documento:** Input field with the value "4206088".
- Fecha de Nacimiento:** Date picker field with the value "19/09/1966".
- Buscar:** Search icon.
- Nombres:** Input field with the value "MARIANA".
- Apellido Paterno:** Input field with the value "ROCHA".
- Apellido Materno:** Input field with the value "MACUAPA".
- Nro Celular:** Empty input field.
- Correo:** Input field with the value "correo@gmail.com".
- Estado:** Dropdown menu.

At the bottom right, there are two buttons: "Guardar" (green) and "Cerrar" (grey).

Para crear un usuario este mismo tiene que estar registrado a Ciudadanía Digital.

Nro Documento: Este campo se refiere al número de carnet del usuario.

Fecha de Nacimiento: Campo que contiene la fecha de nacimiento.



Este botón activará un servicio de contrastación con Ciudadanía Digital el cual nos retornara los datos básicos del usuario como ser: nombres, apellido paterno, apellido materno.

Nombres, Apellido Paterno, Apellido Materno: Estos campos son llenados automáticamente si se encontró el usuario en Ciudadanía Digital.

Estado: Este es el campo proveerá información sobre si el usuario es activo y puede entrar al sistema o si está inactivo y lo bloqueamos.

Sincronización Manual

SISTEMA HEMEROTECA Ejemplo: empresa 4 de julio de 2020 4 de julio de 2020 Todos Buscar

Sincronización Manual

Puede elegir sincronizar desde la fecha en la que se dejó de capturar bien los datos de algun periódico digital en especial.

Nombre	Url	Fecha	Acciones
La Prensa	http://www.laprensa.com.bo	DD/MM/YYYY	Sincronizar
Cambio	http://cambio.bo	No date selected	Sincronizar
Página Siete	https://www.paginasiete.bo		Sincronizar
Los Tiempos	http://www.lostiempos.com		Sincronizar
El Día	https://www.eldia.com.bo		Sincronizar
El Deber	http://www.eldeber.com.bo		Sincronizar
ERBOL	https://erbol.com.bo	DD/MM/YYYY	Sincronizar
FIDES	https://www.radiofides.com	DD/MM/YYYY	Sincronizar
El Diario	http://www.eldiario.net	DD/MM/YYYY	Sincronizar

La sincronización manual nos sirve para recuperar información, puede darse el caso de que un periódico digital cambió su presentación de noticias, gracias a esto puede que no estemos obteniendo los datos. Tenemos el código fuente por lo cual podemos reajustar los xpath's primeramente en la página principal del sistema para luego venir a esta sección donde buscamos la fecha desde donde no se están obteniendo bien los datos para posteriormente volver sincronizar y no perder ni un dato.

Esta operación debe ser realizada con precaución y solo poner la fecha desde

donde dejo de obtenerse correctamente los datos, ya que es una tarea pesada para el servidor.

ANEXOS. MANUAL DE USUARIO

El usuario normal no necesita de una cuenta para poder acceder al sistema, solo acceder al sistema mediante su url.

Página Principal

The screenshot shows the main interface of the 'SISTEMA HEMEROTECA'. At the top, there is a teal header with the system name on the left and a search bar on the right containing 'Ejemplo: empresa', two date fields set to '4 de julio de 2020', a dropdown menu set to 'Todos', and a 'Buscar' button. Below the header, the page title is 'Noticias del 4 de julio de 2020'. The main content area displays a grid of news cards for various digital newspapers. Each card includes the newspaper's name, its URL, and the number of news items for that date.

URL	Nombre del Periódico	Cantidad de Noticias
https://www.paginasiete.bo	PÁGINA SIETE	40
http://www.lostiempos.com	LOS TIEMPOS	25
http://www.eldiario.net	EL DIARIO	22
http://www.laprensa.com.bo	LA PRENSA	9
https://www.la-razon.com	LA RAZÓN	1
http://cambio.bo	CAMBIO	0
https://www.eldia.com.bo	EL DÍA	0
http://www.eldeber.com.bo	EL DEBER	0
https://erbol.com.bo	ERBOL	0
https://www.radicofides.com		
https://jornada.com.bo		
http://www.opinion.com.bo		

Se muestra la cantidad de noticias obtenidas de los diferentes periódicos digitales.

En la parte superior existe un buscador por:

- ▶ Palabra o frase: Podemos ingresar una palabra. Ejemplo "El Alto"
- ▶ Fechas: Se filtraran todas las noticias encontradas en el rango de fecha. Ejemplo: 1 de Marzo de 2020 a 20 de Mayo de 2020
- ▶ Periódico Digital: Podemos elegir un periódico digital en particular. Ejemplo: El Deber

Resultados de Búsqueda

SISTEMA HEMEROTECA Ejemplo: empresa 4 de julio de 2020 4 de julio de 2020 Todos Buscar

Noticias del 4 de julio de 2020

Total: 97 noticias



DT Copito Andrada ayuda a técnicos sin trabajo
4 de julio de 2020
1 / 97 Página Siete



Perú acumula siete días con más pacientes sanados
4 de julio de 2020
2 / 97 Página Siete



Morales dirige el bloqueo en Asamblea Legislativa
4 de julio de 2020
3 / 97 El Diario



El nuevo campo de batalla contra el crimen organizado en México
4 de julio de 2020



EN LA MIRA
"Spin" para el caso respiradores
4 de julio de 2020



Achacollo sale de la cárcel y Gobierno inicia proceso al juez que lo autorizó
4 de julio de 2020

El sistema encontró 97 noticias a primeras horas de la mañana este irá aumentando en el transcurso del día.

Se puede acceder al contenido de una noticia al hacer clic sobre ella

01 DTECA

Publicado el 4 de julio de 2020

Perú acumula siete días con más pacientes sanados



EFE LIMA

Perú acumuló ayer siete días consecutivos donde el número de pacientes recuperados de Covid-19 supera los nuevos contagios, una tendencia que ya hizo que los casos activos de coronavirus estén por debajo de los 100.000 infectados.

Desde el viernes pasado un total de 26.046 enfermos han recibido el alta epidemiológica tras vencer al virus SARS-CoV-2, por encima de los 23.235 nuevos contagios que se han detectado en el mismo periodo.

En el último día se registraron 2.056 casos positivos, lo que representa el mayor número de los últimos siete días, tras haber dado positivo el 17 de los...

Esta noticia se puede visualizar en un modal el cual ocupa toda la pantalla para facilitar la lectura, dicho modal puede cerrarlo y ver otras noticias las cuales son cargadas de 9 en 9 para mejorar la experiencia del usuario.

El Alto, julio del 2020

Señor:

Ing. Enrique Flores Baltazar

TUTOR METODOLÓGICO TALLER II

Presente. -

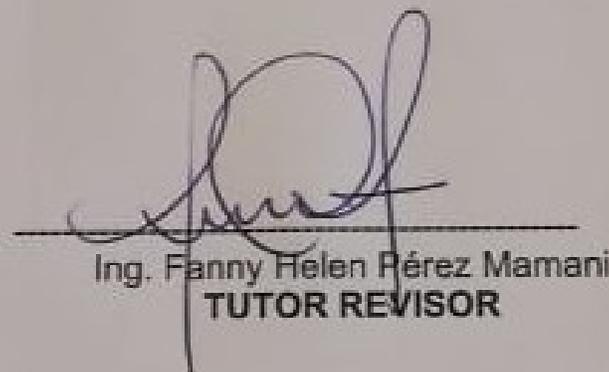
Ref.- Aval de Conformidad

Distinguido ingeniero:

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado "Sistema Web Orientado a la Recolección de Información de Medios Digitales Mediante Técnicas de Web Scraping caso: Agetic – Hemeroteca", que propone el postulante Univ. Roger Ivan Aroa Miranda, con cédula de identidad 7024739 LP para su defensa pública, evaluación correspondiente a la materia Taller de Licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente.



Ing. Fanny Helen Pérez Mamani
TUTOR REVISOR

El Alto, julio del 2020

Señor:

Enrique Flores Baltazar

TUTOR METODOLÓGICO TALLER II

Presente. -

Ref.- Aval de Conformidad

Distinguido ingeniero:

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado **“Sistema Web Orientado a la Recolección de Información de Medios Digitales Mediante Técnicas de Web Scraping caso: Agetic – Hemeroteca”**, que propone el postulante Univ. Roger Ivan Aroa Miranda, con cédula de identidad 7024739 LP para su defensa publica, evaluación correspondiente a la materia Taller de Licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente.



Freddy Salgueiro Trujillo
TUTOR ESPECIALISTA

El Alto, julio del 2020

Señor:

Ing. David Carlos Mamani Quispe

DIRECTOR DE CARRERA INGENIERÍA DE SISTEMAS

Presente. -

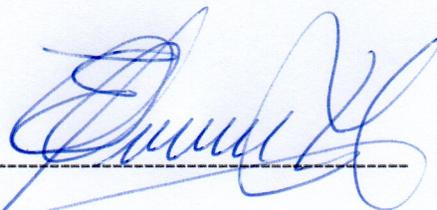
Ref.- Aval de Conformidad

Distinguido ingeniero:

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado "Sistema Web Orientado a la Recolección de Información de Medios Digitales Mediante Técnicas de Web Scraping caso: Agetic – Hemeroteca", que propone el postulante Univ. Roger Ivan Aroa Miranda, con cédula de identidad 7024739 LP para su defensa pública, evaluación correspondiente a la materia Taller de Licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente.



Ing. Enrique Flores Baltazar

TUTOR METODOLÓGICO TALLER II

La Paz, Julio de 2020

Señor:

Ing. Enrique Flores Baltazar

TUTOR METODOLÓGICO TALLER II

Presente.-

Ref.- AVAL DE CONFORMIDAD

Distinguido Ingeniero:

Mediante la presente reciba usted mis más sinceros saludos y deseos de éxito en las funciones que desempeña en bien de los estudiantes de la Universidad Pública de El Alto.

El motivo de la presente es para comunicarle que como supervisor del proyecto de grado denominado **“Sistema Web Orientado a la Recolección de Información de Medios Digitales Mediante Técnicas de Web Scraping caso: Agetic – Hemeroteca”** que propone el postulante **Roger Ivan Aroa Miranda** con cédula de identidad **7024739 LP** doy mi conformidad para su defensa pública y evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la carrera Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular me despido de usted muy cordialmente.

Atentamente,

Elmer Freddy Mendoza Apaza
RESPONSABLE DE INNOVACIÓN Y DESARROLLO
EN GOBIERNO ELECTRÓNICO
AGETIC

CERTIFICADO

A quien corresponda:

EL RESPONSABLE DE INNOVACIÓN Y DESARROLLO EN GOBIERNO ELECTRÓNICO DE LA AGENCIA DE GOBIERNO ELECTRÓNICO Y TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN - AGETIC CERTIFICA QUE:

El Universitario **ROGER IVAN AROA MIRANDA** con Cédula de Identidad **7024739** expedido en la Ciudad de La Paz, estudiante de la Carrera Ingeniería de Sistemas de la Universidad Pública de El Alto, realizó satisfactoriamente su Proyecto de Grado denominado: **“Sistema Web Orientado a la Recolección de Información de Medios Digitales Mediante Técnicas de Web Scraping caso: Agetic – Hemeroteca”**, dando conformidad del mismo, así de esta forma se dió cumplimiento al proyecto presentado por el mencionado.

Es cuanto certifico en honor a la verdad, para fines que convenga al interesado.

La Paz, Julio del 2020

Elmer Freddy Mendoza Apaza
RESPONSABLE DE INNOVACIÓN Y DESARROLLO
EN GOBIERNO ELECTRÓNICO
AGETIC