



Facultad de Ciencias y Tecnologías Computacionales

Desarrollo de la versión 2.0 del Sistema de Obtención y Catalogación de Contenidos de Internet

Trabajo de diploma para optar por el título de
Ingeniera en Ciencias Informáticas

Autora: Lesly Meliza Ferrer Aguado

Tutores:

Ing. José Alberto Varona Labrada

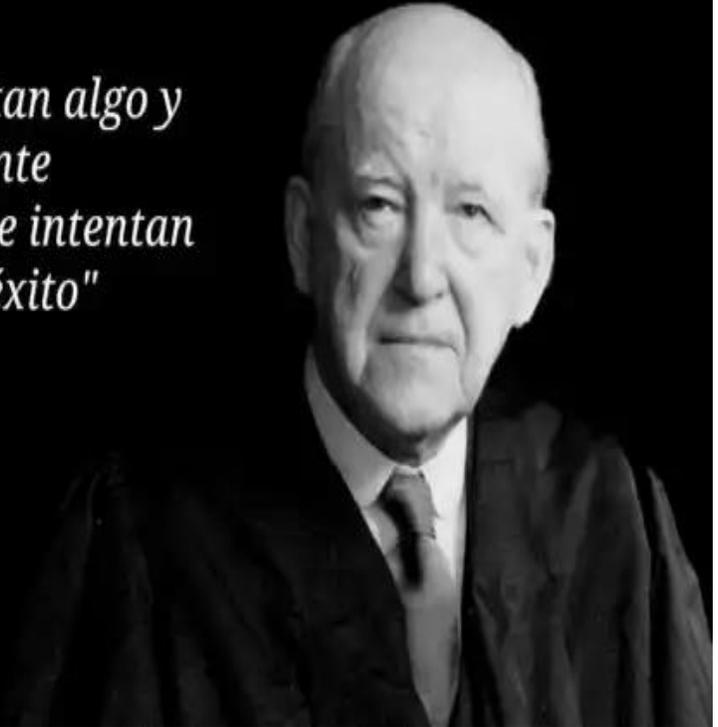
Ing. Ramón Velázquez Rojas

La Habana, noviembre de 2022

“Año 63 de la Revolución”

"Los hombres que intentan algo y fracasan son infinitamente mejores que aquellos que intentan no hacer nada y tienen éxito"

-Lloyd Jones-



DECLARACIÓN DE AUTORÍA

El autor del trabajo de diploma con título ***“Desarrollo de la versión 2.0 del Sistema de Obtención y Catalogación de Contenidos de Internet”***, concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firma la presente a los 28 días del mes de noviembre del año 2022.

Lesly Meliza Ferrer Aguado

Firma del Autor

José Alberto Varona Labrada

Firma del Tutor

Ramón Velázquez Rojas

Firma del Tutor

DATOS DE CONTACTO

Ing. José Alberto Varona Labrada, graduado de Ingeniero en Ciencias Informáticas. Especialista B en Ciencias Informáticas y arquitecto de sistemas en el Centro de Inteligencia de Negocios de la XETID. Posee 5 años de experiencia en el desarrollo de sistemas con aplicaciones de Inteligencia Artificial, javarona@xetid.cu

Ing. Ramón Velázquez Rojas, graduado de Ingeniero en Ciencias Informáticas, RGA. Profesor de Bases de Datos II y profesor guía de grupo en la Facultad de Ciencias y Tecnologías en cuarto año ICI en la Universidad de Ciencias Informáticas, rvojas@uci.cu

AGRADECIMIENTOS

A mis padres, por haberme dado la vida, todo su amor y confianza, por criarme para ser una mujer respetuosa y capaz de luchar por mis sueños.

A mi segundo papá, por siempre soportarme, malcriarme y ayudarme en todo lo que he necesitado, por aportarme todos sus conocimientos a la hora que fuera sin negarse.

A mis abuelos, por la dulce infancia que me hicieron vivir y todo el amor y respeto inculcado.

A mi tata que amo con la vida, por cuidarme y quererme tanto, y por los hermosos sobrinos que me ha dado que me han levantado el ánimo incontables veces.

A mi novio, por el amor, apoyo y comprensión brindado en estos maravillosos años, y toda su ayuda.

A mis suegros, que sin ellos no hubiera podido tener a mi compañero durante estos cuatro años y futuro compañero durante toda mi vida.

A mis amigos de la Universidad, a los que estuvieron al inicio y a los que siguieron a mi lado, por tantos años de apoyo y buenos momentos llenos de felicidad.

A mis amigas de siempre, las que me apoyaron desde la distancia a esforzarme y no rendirme.

A los profesores que me apoyaron en el transcurso de la carrera y en especial a la profe Dagmaris por todos sus consejos y el apoyo en el desarrollo de este trabajo de diploma.

A mis tutores, por aceptarme como tesista, apoyarme en el proceso y confiar en mí.

A mi tribunal, por haberme dado tantos consejos y haber confiado en mí.

A mi oponente Isa, gracias por todo mi vida.

DEDICATORIA

A la persona que me ha soportado durante 24 años y 11 meses. La mujer que me ha apoyado en cada momento, difícil y alegre, de la vida dándome su amor incondicional, mi mami, mi fortaleza, mi guerrera, mi orgullo, mi ejemplo a seguir como madre y mujer. Por haberme convertido en la mujer que soy hoy, por las alas y el apoyo en estos años de esfuerzo y lucha constante. Me has dado las fuerzas que he necesitado para no rendirme y luchar por mi futuro siempre confiando en tu niña, Te Amo mamita.

A mi novio, mi amigo, el compañero de vida que elegí, mi soporte, porque desde que lo conocí me ha ayudado a ser mejor persona, superarme y estudiar a pesar de mis berrinches y malcriadeces me ha comprendido, soportado, mimado y apoyado a lo largo de estos maravillosos 4 años juntos con mucho amor y paciencia. A mi futuro esposo, gracias por ser mi tutor, por todos tus consejos, revisiones y por tu compañía en cada momento de este proceso de tesis, Te amo mi amor.

RESUMEN

La Obtención y Catalogación de Contenidos de Internet, durante el proceso de vigilancia e inteligencia empresarial, es un tema que ha sido investigado y desarrollado en los últimos años. En correspondencia, esta investigación se enfocó en la gestión de la información, y como objetivo general, desarrollar la versión 2.0 de un Sistema Web de Obtención y Catalogación de Contenidos de Internet mediante el uso de las tecnologías, Java Script en el desarrollo del frontend y Python para la implementación del código backend. La investigación se desarrolló empleando un paradigma cuantitativo con estudio descriptivo, donde se emplearon los métodos teóricos: modelación, histórico-lógico, analítico-sintético y sistémico; y empíricos: entrevista semiestructurada, observación, análisis documental, tormenta de ideas y pruebas de calidad, desde el punto de vista de Metodología de la Investigación Científica. Para el desarrollo e implementación del sistema se empleó la metodología de desarrollo de Software ágil Programación Extrema y la arquitectura de software Modelo-Vista-Controlador. Los principales resultados fueron el logro de una versión más completa que permite al usuario disfrutar de una mejor experiencia mediante su interacción con esta nueva versión, ya que se convirtió en un sitio más seguro y con mayor agilidad y precisión a la hora de gestionar información de interés, con mayor rapidez en el tiempo de respuesta.

PALABRAS CLAVE

Catalogación, Gestión de información, Inteligencia empresarial, Tecnologías Web y Vigilancia.

ABSTRACT

The obtaining and cataloguing Internet content during the surveillance and business intelligence process is a topic that has been researched and developed in recent years. Correspondingly, this research focused on information management, and as a general objective, to develop version 2.0 of a Web System for Internet Content Retrieval and Cataloging by using Java Script technologies for the frontend development and Python for the backend code implementation. The research was developed using a quantitative paradigm with descriptive study, where the following theoretical methods were used: modeling, historical-logical, analytical-synthetic and systemic; and empirical methods: semi-structured interview, observation, documentary analysis, brainstorming and quality tests, from the point of view of Scientific Research Methodology. For the development and implementation of the system, the Agile Software development methodology Extreme Programming and the Model-View-Controller software architecture were used. The main results were the achievement of a more complete version that allows the user to enjoy a better experience through its interaction with this new version, since it became a more secure site with greater agility and accuracy when managing information of interest, with faster response time.

KEYWORDS

Business Intelligence, Cataloging, Information Management, Surveillance, Web Technologies.

TABLA DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE EL OBJETO DE ESTUDIO	8
I.1 Gestión del proceso de vigilancia e inteligencia empresarial I.1	8
I.2 Sitio web para la búsqueda y obtención de contenidos y su catalogación I.2	10
I.3 Análisis de soluciones informáticas similares existentes I.3.....	12
I.3.1 Valoración de los estudios realizados	16
I.4 Metodología para dar solución al problema de investigación I.4.....	16
I.5 Herramientas y tecnologías para dar solución al problema de investigación I.5	19
I.5.1. Arquitecturas del sistema	19
Conclusiones del capítulo.....	31
CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO	32
II.1 Modelado de la propuesta de solución II.1	32
II.2 Requisitos, análisis y diseño de la aplicación web II.2	32
II.3 Diseño e implementación del almacenamiento, procesamiento y transmisión de los datos en la aplicación web II.3.....	40
Conclusiones del capítulo.....	55
CAPÍTULO III: VALIDACIÓN DEL SISTEMA DE OBTENCIÓN Y CATALOGACIÓN DE CONTENIDOS DE INTERNET	56
III.1 Verificación y validación del Sistema de Obtención y Catalogación de Contenidos de Internet III.1	56
III.2 Descripción de la estrategia de pruebas a seguir III.2	64
III.3 Realización de las pruebas III.3.....	71
Conclusiones del capítulo.....	80
CONCLUSIONES FINALES	81
RECOMENDACIONES	82
REFERENCIAS BIBLIOGRÁFICAS	83
ANEXOS.....	87

ÍNDICE DE TABLAS

Tabla 1. Historia de Usuario HU-1	36
Tabla 2. Historia de Usuario HU-3	36
Tabla 3. Historia de Usuario HU-7	37
Tabla 4. Historia de Usuario HU-9	37
Tabla 5. Historia de Usuario HU-11.....	38
Tabla 6. Historia de Usuario HU-12.....	38
Tabla 7. Estimación del tiempo de las HU.....	39
Tabla 8. Plan de Iteraciones	40
Tabla 9. Plan de Entrega	42
Tabla 10. Tarjeta CRC 1	42
Tabla 11. Tarjeta CRC 2	43
Tabla 12. Tarjeta CRC 3	43
Tabla 13. Reuniones de seguimiento.....	44
Tabla 14. Historias de usuario iteración 1	57
Tabla 15. Tareas de Ingeniería	57
Tabla 16. Tarea de Ingeniería 1 para historia de usuario 1	58
Tabla 17. Historias de usuario segunda iteración.....	59
Tabla 18. Tareas de ingeniería de la segunda iteración.....	59
Tabla 19. Tarea de Ingeniería 2 para la iteración actual	60
Tabla 20. Tarea de Ingeniería 15 en la iteración presente	60
Tabla 21. Historias de usuario tercera iteración	61
Tabla 22. Tareas de ingeniería de la tercera iteración	61
Tabla 23. Tarea de Ingeniería 1 para la historia de usuario 9	61
Tabla 24. Historias de usuario cuarta iteración	62
Tabla 25. Tareas de ingeniería de la cuarta iteración.....	62
Tabla 26. Tarea de Ingeniería 1 para la historia de usuario 11	63
Tabla 27. Tarea de Ingeniería 4 para la iteración actual	63
Tabla 28. Listado de Pruebas de Aceptación Desarrolladas	65
Tabla 29. Caso de Prueba 1	71
Tabla 30. Caso de Prueba 3	72
Tabla 31. Caso de Prueba 7	72
Tabla 32. Caso de Prueba 9	73
Tabla 33. Caso de Prueba 11	74
Tabla 34. Caso de Prueba 12	74
Tabla 35. Historia de Usuario 4.....	87
Tabla 36. Historia de Usuario 5.....	87
Tabla 37. Historia de Usuario 6.....	87
Tabla 38. Historia de Usuario 8.....	87
Tabla 39. Historia de Usuario 10.....	88
Tabla 40. Historia de Usuario 13.....	88

Tabla 41. Historia de Usuario 14.....	88
Tabla 42. Historia de Usuario 15.....	88
Tabla 43. Historia de Usuario 16.....	88

ÍNDICE DE FIGURAS

Ilustración 1. Vista de Koha (Orex)	13
Ilustración 2. Vista de Google Académico (Google. (s.f.). google scholar)	13
Ilustración 3. Vista de Scielo Data (SciELO. (s.f.))	14
Ilustración 4. Proceso de Vigilancia Informativa del Sistema de Obtención y Catalogación Automática de Contenidos de Internet.....	15
Ilustración 5. Componentes y sus funciones (Varona Labrada & Castellano Leyva, 2018).	16
Ilustración 6. Funcionamiento de XP (Digité, 2022).....	18
Ilustración 7. Visual Paradigm (Visual Paradigm., 2022)	21
Ilustración 8. Logo de Visual Studio Code (Microsoft. (2022). Visual Studio.)	22
Ilustración 9. Logo de Unified Modeling Language (UML. (s.f.))	23
Ilustración 10. Logo de React.js, lenguaje de programación frontend (reactjs.org. (s.f.)).....	25
Ilustración 11. Desarrolladores más comunes de la tecnología miscelánea trabajando con Java Script (mdn. (s.f.). Acerca de JavaScript)	26
Ilustración 12. Python, lenguaje de Programación Backend (python.org. (s.f.). Python.).....	28
Ilustración 13. PostgreSQL, lenguaje de Programación de la Base de Datos (arsys. (s.f.)).....	30
Ilustración 14. Patrones de Arquitectura Modelo Vista Controlador (Aguilar, 2019).....	46
Ilustración 15. Vista Principal de Fuentes	47
Ilustración 16. Vista Principal de Descargas	48
Ilustración 17. Vista Principal de Categorías	49
Ilustración 18. Diagrama Entidad-Relación	54
Ilustración 19. Diagrama de Actividades: Gestionar Categoría.....	66
Ilustración 20. Logo de Jest (jestjs.io)	69
Ilustración 21. Error arrojado en el test de Registrar Usuario	70
Ilustración 22. Código sometido a prueba: Registrar Usuario.....	76
Ilustración 23. Diseño de prueba: Registrar Usuario	77
Ilustración 24. Resultado Satisfactorio Obtenido: Registrar Usuario.....	77
Ilustración 25. Código sometido a prueba: Autenticar Usuario	78
Ilustración 26. Diseño de prueba: Autenticar Usuario.....	79
Ilustración 27. Resultado Satisfactorio Obtenido: Autenticar Usuario	79

OPINIÓN DE LOS TUTORES

<Contenido de la opinión de los tutores>

AVAL DEL CLIENTE

<Contenido del aval del cliente sobre la solución desarrollada>

INTRODUCCIÓN

Los seres humanos evolucionan por medio de la comunicación. Los contenidos digitales es toda la información mostrada en un medio digital. Es información digitalizada cuyo sistema de distribución es a través de internet y su formato es muy amplio ya que pueden estar formadas por textos, imágenes, videos o una combinación de ellos (Pantallea, 2022). La gestión de la información es la denominación convencional de un conjunto de procesos por los cuales se controla el ciclo de vida de la información, desde su obtención, hasta su disposición final. Con el objetivo de garantizar la integridad, disponibilidad y confidencialidad de la información (infomed instituciones, 2017).

El catálogo ha sido aquella herramienta que permite la localización de diversos materiales o recursos existentes en una biblioteca, así como la identificación de eje temático, para que posteriormente el usuario valore la importancia de acorde a sus necesidades de información (IGNITE, 2020).

El proceso de catalogación tanto de libros como de cualquier recurso de información consta de las siguientes fases:

Catalogación descriptiva: Aquí se extraen los datos necesarios para construir las fichas bibliográficas de las obras, tales como autor, título, tipo de formato, editorial, año, entre otros. Es importante mencionar, que para cumplir con esta tarea se requiere de herramientas que permita presentar al usuario los datos de una obra de manera estandarizada, para que no se modifique la presentación de la información bibliográfica.

Catalogación temática: Consiste en la asignación de uno o varios encabezamientos de materia que agrupe los temas o tópicos de que trata la obra en cuestión. Se podría decir que los objetivos de la catalogación temática son: Facilitar la identificación en el catálogo de los recursos que aborden la misma temática y organizar los materiales de acuerdo al tema que desarrollen (IGNITE, 2020).

De acuerdo con Hsieh-Yee, los principios que guían este proceso de catalogación son:

- La selección de los recursos que existen y la selección de aquellos que son relevantes para determinada necesidad de información.
- La descripción de los recursos seleccionados.

- La oferta de los puntos de acceso a estos (control de autoridad).
- El análisis de sus contenidos.
- La entrega a los usuarios de productos meta informativos que permitan la localización de los recursos (IGNITE, 2020).

La Sociedad de la Información está evolucionando a ritmo alarmante, hace unos años se está produciendo una revolución en el análisis y procesamiento de la información que se genera, en este apartado Internet ha tenido mucho que ver. La mayor parte de información diaria en el mundo se publica en Internet por lo que cada día se vuelve más engorroso el seguimiento del contenido publicado referente a un tema o categoría en específico, de ahí la importancia de poder automatizar la obtención y clasificación de esos contenidos.

Es importante mencionar que actualmente existen nuevas opciones para la organización de la información, la cual es representada por los metadatos. El avance en la tecnología ha brindado nuevas opciones para el diseño de los catálogos, la indexación y el acceso temático (Observatorio Tecnológico UA, 2016).

En el ámbito internacional, se aprecian investigaciones asociadas al proceso de la obtención y catalogación de contenidos de internet en diferentes áreas, por ejemplo, en México a través de la investigación para la gestión de contenidos digitales en instituciones académicas; en Argentina mediante su uso en la vigilancia tecnológica e inteligencia competitiva; y en Chile reflejándolo también en el proceso de vigilancia tecnológica.

En Cuba se ve reflejado este proceso a través de la vigilancia tecnológica en diferentes sectores, entre los que se destacan, el Ministerio de la Ciencia, Tecnología y el Medio Ambiente (CITMA), el Ministerio de Educación Superior (MES), el Ministerio de la Industria Básica (MINBAS), el Ministerio del Turismo (MINTUR) y el Ministerio de la Informática y las Comunicaciones (MIC).

La XETID se encuentra inmersa en el desarrollo de una plataforma de análisis e inteligencia, desde hace varios años, que incluye varias herramientas de Inteligencia de Negocios como el Sistema de Obtención y Catalogación Automática de Contenidos de Internet (SOCACI). Este

sistema, constituye una herramienta desarrollada con el objetivo de apoyar las tareas de Vigilancia e Inteligencia Empresarial que desarrollan analistas de diferentes áreas. Permite realizar tanto búsquedas como descargas de forma automática de contenidos de interés publicados en Internet a partir de un conjunto de temáticas previamente definidas y un conjunto de fuentes identificadas. Posibilita, además, visualizar los contenidos descargados según la categoría en que fueron clasificados, la fecha de descarga y el lenguaje del mismo, así como eliminar, ver los detalles de cada contenido y buscar un documento (Varona Labrada & Castellano Leyva, 2018).

El sistema se encuentra en la versión 1.0 actualmente, la cual efectúa la obtención y catalogación de contenidos de Internet como uno de los métodos para lograr el aseguramiento y la disponibilidad de la información primaria. Dicho sitio no está siendo funcional, debido a que los requisitos funcionales y necesidades del cliente para la realización del sitio han cambiado drásticamente y las tecnologías utilizadas en la primera versión del sitio son obsoletas ya que fueron definidas hace varios años, no son eficientes y deben ser remplazadas. Existe un conjunto de requerimientos funcionales y tecnológicos, que harían de él una herramienta de monitoreo a la altura de los procesos de Vigilancia e Inteligencia actuales, su desarrollo implicaría efectuar cambios tanto en el frontend como en el procesamiento interno de la plataforma (backend). Al dejar de ser un sistema independiente, para formar parte de una nueva plataforma aún más abarcadora, se encuentra limitada en cuanto:

- Ausencia de un proyecto que financiara el desarrollo de SOCACI como un producto de software de vigilancia tecnológica, y no solo como un servicio de monitoreo de noticias.
- Necesita mostrar la misma identidad que la plataforma de la cual va a formar parte.
- El diseño actual solo es un producto básico demostrativo sobre las capacidades básicas del servicio de descarga y catalogación.
- No muestra un diseño adaptativo a la identidad visual de la empresa en la cual va a ser desplegada.
- Carencia de opciones de menú impidiendo interactuar de forma intuitiva y agradable con el sitio.
- El sistema de almacenamiento de la información no está capacitado para manejar un alto volumen de datos.

- EL sistema de almacenamiento no cuenta con la velocidad de búsqueda y análisis requerida, demostrando una gestión engorrosa.
- No cuenta con funcionalidades de análisis y difusión, como el control y visualización de las estadísticas de descargas y catalogaciones realizadas.
- No muestra la información de los resultados en cuanto a descargas y evaluaciones realizadas.
- No concreta un proceso a partir del cual se obtiene, analiza, interpreta y finalmente divulga determinada información.
- El sistema actual carece de elementos esenciales para completar el flujo del proceso de Vigilancia Tecnológica, el análisis y la interpretación de los datos recopilados, convirtiéndolos en información estratégica y útil para la toma de decisiones de las entidades.

Lo antes expuesto evidencia la necesidad de realizar una evolución tecnológica en el Sistema de Obtención y Catalogación de Contenidos de Internet hacia una versión 2.0. Por lo cual se evidencia el siguiente **Problema de investigación científica**: ¿Cómo lograr una evolución tecnológica en la versión del Sistema de Obtención y Catalogación de Contenidos de Internet, mediante la implementación de la versión 2.0 del mismo?

Se determina como **objeto de estudio**: Gestión del proceso de vigilancia e inteligencia empresarial basado en la tecnología web.

Campo de acción: Sitio web para la búsqueda y obtención de contenidos y su catalogación basada en la tecnología web.

Objetivo general: Desarrollar el Sistema de Obtención y Catalogación de Contenidos de Internet en su versión 2.0 mediante el uso de la tecnología web.

Para guiar la investigación y garantizar la solución del problema expuesto se proponen los siguientes **objetivos específicos**:

1. Sistematizar los principales referentes teóricos de la gestión del proceso de vigilancia e inteligencia empresarial basado en la tecnología web para el desarrollo del sistema de obtención y catalogación de contenidos de internet en su versión 2.0.
- 2.
3. Diagnosticar los principales requerimientos del sitio web para la búsqueda y obtención de contenidos y su catalogación basada en tecnología web.

4. Identificar la metodología y tecnologías informáticas para el desarrollo del Sistema de Obtención y Catalogación de Contenidos de Internet en su versión 2.0.
5. Validar realizando las pruebas de desarrollo de software al Sistema de Obtención y Catalogación de Contenidos de Internet en su versión 2.0 para comprobar el correcto funcionamiento del sitio web antes de entregarlo al cliente final.

Para garantizar el cumplimiento de los objetivos específicos identificados, se extrajeron las siguientes **tareas de investigación**:

1. Revisión de los referentes literarios para el análisis de los fundamentos teóricos y metodológicos.
2. Caracterización de la gestión del proceso de vigilancia e inteligencia empresarial.
3. Caracterización de la gestión de obtención y catalogación de contenidos de internet.
4. Identificación de los requisitos funcionales y no funcionales sobre el proceso de obtención y catalogación de contenidos de internet.
5. Identificación de la metodología para el desarrollo del software del sitio web.
6. Selección de las tecnologías y herramientas mediante las cuales se va a desarrollar el sitio de obtención y catalogación de contenidos de internet.
7. Selección de la arquitectura mediante la cual se va a llevar a cabo el diseño del sitio web.
8. Realización de las pruebas convenientes y necesarias para validar el sitio web.

La investigación será fundamentalmente de tipo descriptiva porque se especificarán las propiedades más importantes del proceso de gestión de vigilancia e inteligencia empresarial y las variables descriptivas del Sistema. Existe documentación acerca del tema de investigación, de la cual se realizará un análisis vinculado al objeto de estudio.

La investigación se empleará por la vía inductiva, al utilizar en el trabajo una estructura de pregunta científica y tareas de investigación.

Métodos científicos:

Para obtener los conocimientos necesarios con la finalidad de hacer posible el cumplimiento del objetivo trazado, se llevó a cabo una investigación en la que se utilizaron algunos de los métodos científicos existentes, como son los métodos teóricos y empíricos.

Los métodos de investigación empíricos están asociados a la obtención de datos a través de diferentes investigaciones con referencia a la problemática antes planteada y los teóricos permiten revelar las relaciones esenciales del objeto de investigación.

Métodos teóricos utilizados:

Análisis histórico-lógico: Para investigar sobre la evolución de los procesos de vigilancia y evolución empresarial, así como las herramientas informáticas empleadas sobre esta actividad desarrollada tanto a nivel internacional como nacional.

Analítico-Sintético: Utilizada para concretar los requerimientos expuestos por el cliente para satisfacer sus necesidades durante el desarrollo del Sitio de Obtención y Catalogación de Contenidos de Internet.

Modelación: Para modelar los fundamentos, etapas y componentes del Sitio Web a desarrollar.

Sistémico: Para integrar armónicamente los componentes diseñados en la solución propuesta.

Métodos empíricos utilizados:

Observación: Para visualizar el contexto de sitio web en el proceso de obtención y catalogación de contenidos de internet.

Entrevista semiestructurada: Utilizadas para obtener un testimonio oral sobre el sistema que se desea desarrollar y poder conformar un suceso histórico del mismo.

Análisis documental: en la revisión de la literatura especializada, tanto académica como empresarial, para extraer la información necesaria que permitió realizar el proceso de investigación.

Tormenta de ideas: para refinar el diseño de la solución propuesta.

Pruebas de calidad: para evaluar la calidad de la solución desarrollada.

Controles de aplicabilidad, para validar con los profesionales la aplicabilidad de la solución desarrollada.

La presente investigación tendrá como aporte teórico-práctico la investigación e implementación de la versión 2.0 del Sistema de Obtención y Catalogación de Contenidos de Internet, facilitando una mejor experiencia de usuario al corregir los fallos y detalles detectados en su versión actual.

Estructura general del documento

El presente trabajo de diploma se encuentra estructurado por introducción, tres capítulos, conclusiones, recomendaciones, bibliografías y anexos. Los temas trabajados en cada capítulo se describen a continuación:

Capítulo 1. Informatización de la versión 2.0 del Sistema de Obtención y Catalogación de Contenidos de Internet. En este capítulo se exponen los principales conceptos asociados a la solución en cuestión. Se realiza un análisis de las principales soluciones existentes para así identificar posibles características de las mismas que puedan aportar al diseño y desarrollo de la aplicación. Por último, se define la metodología, así como las herramientas que se utilizarán en la implementación y futuro despliegue del sitio web.

Capítulo 2. Análisis y diseño de la solución: Desarrollo de la versión 2.0 del Sistema de Obtención y Catalogación de Contenidos de Internet. Serán definidas las funcionalidades que se informatizarán, el modelado del negocio y los diagramas y tablas para lograr una mejor comprensión del mismo. Además, se completará el modelado del sistema, serán determinados y especificados los requisitos funcionales y no funcionales, se especificarán los patrones utilizados, así como otras características fundamentales como parte del diseño, generándose además un grupo de artefactos teniendo en cuenta la metodología a utilizar.

Capítulo 3. Implementación y pruebas del sistema para la Obtención y Catalogación de los Contenidos de Internet. Se describe el proceso de implementación y pruebas de la solución. Se especifican la selección, diseño y aplicación de las pruebas realizadas en correspondencia a la tipología de la solución. Finalmente se muestran los resultados obtenidos una vez concluida la ejecución de pruebas.

CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE EL OBJETO DE ESTUDIO

Es fundamental referenciar la evolución del desarrollo e industria de cada proyecto, así como del estado actual del mundo al que pertenece. En el presente capítulo, dedicado a comprender el objeto de estudio de la investigación, se abordan los conceptos, propiedades y características del proceso de obtención y catalogación de contenidos de internet. Se exponen las soluciones similares existentes, se define la metodología que se llevará a cabo durante el desarrollo del proyecto y las tecnologías empleadas en el desarrollo de un sistema de obtención y catalogación.

I.1 Gestión del proceso de vigilancia e inteligencia empresarial I.1

Los Sistema de Gestión son considerados valiosas herramientas para la captura, almacenamiento, creación, modificación y análisis de la información. Una gestión eficiente en cualquier proceso a llevar a cabo, permitirá controlar, planificar, organizar o automatizar en gran medida su desarrollo. Logrando así agilizar el tiempo de ejecución de las tareas y otorgar una mayor eficiencia en la obtención de los resultados, disminuyendo el tiempo de respuesta, debido a que todos los datos se encuentran centralizados en el sistema, por lo que garantiza la disponibilidad de los mismos ante cualquier necesidad o consulta en tiempo real. Además, posibilita una mejor rentabilidad al adecuarse en las distintas necesidades de los clientes, incidiendo en las propiedades y la eficiencia de los procesos (Evaluando ERP, 2022).

Al hablar de Vigilancia e Inteligencia, se hace referencia a un sistema que ha venido evolucionando con el paso del tiempo, este proceso se ha ido perfeccionando y aun cuando está dotado de flexibilidad para ser implementado en cualquier tipo de organización, es importante asumirlo como un cambio de paradigma dentro de la propia cultura organizacional (Universidad Politécnica de Valencia, 2022).

La Vigilancia Empresarial es la práctica de los empleadores que monitorean las actividades de sus empleados mientras trabajan los equipos provistos por el empleador. Algunas formas de vigilancia comercial son controvertidas dado que un cuantioso número de empleados afirman que sus derechos de privacidad han sido violados. Los empleadores afirman que la práctica es necesaria por diversas razones, como la seguridad de la información privada y la protección

contra el robo y el vandalismo. Muchas formas de vigilancia empresarial son sencillas y no controvertidas. Los empleadores que participan en la fabricación, el almacenamiento o la distribución están constantemente preocupados por la «reducción», que es la pérdida de inventario debido a hurto o robo total (Spiegato , 2022).

La Inteligencia Empresarial o Business Intelligence (BI), ayuda a las organizaciones a analizar datos históricos y actuales para que puedan extraer conocimientos útiles con rapidez para tomar decisiones estratégicas. Las herramientas de inteligencia empresarial hacen esto posible mediante el procesamiento de grandes conjuntos de datos en varios orígenes y la presentación de las conclusiones en formatos visuales fáciles de comprender y compartir. La inteligencia empresarial sigue cuatro pasos clave para transformar datos sin procesar en conocimientos fáciles de resumir para que los utilicen todos los usuarios de la organización. Los tres primeros (recopilación, análisis y visualización de datos) definen la etapa para el paso final de la toma de decisiones. Antes de utilizar BI, las empresas tenían que realizar manualmente gran parte del análisis, pero las herramientas de BI automatizan muchos de los procesos y ahorran tiempo y esfuerzo a las empresas. Entre los beneficios concretos que se experimenta al utilizar BI se incluyen (Microsoft, 2022):

- Aumento de la eficiencia de los procesos operativos.
- Conocimientos sobre el comportamiento y los patrones de compra del cliente.
- Seguimiento preciso de las ventas, el marketing y el rendimiento financiero.
- Pruebas comparativas claras basadas en datos históricos y actuales.
- Alertas instantáneas sobre anomalías de datos y problemas de los clientes.
- Análisis que se pueden compartir en tiempo real entre departamentos (Microsoft, 2022).

La Vigilancia e Inteligencia Empresarial (VIE) constituye una herramienta de gestión, definida como un sistema integrado de gestión, legal y ético, a través del cual se recolecta y analiza información abierta sobre el entorno, los competidores y la propia organización que se disemina mediante productos de inteligencia para mejorar la toma de decisión, entre otros propósitos. Además, un aspecto nuevo es la vigilancia e inteligencia en red mediante la interacción de diversos nodos de organizaciones diferentes. En Cuba el marco regulatorio de la vigilancia e inteligencia empresarial está relacionado con los documentos programáticos y por resoluciones del Ministerio del Comercio Exterior y la Inversión Extranjera (Pérez, 2019).

I.2 Sitio web para la búsqueda y obtención de contenidos y su catalogación I.2

La búsqueda y obtención de información es el nombre del correspondiente campo de estudios (information seeking and retrieval en inglés). Tiene gran cantidad de puntos en contacto con la recuperación de información (RI), pero es una disciplina independiente y con la que, por supuesto, se complementa en lo que podría consistir el campo más amplio de la búsqueda de información (Hearst, 2013) (Gardner, 2021).

“La búsqueda de información se ocupa del estudio de los sistemas de información existentes, de la oferta de tales sistemas en el mundo real y de las características de sus interfaces de usuario”

A continuación, se exponen los principales conceptos relacionados con el Marco Teórico de la investigación, con el objetivo de profundizar en los distintos puntos de vista y asumir una posición al respecto.

Internet según la Real Academia Española es una red informática mundial, descentralizada, formada por la conexión directa entre computadoras mediante un protocolo especial de comunicación (Real Academia Española, s.f.). Según el diccionario de Oxford es una red informática de nivel mundial que utiliza la línea telefónica para transmitir la información (Oxford). Es un neologismo del inglés que significa red informática descentralizada de alcance global. Se trata de un sistema de redes interconectadas mediante distintos protocolos que ofrece una gran diversidad de servicios y recursos, como, por ejemplo, el acceso a archivos de hipertexto a través de la web (Coelho, 2013).

Un sitio web es la pieza básica de Internet. Se navega por ellos a todas horas, desde el móvil o desde el ordenador. Es un conjunto de páginas web almacenadas en un servidor, interrelacionadas entre sí y casi siempre identificadas con un mismo nombre de dominio (Webolto, 2021).

Contenido según el diccionario de Oxford Language es una cosa contenida en otra. Conjunto de temas o argumento de una obra o escrito (Oxford).

Se debe considerar la información como algo que se da o se lleva (Cevolini, 2016).

Búsqueda es el intento de encontrar algo deseado a través de la atención, el seguimiento, la persecución o cualquier tipo de esfuerzo ya sea algo material o inmaterial (Coelho, 2013).

Las búsquedas de información consisten en un proceso iterativo en el que se realiza una búsqueda, se analizan los resultados y se va modificando la estrategia de búsqueda hasta identificar los términos y las fuentes de información que nos proporcionan los resultados más pertinentes (Universidad Politécnica de Valencia, 2022). Existen dos tipos de información, la analógica y la digital:

- La búsqueda de información analógica se hace a través de libros, personas, entrevistas, pistas u otros medios, cuyos datos son recopilados y elegidos, organizados y clasificados por el criterio de personas y no por la predefinición a través de máquinas o programas (Coelho, 2013).
- La búsqueda de información digital se efectúa, generalmente, a través de motores de búsquedas, también llamados buscadores, que se caracterizan por contener una gran cantidad de información organizada y categorizada programados para arrojar resultados predefinidos según las palabras claves que se introduzcan (Coelho, 2013).

La búsqueda de información se puede realizar a través de varios algoritmos, ya sea por búsqueda parametrizada o por índice.

- Búsquedas parametrizadas: es decir, búsquedas por campos, por secciones o categorías, por tipo de documento, etc.
- Búsqueda por índices: autores, títulos, etc.

Obtención según el diccionario de la Real Academia Española es una acción y resultado de obtener, alcanzar, conseguir y lograr algo que se merece, solicita o pretende. Fabricar o extraer un material o un producto con ciertas cosas o de cierta manera (Real Academia Española, s.f.).

Catalogación es la acción o efecto de catalogar, lo cual significa, según el diccionario de Oxford Language, clasificar libros, documentos, monedas, objetos en venta, para formar un inventario de ellos o incluirlos en uno ya elaborado (Oxford).

La catalogación no es en exclusiva materia de gusto, sino que también debe considerar utilidades, teniendo en cuenta que, en función de su logro, se ahorrará una gran cantidad de tiempo. La catalogación es un asunto de categorías y como tal requiere tratarse con gran cuidado, para aumentar las posibilidades de encontrar un libro, se sugiere desde el principio que se incremente el número de inventarios y que al mismo tiempo se establezcan conexiones entre ellos (Hottinger).

Atendiendo a los conceptos anteriores, se esclarecen varios conceptos de búsqueda y se puede conformar el concepto de obtención y catalogación de Contenidos de Internet, como un catálogo que genere al usuario una búsqueda más confiable y rápida de la información solicitada acertando a sus verdaderos intereses.

I.3 Análisis de soluciones informáticas similares existentes I.3

Existen múltiples entidades que emplean este tipo de sistemas, pero no están conjuntamente desarrolladas. A continuación, se exponen varios ejemplos de sitios web similares que fueron de gran importancia para la construcción de la solución al problema de la presente investigación debido a que, mediante estos, fueron tomados los conceptos fundamentales del proceso de investigación, que posibilitaron además una mejor comprensión sobre el funcionamiento de las herramientas que trabajan en la base de la obtención y catalogación sirviendo de referencias para el análisis del diseño estructural de la investigación.

Koha

Koha es el primer software libre de código abierto de automatización de bibliotecas totalmente web, permitiendo al bibliotecario administrar la biblioteca remotamente utilizando un teléfono móvil o un asistente personal. Este Sistema Integrado de Gestión Bibliotecaria (SIGB) ofrece una completa gestión basada en una solución software, Web 2.0, utilizando un avanzado sistema de Relational Database Management System (RDBMS) y las mejores tecnologías de software a lo largo de su diseño, facilitando la adquisición, mantenimiento y difusión de colecciones bibliográficas y colecciones digitales. Además, cuenta con una plantilla de catalogación rápida en caso que el material no esté ingresado (Open Geek).

Permite la catalogación de nuevos registros bibliográficos vía plantilla de catalogación, consta de un formulario con un significativo número de campos mediante los cuales se definen varias de sus características, se especifica por ejemplo tipo de material, tipo de datos, autor, ilustración, etc., solamente recomendable para registros que no hayan sido localizados en los catálogos de referencia (Orex).

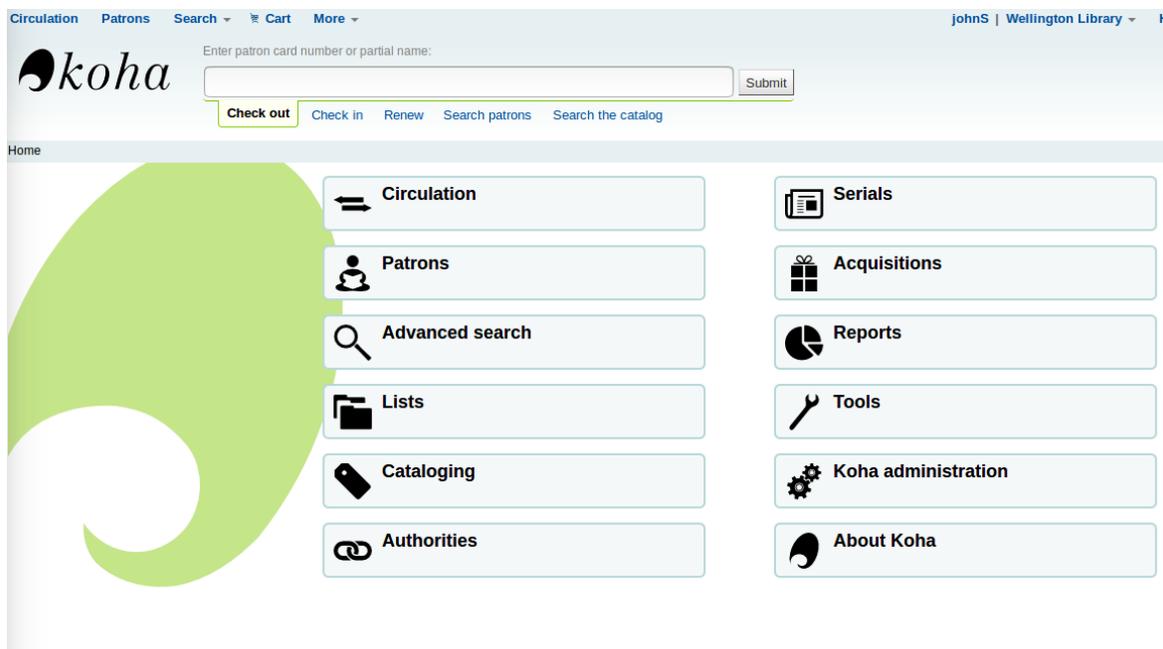


Ilustración 1. Vista de Koha (Orex)

Google Académico

Google Académico es un buscador gratuito que permite localizar documentos académicos como artículos, tesis, libros y resúmenes de fuentes diversas como editoriales universitarias, asociaciones profesionales, repositorios de preprints, universidades y otras organizaciones. Mediante un proceso muy similar al buscador de Google, introduciendo las palabras claves y filtros correspondientes, te permite encontrar las publicaciones agrupadas por el investigador y su institución en un único perfil mejorando su visibilidad e impacto al hacer su perfil público (Google. (s.f.). *google scholar*. Obtenido de scholar.google.com: <https://scholar.google.com>).



Ilustración 2. Vista de Google Académico (Google. (s.f.). *google scholar*. Obtenido de scholar.google.com: <https://scholar.google.com>)

SciELO Data

SciELO Data es un repositorio multidisciplinario para depositar, preservar y difundir datos de investigación de artículos enviados, aprobados para publicación o ya publicados en revistas de la Red SciELO o depositados en SciELO Preprints. Ofrece a la comunidad de autores y editores de revistas de la Red SciELO y usuarios de SciELO Preprints un repositorio de datos de investigación que adopta las mejores prácticas y estándares internacionales para la gestión de datos de investigación. Las revistas SciELO operan un repositorio para depositar, preservar y difundir datos relacionados con los artículos que publican. Desde enero de 2021, el proyecto SciELO promueve la mejora de un grupo técnico en gestión de datos de investigación y su funcionamiento en el sistema Dataverse, que es responsable del desarrollo de SciELO Data (SciELO. (s.f.). *SciELO*. Obtenido de scielo.org: <https://scielo.org>).



Ilustración 3. Vista de SciELO Data (SciELO. (s.f.). SciELO. Obtenido de scielo.org: <https://scielo.org>)

SOCACI

El Sistema de Obtención y Catalogación Automática de Contenidos de Internet permite realizar un constante monitoreo de sitios públicos de información, a través de la descarga y catalogación de forma automática de contenidos de texto de interés, partiendo de un conjunto de temáticas previamente definidas y un conjunto de fuentes identificadas en Internet. El sistema está compuesto por tres componentes fundamentales:

Fuentes: Las fuentes constituyen los sitios públicos que el sistema debe monitorear y que deben ser definidas por el usuario.

Categorías: Las categorías establecen las temáticas en las que el sistema debe clasificar los contenidos descargados de las fuentes. Estas categorías deben ser establecidas por el usuario y para ello debe proporcionar un conjunto de muestras que le permitan al sistema reconocer cuándo un contenido de texto pertenece a dicha categoría.

Descargas: El componente de descarga contiene el registro de contenidos descargados y clasificados por el sistema. Posibilita la obtención de un producto informativo con estadísticas de los resultados obtenidos.

De manera general el sistema implementa un proceso de vigilancia informativa para convertir los datos que se publican en información útil para una organización (Varona Labrada & Castellano Leyva, 2018).



Ilustración 4. Proceso de Vigilancia Informativa del Sistema de Obtención y Catalogación Automática de Contenidos de Internet

Cada componente y función que provee colaboran para desarrollar las actividades siguientes:

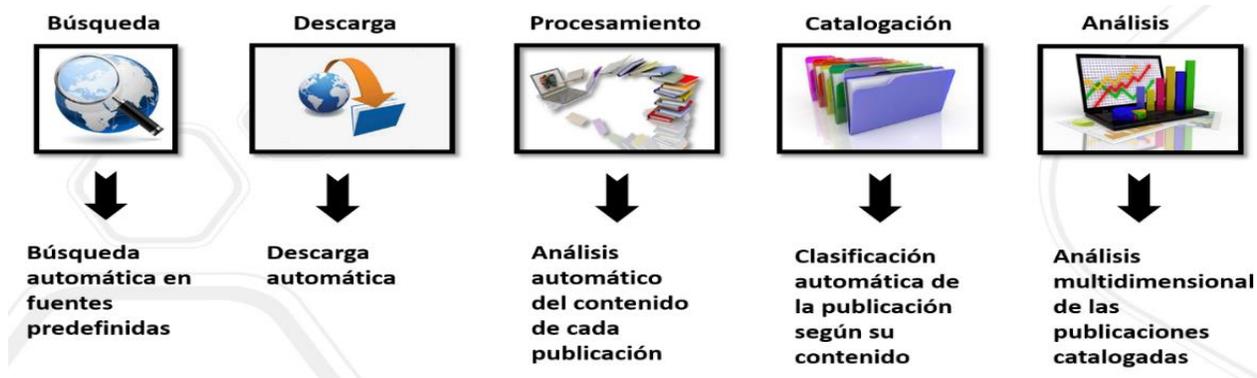


Ilustración 5. Componentes y sus funciones (Varona Labrada & Castellano Leyva, 2018).

I.3.1 Valoración de los estudios realizados

Al culminar los estudios de las soluciones planteadas, se constató, que las mismas, presentan funcionalidades que se pueden asumir dada su utilidad, puesto que proporcionarían importantes cualidades a la solución en cuestión, así como, el proceso de obtención y catalogación analizado en cada una de las soluciones internacionales y la solución nacional analizadas. A demás a la versión actual le serán añadidas nuevas funcionalidades, ya que esta nueva versión contará con la gestión de las estadísticas, y se le realizarán cambios a las tecnologías en las cuales está desarrollado y al sistema de almacenamiento y la parte visual y lógica del sitio actual.

A partir de los análisis realizados se concluye con la necesidad de desarrollar una versión 2.0 al sistema SOCACI que permita agilizar el proceso de obtención y catalogación de la información solicitada, así como gestionar las estadísticas permitiendo cuantificar la interacción con la información, para disponer de los elementos que permitan un análisis profundo.

I.4 Metodología para dar solución al problema de investigación I.4

Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el objetivo de lograr la eficiencia del producto, donde predecir no significa perder la capacidad adaptativa, no significa evitar la introducción de cambios en los requisitos, ni evitar que nuevos requisitos surjan, sino definir un camino reproducible para obtener resultados confiables. Definen, además, una representación que permite facilitar la manipulación de modelos, la comunicación e intercambio de información entre todas las partes involucradas en la construcción de un sistema (Gacitúa, R. ,2013).

Existen dos tipos de metodologías para realizar el proceso de gestión de proyectos, las tradicionales y las ágiles:

Las metodologías tradicionales tienen un enfoque predictivo donde se sigue un proceso secuencial en una dirección única y sin marcha atrás (Tribalyte Technologies).

Las metodologías ágiles surgen como alternativa a las tradicionales porque ayudan a reducir la probabilidad de fracaso por sub-estimación de costos, tiempos y funcionalidades en entornos cambiantes (Tribalyte Technologies). Las metodologías ágiles se centran más en el factor humano. Su base está en la simplificación sin renunciar a las prácticas esenciales que aseguran la calidad del producto, esto se refleja en el hecho de ser la persona, el principal factor de éxito en un proyecto, la habilidad está en centrarse más en los cambios que puedan surgir y no en seguir estrictamente un plan. 'Agile' es mucho más que una metodología para el desarrollo de proyectos que precisan de rapidez y flexibilidad, es una filosofía que supone una forma distinta de trabajar y de organizarse. De tal forma que cada proyecto se 'trocea' en pequeñas partes que tienen que completarse y entregarse en pocas semanas. El objetivo es desarrollar productos y servicios de calidad que respondan a las necesidades de unos clientes cuyas prioridades cambian a una velocidad cada vez mayor (¿Qué es la metodología 'agile'?, 2022).

Las principales ventajas del 'agile' son:

1. Mejora la calidad: Minimiza los errores en los entregables y mejora la experiencia y las funcionalidades para el cliente.
2. Mayor compromiso: Mejora la satisfacción del empleado y genera conciencia de equipo.
3. Rapidez: Acorta los ciclos de producción y minimiza los tiempos de reacción y toma de decisiones.
4. Aumento de la productividad: Al asignar mejor los recursos, y de forma más dinámica, mejora la producción según las prioridades que tenga la empresa.

Xtreme Programation o simplemente XP es una metodología de desarrollo de software que forma parte de lo que se conoce colectivamente como metodologías ágiles. Se empleó para conducir el diseño y desarrollo de la solución informática propuesta. Esta metodología, facilita las modificaciones en el desarrollo del producto, ya que toma como punto de partida la idea de

que ocurrirán cambios naturales durante el ciclo de vida del software, considerados como favorables. Aminora el tiempo de desarrollo del software ya que solo genera los artefactos que el programador determina que son necesarios para el correcto desarrollo de la investigación. Es observable la eficiencia durante el proceso de pruebas y planificación, su tasa de error es muy pequeña, facilita los cambios, origina una programación muy organizada y la satisfacción del programador. Permite estar en constante intercambio con el cliente, ya que fomenta la comunicación entre desarrolladores y clientes (Fernández Escribano, 2002).

XP se basa en valores, principios y prácticas, y su objetivo es permitir que equipos pequeños y medianos produzcan software de alta calidad y se adapten a los requisitos cambiantes y en evolución. Lo que diferencia a XP de las demás metodologías ágiles es que hace hincapié en los aspectos técnicos del desarrollo de software. La programación extrema es precisa sobre cómo trabajan los ingenieros, ya que seguir las prácticas de ingeniería permite a los equipos entregar código de alta calidad. La programación extrema consiste, en las buenas prácticas llevadas al extremo (Digité, 2022).

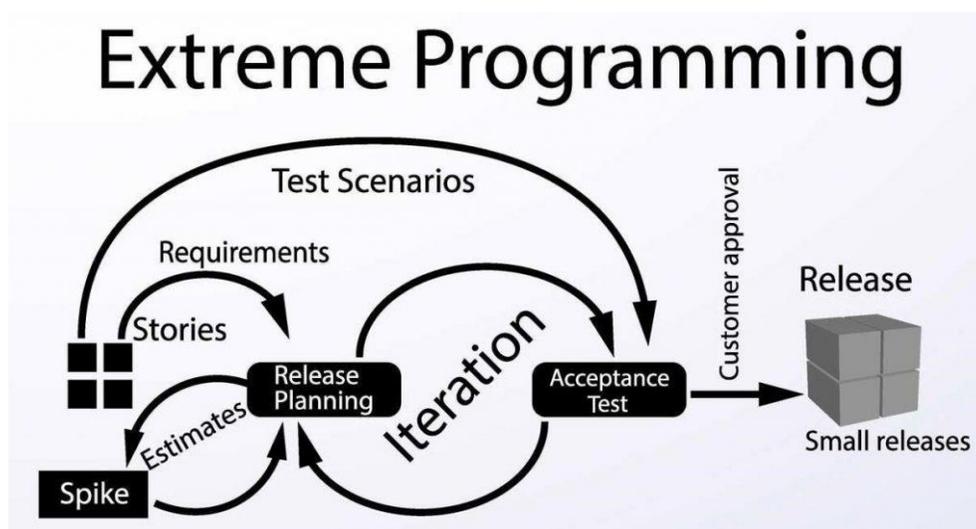


Ilustración 6. Funcionamiento de XP (Digité, 2022)

Esta metodología parece la más útil ya que se basa en la autosimilaridad, ya que se creó con la opinión de que, si una solución determinada funciona en un nivel, también podría funcionar

en un nivel superior o inferior. Por ejemplo, la obtención de una retroalimentación temprana y constante está en juego en varios niveles de la XP (Digité, 2022).

- A nivel de desarrollador, los programadores reciben información de su trabajo utilizando el enfoque de «primero la prueba»;
- A nivel de equipo, el pipeline de integración continua integra, construye y prueba el código varias veces al día;
- A nivel de la organización, los ciclos semanales y trimestrales permiten a los equipos obtener información y mejorar su trabajo según sea necesario.

Seleccionada la metodología que se empleará para dar solución al problema de investigación planteado se analizarán en el siguiente epígrafe las propuestas de tecnologías y herramientas a emplear en dicha solución.

I.5 Herramientas y tecnologías para dar solución al problema de investigación I.5

Actualmente el desarrollo de las tecnologías es cada vez más elevado. Esto trae consigo que exista gran diversidad de herramientas de software para el trabajo de desarrollo de una aplicación informática. Por tal motivo, se hace imprescindible seleccionar cuidadosamente las herramientas a utilizar para desarrollar un buen producto informático. A continuación, se describen las principales herramientas y tecnologías que se utilizarán para el desarrollo de la investigación y la implementación de la solución.

I.5.1. Arquitecturas del sistema

Se utilizaron varias herramientas de modelado como lo es Visual Paradigm y para la edición Visual Studio Code, y tecnologías para el desarrollo de la solución al problema de investigación como el lenguaje de datos PostgreSQL y el lenguaje de programación Java Script para construir el sistema con el que se relaciona la propuesta de solución.

Herramienta de modelado: Visual Paradigm para UML 16.3

La herramienta CASE (Computer Aided Software Engineering, Ingeniería de Software asistida por Computadora para el modelado se selecciona el Visual Paradigm para UML 16.3, la misma soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a

objetos, construcción, pruebas y despliegue y contribuye a una rápida construcción de aplicaciones de calidad, mejores y a un menor costo (Pressman, R, 2002).

Visual Paradigm para UML (Standard Edition) es una potente herramienta que genera código a partir de diagramas y genera documentación. Puede dibujar todo tipo de diagramas de clase, invertir el código, generar código a partir de diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales UML, demostraciones interactivas UML y proyectos UML.

Las principales características que avalan el uso de esta herramienta son:

- Entorno de creación de diagramas para UML 2.0
- Uso de un lenguaje estándar común para el equipo de desarrollo
- Diseño enfocado en el negocio y centrado a casos de uso
- Interoperabilidad entre diagramas permitiendo a partir de un diagrama obtener otro que guarde relación con este
- Disponible en múltiples plataformas
- Capacidad de Ingeniería directa e inversa
- Soporte de análisis OO (OOA), diseño OO (OOD)
- Modelado de casos de uso
- Análisis textual para la identificación de actores candidatos, casos de uso, clases
- Diagrama de tarjetas CRC (Clase-Responsabilidad-Colaborador)
- Diagrama de flujo de trabajo empresarial
- Proceso, Decisión, Actor empresarial, Documento
- Ingeniería incremental de ida y vuelta
- Ingeniería inversa: código a modelo, código a diagrama (Java a modelos UML, Java a diagrama de clases)
- Generación de código: modelo a código, diagrama a código (modelo UML a código, diagrama de clases a Java)
- Sincronización automática entre el código fuente y los diagramas
- Generador de informes para generar documentación

- Disposición automática de diagramas - reordenar las clases y los conectores en los diagramas UML
- Importación de archivos XMI/exportación de archivos XMI
- Importación de archivos de proyecto Rational Rose
- Integración con Microsoft Visio: dibuje diagramas UML con plantillas de Visio
- Exportación de diagramas a imágenes (PNG, JPG, SVG)
- Control de versiones
- Soporte multilingüe



Ilustración 7. Visual Paradigm (Visual Paradigm. (2022). Visual-Paradigm. Obtenido de <https://www.visual-paradigm.com/>)

Editor de Código Fuente Ligero Visual Studio Code

Visual Studio Code es un editor de código fuente ligero, incluye soporte para depuración, control de Git integrado, resaltado de sintaxis, finalización de código inteligente, fragmentos de código y refactorización de código. Se encuentra disponible para sistemas operativos como Windows, macOS y Linux, posee soporte incorporado para JavaScript, TypeScript y Node.js, además de un variado ecosistema de extensiones para otros lenguajes como C ++, C #, Java, Python, PHP, Go y tiempos de ejecución como .NET y Unity. Una de las características más importantes de este software, se puede actualizar con frecuencia y rapidez de actualización de la extensión para que al menos se vea un número de actualizaciones que se suelen ver con las nuevas revisiones en cada nueva versión. Visual Studio Code no sólo es una herramienta

de cifrado, se pueden instalar muchas extensiones para compilar, depurar y embellecer el código (taiwebs).

Características de Visual Studio Code

- La mayoría de los lenguajes de programación
- Capacidad para agregar idioma por defecto de la extensión
- Hay gran cantidad de extensiones con la posibilidad de actualizaciones automáticas
- Las actualizaciones del programa de forma rápida y ordenada
- Entorno de codificación simple y hermoso
- La capacidad de utilizar las extensiones de compilación y depuración de
- Incluir código de mapa
- Código de auto completo
- Herramienta de búsqueda inteligente con el apoyo de Rex
- Texto Terminal de comandos
- Mostrar el parámetro de la función
- La capacidad para saltar directamente a la función y el método de
- Visualización de las Rebajas de archivo en la forma de la traducción
- Capacidad para utilizar el control de código fuente



Ilustración 8. Logo de Visual Studio Code (Microsoft. (2022). Visual Studio. Obtenido de <https://visualstudio.microsoft.com/es/>)

Lenguaje de modelado Unified Modeling Language UML 2.0

Para el desarrollo de la aplicación se utilizará UML 2.0 como el lenguaje de modelado con el que se realizarán los artefactos necesarios en el proceso de desarrollo del software. UML es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. Entrega una forma de modelar sucesos conceptuales como lo son procesos de negocio y funciones de sistema, además de sucesos concretos como lo son escribir clases en un lenguaje determinado, esquema de base de datos y componentes de software reusables (Histchfeld, N., & Salinas Caro, P.).

Entre sus principales características se encuentra:

- Permite modelar sistemas utilizando técnicas de programación orientadas a objeto.
- Permite documentar todos los artefactos de un proceso de desarrollo dígase requisitos, arquitectura, pruebas entre otros.
- Se logra especificar todas las decisiones de análisis, diseño e implementación mediante la construcción de modelos precisos y no ambiguos.

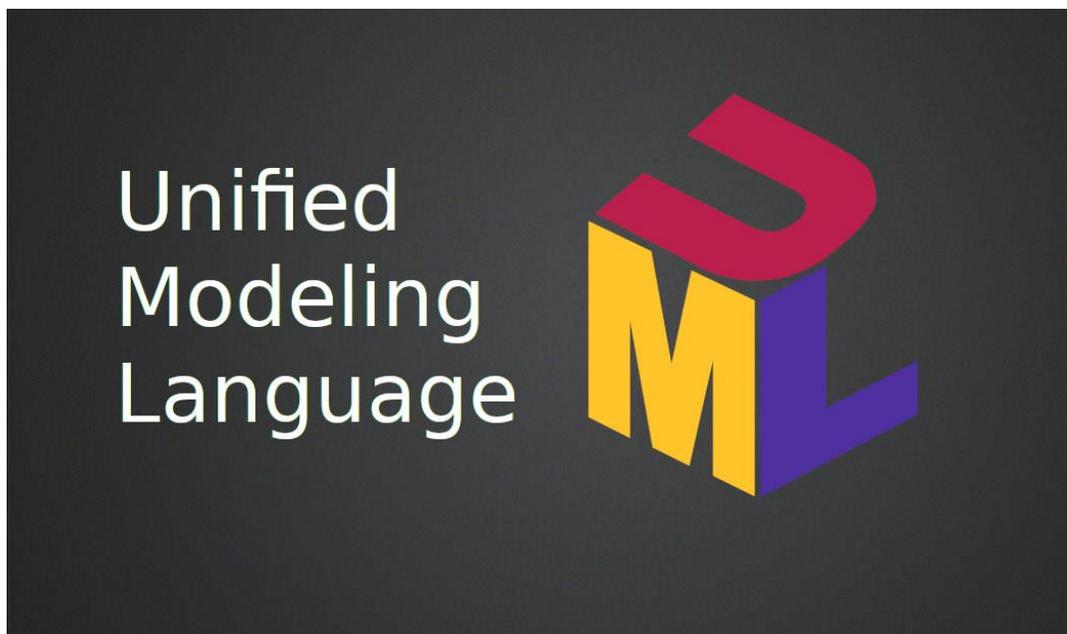


Ilustración 9. Logo de Unified Modeling Language (UML. (s.f.). [uml.org](http://www.uml.org/). Obtenido de <http://www.uml.org/>.)

Lenguaje de programación Java Script

Java Script (JS), originariamente llamado LiveScript, es un lenguaje de programación creado por Netscape y nació en 1995 con un propósito muy claro: Mejorar la navegación del usuario directamente desde el navegador. Es un lenguaje de programación ligero, interpretado, o compilado justo a tiempo (just-in-time) con funciones de primera clase. Si bien es más conocido como un lenguaje de scripting (secuencias de comandos) para páginas web, y es usado en muchos entornos fuera del navegador, tal como Node.js, Apache CouchDB y Adobe Acrobat JavaScript es un lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa, por ejemplo, programación funcional. Se ejecuta en el lado del cliente de la web, y se puede utilizar para estilizar/ programar cómo se comportan las páginas web cuando ocurre un evento. JavaScript es un potente lenguaje de scripts y fácil de aprender, ampliamente utilizado para controlar el comportamiento de las páginas web (mdn).

JavaScript es la tercera pieza fundamental del desarrollo web frontend, junto con los lenguajes HTML y CSS. Cada uno de estos tres lenguajes tiene una función muy concreta en el desarrollo web:

- El HTML se utiliza para conformar el esqueleto y la estructura de los contenidos de una página web.
- El CSS define el estilo y la apariencia web.
- JavaScript rompe con la estaticidad del HTML y permite crear elementos dinámicos e interactivos, mejorando ampliamente la interacción de los usuarios con una página web.

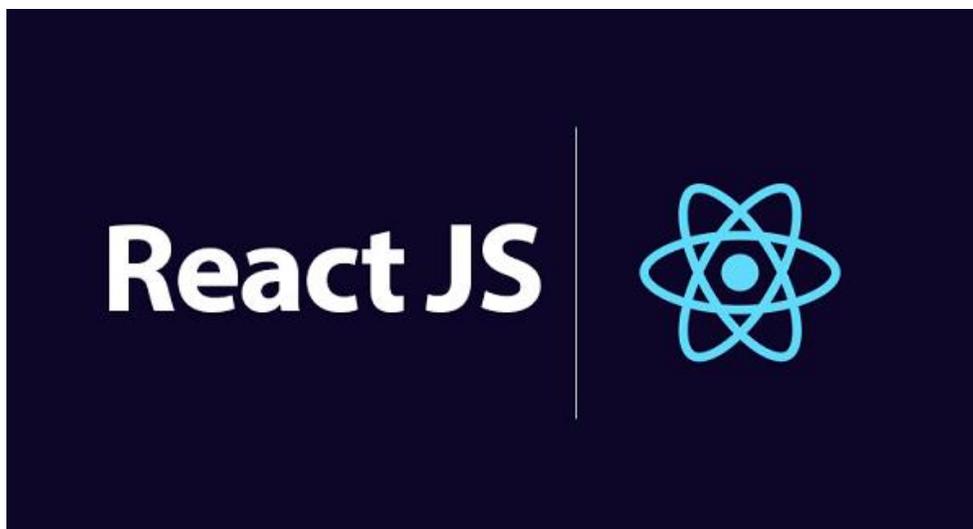


Ilustración 10. Logo de React.js, lenguaje de programación frontend (Obtenido de React: <https://es.reactjs.org/docs/testing.html#tools>, reactjs.org. (s.f.))

Características de JavaScript

- Lenguaje del lado del cliente: Esta categoría de lenguajes se diferencia de la otra gran categoría: los lenguajes del lado del servidor. Estos lenguajes se ejecutan e interpretan por el propio servidor y necesitan un tratamiento antes de mostrarlos al usuario final. Algunos de los lenguajes de programación del lado del servidor más conocidos son PHP, ASP o PERL.
- Lenguaje orientado a objetos: Utiliza clases y objetos como estructuras que permiten organizarse de forma simple y son reutilizables durante todo el desarrollo. Otros lenguajes orientados a objetos son Java, Python o C++.
- De tipado débil o no tipado: No es necesario especificar el tipo de dato al declarar una variable. Esta característica supone una gran ventaja a la hora de ganar rapidez programando, pero puede provocar que cometamos más errores que si tuviéramos esa restricción que poseen los lenguajes de tipado fuerte como C++ o Java.
- De alto nivel: Se le llama de “alto nivel” porque su sintaxis se encuentra alejada del nivel máquina, es decir, del código que procesa una computadora para ejecutar lo que nosotros programamos. Lo cual permite que su barrera de entrada y su curva de aprendizaje se acorte drásticamente. Un ejemplo podría ser que la sentencia condicional empiece por “IF” que significa “si...” en español, permitiendo asociar rápidamente su funcionamiento y

significado. Otro lenguaje de alto nivel muy utilizado y uno de los mejores para iniciarse en programación por esta característica es Python.

- Lenguaje interpretado: Utiliza un intérprete que permite convertir las líneas de código en el lenguaje de la máquina. Esto tiene un gran número de ventajas como la reducción del procesamiento en servidores web al ejecutarse directamente en el navegador del usuario, o que es apto para múltiples plataformas permitiendo usar el mismo código.
- Muy utilizado por desarrolladores: Es en la actualidad uno de los lenguajes más demandados de los últimos años por su versatilidad y su infinita capacidad para crear plataformas cada vez más atractivas (miteris, 2022).

JavaScript developers most common miscellaneous tech worked with

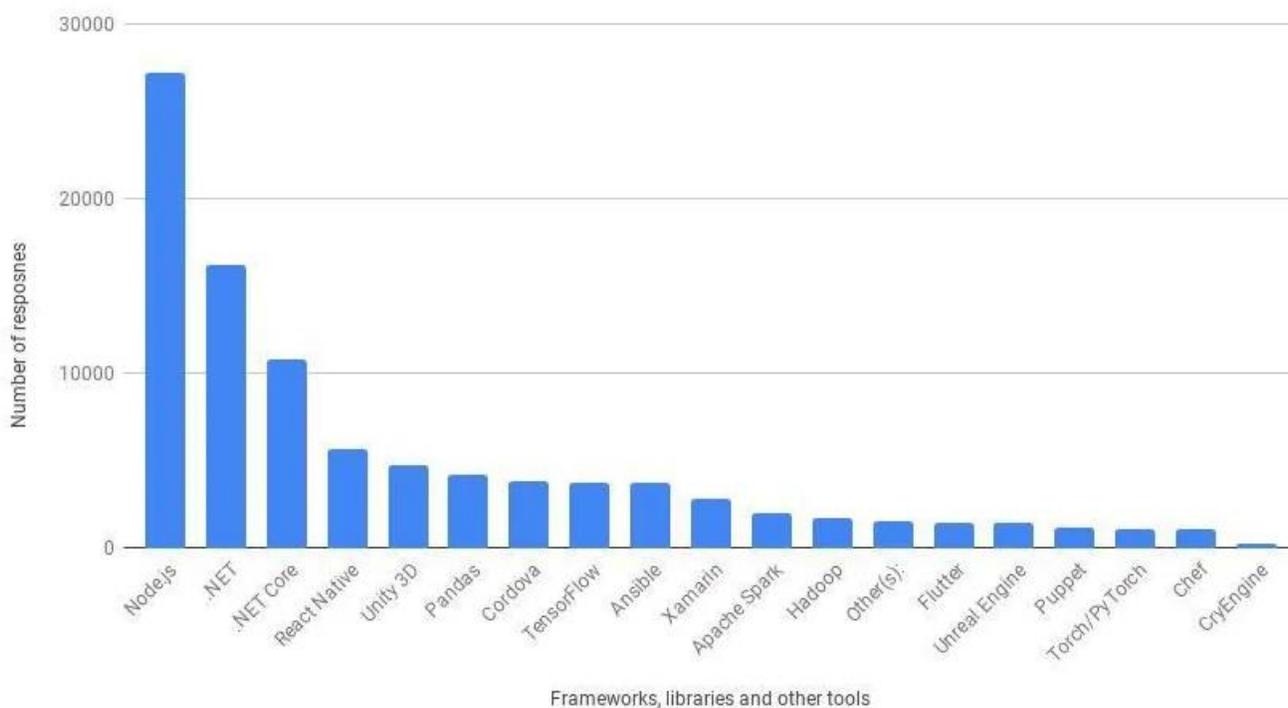


Ilustración 11. Desarrolladores más comunes de la tecnología miscelánea trabajando con Java Script (mdn. (s.f.). Acerca de JavaScript. Obtenido de www.mozilla.org: https://developer.mozilla.org/es/docs/Web/JavaScript/About_JavaScript)

Lenguaje de Programación Python

Python es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de *software*, la ciencia de datos y el *machine learning* (ML). Los desarrolladores utilizan Python porque es eficiente y fácil de aprender, además de que se puede ejecutar en

muchas plataformas diferentes. El *software* Python se puede descargar gratis, se integra bien a todos los tipos de sistemas y aumenta la velocidad del desarrollo.

Desarrollo web del lado del servidor

El desarrollo web del lado del servidor incluye las funciones complejas de backend que los sitios web llevan a cabo para mostrar información al usuario. Por ejemplo, los sitios web deben interactuar con las bases de datos, comunicarse con otros sitios web y proteger los datos cuando se los envía a través de la red.

Python es útil para escribir código del lado del servidor debido a que ofrece muchas bibliotecas que constan de código preescrito para crear funciones de backend complejas. Los desarrolladores también utilizan un amplio rango de marcos de Python que proporcionan todas las herramientas necesarias para crear aplicaciones web con mayor rapidez y facilidad. Por ejemplo, los desarrolladores pueden crear la aplicación web esqueleto en segundos porque no deben escribirla desde cero. Pueden probarla por medio de las herramientas de prueba del marco, sin depender de herramientas de prueba externas (python.org. (s.f.). Python. Obtenido de <https://docs.python.org/es/3/faq/general.html>).

¿Qué beneficios ofrece Python?

Los beneficios de Python incluyen los siguientes:

- Los desarrolladores pueden leer y comprender fácilmente los programas de Python debido a su sintaxis básica similar a la del inglés.
- Python permite que los desarrolladores sean más productivos, ya que pueden escribir un programa de Python con menos líneas de código en comparación con muchos otros lenguajes.
- Python cuenta con una gran biblioteca estándar que contiene códigos reutilizables para casi cualquier tarea. De esta manera, los desarrolladores no tienen que escribir el código desde cero.
- Los desarrolladores pueden utilizar Python fácilmente con otros lenguajes de programación conocidos, como Java, C y C++.

- La comunidad activa de Python incluye millones de desarrolladores alrededor del mundo que prestan su apoyo. Si se presenta un problema, puede obtener soporte rápido de la comunidad.
- Hay muchos recursos útiles disponibles en Internet si desea aprender Python. Por ejemplo, puede encontrar con facilidad videos, tutoriales, documentación y guías para desarrolladores.
- Python se puede trasladar a través de diferentes sistemas operativos de computadora, como Windows, macOS, Linux y Unix.

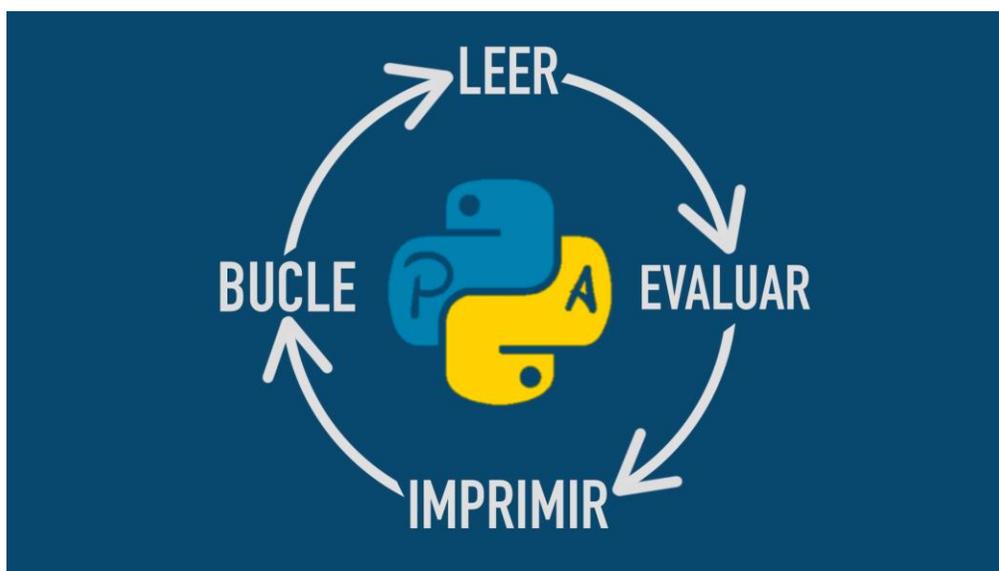


Ilustración 12. Python, lenguaje de Programación Backend (python.org. (s.f.). Python. Obtenido de <https://docs.python.org/es/3/faq/general.html>)

Framework empleado con Python

Django es un marco web de alto nivel en Python que fomenta el desarrollo rápido y el diseño limpio y pragmático de sitios web seguros y mantenibles. Construido por desarrolladores experimentados, se encarga de gran parte de las molestias del desarrollo web, para que usted pueda centrarse en escribir su aplicación sin necesidad de reinventar la rueda. Es gratuito y de código abierto, tiene una comunidad próspera y activa, una gran documentación y muchas opciones de soporte gratuito y de pago. Django fue diseñado para ayudar a los desarrolladores a llevar las aplicaciones desde el concepto hasta su finalización lo más rápidamente posible.

Ayuda a evitar muchos errores de seguridad comunes (Django Software Foundation. (2005-2022). *Meet Django*. Obtenido de [djangoproject.com](https://www.djangoproject.com): <https://www.djangoproject.com>).



Ilustración 13. Django is a popular Web framework written in Python (Unyscape, (2022).)

Sistema de gestión de bases de datos: PostgreSQL

Un sistema de gestión de bases de datos consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a los mismos, donde se permite el almacenamiento, manipulación y consulta de datos pertenecientes a una base de datos organizada en uno o varios ficheros.

Durante el transcurso de la investigación, se emplea PostgreSQL, el cual es un sistema de gestión de bases de datos objeto-relacional. Se encuentra distribuido bajo licencia BSD y tiene su código fuente disponible libremente. Es uno de los sistemas de gestión de bases de datos de código abierto más potente del mercado. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Esto ofrece como ventaja que un fallo en uno de los procesos no afecte al resto, garantizando así que el sistema continúe funcionando (PostgreSQL.ES, 2009).

PostgreSQL es una herramienta libre de gestión de bases de datos relacionales que destaca por su robustez, escalabilidad y cumplimiento de los estándares SQL, contando con versiones para una amplia gama de sistemas operativos, entre ellos: Linux, Windows, Mac OS X, Solaris, BSD, Tru64 entre otros.

Entre algunas de sus características más importantes está el hecho de poseer una alta concurrencia, de forma que el acceso a tablas que están modificándose por otros usuarios no tenga cortes. Posee soporte para una gran cantidad de tipos nativos, permite trabajar con claves foráneas y triggers. Además, PostgreSQL soporta ACID, o lo que es lo mismo, la

realización de transacciones seguras; también, vistas, uniones, procedimientos almacenados, etcétera. Incluye la mayor parte de los tipos de datos especificados en los estándares, como: entero, numérico, booleano, char, varchar, fecha, interval o timestamp (Uptdown).

Algunas de sus principales características son:

- **Alta concurrencia:** Es capaz de atender a muchos clientes al mismo tiempo y entregar la misma información de sus tablas, sin bloqueos.
- **Soporte para múltiples tipos de datos de manera nativa:** Ofrece los tipos de datos habituales en los sistemas gestores, pero además muchos otros que no están disponibles en otros competidores, como direcciones IP, direcciones MAC, Arrays, números decimales con precisión configurable, figuras geométricas, etc.
- **Soporte a triggers:** Permite definir eventos y generar acciones cuando estos se disparan.
- **Trabajo con vistas:** Esto quiere decir que pueden consultar los datos de manera diferente al modo en el que se almacenan.
- **Objeto-relacional:** Otra de sus principales características, que permite trabajar con sus datos como si fueran objetos y ofrece mecanismos de la orientación a objetos, como herencia de tablas.
- **Soporte para bases de datos distribuidas:** Donde el trabajo con transacciones asegura que estas tendrán éxito cuando han podido realizarse en todos los sistemas involucrados.
- **Soporte para gran cantidad de lenguajes:** PostgreSQL es capaz de trabajar con funciones internas, que se ejecutan en el servidor, escritas en diversos lenguajes como C, C++, Java, PHP, Python y Ruby. Además, ofrece interfaces para ODBC y JDBC, así como interfaces de programación para infinidad de lenguajes de programación (arsys, s.f.).

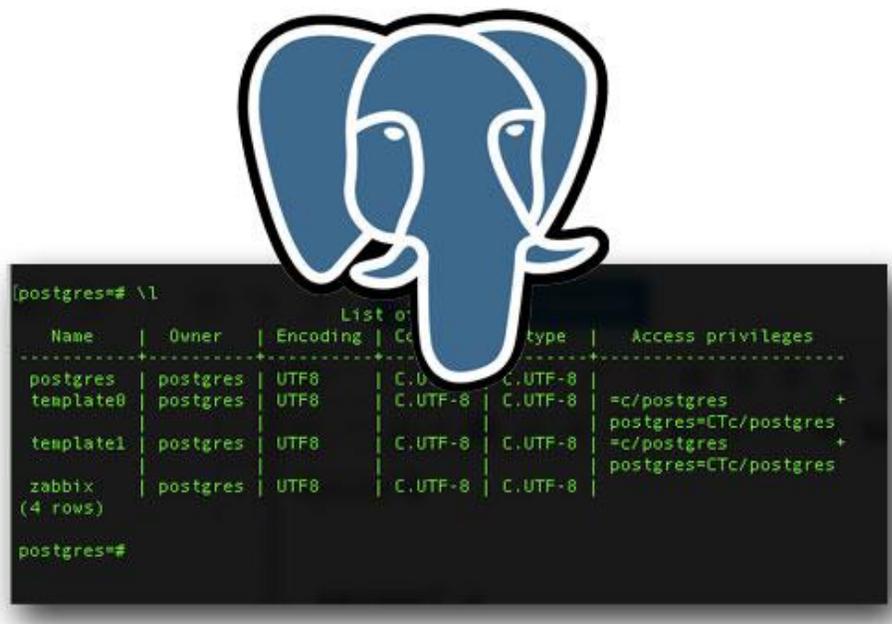


Ilustración 14. PostgreSQL, lenguaje de Programación de la Base de Datos (arsys. (s.f.). ¿Qué es PostgreSQL y por que llevarlo a Cloud? Obtenido de arsys.es:
<https://www.arsys.es/blog/soluciones/postgresql-servidores>

Conclusiones del capítulo

El análisis de la bibliografía y la investigación arrojó elementos suficientes para corroborar la necesidad de implementar una nueva versión del Sistema de Obtención y Catalogación de Contenidos de Internet que sea capaz de ajustarse a solucionar las deficiencias que presentan las aplicaciones analizadas para conformar un sitio web que se complemente en la realización del proceso de obtención y catalogación deseado. Mediante la utilización de la metodología Agile XP y el uso de las herramientas y tecnologías propuestas, se puede afirmar que se cuenta con los elementos necesarios y suficientes para presentar la propuesta de solución al problema de investigación planteado.

CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO

Luego de concluir el estado del arte se hace necesario esclarecer las características y funcionalidades del sistema a desarrollar. En el presente capítulo se realiza una descripción acerca de la estructura y funcionamiento del prototipo elaborado basado en la metodología seleccionada, XP, la cual se aplica de manera dinámica, ya que es capaz de adaptarse a los cambios de requisitos; los individuos e interacciones son más importantes, siendo la base de la metodología. La colaboración y comunicación constante con el cliente es importante. La metodología consta de 4 fases a través de las cuales se podrá lograr un mejor entendimiento de las características del sistema.

II.1 Modelado de la propuesta de solución II.1

Propuesta de Solución

Teniendo como objetivo, desarrollar el Sistema de Obtención y Catalogación de Contenidos de Internet en su versión 2.0 mediante el uso de tecnologías web. Se propone, desplegar un Sistema más completo que su versión actual, para lograr la obtención y catalogación de contenidos de internet, permitiendo al usuario disfrutar de una mejor experiencia mediante su interacción con esta nueva versión. Con esta implementación, el usuario que va a interactuar con dicha solución podrá descargar la información de su interés, gestionar la información según la categoría a la que pertenezca, gestionar la información según la fuente a la cual pertenezca, filtrar la información por categorías, fuente o fecha de descarga, entrenar las categorías manualmente mediante el uso de muestras positivas y negativas, y exportar el contenido buscado como PDF. Se pretende aportar utilidad y ahorrar el tiempo de respuesta, para agilizar el proceso de obtención y catalogación, por ello se deben incluir las características siguientes:

II.2 Requisitos, análisis y diseño de la aplicación web II.2

Los requisitos según la RAE son condiciones necesarias para algo; en la ingeniería de desarrollo de sistemas, un requisito es una necesidad documentada sobre el contenido, forma o funcionalidad de un producto o servicio, utilizándolos como datos de entrada a la etapa de desarrollo del producto, establecen que debe hacer el proyecto, pero no como debe hacerlo.

En este proyecto los requisitos son establecidos por el cliente mediante una conversación con el equipo de desarrollo del proyecto. Los requerimientos se dividen en dos tipos: Funcionales, lo que debe hacer el usuario directamente con el software, y No Funcionales, como se va a ver el software y las características que debe poseer el hardware para el funcionamiento del software.

El análisis de requisitos es una de las tareas más importantes en el ciclo de vida del desarrollo del software, es el proceso de estudio de las necesidades del usuario para llegar a una definición de los requisitos del sistema, hardware o software, así como el proceso de estudio y refinamiento de dichos requisitos.

Requisitos Funcionales

Los requisitos funcionales, como bien dice su nombre, son funciones que el sistema debe cumplir. Describe la interacción entre el sistema y su ambiente independientemente de su implementación. El ambiente incluye al usuario y cualquier otro sistema externo que interactúa con él.

Los requisitos definidos se listan a continuación:

Autenticación

RF 1. Autenticar usuario.

RF 2. Creación de Permiso.

RF 3. Gestionar categorías.

3.1 Adicionar categoría.

3.2 Modificar categoría.

3.3 Eliminar categoría.

3.4 Ver detalles de categoría.

3.5 Listar categorías.

RF 4. Entrenar categoría.

RF 5. Suscripción de categoría.

5.1 Suscribirse categoría.

5.2 Anular suscripción de categoría.

- RF 6. Cargar corpus de entrenamiento.
 - 6.1 Agregar muestras positivas de categoría.
 - 6.2 Agregar muestras negativas de categoría.
- RF 7. Gestionar muestras de categoría.
 - 7.1 Añadir muestra.
 - 7.2 Listar muestras.
 - 7.3 Modificar muestra.
 - 7.4 Eliminar muestra.
- RF 8. Filtrar categoría.
- RF 9. Gestionar fuente de información.
 - 9.1 Adicionar fuente.
 - 9.2 Modificar fuente.
 - 9.3 Eliminar fuente.
 - 9.4 Ver detalles de fuente.
 - 9.5 Listar fuentes.
- RF 10. Filtrar fuentes de información.
- RF 11. Listar contenidos descargados.
- RF 12. Filtrar contenido.
 - 12.1 Filtrar contenido por categoría.
 - 12.2 Filtrar contenido por fuente.
 - 12.3 Filtrar contenido por fecha de descarga.
- RF 13. Ver contenido.
 - 13.1 Ver detalles del contenido.
 - 13.2 Ver noticia original.
- RF 14. Adicionar contenido al corpus.
- RF 15. Eliminar contenido.
- RF 16. Exportar contenido como PDF.

Requisitos No Funcionales

Mediante los cuales se describen aspectos del sistema visibles por el usuario, no incluyen relación directa con el comportamiento funcional del sistema. Hacen que el producto resulte atractivo, usable, rápido o confiable. Incluyen restricciones como el tiempo de respuesta, la precisión, recursos consumidos, seguridad, etc.

Usabilidad:

- El sistema debe proporcionar un alto grado de conexión con el usuario, guiar su transcurso por la página web de manera clara y sencilla, haciendo uso de botones y señalizaciones que aclaren hacia donde debe ir.
- La comunicación de los formularios con el servidor debe permitir que cualquier acción de adicionar información, modificarla o eliminarla se ejecute al seleccionar el botón ENTER una vez se realice la acción de manera correcta.
- El sistema debe estar bien estructurado y cada método debe tener una descripción explicando su funcionalidad para un mejor entendimiento de otros desarrolladores.

Rendimiento:

- El sistema debe contar con un tiempo de respuestas en un marco razonable de 3 segundos.

Portabilidad:

- El sistema podrá ser utilizado en diferentes plataformas, dígase navegadores web.

Disponibilidad:

- El sistema debe garantizar a los usuarios autorizados el acceso a la información de manera permanente durante el horario laboral.

Seguridad:

- El sistema solo permite el acceso a las personas que cuentan con los permisos de autenticación necesarios para ello.
- Cada usuario tendrá acceso a la información dependiendo del rol que lo represente.

Software

- El sistema es compatible con Firefox y Chrome.

Hardware:

- Para el funcionamiento mínimo del sistema es necesario un Microprocesador Intel® Core™ i3-2330M CPU @ 2.2GHz, Memoria RAM 2 GB, Sistema Operativo de 64 bits, Disco Duro interno HDD 500MB.

Fase I: Fase de Planificación

Cumpliendo con la fase I de la metodología empleada, donde el cliente establece la prioridad de cada historia de usuario, y recíprocamente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas.

Story Cards (Historias de Usuario): “Las historias de usuario son más “amigables” que los casos de uso formales”. Técnica utilizada para especificar los requisitos, en la metodología empleada se utilizan las historias de usuario. Representando una breve descripción del comportamiento del sistema, proporcionando el detalle suficiente para hacer razonable la estimación de tiempo que requiere la implementación de la historia, empleando terminología del cliente sin la utilización del lenguaje técnico, se realiza una por cada característica principal del sistema, remplazan un gran documento de requisitos y presiden la creación de las pruebas de aceptación (Echeverry Tobón, L. M. (2007). CASO PRÁCTICO DE LA METODOLOGÍA ÁGIL XP AL DESARROLLO DE. Obtenido de <https://aalbertovargasc.files.wordpress.com/2011/07/proyecto-con-xp.pdf>). A continuación, serán representados los requisitos RF.1, RF.3, RF.7, RF. 11 y RF.12 para conocer las restantes historias de usuario diríjase a los anexos.

Iteración 1

Tabla 1. Historia de Usuario HU-1

Historia de Usuario	
Código: HU-1	Nombre: RF 1. Autenticar Usuario
Modificación de Historia de Usuario (Nro. Y Nombre):	
Usuario: Cliente	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1 semana
Descripción: Permitirá al usuario acceder a la información en dependencia del Rol que posea.	

Iteración 2

Tabla 2. Historia de Usuario HU-3

Historia de Usuario

Código: HU-3	Nombre: RF 3. Gestionar categorías
Modificación de Historia de Usuario (Nro. Y Nombre):	
Usuario: Cliente	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 4
Riesgo en Desarrollo: Alto	Puntos Reales: 4 semanas
Descripción: Permitirá al usuario ejecutar las acciones de adicionar, modificar, eliminar, ver detalles y listar; lo cual le proporcionará un control óptimo sobre las categorías en que se cataloga la información.	
Observaciones: El usuario debe estar autenticado para realizar dicha acción.	

Tabla 3. Historia de Usuario HU-7

Historia de Usuario	
Código: HU-7	Nombre: RF 7. Gestionar muestras de categoría
Modificación de Historia de Usuario (Nro. Y Nombre):	
Usuario: Cliente	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Alto	Puntos Reales: 2 semanas
Descripción: Brinda la posibilidad al usuario de añadir, listar, modificar y eliminar muestras.	
Observaciones: El usuario debe estar autenticado para realizar dicha acción.	

Iteración 3

Tabla 4. Historia de Usuario HU-9

Historia de Usuario	
Código: HU-9	Nombre: RF 9. Gestionar fuente de información
Modificación de Historia de Usuario (Nro. Y Nombre):	
Usuario: Cliente	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Alto	Puntos Reales: 3 semanas
Descripción: Brinda la posibilidad al usuario de añadir, listar, modificar y eliminar fuentes.	
Observaciones: El usuario debe estar autenticado para realizar dicha acción.	

Iteración 4

Tabla 5. Historia de Usuario HU-11

Historia de Usuario	
Código: HU-11	Nombre: RF 11. Listar contenidos descargados
Modificación de Historia de Usuario (Nro. Y Nombre):	
Usuario: Cliente	Iteración Asignada: 4
Prioridad en Negocio: Media	Puntos Estimados: 1
Riesgo en Desarrollo: Bajo	Puntos Reales: 1 semana
Descripción: Permitirá al usuario listar los contenidos descargados, lo cual le proporcionará un control óptimo sobre las descargas realizadas.	
Observaciones: 1. El usuario debe estar autenticado para realizar dicha acción. 2. Debe existir al menos una descarga realizada.	

Tabla 6. Historia de Usuario HU-12

Historia de Usuario	
Código: HU-12	Nombre: RF 12. Filtrar contenidos
Modificación de Historia de Usuario (Nro. Y Nombre):	
Usuario: Cliente	Iteración Asignada: 4
Prioridad en Negocio: Media	Puntos Estimados: 2
Riesgo en Desarrollo: Medio	Puntos Reales: 2 semanas
Descripción: Brinda la posibilidad al usuario de filtrar los contenidos agrupándolos por sus diferentes categorías, fuentes a la que pertenecen y la fecha en la cual fue descargada la información.	
Observaciones: El usuario debe estar autenticado para realizar dicha acción.	

Release Planning (Plan de Publicaciones): La programación bajo la metodología XP basa sus procesos de planificación en estimaciones temporales de las historias de usuario, realizadas por los desarrolladores durante las diversas reuniones de planificación. Una planificación es donde los desarrolladores y clientes establecen los tiempos de implementación ideales de cada historia de usuario que será implementada y la prioridad con la que serán desarrolladas en cada versión del programa. El equipo de desarrollo mantiene un registro de velocidad de desarrollo, establecida en puntos por iteraciones. Las estimaciones de esfuerzo

asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto de estimación, donde un punto de estimación equivale a una semana ideal de programación (Bustamante, D., & Rodríguez, J. (2014). *Metodología Actual Metodología XP*. Barinas.).

Tabla 7. Estimación del tiempo de las HU.

Nombre de la Historia de Usuario	Punto de Estimación
HU-1. Autenticar usuario.	0.5
HU-2. Creación de Permiso	0.5
HU-3. Gestionar categorías.	4
HU-4. Entrenar categoría.	1
HU-5. Suscripción de categoría.	1
HU-6. Cargar corpus de entrenamiento.	1
HU-7. Gestionar muestras de categoría.	2
HU-8. Filtrar categoría.	1
HU-9. Gestionar fuente de información.	3
HU-10. Filtrar fuentes de información.	1
HU-11. Listar contenidos descargados.	1
HU-12. Filtrar contenido.	2
HU-13. Ver contenido.	1
HU-14. Adicionar contenido al corpus.	1
HU-15. Eliminar contenido.	1
HU-16. Exportar contenido como PDF.	1

Las estimaciones realizadas permitieron confeccionar una evaluación puntual del tiempo de implementación de cada historia de usuario para la posterior elaboración del plan de iteración. Es preciso construir un plan de iteración donde se pueden agrupar las historias y dar su cumplimiento paulatinamente. Con la realización del plan de publicaciones queda concluyente la secuencia en el proceso de desarrollo de solución para las distintas actividades de la propuesta de solución.

II.3 Diseño e implementación del almacenamiento, procesamiento y transmisión de los datos en la aplicación web II.3

La arquitectura del software de un programa o sistema de cómputo es la estructura o estructuras del sistema, lo que comprende a los componentes del software, sus propiedades externas visibles y las relaciones entre ellos (Pressman, 2013).

Fase II: Etapa de diseño

En esta Fase se realiza el plan de iteraciones, se confeccionan las tarjetas Clase-Responsabilidad-Colaboración (CRC) para la descripción de las principales clases de los módulos desarrollados. Se define la arquitectura del sistema y los estándares de codificación, así como los patrones de diseño utilizados en el desarrollo de la propuesta.

Plan de Iteraciones

Una iteración es una parte esencial del proyecto ya que al culminar el cliente obtiene un resultado parcial. Está constituido por un conjunto secuencial de actividades y tareas, cada una posee recursos asignados y pueden depender a su vez de otras tareas, realizándose solo una vez por iteración. Los planes de iteración muestran, por lo general, un planteamiento detallado de quién va a realizar una tarea o actividad de acuerdo en conformidad a criterios de evaluación. Al concluir la última iteración, el cliente quedará completamente satisfecho, pues culmina la producción del proyecto (Bustamante, D., & Rodríguez, J. (2014). *Metodología Actual Metodología XP*. Barinas.). En la siguiente tabla se pueden observar las iteraciones que se determinaron necesarias para el desarrollo de la solución:

Tabla 8. Plan de Iteraciones

Iteración	Nombre de la Historia de Usuario	Nro. HU	Duración (Puntos)	
Iteración No.1	RF.1 Autenticar usuario.	HU-1	0.5	1
	RF.2 Creación de Permiso	HU-2	0.5	
	RF.3 Gestionar categorías.	HU-3	2	
	RF.4 Entrenar categoría.	HU-4	1	

Iteración No.2	RF.5 Subscripción de categoría.	HU-5	1	7
	RF.6 Cargar corpus de entrenamiento.	HU-6	1	
	RF.7 Gestionar muestras de categoría.	HU-7	1	
	RF.8 Filtrar categoría.	HU-8	1	
Iteración No.3	RF.9 Gestionar fuente de información.	HU-9	3	4
	RF.10 Filtrar fuentes de información.	HU-10	1	
Iteración No.4	RF.11 Listar contenidos descargados.	HU-11	1	7
	RF.12 Filtrar contenido.	HU-12	2	
	RF.13 Ver contenido.	HU-13	1	
	RF.14 Adicionar contenido al corpus.	HU-14	1	
	RF.15 Eliminar contenido.	HU-15	1	
	RF.16 Exportar contenido como PDF.	HU-16	1	

Realizado el plan de iteraciones, se pudo agrupar las diferentes historias de usuario en 4 iteraciones teniendo en cuenta las características que rigen la metodología XP. Con este plan culminado ya es posible realizar un plan de entrega que será entregado al cliente y que el grupo de desarrollo está obligado a hacer cumplir.

Plan de Entrega

El Plan de Entrega es el compromiso final del equipo de desarrollo con el cliente, pues en él se define cuando será entregado el producto. Representa un factor importante para el proyecto, pues la demora en la entrega del proyecto trae consigo insatisfacciones por parte del cliente (Pressman, 2013). Determinada la duración de cada iteración se presenta el plan de entrega elaborado para la fase de implementación teniendo en cuenta que el desarrollo del

proyecto inicia el 10 de junio del 2022 y culminará el 24 de octubre del 2022 como se muestra en la siguiente tabla:

Tabla 9. Plan de Entrega

	Iteración No.1	Iteración No.2	Iteración No.3	Iteración No.4
Cantidad HU	1	6	2	6
Fecha de inicio	10/05/22	18/05/22	07/07/22	05/08/22
Fecha de entrega	17/05/22	06/07/22	04/08/22	23/09/22

Con el plan de entrega listo se puede confirmar con el cliente la duración del proyecto con un total de 19 semanas, lo cual significa que tendrá un tiempo estimado de 4 meses, 1 semana y 6 días aproximadamente.

Tarjetas CRC

Se elaboran para describir las entidades existentes en la aplicación. Su uso es una técnica de modelado que se divide en tres secciones donde se contiene la información que permite identificar las clases, responsabilidades y colaboradores. EL objetivo es obtener un diseño simple, elegante y fácil de comprender por parte de los programadores. Su objetivo principal es dejar el pensamiento procedimental para enfocarse al orientado a objetos. Cada tarjeta representa una clase con su nombre en la parte superior, en la sección inferior izquierda se especifican las responsabilidades y en la derecha los colaboradores o clases de soporte. Una clase es cualquier persona, cosa, evento, concepto, pantalla o reporte. Las responsabilidades de una clase son las cosas que conoce y las que realizan, sus atributos y métodos. Los colaboradores de una clase son las demás clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades (Bustamante, D., & Rodríguez, J. (2014). *Metodología Actual Metodología XP*. Barinas.). En la siguiente tabla se muestra la distribución de dicha información.

Tabla 10. Tarjeta CRC 1

Categorías	
Responsabilidades	Colaboradores

Crear Categoría Modificar Categoría Eliminar Categoría Listar Categoría Entrenar Categoría Filtrar categoría	Categorías Descargas
---	-------------------------

Tabla 11. Tarjeta CRC 2

Clase: Fuentes	
Responsabilidades	Colaboradores
Crear Fuente Modificar Fuente Eliminar Fuente Listar Fuente Filtrar Fuente	Descargas Fuentes

Tabla 12. Tarjeta CRC 3

Clase: Contenidos	
Responsabilidades	Colaboradores
Listar Contenidos Filtrar Contenidos Adicionar Contenido Ver Contenido Eliminar Contenido Exportar contenido como PDF	Descargas Contenidos

Con la realización de las tarjetas CRC se evidencia la interrelación existente entre cada clase de la solución y sus diferentes funcionalidades.

Reuniones diarias de seguimiento

El planeamiento es esencial independientemente de la metodología a emplear. XP requiere de una revisión continua del plan de trabajo, a pesar de ser una Metodología que evita el uso de documentación exagerada, es muy estricta en la organización del trabajo. Es por ello que el

grupo de desarrollo definió desde el inicio del proyecto espacios en los cuales los Stakeholders realizan encuentros semanales para la evaluación del cumplimiento de los resultados de las diferentes iteraciones, así como el cumplimiento de los resultados en las diferentes iteraciones, ya sea parcial, total o nuevas redefiniciones a los conceptos establecidos, facilitando el trabajo de los desarrolladores y posibilitando el cumplimiento de las necesidades del cliente (Collazo García, A., & Díaz Labrador, M. (2013). *La Programación Extrema*. La Habana: Universidad de las Ciencias Informáticas.).

Tabla 13. Reuniones de seguimiento

Reunión de:	Fecha	Descripción
Plan de entregas	05/05/22	Se realiza entre el equipo de trabajo y el cliente para definir el marco temporal de la realización del sistema. El cliente expone las historias de usuario al equipo en cuestión, los cuales estimarán el grado de dificultad de la implementación de cada historia. A partir de las historias de usuario, el cliente plantea las pruebas de aceptación con las cuales se comprueba que cada una de estas ha sido correctamente implementada.
Inicial de iteración	11/05/22 19/05/22 08/07/22 05/08/22	Dicha reunión es realizada previamente al inicio de una iteración donde son organizadas las actividades de programación a realizar. Las HU son traducidas a tareas y asignadas a los desarrolladores.
Diarias	Al iniciar la jornada de trabajo	Son realizadas al iniciar la jornada laboral, donde todos los miembros del equipo de desarrollo se reúnen para exponer los problemas e ideas que se estén presentando. Es de vital importancia evitar las discusiones largas, ya que se está utilizando tiempo laboral que puede ser destinado a la construcción del sistema.
Fin de iteración	18/05/22 06/07/22 05/08/22 26/09/22	Estos encuentros se realizan al finalizar cada iteración en conjunto con los clientes para presentar los avances en cada una de ella y demostrar la aceptación ante estos resultados como muestra de una correcta implementación.

Teniendo en cuenta la interrelación de las entidades, el grupo de desarrollo está en condiciones de confeccionar una arquitectura del sistema acorde con las exigencias del cliente.

Arquitectura del sistema

La arquitectura de software, como concepto, presenta un gran número de definiciones por parte de reconocidos autores. El patrón arquitectónico Modelo-Vista-Controlador (MVC) es un patrón de arquitectura de software encargado de separar la lógica de negocio de la interfaz de usuario y es el más utilizado en aplicaciones web. Facilita la funcionalidad, estabilidad y escalabilidad del sistema, de forma simple y sencilla, a la vez que permite no mezclar lenguajes de programación en el mismo código (Pressman, R. S. (2002). Ingeniería de software 5ta Edición. McGrawHill.).

MVC: Modelo que expresa como organizar y estructurar los componentes de un sistema de software, sus responsabilidades y las relaciones existentes entre ellos. Propone la separación de los componentes del sitio web o aplicaciones en tres niveles de abstracción o capas principales: modelo, vista y controlador, describiendo la relación entre ellas para mantener una estructura con acoplamiento mínimo entre las capas (Aguilar, 2019).

Patrones de Arquitectura MVC

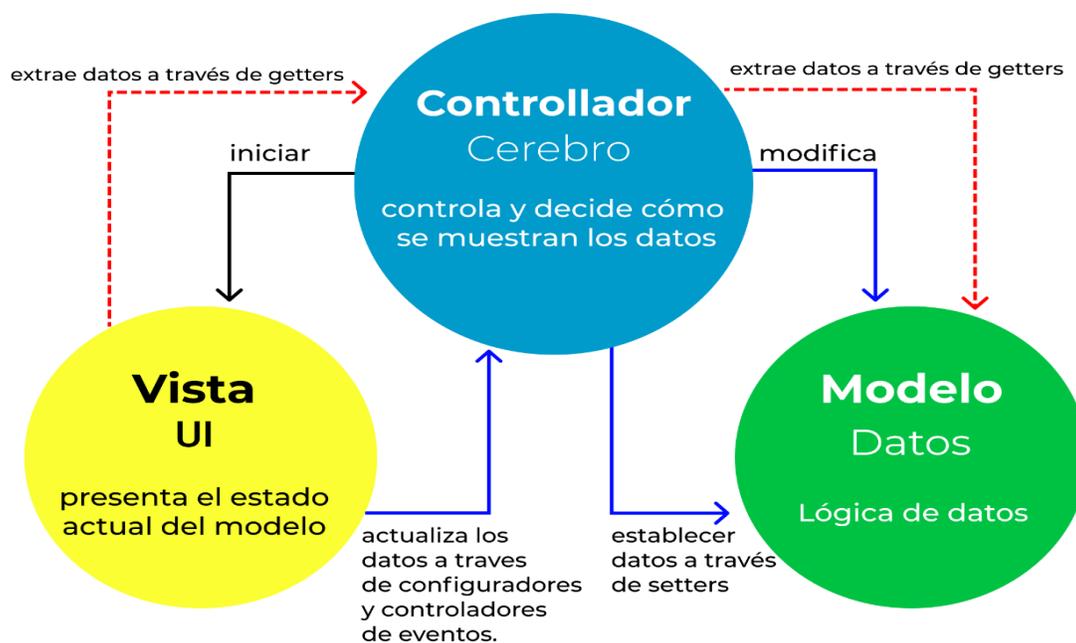


Ilustración 15. Patrones de Arquitectura Modelo Vista Controlador (Aguilar, 2019)

Modelo: La capa de modelo contiene las entidades que representan el dominio, la lógica de negocio y los mecanismos de persistencia del sistema. Se encarga de acceder de forma directa a los datos actuando como intermediario con la base de datos. En ella se encuentra el paquete Monitoreo y un paquete de acceso a datos.

Vista: Es la encargada de mostrar la información al usuario de forma gráfica. En esta clase se interpreta el modelo, se solicita actualizaciones del modelo, envía las acciones del usuario al controlador y permite al controlador seleccionar las vistas. Posee paquetes CSS, JavaScript, HTML y Bootstrap.

Controlador: Intermediario entre el modelo y la vista. Controla las interacciones del usuario, solicitando los datos al modelo y entregándolos a la vista para ser presentados al usuario mediante la misma. En esta clase se define el comportamiento de la aplicación, mapea las acciones del usuario a actualizaciones del modelo, selecciona la vista de respuesta y es una por cada funcionalidad. Contiene un paquete con la clase módulo de seguridad en el componente Controlador.

Realizada la arquitectura del sistema, se define la estructura modular del proyecto, así como la estructura arquitectónica de la solución como guía en el proceso de desarrollo. Definir un grupo de patrones es vital para adoptar buenas prácticas y cumplir con estándares que facilitan la legibilidad del código y el funcionamiento de la aplicación.

Prototipos de Interfaz de Usuario

La creación de los prototipos de interfaz de usuario (UI) es un proceso iterativo, que pasa por un ciclo donde se prueban las características de diseño y funcionalidad, mismas que deben cumplir con criterios bien definidos hasta llegar a los resultados deseados. Los prototipos de UI ayudan a ordenar las ideas, explorar diferentes caminos de conceptos o diseño y detectar posibles problemas o carencias antes de empezar la fase de programación de un proyecto digital. A través de ellos se puede probar, analizar, ajustar y perfeccionar los elementos con los que van a interactuar los usuarios con la aplicación web. Ayudan a tener certeza de los lenguajes de programación que se necesitan para desarrollar una página web, de modo que

se puede ahorrar tiempo y recursos valiosos ya que resultan más baratos que desarrollar un sistema real. Forzosamente pasará por cambios y refinamientos en varias ocasiones. A continuación, se pueden observar algunos de los prototipos UI de la aplicación web a implementar:

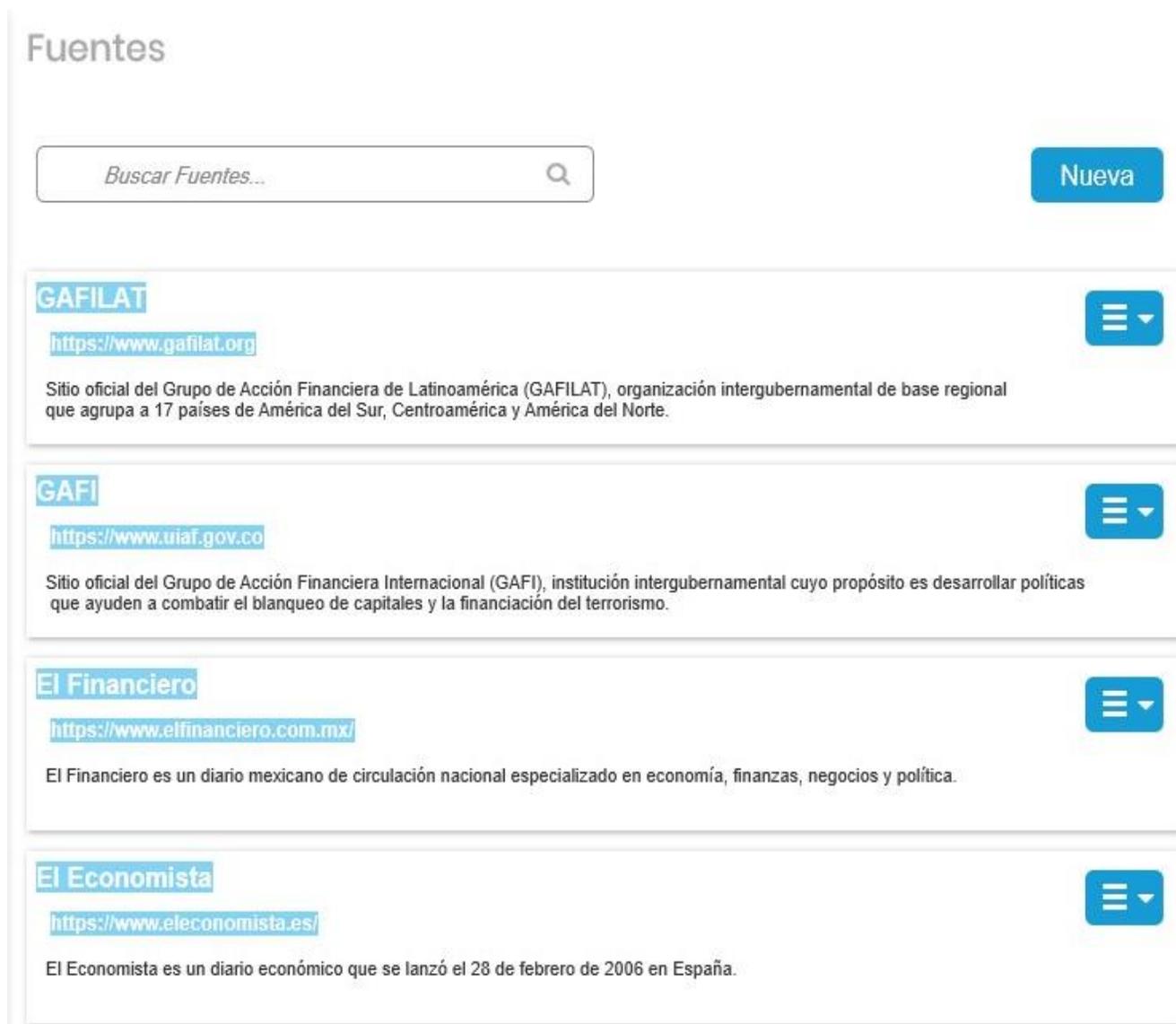


Ilustración 16. Vista Principal de Fuentes

Descargas

Filtrado avanzado

Resultados Plenario del GAFI, 14-17 de junio de 2022

Fecha: 14/06/2022

Fuente: www.gafilat.org Categoría: [Categoría 2](#)

Entre los días 11 y 17 de junio de 2022 el GAFILAT participó de las reuniones del Grupo de Evaluación y Cumplimiento (ECG) y Plenario del GAFI, realizado en Berlín, Alemania. En la misión participaron presencialmente el Dr. Esteban Fullin -quien ejerce el rol de Co-Coordinador...

COMUNICADO GAFILAT

Fecha: 31/05/2022

Fuente: www.gafilat.org Categoría: [Categoría 1](#)

El Grupo de Acción Financiera de Latinoamérica –GAFILAT– condena enérgicamente los recientes actos de violencia acontecidos en países de nuestra región, en contra de funcionarios públicos por el solo hecho de realizar su trabajo con dedicación, responsabilidad y...

GAFILAT inaugura sus nuevas oficinas

Fecha: 28/05/2022

Fuente: www.gafilat.org Categoría: [Categoría 3](#)

La apertura de la nueva sede contó con la visita de la Presidenta Pro Tempore Carla Mera, el Vicepresidente Jorge Chediak, los Coordinadores Nacionales ante el organismo de los países miembros como Argentina, Bolivia, Brasil, Colombia, Cuba, El Salvador...

GAFILAT participó de la reunión ministerial del GAFI

Fecha: 27/05/2022

Fuente: www.gafilat.org Categoría: [Categoría 5](#)

Los ministros miembros del Grupo de Acción Financiera (GAFI), durante esta sesión bienal, se comprometieron a tomar medidas rápidas y decisivas para mejorar la eficacia de la lucha contra el blanqueo de capitales y la financiación del terrorismo y la proliferación...

Conferencia Regional UNOCT-UNCCT-GAFILAT

Fecha: 25/05/2022

Fuente: www.gafilat.org Categoría: [Categoría 2](#)

Se llevó a cabo la Conferencia Regional UNOCT-UNCCT-GAFILAT sobre la implementación de Sanciones Financieras Dirigidas y el fortalecimiento de la Cooperación Internacional y la Coordinación Nacional para Combatir el Financiamiento del Terrorismo en América...

Ilustración 17. Vista Principal de Descargas

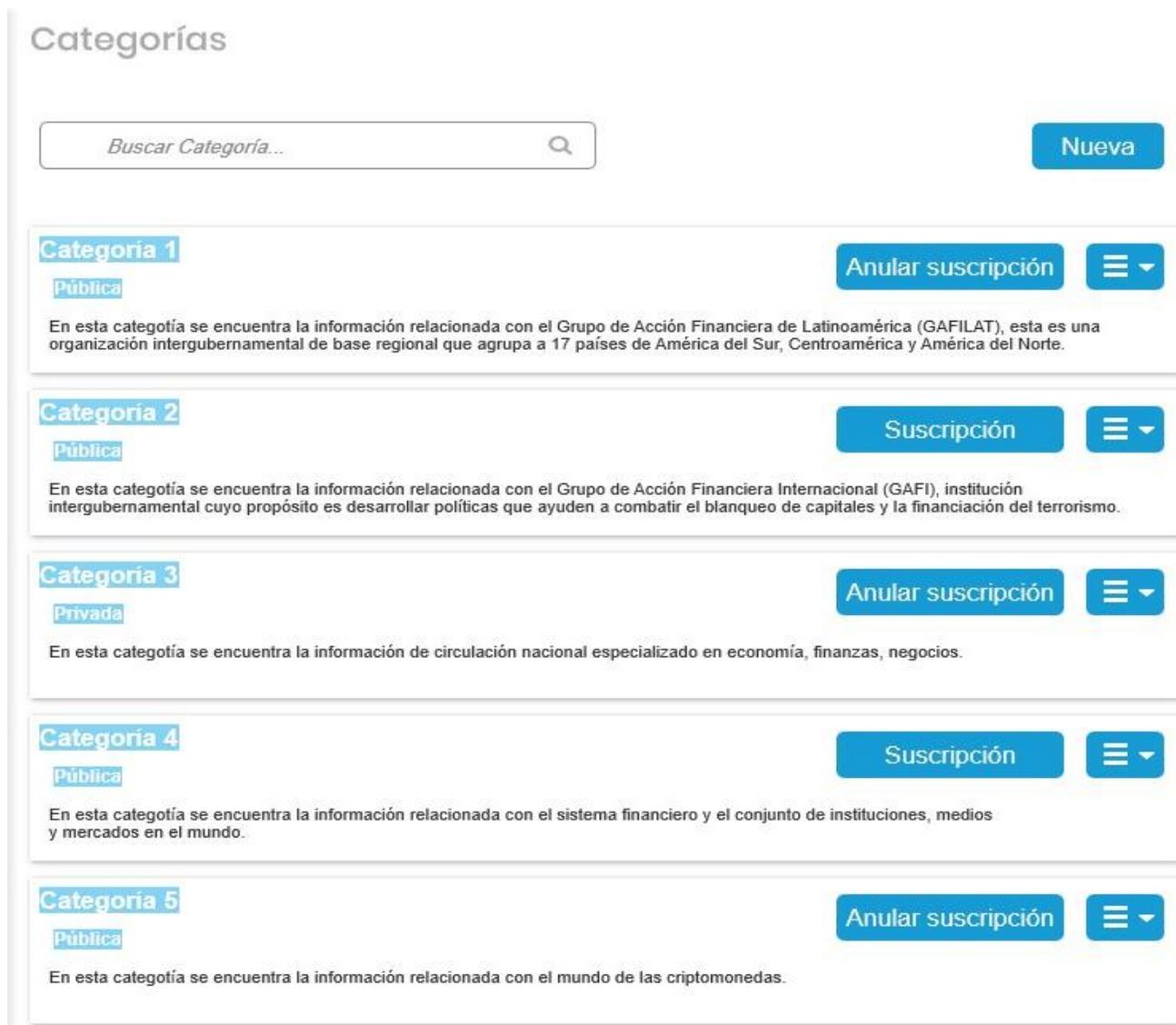


Ilustración 18. Vista Principal de Categorías

Fase III: Fase de implementación

El desarrollo o codificación es un proceso que se realiza en forma paralela con el diseño y la cual está sujeta a varias observaciones por parte de XP. Dentro de esta fase se describen los estándares de codificación, así como los patrones de diseño que se emplearon dentro del sistema.

Patrones de diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular. Comunican los estilos

y soluciones consideradas como buenas prácticas, utilizadas para la creación de sistemas en el diseño orientado a objetos más flexible, elegante y en algunos casos reusables. Ayudan a capturar conocimiento y crear un vocabulario técnico (Ingeniería de Software II, 2011). El objetivo principal de los patrones es facilitar la reutilización de diseños y arquitecturas de software que han sido exitosas, capturando la experiencia y haciéndola accesible a los no expertos. A continuación, se describen los patrones utilizados para el desarrollo de la propuesta de solución:

Patrones GRASP

General Responsibility Assignment Software Patterns (GRASP), son una serie de buenas prácticas de aplicación recomendables en el diseño de software, principios generales para asignar responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objetos. Un patrón es una descripción de un problema y la solución a la que se le da un nombre, y que además se puede aplicar a nuevos contextos. De los patrones existentes, los más empleados son: Experto, Creador, Alta Cohesión, Bajo Acoplamiento y Controlador (Ingeniería de Software II, 2011). A continuación, serán explicadas las que se emplean en el desarrollo del sitio web actual:

Experto: Este patrón es utilizado para dar solución a la siguiente interrogante+: ¿A quién se le debe asignar una responsabilidad? Es el encargado de asignar responsabilidades a una clase, asignar la responsabilidad al experto en la información. El experto en la información es la clase que tiene la información necesaria para cumplir la responsabilidad. Expresa que los objetos deben hacer las cosas relacionadas con la información que poseen. Se observa en la clase Monitoreo.

Creador: Ayuda a identificar quién debe ser el responsable de la creación de nuevos objetos o clases. Contiene la información necesaria para la creación del objeto o utiliza directamente las instancias creadas, almacena o maneja varias instancias de la clase. Proporciona facilidad de mantenimiento, bajo acoplamiento y reutilización. El uso de este patrón se muestra en las clases Descargas, Fuentes y Categorías, responsables de crear una nueva instancia de una entidad y asignarle los atributos.

Alta Cohesión: Busca responder la interrogante: ¿Cómo mantener la complejidad manejable? La información que almacena una clase debe ser coherente y estar relacionada con esta. Poseen la estructura necesaria para el almacenamiento de la información, coherente y de acuerdo a la responsabilidad que tenga asignada cada clase. Su uso se observa en las clases Descargas, Fuentes y Categorías.

Controlador: Responde la interrogante: ¿Quién debe manejar los eventos del sistema? Patrón que sirve como intermediario entre una interfaz y el algoritmo que la implementa, recibe los datos del usuario y los envía a las distintas clases según el método que es llamado. Sugiere que la lógica del negocio esté separada de la capa de presentación, para aumentar la reutilización de códigos y tener un mayor control de estos. Recomienda dividir los elementos del sistema en el mayor número de controladores para aumentar la cohesión y disminuir el acoplamiento. En este caso sería la clase Monitoreo.

Patrones GOF:

The Gang of Four (GOF), constituyen una herramienta fundamental para cualquier programador. Se utilizan para solucionar problemas de creación de instancia, ayudando a encapsular y abstraer dicha creación. Estos patrones fueron estudiados por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides en su mítico libro Design Patterns identificando tres tipos de patrones GOF.

Patrones de creación: Tratan de la inicialización y configuración de clases y objetos.

. **Prototype:** Especifica los tipos de objetos a crear por medio de una instancia prototípica, y crear nuevos objetos copiando este prototipo.

. **Singleton:** Garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella.

Patrones estructurales: Tratan de desacoplar interfaz e implementación de clases y objetos.

. Adapter: Convierte la interfaz de una clase en otra distinta que es la que esperan los clientes. Permiten que cooperen clases que de otra manera no podrían por tener interfaces incompatibles.

. Composite: Combina los objetos en estructura de árbol para representar jerarquías de parte-todo. Permite que los clientes traten de manera uniforme a los objetos individuales y a los compuestos.

. Decorator: Busca añadir responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la subclasificación o especialización mediante herencia, cuando se trata de añadir funcionalidades.

. Facade: Proporciona una interfaz unificada para un subsistema. Define una interfaz de alto nivel que hace que el subsistema sea más fácil de utilizar.

Patrones de comportamiento: Tratan de las interacciones dinámicas entre sociedades de clases y objetos.

. Memento: Representa y externaliza el estado interno de un objeto sin violar la encapsulación, de forma tal que este pueda volver a dicho estado más tarde.

. Observer: Define una dependencia de uno a muchos entre objetos, de forma que cuando un objeto cambia de estado, se notifican y actualizan automáticamente todos los objetos.

. Template Method: Define en una operación el esqueleto de un algoritmo, delegando en las subclases algunos de sus pasos. Permite que las subclases redefinan ciertos pasos del algoritmo sin cambiar su estructura.

. Visitor: Representa una operación sobre los elementos de una estructura de objetos. Permite definir una nueva operación sin cambiar las clases de los elementos sobre los que opera.

Gestión de Datos

La gestión de datos, es el proceso de ingerir, almacenar, organizar, utilizar y mantener los datos creados y recopilados por una organización. La gestión de datos eficaz es una pieza

fundamental de la implementación de los sistemas de TI que ejecutan aplicaciones comerciales y proporcionan información analítica para ayudar a impulsar la toma de decisiones operativas y la planificación estratégica por parte de ejecutivos corporativos, gerentes comerciales y otros usuarios finales.

Las bases de datos son la plataforma más común utilizada para almacenar datos corporativos; contienen una colección de datos que está organizada para que se pueda acceder a ella, actualizarla y administrarla. Se utilizan tanto en sistemas de procesamiento de transacciones que crean datos operativos, como registros de clientes y pedidos de ventas, como en almacenes de datos, que almacenan conjuntos de datos consolidados de sistemas comerciales para BI y análisis (<https://www.computerweekly.com/es/definicion/Gestion-de-datos>).

En cualquier base de datos actual existen cuatro elementos esenciales: tablas, consultas, informes y formularios. Todos son indispensables y necesarios:

- Las tablas permitirán almacenar los datos.
- Con las consultas se podrá acceder a los datos almacenados, ordenados y filtrados por diferentes criterios.
- Los informes serán la forma más eficaz de presentar los datos.
- Con los formularios se facilitan las tareas de introducción de datos.

Modelo de Datos

El modelado de datos es la manera de estructurar y organizar los datos en función de ser utilizados fácilmente por la base de datos. Aporta la base conceptual para diseñar las aplicaciones que hacen uso de datos, la base formal para las herramientas y técnicas empleadas en el desarrollo y uso de sistemas de información (ISSN: 1139-8736). Define, cómo se modela la estructura lógica de una base de datos, cómo los datos se conectan entre sí y cómo se procesan y almacenan dentro del sistema. A través de ello se muestra el formato y definición de los datos involucrados, evitando también la redundancia de los mismos. Se muestran todas las estructuras de las tablas, el nombre de las columnas, el tipo de datos, las restricciones, la clave principal, la clave externa y las relaciones entre las tablas.

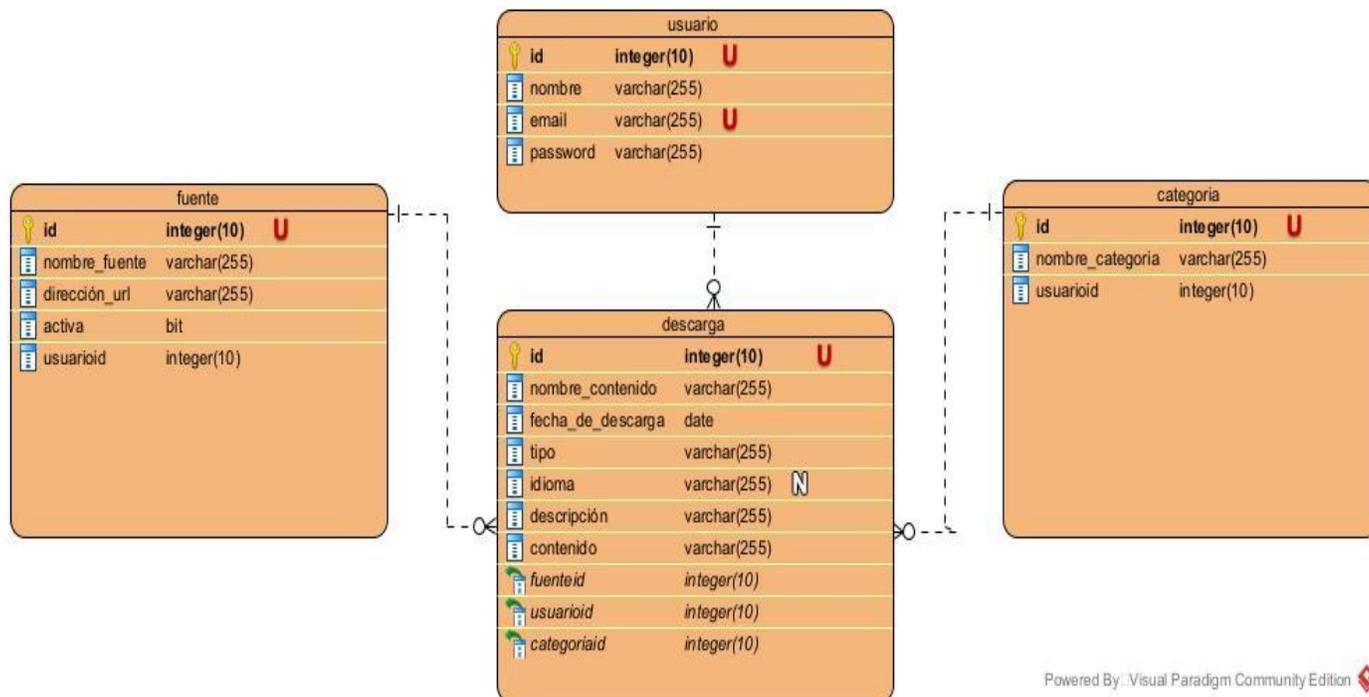


Ilustración 19. Diagrama Entidad-Relación

El diagrama Entidad-Relación muestra el conjunto de tablas que conforman la base de datos. Se observan las tablas usuario, la cual contiene los datos de cada usuario, dígame nombre, email y contraseña, dicha tabla mantiene relación de uno a muchos mediante el id_usuario con la tabla descarga, que contiene la información recopilada de cada archivo, tal como, nombre del contenido, fecha de descarga, tipo: si es texto o imagen, idioma: que se puede o no definir, descripción, contenido que trata. A su vez, la tabla descarga mantiene relación de uno a muchos mediante el id_categoria con la tabla categoría, la cual muestra el nombre de cada categoría, y mediante el id_fuente con la tabla fuente, la cual muestra el nombre de la fuente, la dirección_url y activa: para activar o desactivar la búsqueda en ella sin tener que eliminarla del sistema.

Los datos son ingresados y modificados en la base de datos por los usuarios que cuentan con el permiso para gestionar la información almacenada en la misma a través de la interfaz de usuario que brinda el sitio web con este propósito. Dígame la gestión, entrenamiento, suscripción y filtrado de las categorías, gestión de las muestras y fuentes de información, filtrado de fuentes de información y contenidos según la categoría, fuente y fecha de descarga, eliminar contenido y exportar contenido como PDF.

Conclusiones del capítulo

Con la culminación del presente capítulo, apoyándose en lo recopilado por cada fase de la metodología, se han definido las bases para el desarrollo del sistema. Como parte del desarrollo del sistema, la etapa de diseño constituye la pieza fundamental para llevar a cabo su implementación. La generación de los artefactos en la etapa de implementación de la plataforma, la reducción de errores y riesgos. A partir del análisis de la propuesta de solución y la comprensión de los requisitos funcionales se elaboraron 16 Historias de Usuario, así como se definió el plan de iteraciones a tener en cuenta para cada entrega de la solución. Además, se confeccionaron las tarjetas CRC lo que permitió detallar cada funcionalidad de las clases a utilizar.

CAPÍTULO III: VALIDACIÓN DEL SISTEMA DE OBTENCIÓN Y CATALOGACIÓN DE CONTENIDOS DE INTERNET

En el presente capítulo se aborda la implementación y los flujos de pruebas realizadas al sistema exigidas por la metodología XP para la certificación del correcto funcionamiento y la calidad de la propuesta desarrollada. Se especifican las tareas de ingeniería, la estrategia de pruebas a seguir y finaliza con la definición del modelo de pruebas unitarias y de aceptación, donde se refleja el resultado de la ejecución de dichas pruebas.

III.1 Verificación y validación del Sistema de Obtención y Catalogación de Contenidos de Internet III.1

Además del ciclo de vida común de XP, se puede encontrar otro ciclo de vida dentro de este que indica como trabajan los programadores para poder codificar.

1. Elección de Pares:

- Toda la producción se realiza en pares.
- El encargado de escribir piensa tácticamente, su pareja piensa estratégicamente.
- Se rotan los roles periódicamente.

2. Pruebas:

- Se escriben las pruebas unitarias.
- Se verifica que las pruebas fallen antes de codificar.
- Se testea todo lo que posiblemente puede fallar.

3. Codificación:

- Hacer algo que funcione de la manera más sencilla.
- Implementar lo suficiente para hacer que la prueba pase.
- Seguir el estándar de codificación.

4. Refactorizar:

- Eliminar toda porción de código que no parezca estar bien y luego verificar si el código aún pasa las pruebas unitarias.
- El código debe ser claro, no tener partes duplicadas y tener la menor cantidad de clases y métodos posibles.

- Realizar cambios pequeños y luego realizar las pruebas unitarias.

Task Card (Tareas de Ingeniería): Usadas para describir actividades que se realizan en el proyecto para cada historia de usuario. Una historia de usuario se descompone en varias tareas de ingeniería, así mismo las tareas de ingeniería se vinculan más al desarrollo, ya que permite tener un acercamiento con el código. Basado en las historias de usuario definidas para el desarrollo del sistema web, se han elaborado las tareas de ingeniería según el plan de iteraciones definido en el capítulo anterior (Bustamante, D., & Rodríguez, J. (2014). *Metodología Actual Metodología XP*. Barinas.

Las siguientes tablas muestran un resumen de las historias de usuarios definidas anteriormente y las tareas de ingeniería desarrolladas por cada una de las historias de usuario en cada iteración:

Iteración 1

Para la siguiente iteración fueron desarrollados los módulos de Sesión Administración, para lograr su progreso se han utilizado y aplicado cada una de las herramientas que se destacan en la metodología de programación extrema XP.

En la siguiente tabla se muestran de forma general las historias de usuario de esta etapa:

Historias de Usuario

Tabla 14. Historias de usuario iteración 1

Número	Nombre
1	Acceso al Sistema
2	Creación de Permiso

Tareas de Ingeniería

En la siguiente tabla se muestran de manera general las correspondientes tareas de ingeniería y a continuación de la misma se detallan cada una de ellas.

Tabla 15. Tareas de Ingeniería

Número de Tarea	Número de Historias	Nombre de la Tarea
1	1	Diseño de Interfaz de Acceso al Sistema
2	1	Validación de Usuarios en la Base de Datos
3	1	Adaptación a la Base de Datos para los Usuarios
4	2	Diseño de Interfaz de Usuario para la Creación de Permisos
5	2	Diseño de la Base de Datos de Creación de Permisos
6	2	Grabar Creación de Permiso en la Base de datos

Tarea de Ingeniería

Tabla 16. Tarea de Ingeniería 1 para historia de usuario 1

Tarea de Ingeniería	
Número de Tarea: 1	Nombre: HU 1. Acceso al sistema
Nombre de Tarea: Diseño de Interfaz de Acceso al Sistema	
Tipo de Tarea: Revisión	Puntos Estimados: 0.2
Fecha de inicio: 17/05/22	Fecha de fin: 18/05/22
Programador Responsable: Lesly Meliza Ferrer Aguado	
Descripción: Permitirá al usuario tener acceso a la aplicación en la sección de Acceso al Sistema, donde podrán introducir nombre, email y password.	

Como resultado de la entrega de la primera iteración del sistema de obtención y catalogación de contenidos de internet, el cliente quedó satisfecho con las funcionalidades de los módulos que se desarrollan, pero solicitó la realización de cambios, lo cual será una prioridad fundamental en la siguiente iteración.

Iteración 2

Para la siguiente iteración fueron desarrollados los módulos de Categoría, los cuales para lograr su progreso se ha utilizado y aplicado cada una de las herramientas que se destacan en la metodología de programación extrema XP.

Historias de Usuario

Tabla 17. Historias de usuario segunda iteración

Número	Nombre
3	Gestionar Categoría
4	Entrenar Categoría
5	Subscripción de Categoría
6	Cargar corpus de entrenamiento
7	Gestionar muestras de categoría
8	Filtrar categoría

Tareas de Ingeniería

Tabla 18. Tareas de ingeniería de la segunda iteración

Número de Tarea	Número de Historias	Nombre de la Tarea
1	3	Diseñar la interfaz de Categoría
2	3	Verificar integración de las categorías con la base de datos
3	3	Validar campos de categorías
4	3	Diseñar la interfaz de detalles de categoría
5	3	Validar datos de formulario de detalles
6	4	Diseñar la interfaz de entrenamiento
7	4	Validar campos de entrenamiento
8	4	Guardar en la Base de Datos los archivos de entrenamiento
9	5	Diseñar botón de subscripción
10	5	Validar funcionamiento del botón de subscripción
11	6	Diseñar la interfaz de cargar corpus

12	6	Agregar a la base de datos las muestras positivas
13	6	Agregar a la base de datos las muestras negativas
14	7	Diseñar la interfaz de muestras de categoría
15	7	Verificar integración de las muestras con la base de datos
16	8	Diseñar botón de filtrado
17	8	Validación del campo filtrado

Tabla 19. Tarea de Ingeniería 2 para la iteración actual

Tarea de Ingeniería	
Número de Tarea: 2	Nombre: HU 3. Gestionar categorías
Nombre de Tarea: Verificar integración de las categorías con la base de datos	
Tipo de Tarea: Revisión	Puntos Estimados: 0.2
Fecha de inicio: 18/05/22	Fecha de fin: 20/05/22
Programador Responsable: Lesly Meliza Ferrer Aguado	
Descripción: Permitirá al usuario crear las categorías en que se cataloga la información.	

Tabla 20. Tarea de Ingeniería 15 en la iteración presente

Tarea de Ingeniería	
Número de Tarea: 15	Nombre: HU 7. Gestionar muestra de categoría
Nombre de Tarea: Verificar integración de las muestras con la base de datos	
Tipo de Tarea: Revisión	Puntos Estimados: 0.1
Fecha de inicio: 07/07/22	Fecha de fin: 08/07/22
Programador Responsable: Lesly Meliza Ferrer Aguado	
Descripción: Permitirá al usuario crear las muestras para catalogar la información agregándolo a la base de datos.	

Como resultado de entrega de la segunda iteración del sistema de obtención y catalogación, el cliente quedó conforme con las funcionalidades de los módulos que se desarrollaron, así mismo se realizaron los cambios que fueron solicitados por el cliente en la primera iteración.

Iteración 3

Para la siguiente iteración se ha desarrollado el módulo Fuentes, el cual está diseñado y codificado mediante las herramientas que brinda la metodología XP.

Historias de Usuario

Tabla 21. Historias de usuario tercera iteración

Número	Nombre
9	Gestionar fuente de información
10	Filtrar fuente de información

Tareas de Ingeniería

Tabla 22. Tareas de ingeniería de la tercera iteración

Número de Tarea	Número de Historias	Nombre de la Tarea
1	9	Diseño de Interfaz de fuentes
2	9	Verificar integración de las fuentes en la base de datos
3	9	Validar campos de fuentes
4	9	Diseñar la interfaz de detalles de fuentes
5	9	Validar datos de formulario de detalles
6	10	Diseño de Interfaz de filtrar fuentes
7	10	Validar campos de filtrar fuentes

Tabla 23. Tarea de Ingeniería 1 para la historia de usuario 9

Tarea de Ingeniería	
Número de Tarea: 1	Nombre: HU 9. Gestionar fuente de información
Nombre de Tarea: Diseño de Interfaz de fuentes	
Tipo de Tarea: Revisión	Puntos Estimados: 0.2
Fecha de inicio: 18/05/22	Fecha de fin: 20/05/22
Programador Responsable: Lesly Meliza Ferrer Aguado	
Descripción: Permitirá al usuario visitar la interfaz de fuentes y así poder ver la información descargada por cada fuente.	

Como resultado de entrega de la tercera iteración del sistema de obtención y catalogación, el cliente quedó conforme con las funcionalidades de los módulos que se desarrollaron, así mismo se realizaron los cambios que fueron solicitados por el cliente en la segunda iteración.

Iteración 4

Para esta última iteración se ha desarrollado el módulo Contenido, el cual está diseñado y codificado mediante las herramientas que brinda la metodología XP.

Historias de Usuario

Tabla 24. Historias de usuario cuarta iteración

Número	Nombre
11	Listar contenidos descargados
12	Filtrar contenido
13	Ver contenido
14	Adicionar contenido al corpus
15	Eliminar contenido
16	Exportar contenido como PDF

Tareas de Ingeniería

Tabla 25. Tareas de ingeniería de la cuarta iteración

Número de Tarea	Número de Historias	Nombre de la Tarea
1	11	Diseñar interfaz de listar contenidos
2	11	Validar campos listados
3	12	Diseñar la interfaz para filtrar contenidos
4	12	Validar campos de contenidos por categoría en la base de datos
5	12	Validar campos de contenidos por fuente en la base de datos
6	12	Validar campos por fecha de descarga en la base de datos

7	13	Diseñar la interfaz de detalles del contenido
8	13	Ir a través del enlace a la noticia original
9	13	Validar campos de detalle
10	14	Diseñar la interfaz adicionar contenido al corpus
11	14	Verificar la adición del contenido en la base de datos
12	15	Diseñar la interfaz de eliminar contenido
13	15	Comprobar la eliminación del contenido en la base de datos
14	16	Exportar contenido como PDF

Tabla 26. Tarea de Ingeniería 1 para la historia de usuario 11

Tarea de Ingeniería	
Número de Tarea: 1	Nombre: HU 11. Listar contenido descargado
Nombre de Tarea: Diseñar interfaz	
Tipo de Tarea: Revisión	Puntos Estimados: 1
Fecha de inicio: 08/08/22	Fecha de fin: 16/08/22
Programador Responsable: Lesly Meliza Ferrer Aguado	
Descripción: Permitirá al usuario observar las listas de contenido de la información descargada.	

Tabla 27. Tarea de Ingeniería 4 para la iteración actual

Tarea de Ingeniería	
Número de Tarea: 4	Nombre: HU 12. Filtrar contenidos
Nombre de Tarea: Validar campos de contenidos por categoría en la base de datos	
Tipo de Tarea: Revisión	Puntos Estimados: 2
Fecha de inicio: 16/08/22	Fecha de fin: 1/09/22
Programador Responsable: Lesly Meliza Ferrer Aguado	
Descripción: Permitirá al usuario filtrar el contenido de la información descargada por categorías.	

Como resultado de la cuarta iteración del proyecto llevado a cabo, una vez más, el cliente quedó satisfecho con la funcionalidad e integración de los módulos desarrollados.

Una vez determinadas las tareas de ingeniería a implementar por cada iteración, se posee un espectro más detallado para el desarrollo de la solución planteada.

Fase IV: Pruebas

La metodología XP en su última fase propone la realización de las pruebas de software, con el objetivo de desarrollar una herramienta que cumpla con los requisitos previamente identificados. Los desarrolladores tienden a predecir las necesidades futuras e implementarlas antes. En XP sólo se analiza lo que se desarrollará en la iteración actual, olvidando por completo cualquier necesidad que se pueda presentar en el futuro, lo que supone uno de los preceptos más radicales de la programación extrema (Meléndez Valladarez, S. M. (28 de enero de 2016). METODOLOGIA ÁGIL DE DESARROLLO DE SOFTWARE PROGRAMACION. Obtenido de <https://repositorio.unan.edu.ni/1365/1/62161.pdf>

III.2 Descripción de la estrategia de pruebas a seguir III.2

Uno de los pilares de la eXtreme Programming es el proceso de pruebas. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. Además permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones.

Las pruebas de software comprenden el conjunto de actividades que se realizan para identificar posibles fallos de funcionamiento, configuración o usabilidad, de un programa o aplicación, mediante pruebas sobre el comportamiento del mismo. Las pruebas son uno de los elementos de XP y en tal sentido surgieron algunas consideraciones con sus características, en el presente epígrafe se describen las pruebas de la propuesta de solución, definidas a través de los casos de pruebas, conformados con el objetivo de probar las funcionalidades definidas en el sistema. En este proceso no solo participa el equipo de desarrollo, también es importante los aportes del cliente, sobre todo en las pruebas de aceptación (Echeverry Tobón, L. M. (2007). CASO PRÁCTICO DE LA METODOLOGÍA ÁGIL XP AL DESARROLLO DE. Obtenido de <https://aalbertovargasc.files.wordpress.com/2011/07/proyecto-con-xp.pdf>).

XP divide las pruebas en dos grupos: pruebas unitarias, encargadas de verificar el código y pruebas de aceptación, destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñada por el cliente final.

Pruebas de Aceptación

Las pruebas de aceptación, también llamadas pruebas funcionales son una parte integral del desarrollo incremental. Son pruebas creadas en base a las historias de usuario, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar por cada iteración definida que una historia de usuario ha sido correctamente implementada y ninguna malinterpretada. Asimismo, en caso que fallen varias pruebas, se debe indicar el orden de prioridad de resolución.

Las pruebas de aceptación son pruebas de caja negra, que representan un resultado esperado de determinada transacción con el sistema. Una historia de usuario no se puede considerar terminada o aprobada hasta que pase correctamente todas las pruebas de aceptación elaboradas para ella (Echeverry Tobón, L. M., 2007).

La siguiente tabla muestra el número de pruebas de aceptación desarrolladas a la aplicación web, el número de historia de usuario a la cual pertenece y el nombre de la prueba:

Tabla 28. Listado de Pruebas de Aceptación Desarrolladas

Número de prueba	Número de Historia	Nombre de la Prueba
1	HU_1	Autenticar Usuario
2	HU_3	Gestionar Categoría
3	HU_7	Gestionar Fuente de Información
4	HU_9	Gestionar Muestra de Categoría
5	HU_11	Listar Contenido Descargado
6	HU_12	Filtrar Contenido

Diagrama de Actividad

El UML de diagrama de actividades se emplea para mostrar la secuencia de actividades. Los diagramas de actividades muestran el flujo de trabajo desde el punto de inicio hasta el punto final detallando muchas de las rutas de decisiones que existen en el progreso de eventos contenidos en la actividad. Los diagramas de actividades son útiles para el modelado del negocio, donde son empleados para detallar el proceso involucrado en las actividades de negocio.

Una actividad es la especificación de una secuencia parametrizada de comportamiento, se muestra mediante un rectángulo con las puntas redondeadas adjuntando todas las acciones, flujos de control y otros elementos que constituyen la actividad.

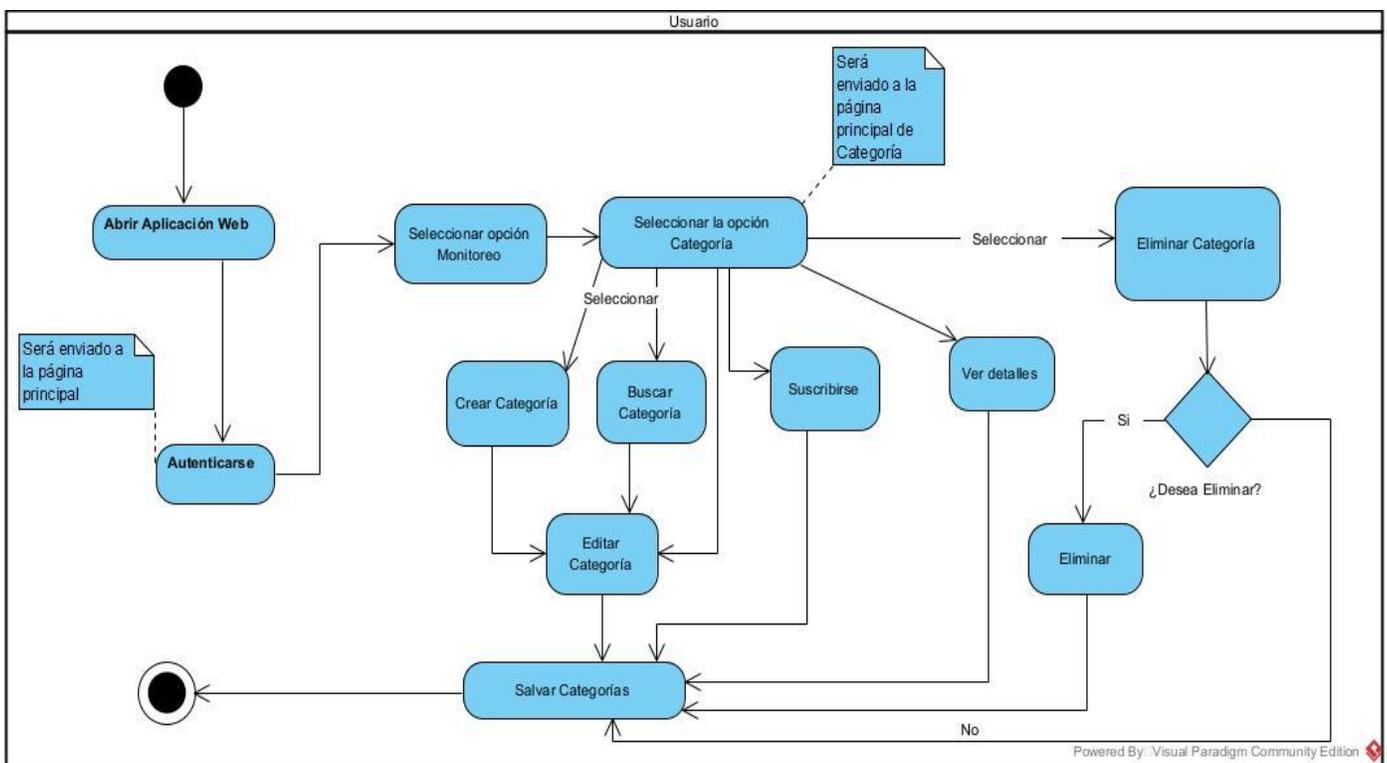


Ilustración 20. Diagrama de Actividades: Gestionar Categoría

Pruebas Unitarias

El objetivo de las pruebas del sistema es verificar los requisitos. Por este motivo, los propios requisitos del sistema son la principal fuente de información a la hora de construir pruebas del sistema. En un desarrollo XP los requisitos se recogen en historias de usuario durante la etapa de planificación. Una historia de usuario recoge la descripción natural, en general ambigua,

incompleta e informal, de un fragmento de la funcionalidad del sistema. Aunque las historias de usuario pueden ser completadas con notas, el tamaño recomendado para la tarjeta no permite añadir información clara ni exhaustiva (Meléndez Valladarez, S. M. (28 de enero de 2016). METODOLOGIA ÁGIL DE DESARROLLO DE SOFTWARE PROGRAMACION. Obtenido de <https://repositorio.unan.edu.ni/1365/1/62161.pdf>).

Por tanto, para la realización de las pruebas unitarias, XP propone la utilización de una herramienta que las realice de forma automática. Todos los módulos deben pasar las pruebas unitarias antes de ser liberados o publicados. Por otra parte, las pruebas deben ser definidas antes de realizar el código (“Test-Driven Programming”). Que todo código liberado pase correctamente las pruebas unitarias, es lo que habilita que funcione la propiedad colectiva del código. Estas pruebas se realizan a los métodos no triviales de todas las clases del proyecto con la condición que no se libraré ninguna clase que no tenga asociada su correspondiente paquete de pruebas. El empleo de las pruebas unitarias completas facilita la liberación continua de versiones por cuanto al implementar algo nuevo y actualizar la última versión, solo es cuestión de ejecutar de forma automática las pruebas unitarias ya creadas para saber que la nueva versión no contiene errores (Echeverry Tobón, L. M. (2007). CASO PRÁCTICO DE LA METODOLOGÍA ÁGIL XP AL DESARROLLO DE. Obtenido de <https://aalbertovargasc.files.wordpress.com/2011/07/proyecto-con-xp.pdf>

Herramienta de Prueba

Para la realización de dichas pruebas se emplea la herramienta de prueba Jest, ya que fue diseñada principalmente para aplicaciones basadas en React.

Jest es la herramienta más popular recomendada para React. Es un marco de trabajo creado por Facebook, una biblioteca de JavaScript para la ejecución de pruebas que permite acceder al DOM mediante jsdom. Aunque jsdom solo se aproxima a como realmente los navegadores web trabajan, usualmente es suficiente para probar componentes de React. Jest brinda una gran velocidad de iteración combinada con potentes funcionalidades como simular módulos y temporizadores, esto permite tener mayor control sobre cómo se ejecuta el código (Obtenido de React: <https://es.reactjs.org/docs/testing.html#tools>, reactjs.org. (s.f.).

Ventajas de usar Jest

Compatibilidad: Es compatible con Angular, ReactJS, NodeJS, VueJS y otros proyectos basados en babel.

Rapidez: Es una herramienta de prueba muy rápida. Cuando la prueba está vinculada a la CPU, puede ahorrar un tiempo significativo de nuestras ejecuciones de prueba. La prueba local solía tomar 45 minutos, Jest lo redujo a valores de entre 14 y 15 minutos.

Basado en Jasmine: Se basaba en Jasmine, por lo que ha heredado todas sus buenas cualidades. Aunque se ha apartado de Jasmine, últimamente, ha mantenido la misma funcionalidad, pero ahora mejorada.

Compatible con TypeScript: TypeScript es un superconjunto de JavaScript que se compila en JS simple. Admite TypeScript a través del paquete ts-jest.

Pruebas instantáneas: Es una gran herramienta, garantiza que la interfaz de usuario de una aplicación no cambie inesperadamente entre versiones.

Simulacros de temporizador: Los usuarios pueden manipular el tiempo. Por lo tanto, la prueba no será lenta.

- `setTimeout ()`
- `setInterval ()`
- `clearTimeout ()`
- `clearInterval ()`

Configurable: Es fácil de instalar usando directamente npm o hilo.



Ilustración 21. Logo de Jest (jestjs.io)

Detección y Corrección de Errores

Cuando se encuentra un error (“Bug”), este debe ser corregido inmediatamente, y se debe tener precauciones para que errores similares no vuelvan a ocurrir. Asimismo, se generan nuevas pruebas antes de intentar corregirlo. De esta forma tanto el cliente logrará tener completamente claro cuál fue y dónde se encontraba el mismo como el equipo de desarrollo podrá enfocar mejor sus esfuerzos para solucionarlo. Por otro lado, se evitará volver a cometerlo. Si el error fue reportado por el cliente y este creó la correspondiente prueba de aceptación junto al equipo de desarrollo, el programador encargado podrá a su vez producir nuevas pruebas unitarias que le permita ubicar la sección específica donde se encuentra el error (Echeverry Tobón, L. M. (2007). CASO PRÁCTICO DE LA METODOLOGÍA ÁGIL XP AL

```

FAIL src/Test/RegisterCard.test.js
Funciones a testear dentro del componente auth
Comprobar coincidencia de las contraseñas proporcionadas
  ✓ Debe retornar un boolean (2 ms)
Comprobar si el tamaño de la contraseña proporcionada es mayor de 5 caracteres
  ✗ Debe retornar un boolean
Comprobar si el correo proporcionado es valido mediante la librería validate
  ✓ Debe retornar un boolean (4 ms)

• Funciones a testear dentro del componente auth › Comprobar si el tamaño de la contraseña proporcionada es mayor de 5 caracteres › Debe retornar un boolean

TypeError: Cannot read properties of undefined (reading 'length')

   105 |   };
   106 |   export const passwordLength = (password) => {
>  107 |     if (password.length > 5) {
       |           ^
   108 |         return true;
   109 |     } else {
   110 |         return false;

at passwordLength (src/components/auth/RegisterScreen.js:107:18)
at Object.<anonymous> (src/Test/RegisterCard.test.js:12:28)

Test Suites: 1 failed, 1 total
Tests:      1 failed, 2 passed, 3 total
Snapshots: 0 total
Time:      3.871 s
Ran all test suites.

```

DESARROLLO DE. Obtenido de

<https://aalbertovargasc.files.wordpress.com/2011/07/proyecto-con-xp.pdf>

Ilustración 22. Error arrojado en el test de Registrar Usuario

El error sucedió, al usuario no proporcionar el password en el formulario, por tanto, no se pudo validar el número de caracteres del campo, arrojando un error al realizar el test. El mismo, fue

corregido imponiendo una verificación en el campo, impidiendo que se envíe vacío o nulo, al culminar el formulario.

III.3 Realización de las pruebas III.3

XP enfatiza en la realización de un sin número de pruebas a lo largo del proyecto, con el fin de asegurar en todo momento la realización de lo planteado en el diseño.

A continuación, se comenzará ejemplificando con las pruebas de aceptación a varias de las diferentes historias de usuario existentes en cada una de las iteraciones:

Pruebas de aceptación

Las pruebas funcionales son definidas y escritas por el cliente al inicio de cada iteración para cada una de las historias establecidas. El objetivo es demostrar al cliente que el requerimiento implementado realmente funciona como él lo desea (Echeverry Tobón, L. M. (2007). CASO PRÁCTICO DE LA METODOLOGÍA ÁGIL XP AL DESARROLLO DE. Obtenido de <https://aalbertovargasc.files.wordpress.com/2011/07/proyecto-con-xp.pdf>). A continuación, se muestran las pruebas de aceptación, mediante una serie de tablas, llevadas a cabo por cada historia de usuario expuesta en el capítulo anterior:

Iteración 1

Tabla 29. Caso de Prueba 1

Caso de Prueba	
Código: HU-1_P-1	No. Historia de Usuario: HU-1
Historia de Usuario: Autenticar Usuario	
Pre condición: El usuario debe estar sin autenticar.	
Condiciones de ejecución: El usuario debe acceder al sitio en la página de autenticación.	
Pasos de Ejecución: 1. Ingresar a la aplicación. 2. Completar los datos requeridos. 3. Seleccionar el botón Acceder.	
Resultado esperado: 1. El usuario es enviado a la página de inicio.	
Evaluación de la prueba: La prueba se concluyó satisfactoriamente.	

Iteración 2

Tabla 30. Caso de Prueba 3

Caso de Prueba	
Código: HU-3_P-3	No. Historia de Usuario: HU-3
Historia de Usuario: Gestionar Categoría	
Pre condición: El usuario debe estar autenticado.	
Condiciones de ejecución: El usuario podrá acceder a gestionar las categorías a través del menú que se encuentra en la página inicial de la aplicación.	
Pasos de Ejecución: 1. El usuario debe autenticarse 2. Será enviado a la página de inicio 3. A través del menú desplegado a su izquierda podrá acceder a Monitoreo 4. Dentro de la opción Monitoreo encontrará la opción Categorías 5. En la página principal de Categorías podrá: <div style="margin-left: 100px;"> Crear nueva categoría Buscar categoría Eliminar categoría Editar categoría Ver detalles de la categoría Suscribirse </div>	
Resultado esperado: El usuario podrá gestionar las Categorías	
Evaluación de la prueba: La prueba se concluyó satisfactoriamente.	

Tabla 31. Caso de Prueba 7

Caso de Prueba	
Código: HU-7_P-7	No. Historia de Usuario: HU-7
Historia de Usuario: Gestionar muestra de Categoría	
Pre condición: El usuario debe estar autenticado.	
Condiciones de ejecución: El usuario podrá acceder a gestionar fuente de información a través del menú que se encuentra en la página inicial de la aplicación.	

<p>Pasos de Ejecución: 1. El usuario debe autenticarse</p> <ol style="list-style-type: none"> 2. Será enviado a la página de inicio 3. A través del menú desplegado a su izquierda podrá acceder a Monitoreo 4. Dentro de la opción Monitoreo encontrará la opción Categorías 5. En la página principal de Fuentes podrá: <ul style="list-style-type: none"> Crear muestra Eliminar muestra Editar muestra Listar muestra
<p>Resultado esperado: El usuario podrá gestionar las Fuentes de Información</p>
<p>Evaluación de la prueba: La prueba se concluyó satisfactoriamente.</p>

Iteración 3

Tabla 32. Caso de Prueba 9

<p>Caso de Prueba</p>	
<p>Código: HU-9_P-9</p>	<p>No. Historia de Usuario: HU-9</p>
<p>Historia de Usuario: Gestionar Fuente de Información</p>	
<p>Pre condición: El usuario debe estar autenticado.</p>	
<p>Condiciones de ejecución: El usuario podrá acceder a gestionar fuente de información a través del menú que se encuentra en la página inicial de la aplicación.</p>	
<p>Pasos de Ejecución: 1. El usuario debe autenticarse</p> <ol style="list-style-type: none"> 2. Será enviado a la página de inicio 3. A través del menú desplegado a su izquierda podrá acceder a Monitoreo 4. Dentro de la opción Monitoreo encontrará la opción Fuentes 5. En la página principal de Fuentes podrá: <ul style="list-style-type: none"> Crear nueva fuente Eliminar fuente Editar fuente Ver detalles de la fuente Listar fuentes 	
<p>Resultado esperado: El usuario podrá gestionar las Fuentes de Información</p>	

Evaluación de la prueba: La prueba se concluyó satisfactoriamente.
--

Iteración 4

Tabla 33. Caso de Prueba 11

Caso de Prueba	
Código: HU-11_P-11	No. Historia de Usuario: HU-11
Historia de Usuario: Listar Contenido Descargado	
Pre condición: El usuario debe estar autenticado.	
Condiciones de ejecución: El usuario encontrará la lista de contenidos descargados a través del menú desplegado en la parte izquierda de la página inicial de la aplicación.	
Pasos de Ejecución: 1. El usuario debe autenticarse 2. Será enviado a la página de inicio 3. A través del menú desplegado a su izquierda podrá acceder a Monitoreo 4. Dentro de la opción Monitoreo encontrará la opción Descargas 5. En la página principal de Descargas se muestra la lista de contenidos	
Resultado esperado: El usuario podrá tener acceso a la lista de los contenidos descargados con anterioridad.	
Evaluación de la prueba: La prueba se concluyó satisfactoriamente.	

Tabla 34. Caso de Prueba 12

Caso de Prueba	
Código: HU-12_P-12	No. Historia de Usuario: HU-12
Historia de Usuario: Filtrar Contenido	
Pre condición: El usuario debe estar autenticado.	
Condiciones de ejecución:	
Pasos de Ejecución: 1. El usuario debe autenticarse 2. Será enviado a la página de inicio 3. A través del menú desplegado a su izquierda podrá acceder a Monitoreo 4. Dentro de la opción Monitoreo encontrará la opción Descargas	

5. En la página principal de Descargas seleccione la opción Filtrado Avanzado
Resultado esperado:
Evaluación de la prueba: La prueba se concluyó satisfactoriamente.

Pruebas Unitarias

Las pruebas unitarias son implementadas por el programador antes de comenzar con la implementación. Se establecen todos los casos en el que el código puede fallar. Una vez que se termina con la implementación se corren dichas pruebas y se verifica que no haya ningún test que falle. Hasta que pasen todos los test el programador debe seguir mejorando el código (Echeverry Tobón, L. M. (2007). CASO PRÁCTICO DE LA METODOLOGÍA ÁGIL XP AL DESARROLLO DE. Obtenido de <https://aalbertovargasc.files.wordpress.com/2011/07/proyecto-con-xp.pdf>).

Para comenzar a realizar dichas pruebas, primero fue creado un archivo nombrado Test, donde se localizó el archivo RegisterCard.test.js, ya que fue una prueba realizada al proceso de registrar usuario, donde se ubicaron los diseños de prueba por componentes, dentro del cual se procede a realizar la descripción de las pruebas a nivel de componente y dentro de la descripción general se hace la particular a la función que será sometida a prueba, luego se obtiene el resultado de la función y se describe la expectativa de del resultado de la prueba. Posteriormente mediante el comando npm test se procede a ejecutar las pruebas indicando la prueba que se va a testear específicamente. A continuación, se muestran las pruebas realizadas al componente registrar:

```
RegisterScreen.js M X RegisterCard.test.js U
src > components > auth > RegisterScreen.js > validateEmail
97   }
98
99   export const passwordConicidence = (password, confirmPassword) => {
100     if (password !== confirmPassword) {
101       return false;
102     } else {
103       return true;
104     }
105   };
106   export const passwordLength = (password) => {
107     if (password.length > 5) {
108       return true;
109     } else {
110       return false;
111     }
112   };
113   export const validateEmail = (email) => {
114     if (!validator.isEmail(email)) {
115       return false;
116     } else {
117       return true;
118     }
119   };
120
```

Ilustración 23. Código sometido a prueba: Registrar Usuario

```

RegisterScreen.js M  RegisterCard.test.js U X
Funciones a testear dentro del componente auth' callback > describe('Comprobar si el correo proporcionado es valido mediante la librería validate') callback > test('De
1  import {passwordCoincidencia, passwordLength, validateEmail} from '../components/auth/RegisterScreen'
2
3  describe('Funciones a testear dentro del componente auth', () => {
4    describe('Comprobar coincidencia de las contraseñas proporcionadas', () => {
5      test('Debe retornar un boolean', () => {
6        const result = passwordCoincidencia('12345','12345');
7        expect(typeof result).toBe('boolean');
8      });
9    });
10   describe('Comprobar si el tamaño de la contraseña proporcionada es mayor de 5 caracteres', () => {
11     test('Debe retornar un boolean', () => {
12       const result = passwordLength('446464');
13       expect(typeof result).toBe('boolean');
14     });
15   });
16   describe('Comprobar si el correo proporcionado es valido mediante la librería validate', () => {
17     test('Debe retornar un boolean', () => {
18       const result = validateEmail('lmferrer@estudiantes.uci');
19       expect(typeof result).toBe('boolean');
20     });
21   });
22 });

```

Ilustración 24. Diseño de prueba: Registrar Usuario

```

PASS src/Test/RegisterCard.test.js
Funciones a testear dentro del componente auth
  Comprobar coincidencia de las contraseñas proporcionadas
    ✓ Debe retornar un boolean (2 ms)
  Comprobar si el tamaño de la contraseña proporcionada es mayor de 5 caracteres
    ✓ Debe retornar un boolean (1 ms)
  Comprobar si el correo proporcionado es valido mediante la librería validate
    ✓ Debe retornar un boolean

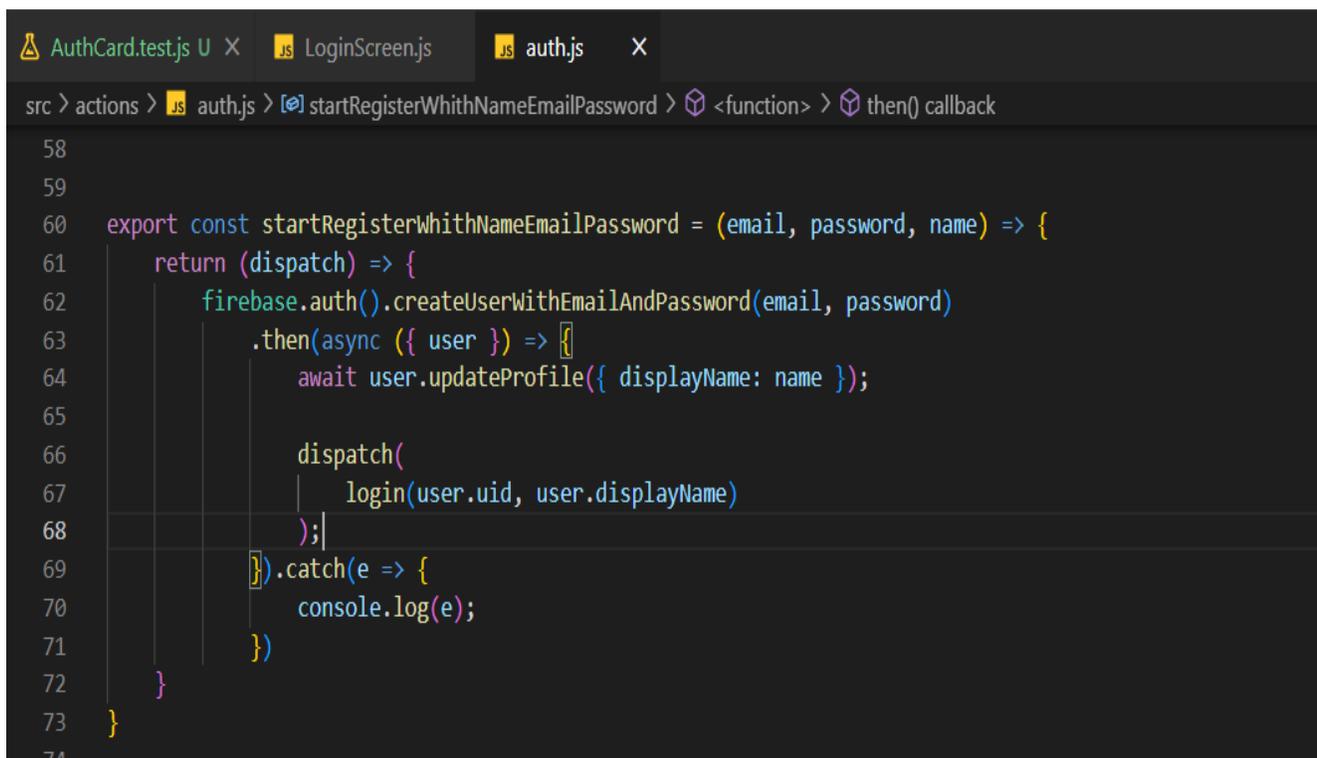
Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:  0 total
Time:        3.682 s
Ran all test suites.

Watch Usage: Press w to show more.

```

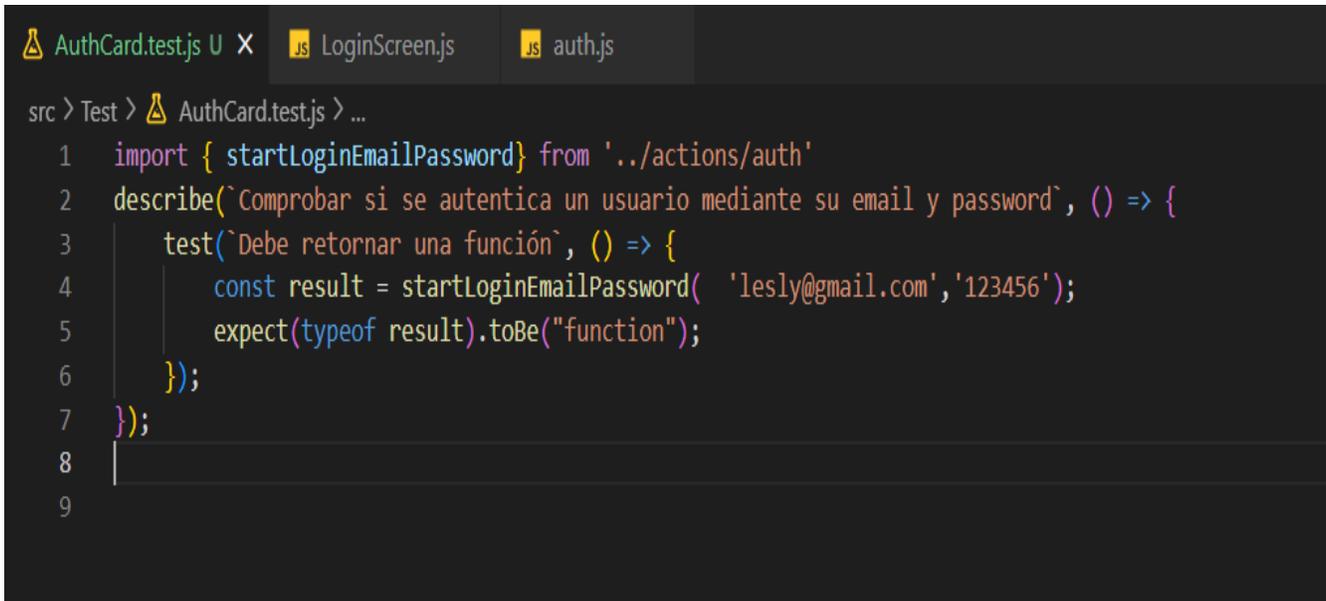
Ilustración 25. Resultado Satisfactorio Obtenido: Registrar Usuario

Para comenzar a realizar dichas pruebas, primero fue creado un archivo nombrado Test, donde se localizó el archivo AuthCard.test.js, ya que fue una prueba realizada al proceso de autenticación de usuario, donde se ubicaron los diseños de prueba por componentes, dentro del cual se procede a realizar la descripción de las pruebas a nivel de componente y dentro se hace la particular a la función que será sometida a prueba, luego se obtiene el resultado de la función y se describe la expectativa del resultado de la prueba. Posteriormente mediante el comando npm test se procede a ejecutar la prueba que se va a testear específicamente. A continuación, se muestran las pruebas realizadas al componente autenticación:

A screenshot of a code editor with a dark theme. The editor shows a file named 'auth.js' with a breadcrumb path 'src > actions > auth.js > startRegisterWhithNameEmailPassword > <function> > then() callback'. The code is as follows:

```
58
59
60 export const startRegisterWhithNameEmailPassword = (email, password, name) => {
61   return (dispatch) => {
62     firebase.auth().createUserWithEmailAndPassword(email, password)
63       .then(async ({ user }) => {
64         await user.updateProfile({ displayName: name });
65
66         dispatch(
67           login(user.uid, user.displayName)
68         );
69       }).catch(e => {
70         console.log(e);
71       })
72   }
73 }
```

Ilustración 26. Código sometido a prueba: Autenticar Usuario

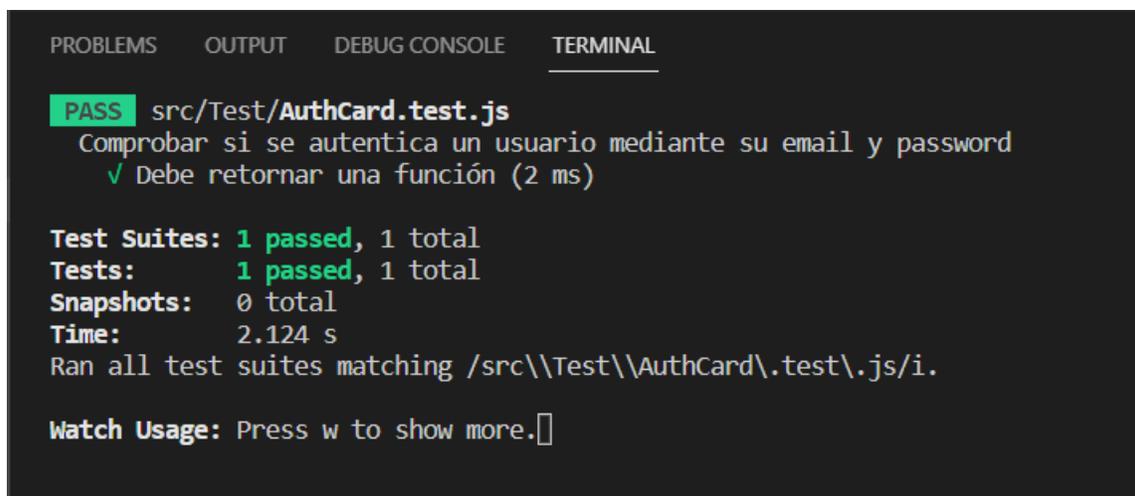


```

AuthCard.test.js U X LoginScreen.js auth.js
src > Test > AuthCard.test.js > ...
1 import { startLoginEmailPassword } from '../actions/auth'
2 describe(`Comprobar si se autentica un usuario mediante su email y password`, () => {
3   test(`Debe retornar una función`, () => {
4     const result = startLoginEmailPassword( 'lesly@gmail.com', '123456');
5     expect(typeof result).toBe("function");
6   });
7 });
8
9

```

Ilustración 27. Diseño de prueba: Autenticar Usuario



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PASS src/Test/AuthCard.test.js
  Comprobar si se autentica un usuario mediante su email y password
    ✓ Debe retornar una función (2 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        2.124 s
Ran all test suites matching /src\\Test\\AuthCard\\.test\\.js/i.

Watch Usage: Press w to show more.

```

Ilustración 28. Resultado Satisfactorio Obtenido: Autenticar Usuario

Como resultado de la entrega de la aplicación web, el cliente quedó satisfecho con las funcionalidades e integraciones desarrolladas en cada iteración, ya que contará con lo necesario para agilizar el proceso de obtención y catalogación de contenidos de internet tal cual fueron descritas mediante las historias de usuario al inicio del proyecto.

Conclusiones del capítulo

El proceso de desarrollo de las pruebas ayudó al cliente a clarificar y concretar la funcionalidad de la historia de usuario y favorece la comunicación entre el cliente y el equipo de desarrollo. En este capítulo se representaron las tareas de ingeniería llevadas a cabo por las historias en cada iteración, se realizó el diagrama de actividad para obtener pruebas de aceptación. Fue probada la solución informática con los niveles de pruebas unitaria y aceptación, obteniendo resultados satisfactorios.

Las pruebas unitarias se llevaron a cabo mediante el marco de prueba Jest, el cual permitió la realización de las comprobaciones según dicta la metodología que rige el desarrollo de la aplicación. Se obtuvieron resultados que arrojaron error tras verificar el código mediante las pruebas diseñadas por los desarrolladores para los distintos escenarios evaluados, los cuales fueron solucionados.

Las pruebas de aceptación se realizaron de manera independiente del desarrollo del código, generadas desde los requisitos funcionales. Se especificaron por el cliente los diversos escenarios para comprobar que cada historia de usuario haya sido correctamente implementada.

CONCLUSIONES FINALES

Al concluir la presente investigación, y una vez implementada la propuesta de solución, se concluye que:

La definición de los elementos teóricos y metodológicos asociados a la gestión del proceso de vigilancia e inteligencia empresarial, permitió profundizar en los aspectos fundamentales para el desarrollo de la investigación.

El análisis de los lenguajes, herramientas y tecnologías, permitió la selección de los más idóneos, los cuales, unidos a la metodología de desarrollo XP, posibilitaron la implementación de la aplicación propuesta.

Se puede afirmar que con la comprensión del proceso de desarrollo de software utilizando la metodología de programación extrema XP, las técnicas que aporta, fueron de vital importancia para llevar a cabo la creación del sistema de obtención y catalogación de contenidos de internet, pues con esto, fue permitido al equipo de trabajo realizar cada una de las tareas a cumplir en forma ordenada y a su tiempo de entrega, así mismo brindar al cliente un sistema de acuerdo a sus necesidades requeridas.

Con el diseño del sistema se corroboró la necesidad de implementar una nueva versión que fuese capaz de ajustarse a solucionar las deficiencias presentadas por las aplicaciones analizadas para conformar un sitio web que se complemente en la realización del proceso estudiado. Quedando así definidas las bases para su implementación.

Mediante el proceso de desarrollo de las pruebas se fortaleció el vínculo entre el cliente y el equipo de trabajo. La estrategia de pruebas definidas para validar el sistema de obtención y catalogación de contenidos de internet, permitió la identificación de inconformidades existentes en la aplicación y su posterior corrección.

RECOMENDACIONES

Para que el sistema de obtención y catalogación de contenidos de internet tenga un buen funcionamiento, se realizan las siguientes recomendaciones:

- Se recomienda brindar capacitación al personal que hará uso del sistema web.
- Establecer políticas que permitan el respaldo de la información en caso de fallar el software, hardware o por mala manipulación del sistema.
- Se sugiere brindar el debido mantenimiento al sistema periódicamente.
- El sistema será protegido a través de contraseñas que restrinjan el inicio de sesión y que impida que un tercero pueda acceder a él.
- Solo los usuarios autorizados manipularán cada función del sistema web, una vez que esté en marcha.
- Plantear una estrategia para afrontar el diseño de datos en XP.
- Plantear una estrategia especial de refactorización para las bases de datos.
- Debe hacerse lo posible por no realizar modificaciones a XP demasiado drásticas ya que se corre el riesgo de alterar la esencia de la metodología.

REFERENCIAS BIBLIOGRÁFICAS

1. *¿Qué es la metodología 'agile'?* (2022). Obtenido de www.bbva.com: <https://www.bbva.com/es/metodologia-agile-la-revolucion-las-formas-trabajo/>
2. arsys. (s.f.). *¿Qué es PostgreSQL y por que llevarlo a Cloud?* Obtenido de arsys.es: <https://www.arsys.es/blog/soluciones/postgresql-servidores>
3. Cevolini, A. (12 de marzo de 2016). *Historia y Grafía*. Obtenido de El archivo como máquina histórica: el sistema de selección e información en las prácticas de catalogación: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-09272016000200251&lang=es#fn21
4. Coelho, F. (2013). *significados*. Obtenido de significados.com: <https://www.significados.com/>
5. Deepl. (s.f.).
6. Digité. (2022). *¿Qué es la programación extrema (XP) y*. Obtenido de www.digite.com: <https://www.digite.com/es/agile/programacion-extrema-xp/>
7. Evaluando ERP. (2022). *evaluandoerp.com*. Obtenido de <https://www.evaluandoerp.com/software-erp/sistema-de-gestion/>
8. Fernández Escribano, G. (2002). *Introducción a Extreme Programming*.
9. Gardner, T. e. (julio de 2021). *How readers discover content in scholarly publications*. Obtenido de renewconsultants.com: <https://renewconsultants.com/wp-content/uploads/2018/08/How-Readers-Discover-Content-2018-Published-180903.pdf>
10. Hearst, M. A. (marzo de 2013). Obtenido de <https://doi.org/10.1017/CBO9781139644082>
11. Hottinger, J. H. (s.f.). *Las cursivas son más*. Zúrich.
12. IGNITE. (2020). *El proceso de catalogación bibliográfica y de recursos digitales*. Obtenido de Ignite: <https://igniteonline.la>
13. infomed instituciones. (16 de abril de 2017). *¿QUE ES GESTIÓN DE LA INFORMACIÓN?* Obtenido de instituciones.sld: <https://www.instituciones.sld.cu>
14. mdn. (s.f.). *Acerca de JavaScript*. Obtenido de www.mozilla.org: https://developer.mozilla.org/es/docs/Web/JavaScript/About_JavaScript

15. Mendeley. (s.f.). *Mendeley Desktop*. Obtenido de mendeley-desktop.uptodown.com: <https://mendeley-desktop.uptodown.com/windows>
16. Microsoft. (2022). *Power BI*. Obtenido de powerbi.microsoft.com: <https://powerbi.microsoft.com/es-es/what-is-business-intelligence/>
17. miteris. (8 de junio de 2022). *¿Qué es Javascript? Características y Librerías*. Obtenido de www.miteris.com: <https://www.miteris.com/blog/que-es-javascript-caracteristicas-librerias/>
18. Observatorio Tecnológico UA. (2016). *Guía de Vigilancia e Inteligencia Tecnológica*. Obtenido de ovtt.org: <https://www.ovtt.org/guias/guia-de-inteligencia-tecnologica/>
19. Open Geek. (s.f.). *Koha software de gestión bibliotecaria*. Obtenido de opengeekservice.cl: <https://www.opengeekservice.cl/sitio/es/bibliotecas/koha.html>
20. Orex. (s.f.). *Catálogo Nacional en Publicaciones de Ciencias de la Salud*. Obtenido de cncs.isciii.es: <https://cncs.isciii.es/wp-content/uploads/2019/01/CNCS-Manualv01.pdf>
21. Oxford. (s.f.). *Lexico*. Obtenido de lexico.com: <https://www.lexico.com/es/definicion/internet>
22. Pantallea. (2022). *Contenidos digitales: qué son, tipos y ejemplos*. Obtenido de pantallea.com: <https://www.pantallea.com>
23. Pérez, J. y. (2019). *SURVEILLANCE AND INTELLIGENCE AS A SUCCESS STRATEGY IN BUSINESS*. Venezuela: Qualitas.
24. Real Academia Española. (s.f.). *dle.rae.es*. Obtenido de <https://www.rae.es/>
25. Santana Pérez, E. (2021). *Surveillance and Business Intelligence in Cuban Conditions*. La Habana: UH.
26. Spiegato . (2022). *¿Qué es la vigilancia empresarial?* Obtenido de spiegato.com: <https://spiegato.com/es/que-es-la-vigilancia-empresarial>
27. taiwebs. (s.f.). *Descargar Visual Studio Code - Herramienta de cifrado de Microsoft ha cumplido con la mayoría de las necesidades del desarrollador*. Obtenido de taiwebs.com: <https://es.taiwebs.com/windows/download-visual-studio-code-3182.html>
28. Tribalys Technologies. (s.f.). *Metodología ¿tradicional vs ágil?* Obtenido de www.tribalyte.com: <https://tech.tribalyte.eu/blog-metodologias-tradicional-vs-agil>
29. Universidad Politécnica de Valencia. (2022). *BiblioGuías*. Obtenido de biblioguias.webs.upv.es: <https://biblioguias.webs.upv.es/bg/index.php/es/como-buscar-informacion>.

30. Uptdown. (s.f.). *PostgreSQL*. Obtenido de postgresql.uptodown.com: <https://postgresql.uptodown.com/windows>
31. Varona Labrada, I. A., & Castellano Leyva, I. (2018). *Sistema de Obtención y Catalogación Automática de Contenidos de Internet*. La Habana.
32. *Visual Studio Code*. (s.f.). Recuperado el 8 de junio de 2022, de <https://code.visualstudio.com/>
33. Webolto. (24 de marzo de 2021). *Qué es un sitio web: definición, tipos y cómo crearlo*. Obtenido de [webolto.com](https://www.webolto.com/es/blog/que-es-un-sitio-web/): <https://www.webolto.com/es/blog/que-es-un-sitio-web/>
34. PostgreSQL.ES. 2009. [En línea] 7 de diciembre de 2009. [Citado el: 10 de junio de 2022.] http://www.postgresql.org.es/sobre_postgresql.
35. Histchfeld, N., & Salinas Caro, P. (20 de abril de 2014). Tutorial de UML. Obtenido de <http://users.dcc.uchile.cl/psalinas/uml/introduccion.html>
36. Pressman, R. S. (2002). *Ingeniería de software 5ta Edición*. McGrawHill.
37. Gacitúa, R. (2013). *Métodos de desarrollo de software. El desafío pendiente de la estandarización*.
38. Google. (s.f.). *google scholar*. Obtenido de scholar.google.com: <https://scholar.google.com>
39. Bustamante, D., & Rodríguez, J. (2014). *Metodología Actual Metodología XP*. Barinas.
40. Collazo García, A., & Díaz Labrador, M. (2013). *La Programación Extrema*. La Habana: Universidad de las Ciencias Informáticas.
41. Hernandez, R. D. (28 de junio de 2021). *Patrones de Diseño*. Obtenido de El Patrón modelo-vista-controlador: Arquitectura y Frameworks explicados: <https://www.freecodecamp.org/espanol/news/el-modelo-de-arquitectura-view-controller-pattern/>.
42. SciELO. (s.f.). *SciELO*. Obtenido de scielo.org: <https://scielo.org>
43. Meléndez Valladarez, S. M. (28 de enero de 2016). *METODOLOGIA ÁGIL DE DESARROLLO DE SOFTWARE PROGRAMACION*. Obtenido de <https://repositorio.unan.edu.ni/1365/1/62161.pdf>
44. Echeverry Tobón, L. M. (2007). *CASO PRÁCTICO DE LA METODOLOGÍA ÁGIL XP AL DESARROLLO DE*. Obtenido de <https://aalbertovargasc.files.wordpress.com/2011/07/proyecto-con-xp.pdf>

45. Obtenido de React: <https://es.reactjs.org/docs/testing.html#tools>, reactjs.org. (s.f.).
46. UML. (s.f.). uml.org. Obtenido de <http://www.uml.org/>
47. Microsoft. (2022). Visual Studio. Obtenido de <https://visualstudio.microsoft.com/es/>
48. Visual Paradigm. (2022). Visual-Paradigm. Obtenido de <https://www.visual-paradigm.com/>
49. python.org. (s.f.). Python. Obtenido de <https://docs.python.org/es/3/faq/general.html>
50. Django Software Foundation. (2005-2022). *Meet Django*. Obtenido de [djangoproject.com](https://www.djangoproject.com):
<https://www.djangoproject.com>
51. Unyscape. (2022). Django is a popular Web framework written in Python. Obtenido de [unyscape.com](https://unyscape.com/django-a-python-based-free-and-open-source-web-framework/): <https://unyscape.com/django-a-python-based-free-and-open-source-web-framework/>

ANEXOS

En las siguientes tablas se muestran las restantes historias de usuario que fueron creadas por el cliente con apoyo del equipo de trabajo para definir los requisitos funcionales de la aplicación web:

Tabla 35. Historia de Usuario 4

Historia de Usuario	
Código: HU-4	Nombre: RF 4. Entrenar categoría
Modificación de Historia de Usuario (Nro. Y Nombre):	
Usuario: Cliente	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1 semana
Descripción: Permitirá al usuario.	
Observaciones:	

Tabla 36. Historia de Usuario 5

Historia de Usuario	
Código: HU-5	Nombre: RF 5. Suscripción de categoría
Modificación de Historia de Usuario (Nro. Y Nombre):	
Usuario: Cliente	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1 semana
Descripción: Permitirá al usuario suscribirse a la categoría seleccionada.	
Observaciones:	

Tabla 37. Historia de Usuario 6

Historia de Usuario	
Código: HU-6	Nombre: RF 6. Cargar corpus de entrenamiento
Modificación de Historia de Usuario (Nro. Y Nombre):	
Usuario: Cliente	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1 semana
Descripción: Permitirá al usuario.	
Observaciones:	

Tabla 38. Historia de Usuario 8

Historia de Usuario	
Código: HU-8	Nombre: RF 8. Filtrar categoría
Modificación de Historia de Usuario (Nro. Y Nombre):	
Usuario: Cliente	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1 semana
Descripción: Permitirá al usuario filtrar por categoría el contenido descargado.	
Observaciones:	

Tabla 39. Historia de Usuario 10

Historia de Usuario	
Código: HU-10	Nombre: RF 10. Filtrar fuente de información
Modificación de Historia de Usuario (Nro. Y Nombre):	
Usuario: Cliente	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1 semana
Descripción: Permitirá al usuario filtrar por fuente de información el contenido descargado.	
Observaciones:	

Tabla 40. Historia de Usuario 13

Historia de Usuario	
Código: HU-13	Nombre: RF 13. Ver contenido
Modificación de Historia de Usuario (Nro. Y Nombre):	
Usuario: Cliente	Iteración Asignada: 4
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1 semana
Descripción: Permitirá al usuario	
Observaciones:	

Tabla 41. Historia de Usuario 14

Historia de Usuario	
Código: HU-14	Nombre: RF 14. Adicionar contenido al corpus
Modificación de Historia de Usuario (Nro. Y Nombre):	
Usuario: Cliente	Iteración Asignada: 4
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1 semana
Descripción: Permitirá al usuario	
Observaciones:	

Tabla 42. Historia de Usuario 15

Historia de Usuario	
Código: HU-15	Nombre: RF 15. Eliminar contenido
Modificación de Historia de Usuario (Nro. Y Nombre):	
Usuario: Cliente	Iteración Asignada: 4
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1 semana
Descripción: Permitirá al usuario eliminar el contenido descargado en alguna ocasión que ya no le es útil.	
Observaciones:	

Tabla 43. Historia de Usuario 16

Historia de Usuario	
Código: HU-16	Nombre: RF 16. Exportar Contenido como PDF
Modificación de Historia de Usuario (Nro. Y Nombre):	
Usuario: Cliente	Iteración Asignada: 4

Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1 semana
Descripción: Permitirá al usuario exportar el contenido descargado a PDF para poderlo guardar.	
Observaciones:	