Facultad 1

Sistema para la recopilación automática de la información publicada en los sitios web de navegación nacional

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

Autor(es): Gabriel García Hernández
Tutor(es): MSc. Sahilyn Delgado Pimentel
Ing. Claudio Fernández Cabrera
MSc. Waldo Barrera Martínez

La Habana, noviembre de 2021
"Año 63 de la Revolución"

DELARACIÓN DE AUTORÍA

Declaro por este medio que yo, Gabriel García Hernández, con carné de identidad 97022506768, soy el autor principal del trabajo titulado Herramienta para la recopilación automática de la información publicada en los medios de difusión web y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste firmamos el presente documento a los __ días del mes de _____ del año 2021.

Gabriel García Hernández

Firma del Autor

MSc. Sahilyn Delgado Pimentel

Firma del Tutor

Ing. Claudio Fernández Cabrera

Firma del Tutor

MSc. Waldo Barrera Martínez

Firma del Tutor

AGRADECIMIENTOS

Le agradezco a mis padres por haberme apoyado para estudiar la carrera y enseñarme a ser un hombre independiente, integro y solidario.

A mi hermano que aunque este en la distancia se ha encargado de apoyarme en todo momento y preocupandose porque me una al club de los universitarios de la familia.

Agradecerle eternamente a los que se han ido fisicamente y fueron participes de que hoy en dia sea el hombre que soy. Siempre estaran presentes en cada acción que haga.

A mis amistades Yohan, Marylaura, Luis Javier y Daylin por haber sido ademas de grandes compañeros de aula durante todo estos año, ser una fuente de apoyo para la realización de este trabajo de culminación de estudio.

Quiero agradecerle a mis tutores Sahilyn y Claudio por haberme guiado en esta etapa final de mi carrera.

A todos muchas gracias.

RESUMEN

En la presente investigación se propone un sistema de recopilación automática de información de los sitios web de navegación nacional. Se analizaron diferentes fuentes bibliográficas relacionadas con las características y la utilización de herramientas que están enfocadas en la recopilación de información a nivel nacional e internacional. La solución consiste en un sistema capaz de recopilar información de forma automática obtenidas de los sitios web que sean de navegación nacional para el grupo de trabajo de las redes sociales, ya que esta recopilación es realizada actualmente de manera manual. El proceso de desarrollo estuvo guiado por la metodología de software AUP en su versión UCI, seleccionándose como principales tecnologías: el marco de trabajo Laravel 8.0, el lenguaje de programación PHP 8.0, el sistema gestor de base de datos PostgreSQL 13.1 y como herramienta para el modelado Visual Paradigm 10.0. Se plantearon las pruebas de software que deben ser aplicadas al sistema de recopilación automática de información, con el objetivo de demostrar que sea una solución funcional, segura y con un rendimiento adecuado.

Palabras claves: información, sitios web, recopilación automática.

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

Los avances obtenidos en las Tecnologías de la Información y la Comunicaciones (TIC) en la actualidad, se han convertido en un componente esencial de la cotidianidad humana, generando nuevas formas de socialización, educación, producción de conocimiento y acceso a la información; debido a esto se ha manifestado una creciente búsqueda de alternativas de herramientas de conectividad, mayor demanda de dispositivos inteligentes y consumo de contenidos digitales en la web. Dado que las TIC agrupan los elementos y las técnicas usadas en el tratamiento y la transmisión de las informaciones, principalmente de informática, internet y telecomunicaciones, se hace necesario administrarlas de manera rápida y confiable (Linares, Verdecia y Álvarez, 2014).

Debido a los vertiginosos avances que en los últimos años se han dado en el ámbito de las TIC, continuamente existe una mayor preocupación en todo el mundo por incrementar su uso y aprovechamiento y, con ello, conformar nuevos paradigmas sociales en los que se vean mayormente beneficiados todas las sociedades. Aunado a esto, hoy en día, prácticamente en todos los países existen programas nacionales o líneas de acción gubernamentales que buscan incentivar el acceso generalizado entre todos los individuos a este tipo de tecnologías, intentando con ello, disminuir la llamada brecha digital en la que actualmente vivimos, en donde sólo los grupos mejor posicionados, son los que tienen acceso a ellas (Ortí, 2017).

El volumen de información publicada en internet aumenta diariamente debido a las múltiples noticias en diarios, revistas digitales y los sistemas informáticos creados agrupar, procesar, transmitir y diseminar datos que representen información para los usuarios. Estos sistemas comprenden maquinas, personas y/o métodos organizados y pueden ser ejecutados de manera manual o automática (Gonzalez, 2012).

En Cuba se está llevando a cabo un proceso de informatización de la sociedad, donde es cada vez más común que aumente la cantidad de información generada desde la isla con el fin de satisfacer las necesidades de todas las esferas de la sociedad, en su esfuerzo por lograr cada vez más eficacia y eficiencia en todos los procesos y por consiguiente, mayor generación de riqueza y aumento en la calidad de vida de los ciudadanos (Meneses, 2018). Varias serían las vías que se llevarían a cabo mediante este proceso para generar y hacer llegar la información a los usuarios de internet:

- Las revistas digitales: aportan contenidos variados para el consumo de sus usuarios.

- Las redes sociales: son las más usadas para trasmitir informaciones, ya sea por empresas, organizaciones y personas.
- Gobierno electrónico: mejora la información y los servicios que ofrece a los ciudadanos.
- Portales web: se almacenan informaciones de interés para diferentes sectores de la sociedad.

La Universidad de las Ciencias Informáticas (UCI) surge como parte del proceso anteriormente descrito con la idea de servir de soporte a la industria cubana de la informática, formando profesionales capaces de producir aplicaciones y servicios informáticos de calidad. Dentro de esta entidad se encuentra el Centro de Innovación y Desarrollo para la Internet (CIDI), el cual se encarga de proveer soluciones integrales, productos y servicios relacionados con las tecnologías de internet, en función de la defensa de la ideología socialista a través de la red de redes y la *web* (UCI, 2021a).

Los medios internos de difusión nacionales generan numerosa información importante cada día, que necesita ser recopilada por el grupo de trabajo en redes sociales perteneciente a CIDI. La información recopilada permite que luego de un análisis por parte de los especialistas, se puedan establecer tendencias de opinión, hacer modelos predictivos de cómo se ha comportado la publicación de contenidos y de cómo se comportarán en el futuro teniendo en cuenta el análisis de periodos anteriores. Además, les permite predecir qué noticias pueden causar un mayor impacto y detectar a tiempo posibles noticias falsas.

La recopilación de esta información se realiza de forma manual, lo que implica mayor esfuerzo y cantidad de tiempo dedicado a la actividad por los especialistas del grupo, retrasando sensiblemente la capacidad de actuar de forma oportuna sobre los problemas que se detecten al someter esa información a análisis posteriores. El especialista tampoco puede optimizar su tiempo y dedicárselo a otras tareas igual de importantes que componen su contenido de trabajo, de ahí la necesidad de que la información publicada en los sitios de navegación nacional pueda ser obtenida de manera más eficiente.

Aunque existen herramientas que son utilizadas para realizar este trabajo de forma automática, en el grupo no se cuenta con ninguna. Las que se encuentran disponibles actualmente tampoco se pueden configurar para adecuarse al ámbito de la Universidad de las Ciencias Informáticas y algunas de estas son propietarias, otras solo hacen una recopilación parcial de los datos, no tienen incorporada o tienen en modo pago la opción de la automatización y selección de los sitios y páginas web donde se hará el *webscraping*, el cual es un proceso de recolección automático de datos en páginas web, que convierte datos no estructurados en datos estructurados que pueden almacenarse en su computadora local o en database. Además, la forma de clasificar la información no se ajusta a las necesidades del grupo de trabajo en redes sociales, debido fundamentalmente a que extraen de forma parcial y obvian datos que se necesitan (Hillier, 2021).

A partir de la situación problemática anterior, se propone como **problema de investigación:** ¿Cómo recopilar de forma automática contenidos publicados en los sitios web de navegación nacional?
Para dar solución al problema en cuestión se define como **objeto de estudio:** el proceso de recopilación de información que se encuentra enmarcado en el **campo de acción**: la recopilación automática de información de los sitios web de navegación nacional.
Para dar respuesta al problema a resolver se plantea como **objetivo general**: Desarrollar un sistema que permita el proceso de recopilación automática de información publicada en los sitios web de navegación nacional.

Objetivos específicos:
1-Fundamentar los conceptos, características y antecedentes de los sistemas que se utilizan para recopilar información automática de sitios web.
2-Diseño e implementación de la herramienta que permita la recopilación automática de información publicada en los sitios web de navegación nacional.
3-Validación de la herramienta implementada.

Tareas de Investigación:

I. Determinación de los referentes teóricos – metodológicos que sustentan el proceso de recopilación automática de información.

II. Selección de las tecnologías para la implementación del sistema.
III. Diseño del sistema de recopilación de información de los sitios web de navegación nacional.
IV. Implementación del sistema de recopilación de información de los sitios web.
V. Análisis y documentación de las pruebas realizadas al sistema de recopilación de información de los sitios web de navegación nacional.

Después de establecer los elementos del área de la ciencia a incidir y los objetivos fundamentales, se formula como **idea a defender** la creación de un sistema que permita la recopilación automática de la información en los sitios web de navegación nacional contribuirá a disminuir el tiempo y el esfuerzo empleado por los especialistas del grupo de trabajo en las redes sociales para realizar esta tarea.

Para la caracterización del problema y los conocimientos precedentes asociados a la información de los sitios web se utilizaron métodos teóricos y empíricos.

Método Teóricos:

Histórico-lógico: Se utilizó para analizar y estudiar las características actuales, conceptos y evolución de los elementos relacionados con la recopilación automática de la información.

Analítico-sintético: Se realizaron estudios y análisis referente a las herramientas que realizan extracción de información, así como se enunció y describió la técnica que será utilizada para realizar la extracción automática de la información para la aplicación a desarrollar.

Inductivo-deductivo: Se utilizó para desarrollar razonamientos lógicos que permitieron arribar a conclusiones generales a partir de premisas vinculadas al proceso de recopilación automático de la información.

Modelación: Se utilizó para la realización de los diagramas necesarios en el proceso de desarrollo del software, haciendo una representación abstracta de la solución, facilitando así el desarrollo de la misma.

Métodos Empíricos:

- **Observación:** Se aplicó para comprender como funciona el proceso de recopilación automática de la información de sitios web.
- **Entrevista:** Se realizaron entrevistas a trabajadores del centro CIDI, en especial a los del grupo de trabajo en las redes sociales para familiarizarse con el proceso de recopilación manual realizado en el centro, obtener los requisitos funcionales, y recoger criterios y recomendaciones para la creación de la nueva herramienta de recopilación automática de la información en los sitios web de navegación nacional. La entrevista realizada se encuentra en el A Entrevista realizada a especialista del grupo de trabajo en las redes sociales para obtener información sobre el proceso de recopilación de la información por el grupo de trabajo de las redes sociales del centro CIDI.

El presente trabajo está estructurado en 3 capítulos, a continuación, se describe el contenido de estos:

Capítulo 1: Fundamentación teórica sobre el proceso de recopilación automática de la información.

En este capítulo se fundamenta la base teórica de la presente investigación, se realiza un estudio de soluciones similares a nivel global, así como la metodología de desarrollo de software a usar y las herramientas para el desarrollo de la solución propuesta. Se caracterizan además los lenguajes de programación que serán implementados para realizar el sistema de recopilación automática de la información en los sitios web de navegación nacional.

Capítulo 2: Análisis y diseño del sistema de recopilación automática de noticias de los sitios web de navegación nacional.

En este capítulo se realiza el levantamiento de requisitos, en la cual se engloban los requisitos funcionales y no funcionales de la propuesta de solución. Se obtienen además los artefactos correspondientes a la disciplina de análisis y diseño, aplicando los patrones de diseño definidos como buenas practicas del ciclo de desarrollo del software, el modelo de datos, el diagrama de clases de diseño y el patrón arquitectónico que da soporte a la propuesta de solución.

Capítulo 3: **Implementación y prueba de un sistema de recopilación automática de noticias de los sitios web de navegación nacional.**

En este capítulo se evalúa el grado de calidad y fiabilidad de la propuesta de solución. Esto se lleva a cabo mediante la validación del diseño a través de las métricas: tamaño operacional de las clases y relación entre clases. Se aplican las pruebas internas y pruebas de liberación que propone la metodología que guía el desarrollo de la solución, con el fin de verificar la calidad del producto antes de entregárselo al cliente.

Como **posible resultado** al concluir esta investigación se pretende contar con un sistema de recopilación automática de la información en los sitios web de navegación nacional para ser utilizada por el grupo de trabajo con redes sociales de centro CIDI.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA SOBRE EL PROCESO DE RECOPILACIÓN AUTOMÁTICA DE LA INFORMACIÓN

En este capítulo serán abordados los principales conceptos y las terminologías utilizadas en el soporte teórico de la herramienta que se va a diseñar y desarrollar. Se realizará además un estudio de los homólogos existentes, así como de los lenguajes de programación, herramientas y tecnologías que serán implementadas para el desarrollo del sistema para la recopilación automática de la información en los sitios web de navegación nacional.

1.1 Recopilación de la información

La recopilación de información es el área de la ciencia y la tecnología que trata la identificación, clasificación y estructuración en clases semánticas de información específica encontrada en fuentes no estructuradas, para así permitir su posterior tratamiento automático en tareas de procesamiento de la información (Grishman y Sundheim, 2010).

1.1.1 Métodos de recopilación de la información

Las tecnologías de recopilación de datos e información de la Web han supuesto una revolución en el campo de la interacción usuario-ordenador. Facilitan el acceso a una cantidad impresionante de información que puede ser transformada, convirtiéndose en beneficios para los intereses del usuario final. Con el trascurso del tiempo se han desarrollado un conjunto de métodos y técnicas para la recopilación de información en la Web y su posterior uso (Pérez, 2017).

A continuación, la tabla 1 muestra métodos de recopilación de la información (Pérez, 2017):

Tabla 1 Métodos de recopilación de información (Elaboración propia).

Método	**Descripción**
Manual	Método más primitivo y básico de recopilar los datos de las páginas web, mediante el copiar y pegar (Ctrl+C y Ctrl+V) de los datos seleccionados. Este proceso es lento y por lo general, solo se realiza una vez.
Búsquedas en Internet	Método mediante el cual buscadores web, como Google.com, Yahoo.es y otros, obtienen

	la información asociada a un parámetro de búsqueda para su posterior recopilación.
Programación HTTP	Permite la creación de formas para facilitar el acceso y obtención de la información contenida en la Web. Entre sus principales usos en sitios web encontramos los RSS un formato XML para sindicar o compartir contenido en la Web. Se utiliza para difundir información actualizada frecuentemente a usuarios que se han suscrito. Este método a pesar de ser bastante utilizado, depende totalmente de la información que el sitio desee brindar o compartir y si el sitio presenta o no el uso de RSS, por lo que esta opción no cumple lo requerido para la solución.

Los métodos anteriormente descritos son factibles para la obtención de información como un conjunto o un todo, pero si se desea obtener algunos datos específicos de dicha información, se debería recurrir al método manual para su realización.

1.2 Extracción automática de la información desde la web

Se refiere a la aplicación de algún método que recupere de forma automática la información, con el fin de extraer datos que sea legible para una computadora. Dicha información debe haber sido escrita en lenguaje natural y van desde los artículos de prensa hasta los informes científicos. Este se encuentra caracterizado por los siguientes elementos(Delgado, 2020):

- La información obtenida se devuelve de forma estructurada.
- Ha de establecerse a priori que constituye un hecho/relación.
- Sistemas muy especializados de dominio acotado.

El tratamiento automático de la información facilita a los usuarios la manipulación, evaluación y utilización de grandes cantidades de documentos, tarea que mediante técnicas completamente manuales no se podría realizar. Esta situación se encuentra afectada por la alta disponibilidad de documentos en Internet y la existencia de fuentes de información “en línea”, como por ejemplo las agencias de noticias y periódicos digitales(Delgado, 2020).
La extracción automática de la información implica que se reduzca al mínimo la participación de los usuarios en la recopilación de contenidos desde la red. Además, provoca que el análisis y procesamiento de la misma se realice de una manera más correcta y dinámica.

1.2 Técnicas para la extracción automática de información desde la web

Los métodos de *webscraping* permiten realizar la búsqueda, descarga y procesamiento de información de manera programada y automática. Existe una amplia gama de empresas que utilizan programas de *webscraping*, en español también se conoce como raspado web, para el proceso de obtención de la información contenida en el HTML de una página web. También se usan para realizar diferentes actividades como la investigación en línea, seguimiento de los cambios de datos del sitio web, la extracción de datos desde diferentes sitios web, entre otros. De manera general, en este proceso intervienen los conceptos de *crawler*, *scraper* y *spider*. Aunque no existe una definición precisa para estos términos, hay algunas diferencias en su funcionamiento y los casos en que son usados (Riquelme, Ruiz y Gilbert, 2006). A continuación, se abordan con más detalles los términos mencionados(Pérez, 2017):

Crawler: Recorren los enlaces en la Web usando un sitio de partida y permite crear copias del contenido de los sitios visitados, de manera similar a un motor de búsqueda.

Spider: Conocidos en español como arañas web, permiten iterar a través de los enlaces en las páginas web hasta el nivel de profundidad indicado. Los enlaces son identificados mediante sus etiquetas <a> por lo que es requerido un análisis sintáctico del HTML. Entre las tareas más comunes de los Spider o Arañas Web se encuentran:

- Crear el índice de una máquina de búsqueda.

- Analizar los enlaces de un sitio para buscar links rotos.

- Recolectar información de un cierto tipo, como precios de productos para recopilar un catálogo o el texto de publicaciones o noticias en la Web para su posterior análisis.

- Indexar páginas web, en tareas de mantenimiento comprobando enlaces o validando el código HTML.

- Reunir tipos específicos de información procedente de páginas web, como es el caso de cosechar direcciones de correo electrónico para enviar correos basura.

Scraper: Realizan la extracción de información de sitios específicos, buscando expresiones regulares, palabras clave, elementos, atributos, entre otros.

Estos procesos antes mencionados tienen que trabajar en conjunto para llevar a cabo la automatización de la búsqueda y procesamiento de la información presente en los sitios web, para de esta manera darle origen a la aplicación de *webscraping.* El funcionamiento de esta técnica consiste en obtener los datos desde las páginas web, insertarlos en la base de datos de la entidad y monitorizar toda la operación realizada. En el Anexo B Representación del funcionamiento del webscraping.

1.3 Estudio de Homólogos

En el mundo se han desarrollado diferentes aplicaciones que recopilan información de manera automatizada. En la investigación se realiza un análisis de alguna de estas soluciones con el fin de obtener una aproximación de las funcionalidades que debe tener la solución. A continuación, en la tabla 2 se detallan sus principales características y se hace una comparación entre estos atreves de aspectos como la funcionalidad, ventajas y desventajas (Octoparse, 2020):

Tabla 2 Relación de homólogos que realizan webscraping (Elaboración propia).

Homólogo	Público	Uso	Ventajas	Desventajas
Diffbot	Desarrolladores y empresas	Es una herramienta que utiliza aprendizaje automático, algoritmos y API públicas para extraer datos de páginas web. Se Puede usar	-Información precisa actualizada. -API (Application Programming	-La salida inicial fue en general bastante complicada, lo que requirió

		para el análisis de la competencia, el monitoreo de precios, analizar el comportamiento del consumidor y muchos más.	Interface o en español interfaz de programación de aplicaciones) confiable. -Permite crear tus propios *bots.*	mucha limpieza antes de ser utilizable. -Puede ser usado de forma gratuita solo por 14 días, el resto del tiempo es por pago.
Parsehub	Analista de datos, comercializadores e investigadores.	Es un software visual que puede usar para obtener datos de la web. Puede extraer los datos haciendo clic en cualquier campo del sitio web. Tiene una rotación de IP que ayudaría a cambiar su dirección IP cuando se encuentre con sitios web agresivos con una técnica anti-scraping.	-Tener un excelente boarding que te ayude a comprender el flujo de trabajo y los conceptos dentro de las herramientas. -Plataforma cruzada, para Windows, Mac y Linux. -No necesita conocimientos básicos de programación para comenzar. -Soporte al usuario de muy alta calidad.	-No se puede importar / exportar la plantilla. -Tiene una integración limitada de javascript (JS) / regex solamente.
Dexi.io	Programadores	Es un web *spider* basado en navegador.	-Fácil de empezar.	La página de ayuda y

		Proporciona tres tipos de robots: extractor, rastreador y tuberías, este último tiene una función de robot maestro, donde un robot puede controlar múltiples tareas. Admite muchos servicios de terceros (solucionadores de captcha, almacenamiento en la nube, etc.) que puede integrar fácilmente en sus robots.	-El editor visual hace que la automatización web sea accesible para las personas que no están familiarizadas con la codificación. -Integración con Amazon S3.	soporte del sitio no cubre todo. -Carece de alguna funcionalidad avanzada.
Mozenda	Empresas negocios que manejen datos en tiempo.	Proporciona una herramienta de extracción de datos que facilita la captura de contenido de la web. También proporcionan servicios de visualización de datos. Elimina la necesidad de contratar a un analista de datos.	-Creación dinámica de agentes. -Interfaz gráfica de usuario es limpia para el diseño de agentes. -Excelente soporte al cliente cuando sea necesario.	-La interfaz de usuario para la gestión de agentes se puede mejorar. -Cuando los sitios web cambian, los agentes podrían mejorar en la actualización dinámica -Solo Windows. -Es por pago.
Import.io	Empresas que buscan una	Es una plataforma de datos web *Software as a*	-Colaboración con un equipo.	-Es necesario reintroducir

	solución de integración en datos web.	*Service* (SaaS). Proporciona un software de *webscraping* que le permite extraer datos de una web y organizarlos en conjuntos de datos. Pueden integrar los datos web en herramientas analíticas para ventas y marketing para obtener información.	-Muy eficaz y preciso cuando se trata de extraer datos de grandes listas de URL. -Rastrear páginas y raspar según los patrones que especificas a través de ejemplos.	una aplicación de escritorio, ya que recientemente se basó en la nube. -Demora un tiempo para comprender cómo usar la herramienta y luego dónde usarla. -Es por pago.

Considerando los elementos analizados en el estudio se concluye que las mismas no satisfacen las necesidades del cliente por las siguientes razones:

- Dexi.io, aunque es fácil de usar y proporciona *robots* que pueden realizar varias tareas a la vez, tiene alguna dificultad técnica aprender a usar cada una de las herramientas que esta emplea, ya que carece de servicio técnico, por lo que es necesario apoyarse los tutoriales y documentación, los cuales están completamente en inglés.
- Parsehub es muy sencillo de usar ya que solo dando clics en las opciones se podrá programarlo que se debe extraer y clasificar, aunque su integración se ve limitada por ser única y exclusivamente a JS y regex, y tiene la opción de trabajar de manera gratuita, aplica la opción de pago en caso de requerir más funcionalidades aplicables, es decir, que si los usuarios desean mejorar las opciones que presenta la herramienta por default deben pagar una suscripción mensual.
- La herramienta Import.IO se destaca por su eficacia a la hora de extraer datos de grandes listas de sitios web, aunque devuelve los datos en tabla lo que resulta engorroso realizar el procesamiento de la información.

-Diffbot tiene una API muy confiable y permite la creación de *bots* por parte de los usuarios, pero tiene problemas a la hora de extraer y procesar la información, por lo que requiere de una revisión por parte del usuario; estas 2 herramientas tienen como limitante además que usan la modalidad de pago para su funcionamiento.

-Quizás Mozenda sea la herramienta que más se acerque a lo necesitado a que tiene un excelente soporte técnico, permite la extracción y visualización de la información, pero aplica la misma modalidad de las 2 herramientas antes mencionadas y estas cifras ascienden a valores significativos.

Aunque todos estos artilugios no satisfacen las necesidades del grupo de redes sociales, serán utilizadas como guía de referencia para el desarrollo de la propuesta de solución (Octoparse, 2020).

1.4 Metodología de Desarrollo de Software

Son un conjunto de técnicas y métodos organizativos que se aplican para diseñar soluciones de software informático. El objetivo de las distintas metodologías es el de intentar organizar los equipos de trabajo para que estos desarrollen las funciones de un programa de la mejor manera posible (Santander, 2020).

1.4.1 Proceso Unificado Ágil (AUP-UCI)

El Proceso Unificado Ágil (Agile Unified Process o según sus siglas AUP) es una versión simplificada del Proceso Unificado de Racional (Rational Unified Process o según sus siglas RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP(Rodríguez, 2015).

AUP-UCI fue creada en la Universidad de las Ciencias Informáticas, como una variante de la AUP, como consecuencia de la no existencia de una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable. A continuación, la figura 1 muestra cómo se manifiestan las fases en esta metodología (Rodríguez, 2015):

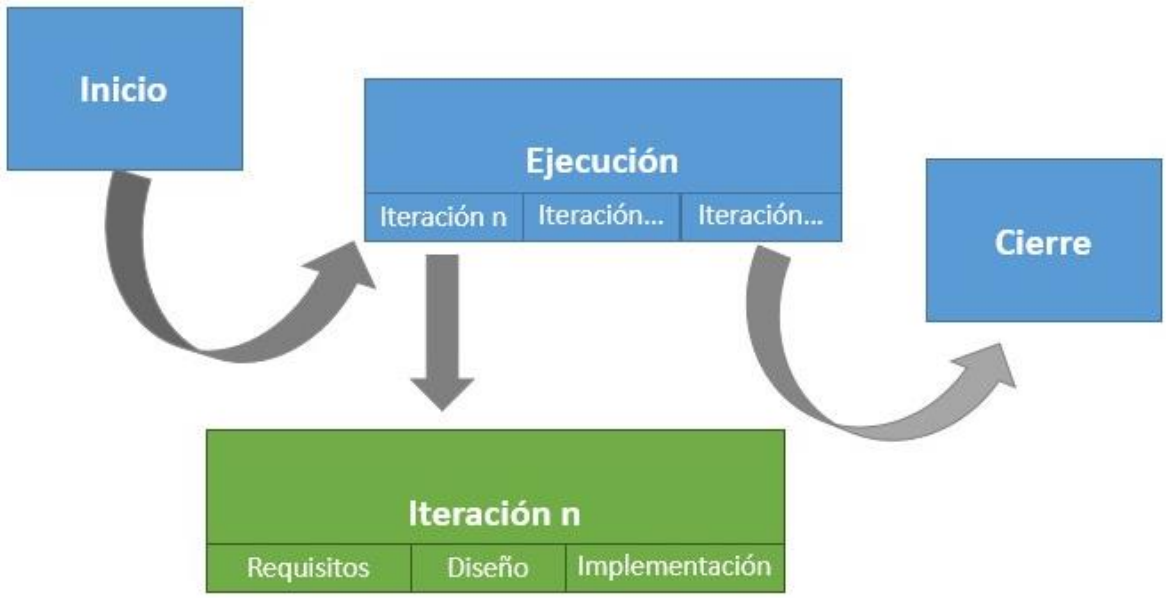


Figura 1 Fases de iteración de la metodología AUP-UCI

Fases:

Inicio: Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.

Ejecución: En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.

Cierre: En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Disciplinas:

1. Modelado de negocio
2. Requisitos
3. Análisis y diseño
4. Implementación

5. Pruebas internas
6. Pruebas de liberación
7. Pruebas de aceptación

Escenarios:

Existen tres formas de encapsular los requisitos Casos de Uso del Sistema (CUS), Historias de usuario (HU) y Descripción de requerimientos por proceso (DRP), los cuales se agrupan en cuatro escenarios para modelar el sistema en los proyectos. A continuación, la figura 2 muestra la representación de estos escenarios:

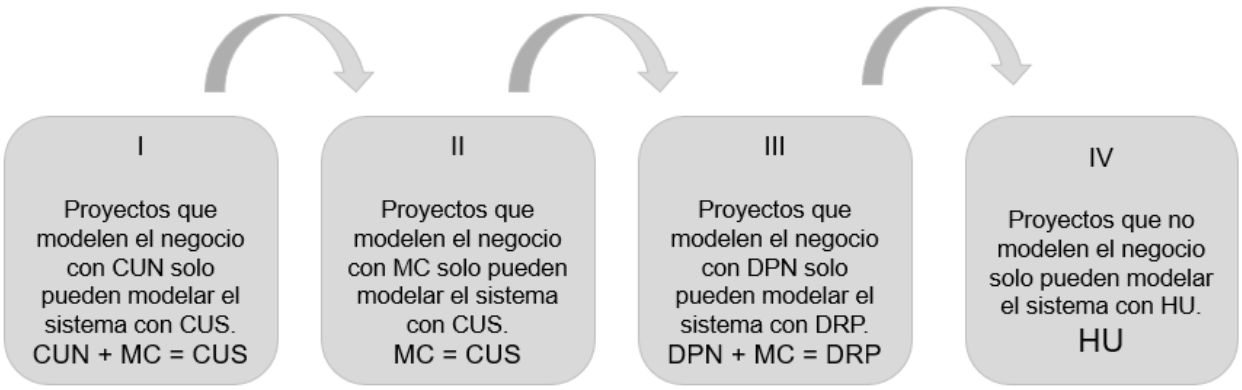


Figura 2 Escenarios en el AUP-UCI (Elaboración propia).

Leyenda:

CUN: Casos de uso del negocio.

MC: Modelo Conceptual.

DPN: Descripción del proceso de negocio.

Luego de analizar los elementos antes expuestos se decidió aplicar para el desarrollo del sistema de recopilación automática de la información en los sitios web de navegación nacional el escenario cuatro. Esta selección está justificada con la presencia del cliente en el desarrollo acompañando al equipo de desarrollo, con el objetivo de tener un tratamiento de requerimientos del sistema bien definidos para de esta manera implementarlos, probarlos y validarlos. Además, el tratamiento de las HU se realiza de una manera dócil y bastante dinámica, permitiendo que puedan ser añadidas, modificadas y eliminadas sin necesitar usar mucho tiempo en el proceso.

1.5 Lenguajes y Herramientas para el modelado

A continuación, se describirán el lenguaje y la herramienta que serán utilizados para la modelación de la herramienta para la recopilación automática de la información publicada en los medios de difusión.

1.5.1 Lenguaje unificado de modelado (UML 2.0)

El lenguaje UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos, etc., hasta la implementación y configuración con los diagramas de despliegue (Orallo, 2012).

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones (Orallo, 2012):

• Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.

• Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.

• Construir: A partir de los modelos especificados se puede construir los sistemas diseñados.

• Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

UML es además un método formal de modelado. Esto aporta las siguientes ventajas (Orallo, 2012):

- Mayor rigor en la especificación.
- Permite realizar una verificación y validación del modelo realizado.
- Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y a la inversa (a partir del código fuente generar los modelos). Esto permite que el modelo y el código estén actualizados, con lo que siempre se puede mantener la visión en el diseño, de más alto nivel, de la estructura de un proyecto.

1.5.2 Herramienta para el modelado Visual Paradigm 10.0

Visual Paradigm ayuda a los equipos de desarrollo de softwares a capturar los requisitos correctos y transformarlos en diseños precisos, lo que ayuda a los desarrolladores a crear el software adecuado según los requisitos. Esta herramienta permite aumentar la calidad del software, a través de la mejora de la productividad en el desarrollo y mantenimiento del software. Aumenta el conocimiento informático de una empresa ayudando así a la búsqueda de soluciones para los requisitos. También permite la reutilización del software, portabilidad y estandarización

de la documentación, además del uso de las distintas metodologías propias de la Ingeniería de Software (Peña, 2016).

1.6 Tecnología para la Implementación de la Herramienta

A continuación, se describirán cada una de las tecnologías a usar para el desarrollo de la herramienta para la recopilación automática de la información publicada en los medios de difusión web de la UCI.

1.6.1 Lenguajes de programación del lado del cliente

HTML 5

HyperText Markup Language (Lenguaje de Marcado de Hipertexto) según sus siglas en inglés, define los nuevos estándares de desarrollo web, rediseñando el código para resolver problemas y actualizándolo así a nuevas necesidades. No se limita solo a crear nuevas etiquetas o atributos, sino que incorpora muchas características nuevas y proporciona una plataforma de desarrollo de complejas aplicaciones web (Garro, 2014).

Define una estructura básica y un código HTML para la definición de contenido de una página web, como texto, imágenes, entre otros y se basa en el referenciación por hipertextos o enlaces entre páginas. Permite a las páginas web almacenar datos localmente en el lado del cliente y operar sin conexión de manera más eficiente. Otorga además un excelente soporte para utilizar contenido multimedia nativamente, ya sean audios y videos. Con respecto al rendimiento y la integración, este proporciona un mejor uso de hardware y una mayor optimización de la velocidad (Delgado 2020).

Este lenguaje de maquetado fue seleccionado debido a ofrecer una gran variedad de herramientas que pueden diseñar y presentar sitios y aplicaciones Web que superen las expectativas de los usuarios. Además, este fue creado para ofrecer un código más limpio, que soluciona el tan aquejado problema de tener que hacer modificaciones en un sitio Web. Su compatibilidad con los navegadores está más que garantizada, por lo que favorece a los sitios creados con este, permitiendo que estos sean accesibles desde cualquier dispositivo que tenga conexión a internet (Pimentel, 2015).

CSS 3

Las Hojas de estilo en cascada (del inglés ***Cascading Stylesheets***) se usan para dar estilo y posicionar visualmente los elementos pertenecientes a una página web. CSS se puede usar, por ejemplo, para cambiar la fuente, el color, el tamaño y el espaciado del contenido, para formar múltiples columnas, añadir animaciones y otros elementos decorativos. Este lenguaje ofrece una nueva gran variedad de opciones para hacer diseños más sofisticados, mejora la accesibilidad de los documentos, reduce la complejidad de su mantenimiento y permite visualizar los mismos documentos en infinidad de diferentes dispositivos(MDN, 2021).

CSS 3 fue escogido para darle el estilo al sitio ya que no necesitas de ningún software costoso para empezar a codificar, permite vincular un solo archivo a diversas páginas, de modo que puedes definir todos los estilos de un sitio web y vincularlos mediante las etiquetas respectivas según corresponda. No sólo puede mejorar tu productividad, sino que ayuda a mejorar el tiempo de respuesta de tu sitio. Ya que todos los estilos se encuentran en un solo archivo CSS, evita que tengas que repetir código en los archivos HTML (Formativa, 2017).

Bootstrap 4

Es un kit de herramientas de código abierto para desarrollos web *responsive* con HTML, CSS y JavaScript. Con él puedes darle forma a tu sitio web a través del uso de sus librerías CSS y JavaScript. Incluye diferentes componentes: ventanas modales, menús, cuadros, botones, formulario, entre otros elementos necesarios para maquetar una página web. Bootstrap permite crear interfaces de usuario limpias y totalmente adaptables a todo tipo de dispositivos y pantallas, sea cual sea su tamaño y presenta una alta compatibilidad con la mayoría de los navegadores existentes(Raiola, 2020).

Se escogió Bootstrap 4 para la realización ya contiene elementos compuestos de HTML 5, CSS 3 y JavaScript para diseñar una web. Permite maquetar por columnas, lo que ayuda a diseñar el sitio de una manera más estructurada. Su documentación es sumamente abarcadora, permitiendo que las consultas a estas aclaren cualquier duda que pudiera surgir.

1.6.2 Lenguaje de programación PHP 8.0

Un lenguaje de programación es lo que le proporciona a un programador, la capacidad de escribir una serie de instrucciones o secuencias de órdenes en forma de algoritmos con el fin de controlar el comportamiento físico o lógico de un sistema informático, de manera que se puedan obtener diversas clases de datos o ejecutar determinadas tareas.

PHP 8.0 es una actualización importante del lenguaje PHP (Hypertext Preprocessor) que contiene nuevos recursos y optimizaciones incluyendo argumentos nombrados, tipos de uniones, atributos, promoción de propiedades constructivas, expresiones match, operador nullsafe, JIT (traducción dinámica) y también mejoras en el sistema de tipos, manejo de errores y consistencia en general (IONOS, 2020). Es un lenguaje de lado del servidor, por lo que se envía justo antes que se envíe la página a través de internet al cliente.

Las principales características de este lenguaje son:

- Está concebido principalmente para la creación de sitios web dinámicos que necesiten del uso de bases de datos.

- Su programación es confiable y segura ya que el código fuente no está al alcance de los navegadores web.
- Fácil conexión con la gran mayoría de los motores de bases de datos.
- Es multiplataforma.

Se decidió usar este lenguaje debido a las siguientes ventajas que brinda este lenguaje para la programación orientada (Tapia, 2018) :

-Lenguaje totalmente libre y abierto.

-Posee una curva de aprendizaje muy baja.

-Los entornos de desarrollo son de rápida y fácil configuración.

-Fácil de instalar: existen paquetes autoinstalables que integran PHP rápidamente.

-Fácil acceso e integración con las bases de datos.

-Posee una comunidad muy grande.

-Es el lenguaje con mayor usabilidad en el mundo.

-Es un lenguaje multiplataforma.

-Completamente orientado al desarrollo de aplicaciones web dinámicas y/o páginas web con acceso a una Base de Datos.

-El código escrito en PHP es invisible al navegador ya que se ejecuta al lado del servidor y los -resultados en el navegador es HTML.

-Posee una versatilidad para la conexión con la mayoría de base de datos que existen en la actualidad.

En el C Uso de los distintos lenguajes de programación orientados a *backend.* se puede evidencia un gráfico que demuestra el uso a nivel global de varios lenguajes de programación a del lado del servidor (*backend* según se le denomina en inglés).

1.6.3 Marco de Trabajo Laravel 8

Un marco de trabajo o framework es un diseño abstracto orientado a objetos para un determinado tipo de aplicación, es un patrón arquitectónico que proporciona una plantilla extensible para un tipo específico de aplicaciones. Consiste en un subsistema expandible de un conjunto de servicios, es un conjunto cohesivo de interfaces y clases que colaboran para proporcionar los servicios de la parte central e invariable de un subsistema lógico (Anglada y Garófalo, 2013).

Laravel es un framework PHP y a su vez uno de los más utilizados y de mayor comunidad en el mundo de Internet. Se caracteriza por (Desarrolloweb, 2015):

- Trabajar con una arquitectura de carpetas avanzada, de modo que promueve la separación de los archivos con un orden correcto y definido, que guiará a todos los integrantes del equipo de trabajo y será un estándar a lo largo de los distintos proyectos.
- Dispone de una arquitectura de clases muy adecuada, que promueve la separación del código por responsabilidades.
- Su estilo arquitectónico es Modelo-Vista-Controlador.
- Un sistema de rutas, mediante las cuales es fácil crear y mantener todo tipo de URLs amistosas a usuarios y buscadores, rutas de API, etc.
- Un sistema de abstracción de base de datos, con un ORM (mapeo objeto-relacional) potente pero sencillo de manejar, mediante el que podemos tratar los datos de la base de datos como si fueran simples objetos.
- Un sistema para creación de colas de trabajo, de modo que es posible enviar tareas para ejecución en background y aumentar el rendimiento de las aplicaciones.
- Varias configuraciones para envío de email, con proveedores diversos.
- Un sistema de notificaciones a usuarios, mediante email, base de datos y otros canales.
- Una abstracción del sistema de archivos, mediante el cual podemos escribir datos en proveedores cloud, y por supuesto en el disco del servidor, con el mismo código.
- Gestión de sesiones.
- Sistema de autenticación, con todo lo necesario como recordatorios de clave, confirmación de cuentas, recordar un usuario logueado, etc.
- La posibilidad de acceder a datos en tiempo real y recibir notificaciones cuando éstos se alteran en la base de datos.

Este frameworks será el usado ya que permite a un desarrollador aprovechar una gran biblioteca de funcionalidad pre programada, que simplifica la construcción de aplicaciones web robustas de forma rápida y minimiza la cantidad de codificación necesaria. Las aplicaciones creadas con Laravel son altamente escalables y tienen bases de código fáciles de mantener. Además, permite añadir funcionalidad a sus aplicaciones sin problemas, gracias al sistema de empaquetado modular de Laravel y a la sólida gestión de dependencias (Kinsta, 2021).

1.6.4 Entorno de Desarrollo Integrado(IDE) PhpStorm 2021.1

Es un IDE multiplataforma creado por la empresa JetBrains que capta el código y comprende su estructura en profundidad, además de ser compatible con todas las funcionalidades del lenguaje PHP para proyectos tanto nuevos como heredados. Proporciona la mejor finalización de código, refactorizaciones, prevención de errores sobre la marcha y más. Permite el uso de tecnologías de front-end (el cual no es más que es aspecto de un sitio) tales como HTML5, CSS, Sass, JavaScript, entre otros, con disponibilidad de refactorizaciones, depuración y pruebas de unidad (JetBrains, 2021).

La asistencia inteligente a la codificación es uno de los elementos más destacables de este IDE ya que posee cientos de inspectores que se encarga de verificar su código a medida que es escrito, analizando el proyecto entero. La compatibilidad con la documentación oficial de PHP, el organizador, reorganizador y formateador de código, los arreglos rápidos y otras funcionalidades, le ayudan a escribir un código limpio y fácil de mantener (JetBrains, 2021).

No solo por las funcionalidades antes descritas que son 100% compatibles con las necesidades del proyecto, sino por la presencia en internet de abundantes bibliografías en forma de texto y video que complementan el uso y aprendizaje de esta popular herramienta, fue seleccionado PhpStorm para desarrollar la solución propuesta en este proyecto.

1.7 Gestor de Base de Datos PostgreSQL 13.1

PostgreSQL es un sistema para gestionar bases de datos de muy alto nivel, completamente de software libre, compatible con cualquier uso, ya sea personal o comercial. Se caracteriza por:

- **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, arreglos, 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 distribuida:** 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:** es capaz de trabajar con funciones internas, que se ejecutan en el servidor, escritas en diversos lenguajes como C, C++, Java, PHP o Python.

PostgreSQL fue elegido como herramienta ya que es el sistema gestor de bases de datos de código abierto más avanzado, multiplataforma y capaz de trabajar con proyectos grandes sin aumentar su complejidad. Su funcionalidad y capacidad de trabajar con mayores cantidades de datos hacen que sea una de las más fiables y populares a nivel mundial.

Conclusiones parciales

Como resultado obtenido en la investigación realizada en este capítulo puede concluirse que:

- Con el estudio realizado a los diferentes sistemas queda acentuada la necesidad del desarrollo de un sistema que permita la recopilación automática de información desde los sitios web de navegación nacional.

- Con el análisis de las tecnologías informáticas se definió la base tecnológica que se utilizará en el desarrollo del sistema, seleccionando a AUP-UCI como metodología para guiar los pasos del desarrollo, UML como lenguaje de representación visual y Visual Paradigm como herramienta CASE para el modelado del sistema. Se escogió PHP como lenguaje para el desarrollo y para el trabajo con base de datos de la herramienta, PostgreSQL. Se definió como entorno de desarrollo integrado PhpStorm y como marco de trabajo, Laravel. Para el maquetado se seleccionó HTML y CSS para darle estilo a la solución.

CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA PARA LA RECOPILACIÓN AUTOMÁTICA DE LA INFORMACIÓN EN LOS SITIOS WEB DE NAVEGACIÓN NACIONAL

En este capítulo se realizará una descripción de las características de la herramienta para la recopilación automática de la información en los medios de difusión web, las cuales son necesarias para comenzar con el proceso de desarrollo de dicha aplicación. Dichas características nos permiten tener una idea objetiva de cómo debería ser el producto a elaborar, además de permitirnos hacer la selección de los requisitos funcionales y no funcionales que deba presentar la herramienta. Se realizará además la modelación de los artefactos necesarios para la implementación del sistema.

2.1 Propuesta de solución

Se propone un sistema de recopilación automática de información en los sitios web de navegación nacional, con el objetivo de almacenar en un único espacio todas las informaciones requeridas por el grupo de trabajo en las redes sociales del centro CIDI, extraídas de distintos medios de difusión web sin depender de la participación de algún trabajador de dicho centro. Este permitirá tener el link, título, resumen, autor, fecha, cuerpo y comentarios, para ello los usuarios deberán escoger algunas las fuentes y categorías establecidas previamente en el sistema, ya que estas tendrán bien definidas las estructuras que compones a el sitio web de que se desee recopilar la información. Para poder usar esta herramienta es necesario que el usuario se autentique; estando dentro de la este podrá crear una nueva tarea para el sistema, modificar cualquiera de las ya existentes, consultar y eliminar las mismas. Todo esto antes descrito le será posible al usuario mediante una interfaz web.

2.2 Especificación de los requisitos de software

El proceso que permite comprobar si se alcanzan o no los objetivos planteados en un proyecto, según lo solicitado por el cliente es denominado como especificación de requisitos. Su función es permitir la correcta comprensión de las tareas haciendo un análisis profundo análisis de la problemática en cuestión, lo que facilita a su vez una mejor identificación de las funcionalidades que serán implementadas(García, García y Vázquez, 2019).

2.2.1 Requisitos funcionales

Tabla 3: Requisitos funcionales [Elaboración propia]

No.	**Nombre**	**Prioridad para el cliente**	**Complejidad**
RF1	Autenticar usuario	Media	Baja
RF2	Mostrar usuario	Media	Media
RF3	Registrar usuario	Media	Baja
RF4	Eliminar usuario	Media	Baja
RF5	Modificar usuario	Media	Media
RF6	Insertar tarea	Alta	Alta
RF7	Mostrar tarea	Media	Media
RF8	Modificar tarea	Alta	Alta
RF9	Eliminar tarea	Baja	Baja
RF10	Listar tareas	Media	Media
RF11	Extraer código HTML de los artículos	Alta	Alta
RF12	Extraer automáticamente los contenidos de los campos del artículo	Alta	Alta
RF13	Guardar código HTML de los artículos	Alta	Alta
RF14	Guardar automáticamente los contenidos de los campos del artículo	Alta	Alta

2.2.2 Requisitos no funcionales

Software

RnF1: Se requiere la instalación de PHP en su versión 8.0.

RnF2: Se requiere la instalación del servidor de base de datos PostgreSQL en su versión 13.1.

RnF3: Se requiere la instalación de Composer en su versión 2.0.

Hardware

RnF4: El servidor de base de datos debe poseer como mínimo un disco duro de 50 GB, 4GB de RAM y un núcleo con velocidad de procesamiento a 2.20 GHZ.

RnF5: El servidor donde se alojará el sistema requiere de al menos 10 GB de disco duro, 4GB de RAM y un núcleo con velocidad de procesamiento a 2.20 GHZ.

Disponibilidad

RnF6: Solo los usuarios autorizados tendrán acceso a la herramienta de recopilación de la información.

Eficiencia

RnF7: La herramienta debe permitir que varios usuarios interactúen a la vez.

RnF8: El tiempo de respuesta al usuario por parte de la herramienta debe ser rápido, no deberá superar los 5 segundos de duración.

Seguridad

RnF9: Realizar salvas de seguridad a la base de datos para prevenir perdida de información.

Interfaz

RnF10: Crear un interfaz amigable con el usuario, que le permita interactuar de manera fácil y dinámica con el sistema.

Confiabilidad

RnF11: El sistema debe permitirle al usuario recuperar contraseña.

Usabilidad

RnF12: Indicar correctamente a el usuario el título de la sección de la página en que se encuentre.

RnF13: Verificar que todos los campos de los formularios con los que interactúe el usuario sean llenados de manera correcta, en caso contrario indicárselo mediante un aviso.

2.3 Historia de Usuario

Las historias de usuario (HU) son parte de un enfoque ágil que ayuda a cambiar el enfoque de escribir sobre los requisitos a hablar sobre ellos**.** Son descripciones cortas y simples de una característica contada desde la perspectiva de la persona que desea la nueva capacidad, generalmente un usuario o cliente del sistema(SCRUM México, 2018).

Tabla 4: HU Gestionar Usuario [Elaboración propia]

Historia de Usuario	
Número: 1	**Usuario:** Cliente
Nombre de historia: Gestionar Usuario	
Prioridad en negocio: Alta	**Riesgo en desarrollo: Medio**
Puntos estimados:	**Iteración asignada: 1**
Programador Responsable: Gabriel García Hernández	
Descripción: Hace alusión a los requisitos funcionales RF1, RF2, RF3, RF4 y RF5. Primeramente, para acceder al sistema es necesario autenticarse, por lo es necesario registrarse en caso de no haberlo hecho anteriormente. Estando dentro del sitio, el usuario tendrá la posibilidad de poder observar su cuenta de usuario y mediante esta tendrá las opciones de modificar los datos personales y eliminar dicha cuenta.	
Validación: Las credenciales de los usuarios deben cumplir con las especificaciones del sistema en cuanto a seguridad.	

Tabla 5: HU Recopilación automática de la información [Elaboración propia]

Historia de Usuario	
Número: 3	**Usuario:** Cliente
Nombre de historia: Recopilación automática de la información	
Prioridad en negocio: Alta	**Riesgo en desarrollo: Medio**
Puntos estimados:	**Iteración asignada: 3**
Programador Responsable: Gabriel García Hernández	
Descripción: Hace alusión a los requisitos funcionales RF11, RF12, RF13 y RF14. Para este proceso la herramienta consultará las tareas activas y realizará *webscraping* para extraer y guardar de estos artículos datos como URL, título, resumen, autor, fecha, cuerpo y comentarios del mismo.	
Validación: Realizar el *webscraping* a las tareas pendientes que lleven más tiempo en este estado.	

Tabla 6: HU Gestionar Tarea [Elaboración propia]

Historia de Usuario	
Número: 2	**Usuario:** Cliente
Nombre de historia: Gestionar Tarea	
Prioridad en negocio: Alta	**Riesgo en desarrollo: Medio**
Puntos estimados:	**Iteración asignada: 2**
Programador Responsable: Gabriel García Hernández	
Descripción: Hace alusión a los requisitos funcionales RF6, RF7, RF8, RF9 y RF10. La interfaz web de la herramienta posee un listado de tareas realizadas anteriormente por el usuario y además una serie de operaciones a las que este puede acceder como es el caso de agregar una nueva tarea, modificar y eliminar una tarea existente. Para la primera operación antes descrita, el usuario deberá crear una tarea, lo cual consiste en declarar mediante un formulario la fuente y categoría de la información que desea recopilar. Para modificar una tarea el usuario solo deberá seleccionar la tarea a modificar y activar dicha opción, y reescribir los datos que desee modificar. Para eliminar sería muy similar, selecciona en el listado la tarea a eliminar, activa dicha opción y esta tarea será eliminada.	
Validación: No se pueden realizar 2 tareas sobre la misma fuente y categoria.	

2.3.1 Estimación de HU

La estimación de esfuerzo de software es una de las áreas más complejas e importantes del desarrollo y la administración de proyectos de software. La dificultad recae en realizar pronósticos de parámetros al inicio del ciclo de vida del proyecto, en donde el alcance del mismo aún no está completamente claro y en donde la incerteza respecto a las funcionalidades que lo compondrán es muy alta. Un pronóstico más preciso al inicio de los proyectos tiene un impacto significativo en la probabilidad de que el proyecto se pueda completar (Morales, 2019). Mediante la estimación podemos organizar como se realizará el desarrollo del software a implementar, así como planificar el tiempo que durará este este proceso, permitiendo así tener un mayor control sobre el mismo.

Estimación por puntos de caso de uso

Este método de estimación de proyectos de software fue desarrollado en 1993 por Gustav Karner de Rational Software y está basado en una metodología orientada a objetos, dándole el nombre de “estimación de esfuerzos con casos de uso”. Surgió como una mejora al método de puntos de función, pero basando las estimaciones en el modelo de casos de uso, producto del análisis de requerimientos. Según su autor, la funcionalidad vista por el usuario (modelo de casos de uso) es la base para estimar el tamaño del software (Orea, 2010). En el Anexo D encontramos la estimación por puntos de caso de uso del sistema a implementar.

2.4 Análisis y diseño

El análisis y diseño de sistemas es un proceso que consiste en estudiar su situación de manera que, al observar cómo se trabaja se pueda detectar si es necesario realizar alguna mejora. Antes de comenzar el desarrollo de cualquier proyecto, se realiza un estudio de sistemas para detectar todos los detalles de la situación actual en la empresa. La información reunida con este estudio sirve como base para crear varias estrategias de diseño(E. Kendall y E. Kendall, 2011).

El análisis y diseño de sistemas se refiere al proceso de examinar la situación de una empresa con el propósito de mejorar con métodos y procedimientos más adecuados. El desarrollo de sistemas tiene dos componentes(E. Kendall y E. Kendall, 2011):

-**Análisis** Es el proceso de clasificación e interpretación de hechos, diagnóstico de problemas y empleo de la información para recomendar mejoras al sistema.

-**Diseño:** Especifica las características del producto terminado.

2.4.1 Diseño arquitectónico

El diseño arquitectónico nos permite definir cómo debe organizarse un sistema y cómo tiene que diseñarse la estructura global de esté. Es el enlace entre el diseño y la ingeniería de requerimientos, que identifica los principales componentes estructurales en un sistema y la relación entre ellos. La salida del proceso de diseño arquitectónico consiste en un modelo arquitectónico que describe la forma en que se organiza el sistema como un conjunto de componentes en comunicación(E. Kendall y E. Kendall, 2011).

Patrón arquitectónico Modelo-Vista-Controlador (MVC)

MVC es un patrón arquitectónico de software que separa una aplicación en tres capas descritas como su acrónimo lo indica. Laravel, así como la mayoría de frameworks en PHP implementan este patrón de diseño en donde cada capa maneja un aspecto de la aplicación. Cada una de sus partes se define de la siguiente manera(Rivera, 2019):

- **Modelo**: Hace referencia a la estructura de datos de la aplicación. Los datos pueden ser transferidos desde la base de datos, una clase, un servicio, u otros, directamente a la vista o ser transformados en el controlador para ser actualizados nuevamente al origen.
- **Vista**: Es la representación de la información en una interfaz de usuario. Por lo general en interfaces no estáticas se representan los datos que vienen directamente del modelo o estos son transformados en un proceso intermedio en el controlador. En vistas estáticas por lo general no hace falta que las vistas sean renderizadas con datos enviados del controlador.
- **Controlador**: Es el lugar en donde se implementa la lógica de la aplicación, los procedimientos, algoritmos y rutinas que hacen que funcione el software. Actúa como interfaz entre los componentes de modelo y vista aplicando las transformaciones y lógica necesarias.

El MVC en Laravel está implementado de la siguiente manera. En una aplicación web, los controladores estarán situados en la carpeta **app/Http/Controllers**, los modelos directamente en **app/Models** y las vistas en **resources/views**(Rivera, 2019).

La figura 1 representa la separación de las clases de la herramienta de recopilación automática de la información de los medios de difusión web según el *framework* Laravel; en el paquete Controlador se encuentran las clases controladoras AutenticacionController, TareasController y RecopilacionController, en el paquete Modelo se encuentra las entidades Usuario, Tarea y Resultado que se corresponden con las tablas que se encuentran en la base de datos de la aplicación, y en el paquete Vista se encuentra list_tareas, mostrar_tarea, form_tarea, login y registro.

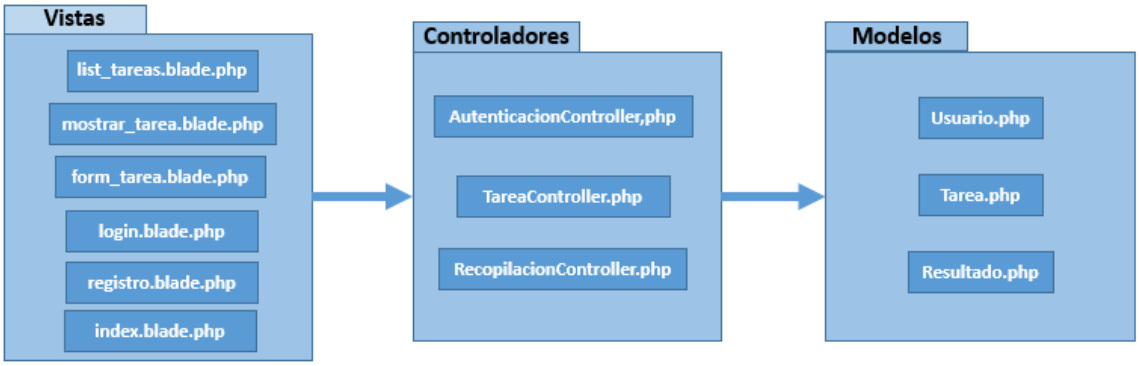


Figura 3: Uso del patrón arquitectónico MVC en la herramienta (Elaboración Propia).

2.4.2 Patrones de diseño

En ingeniería de software, un **patrón de diseño** es una solución general repetible a un problema común en el diseño de software. Un patrón de diseño no es un diseño terminado que se puede transformar directamente en código. Es una descripción o plantilla de cómo resolver un problema que se puede utilizar en muchas situaciones diferentes(Shvets, 2020).

Patrones Grasp

Grasp, General Responsibility Assignment Software Patterns por sus siglas en inglés, describe los principios fundamentales de la asignación de responsabilidades a objetos. Para la herramienta de recopilación automática de la información se implementó dicho patrón de la siguiente manera(LaravelTips, 2021):

- Experto: Este patrón nos dice que la responsabilidad de hacer una acción debe ser asignada a la clase que tiene la información necesaria para realizar dicha acción. En la herramienta, este se evidencia en el modelo Tarea donde se encuentran presentes todas las informaciones necesarias para acceder al sitio.
- Alta cohesión: Asigna una responsabilidad de manera que la cohesión permanezca alta, es decir, asignar a las clases responsabilidades que trabajen sobre una misma área de la aplicación y que no tengan mucha complejidad.
- Bajo acoplamiento: Asigna una responsabilidad de manera que el acoplamiento permanezca bajo, es decir, se basa asignar responsabilidades de forma tal que cada clase se comunique con el menor número de clases. Este patrón se evidencia mediante el controlador Recopilación que se encarga de listar las informaciones guardadas en base de datos y mostrar el contenido de las mismas.

Patrones GoF

El patrón estructural utilizado fue el Decorador que se pone de manifiesto en las vistas realizadas en Laravel mediante el uso de la plantilla index.base.php, la cual contiene el diseño común para toda la herramienta que se realiza a través de herencia entre plantillas.

2.5 Diseño de Clases

2.5.1 Diagramas de clases de diseño

Los diagramas de clases del diseño representan las especificaciones de las clases e interfaces de software. Entre la información que representa se encuentran clases, interfaces con sus operaciones y constantes, métodos, navegabilidad y dependencias. Durante el diseño del sistema, el diagrama de clase del diseño se enfoca a los detalles de la implementación y refleja el funcionamiento de la aplicación en términos lógicos(Martínez y Hernández, 2013).

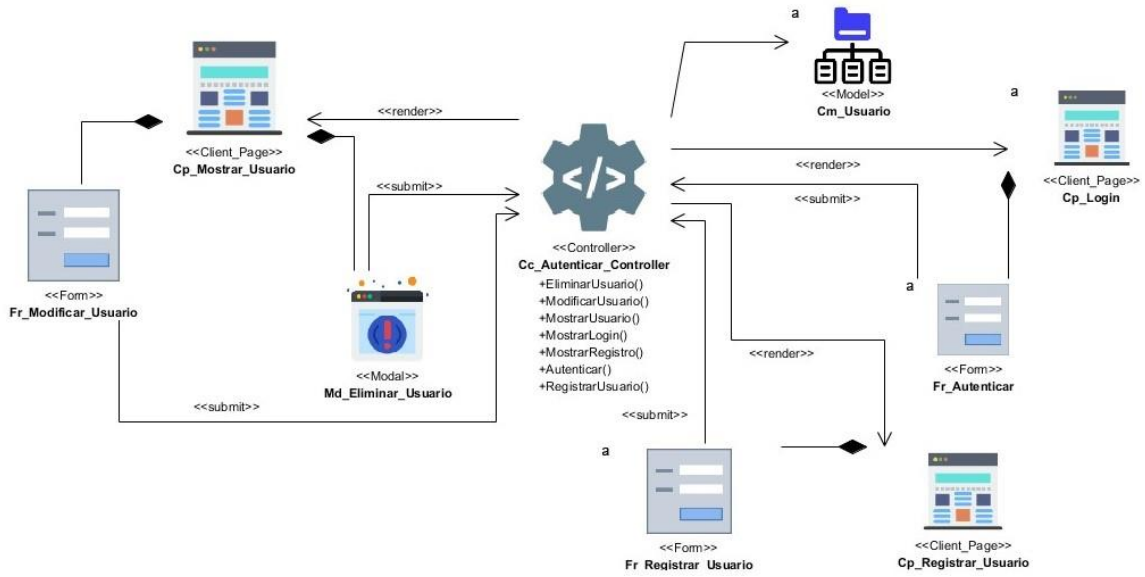


Figura 4: Diagrama de casos de diseño de la HU Gestionar Usuario (Elaboración Propia).

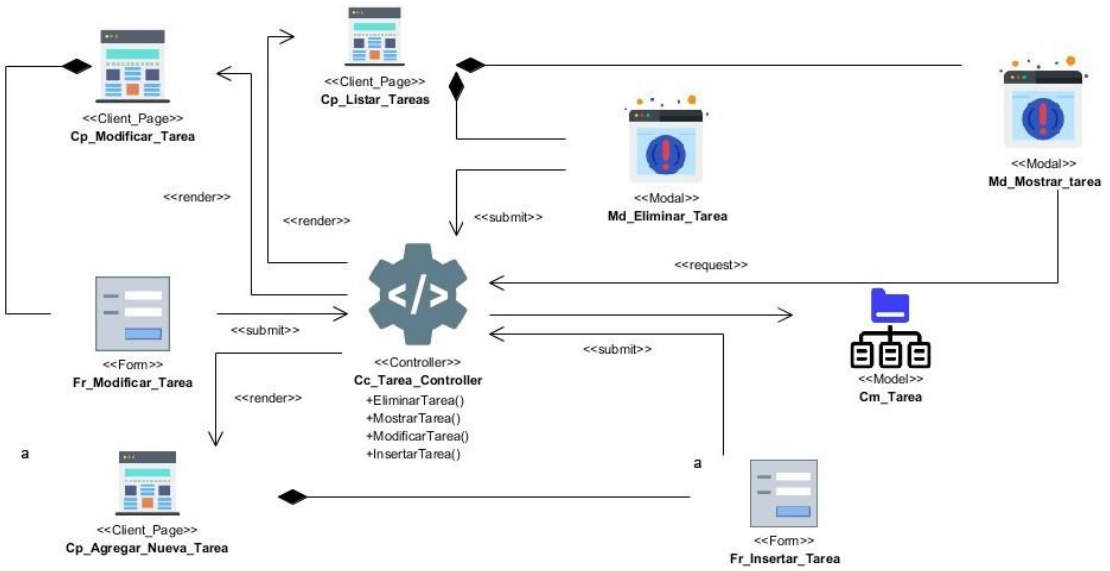


Figura 5: Diagrama de casos de diseño de la HU Gestionar Tarea (Elaboración Propia).

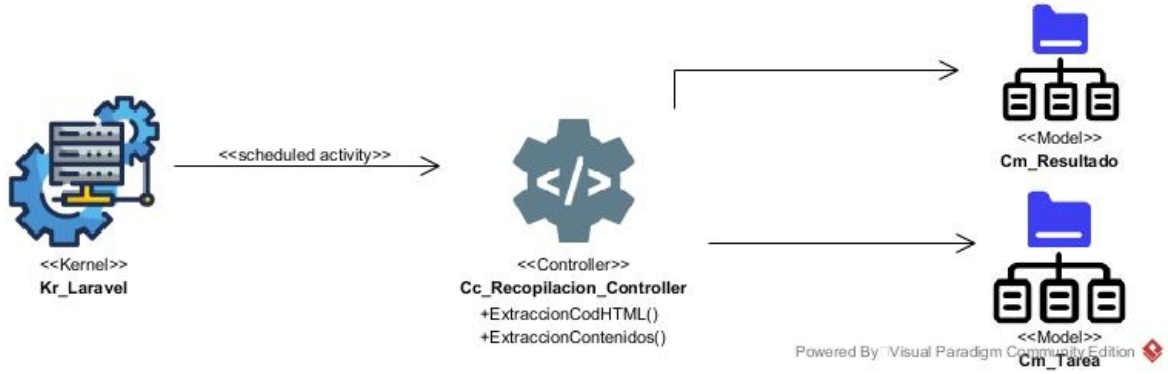


Figura 6: Diagrama de casos de diseño de la HU Recopilación Automática de la información (Elaboración Propia).

2.5.2 Diagrama entidad-relación

Un modelo de datos es un conjunto de conceptos que sirven para describir la estructura de una base de datos: los datos, las relaciones entre los datos y las restricciones que deben cumplirse sobre los datos. Los modelos de datos contienen también un conjunto de operaciones básicas para la realización de consultas (lecturas) y actualizaciones de datos. Además, los modelos de datos más modernos incluyen conceptos para especificar comportamiento, permitiendo especificar un conjunto de operaciones definidas por el usuario(Cabello, 2010).

A continuación, se muestra el modelo de datos que representa físicamente las tablas de la base de datos que contiene el subsistema y las relaciones entre las tablas. La tabla Tarea representa cada tarea que sea programada por un usuario y la tabla Resultado sería todos los elementos recolectados durante el proceso de recopilación automática de la información.

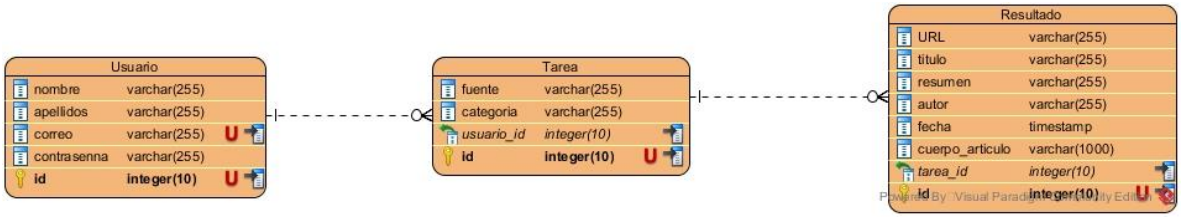


Figura 7: Diagrama Entidad-Relación del sistema para la recopilación automática de la información en los sitios web de navegación nacional (Elaboración propia).

2.6 Diagramas de secuencia

Los diagramas de secuencia modelan el flujo de la lógica dentro del sistema de forma visual, permitiendo documentarla y validarla. Pueden usarse tanto en análisis como en diseño, proporcionando una buena base para identificar el comportamiento del sistema. Usualmente se usan para modelar los escenarios de uso del sistema, describiendo de qué formas puede usarse(IONOS, 2019).

En la figura 6 se representa el Diagrama de Secuencia correspondiente a la HU Gestionar Tarea. Los puntos 1 y 2 corresponden al RF6 (Insertar tarea), el 3 al RF10 (Listar tarea), el 4 al RF7 (Mostrar tarea), el 5 y 6 al RF8 (Modificar tarea) y el 7 al RF9 (Eliminar tarea).

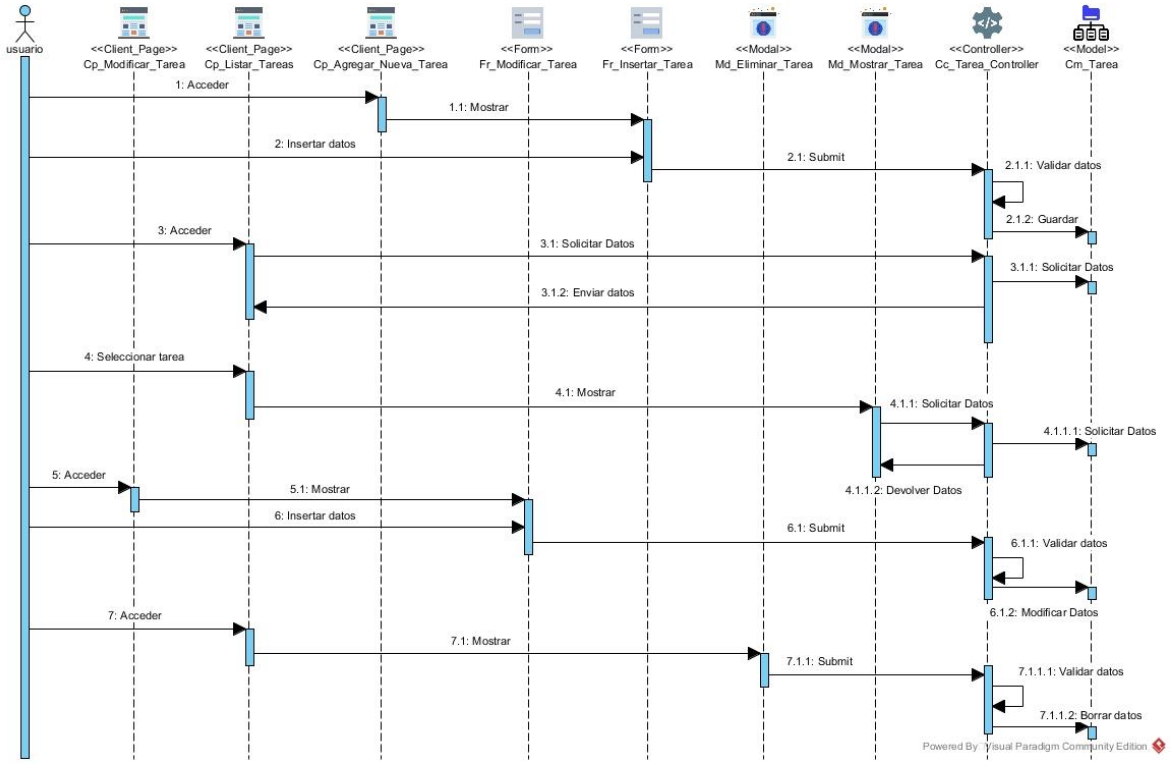


Figura 8: Diagrama de secuencia de Tarea (Elaboración propia).

En la figura 7 se representa el Diagrama de Secuencia correspondiente a la HU Gestionar Usuario. Los puntos 1 y 2 corresponden al RF1 (Autenticar), el 3 y 4 al RF3 (Registrar usuario), el 5 al RF7 (Mostrar usuario), el 6 y 7 al RF8 (Modificar usuario) y el 8 al RF9 (Eliminar usuario).

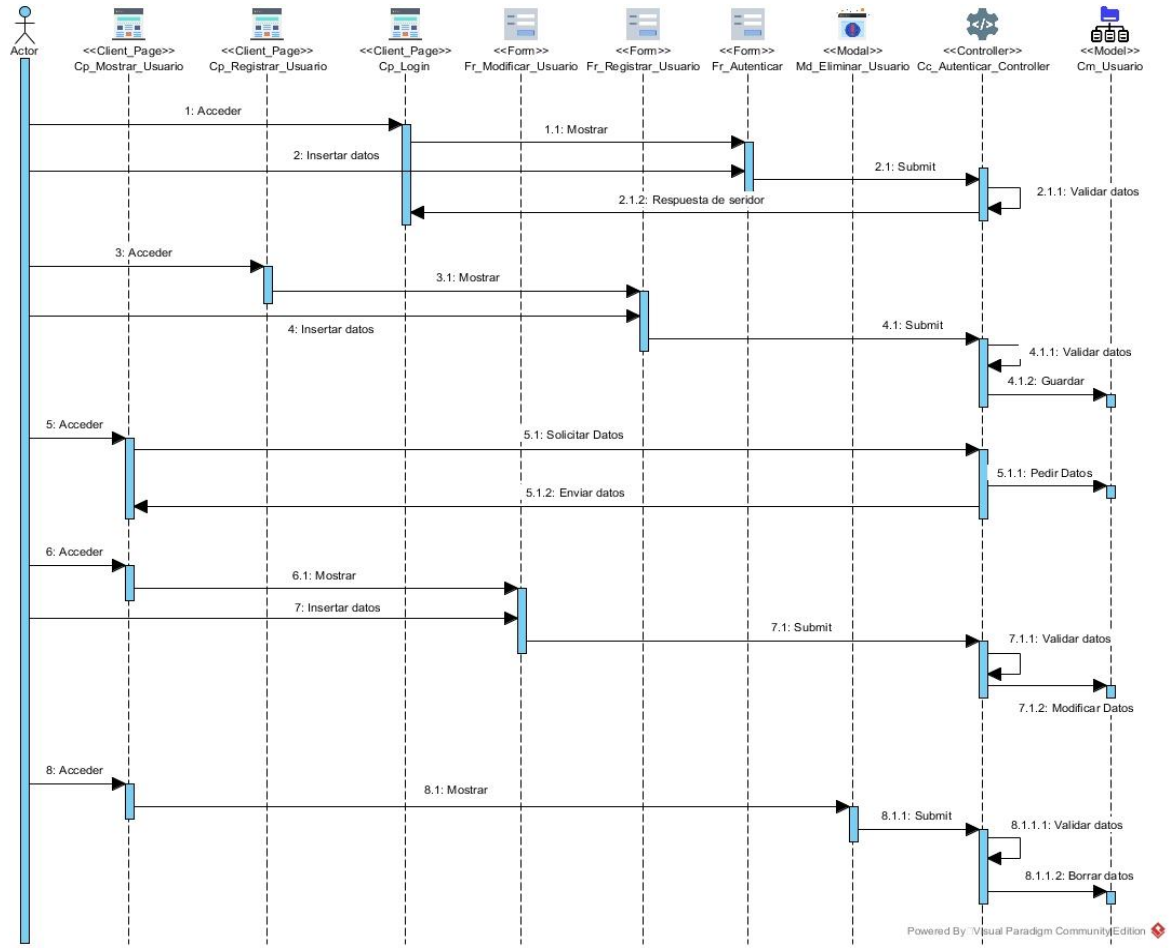


Figura 9: Diagrama de secuencia de Usuario (Elaboración propia).

En la figura 8 se representa el Diagrama de Secuencia correspondiente a la HU Recopilación Automática de la Información. En este se representa el proceso con el cual se llevarían a cabo los requisitos funcionales RF11 (Extraer código HTML de los artículos), RF12 (Extraer automáticamente los contenidos de los campos del artículo), RF13 (Guardar código HTML de los artículos) y RF14 (Guardar automáticamente los contenidos de los campos del artículo).

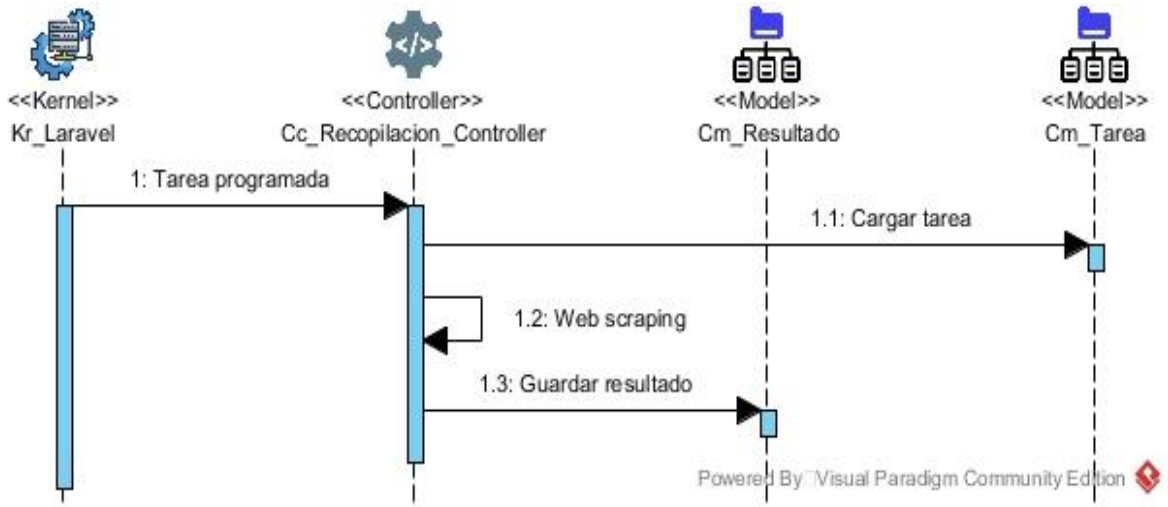


Figura 10: Diagrama de secuencia de la recopilación automáticamente de la información (Elaboración propia).

2.7 Diagrama de despliegue

- Es utilizado para representar la distribución física de los componentes software en los distintos nodos físicos de la red. Se caracteriza por(DiagramasUML, 2021):
- Identificar los nodos en los que trabajará o utilizarán el sistema de información, identificando a su vez agentes externos e internos que interactúen con el sistema.
- Permite representar de forma clara la arquitectura física de la red, así como la distribución del componente software. UML no tiene un tipo de diagramas específico para mostrar la arquitectura de la red, así que se utiliza este tipo de diagrama que cumple efectivamente este cometido, aunque se le suele hacer alguna modificación gráfica.
- Lo más normal es utilizarlo para dar una visión global, pero es posible utilizarlo para representar partes específicas de la implementación.

A continuación, se muestra el diagrama de despliegue de la herramienta de recopilación automática de información automática de la información. El diagrama muestra la disposición física de los nodos que componen el sistema, los componentes que se encuentran en cada nodo, así como la relación de protocolo y puerto entre los nodos.

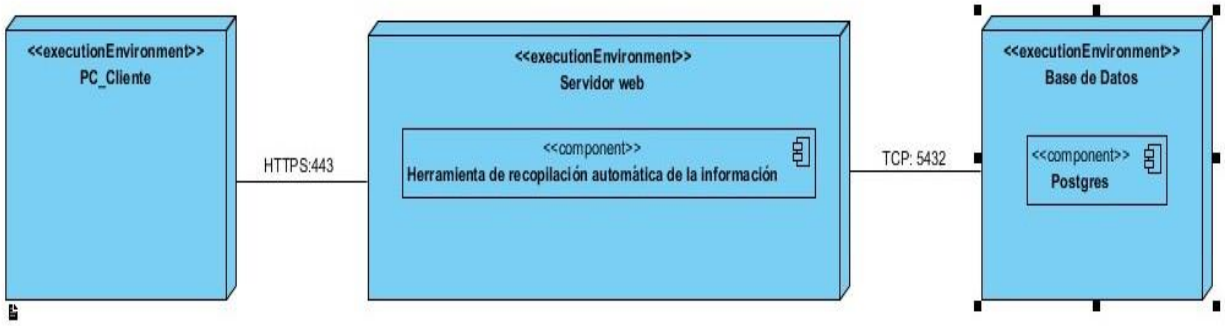


Figura 11: Diagrama de despliegue del sistema de recopilación automática de la información (Elaboración propia).

Conclusiones parciales

En este capítulo, se tuvieron en cuenta varios aspectos como son:

- la identificación de los requisitos funcionales y no funcionales que permitió definir las características y las condiciones que debe presentar el sistema de recopilación automática de la información de los sitios web de navegación nacional.
- Se definió la arquitectura y los patrones de diseño a utilizar, que permitieron establecer las bases para fomentar la reutilización y las buenas prácticas de programación durante la fase de implementación, así como disminuir el impacto de los cambios futuros en el código fuente.
- Se realizó el diagrama de secuencia con el objetivo de conocer el flujo con que se desarrollan las actividades en la aplicación.
- Se elaboró el diagrama de despliegue que permitió identificar la disposición física de los artefactos de la propuesta de solución.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA DE EL SISTEMA PARA LA RECOPILACIÓN AUTOMÁTICA DE LA INFORMACIÓN DE LOS SITIOS WEB DE NAVEGACIÓ NACIONAL

En el presente capítulo se describe el proceso de implementación y de validación desarrollado para evaluar la propuesta de solución. Mediante la definición de las estrategias de pruebas de software se verifica la calidad del resultado de la implementación, mediante los productos de trabajos generados durante las disciplinas de pruebas, exponiendo los resultados obtenido.

3.1 Estándares de codificación

Los estándares de código, son parte de las llamadas buenas practicas o mejores prácticas, estas son un conjunto no formal de reglas, que han ido surgiendo en las distintas comunidades de desarrolladores con el paso del tiempo y las cuales, bien aplicadas pueden incrementar la calidad de tu código, notablemente. Entendemos como estándar de código a un conjunto de convenciones establecidas de ante mano (denominaciones, formatos, etc.) para la escritura de código. Estos estándares varían dependiendo del lenguaje de programación elegido y además varían en cobertura, algunos son más extensos que otros(Merkury, 2017).

Laravel sigue los estándares PSR-1 y PSR-4, los cuales son los siguientes:

- Usa siempre el tag de apertura largo de PHP <?php.
- La declaración del namespace debe estar en la misma línea que <?php. Ejemplo:
  <?php namespace Curso\Http\Controllers;

- Las llaves de apertura de las clases deben ir en la misma línea que el nombre de la clase. Ejemplo:
  class SQLiteConnection extends Connection {

  protected function getDefaultQueryGrammar()
  {
  return $this->withTablePrefix(new QueryGrammar);
  }
  (...)

- Las funciones y estructuras de control deben seguir el estilo de llaves Allman. El estilo Allman define que las llaves de apertura de las estructuras de control deben ir en la línea siguiente. La llave de cierre debe estar al mismo nivel que la de apertura. Y el cuerpo de la estructura debe estar indentado. Ejemplo:

  ```
  (...)
   if ($previous instanceof PDOException)
  {
  $this->errorInfo = $previous->errorInfo;
  }
   (...)
  ```

- Intenta mantener el límite de tus líneas en 80 caracteres. Si es necesario, puedes dividir las líneas en varias e indentarlas, ya que PHP lo soporta perfectamente. Ganarás en facilidad de lectura y comprensión de tu código.
- Declara las propiedades de las clases antes de los métodos.
- Declara los métodos en este orden: public, protected y private
- Utiliza espacios para mejorar la lectura de los operadores. Ejemplo:

  ```
  if (! isset($config[$setting])) {
  continue;
  }
  ```

- Usa comillas sencillas habitualmente y comillas dobles cuando quieras expandir una variable.
- Nombres de las rutas. Usar nombres en línea con las convenciones internas de Laravel: users.index

3.2 Estrategia de pruebas

Una estrategia de prueba del software integra los métodos de diseño de casos de pruebas del software en una serie bien planeada de pasos que desembocará en la eficaz construcción del software. La estrategia proporciona un mapa que describe los pasos que se darán como parte de la prueba indica cuándo se planean y cuándo se dan estos pasos, cuánto esfuerzo, tiempo y recursos consumirán un enfoque estratégico para la prueba del software(canzion23, 2016).

3.2.1 Cronograma de planificación de pruebas

A continuación, la tabla 2 muestra el cronograma que se llevara a cabo para realizar las pruebas a la herramienta de recopilación automática de la información en los medios de difusión web:

Tabla 7 Cronograma de planificación de pruebas (Elaboración propia).

Actividades	Días
Elaborar la estrategia de prueba	Día 1
Diseñar la prueba	Día 2
Realizar el montaje del entorno de prueba	Día 3
Primera iteración de las pruebas	Día 4-5
Corrección de defectos en los artefactos en prueba y actualización de los CP y artefactos de apoyo	Día 6-7-8
Segunda iteración de las pruebas	Día 9
Corrección de defectos en los artefactos en prueba y actualización de los CP y artefactos de apoyo	Día 10-11
Tercera iteración de las pruebas	Día 12
Evaluación de los resultados de las pruebas	Día 13

A cada una de las iteraciones antes descritas le corresponde una HU y existirá una última iteración donde se realizará una prueba de integración, representadas de la siguiente manera:

-HU Gestionar usuario: 1 iteración.

-HU Gestionar tarea: 2 iteración.

-HU Recopilación automática de la información: 3 iteración.

-Prueba de integración: 4 iteración.

3.2.2 Pruebas funcionales

Las pruebas funcionales, también denominadas pruebas de comportamiento se centran en los requisitos funcionales del software. Estas permiten al desarrollador obtener un conjunto de datos de entrada que evalúan todos los requisitos funcionales del sistema. Por esto se denominan pruebas funcionales, donde se suministran datos de entrada y se observa la salida, sin necesidad de conocer el funcionamiento interno del software(Tester, 2019).

De los tipos de pruebas funcionales que se implementaran en para comprobar la funcionalidad de la herramienta son las pruebas de caja negra, la cual es una técnica de pruebas de software en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna de código, detalles de implementación o escenarios de ejecución internos en el software. Estas se centran en las entradas y salidas del sistema, sin preocuparnos en tener conocimiento de la estructura interna del programa de software. Para obtener el detalle de cuáles deben ser esas entradas y salidas, se basa únicamente en los requerimientos de software y especificaciones funcionales(Terrera, 2017).

Dentro de las pruebas de caja negra que existen se implementará la partición equivalente, ya que es una técnica que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. La partición equivalente busca obtener casos de prueba ideales que descubran de forma inmediata clases de errores, reduciendo el número total de casos de prueba a desarrollar. Una clase de equivalencia representa un conjunto de estados válidos o no válidos para condiciones de entrada. Una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica (Pressman, 2013).

A continuación, en la figura 10 se usará el caso de prueba Autenticar usuario asociado a la HU Gestionar usuario para generar el diagrama de caso de prueba referente a este.

Descripción general

RF1 El sistema permite autenticar usuario

Condiciones de ejecución

El usuario debe estar registrado en el sistema con anterioridad.

SC RF1_Autenticar usuario

Escenario	Descripción	Cuenta de correo	Contraseña	Respuesta del sistema	Flujo central
EC 1.1 Autenticar usuario de forma correcta	El sistema autentica un usuario de forma	V qghernandez@estudiantes.uci.cu	V Gabriel.97	El sistema autentica al usuario y le permite el acceso a la herramienta.	1-El usuario accede a la url del portal web y selecciona la opción ¨Iniciar sesión¨. 2- El sistema muestra una interfaz con el formulario de autenticación. 3- El usuario introduce la información y presiona el botón: "Entrar".
EC 1.2 Autenticar usuario de forma incorrecta	El sistema no autentica un usuario de forma incorrecta.	V qghernandez@estudiantes.uci.cu I mmartel@estudiantes.uci.cu I mmartel@estudiantes.uci.cu	I Gabriel.9 V Gabriel.97 I Gabriel.9	El sistema no autentica al usuario y le indica que sus credenciales son incorrectas.	
EC 1.3 Autenticar usuario dejando campos vacíos.	El sistema no autentica un usuario dejando campos obligatorios vacíos.	V qghernandez@estudiantes.uci.cu I I	I V Gabriel.97 I	El sistema no autentica al usuario y le muestra un mensaje indicando que son obligatorios.	

[Las celdas de la tabla contienen V, I, N/A. V indica válido, I indica inválido y N/A indica que no es necesario proporcionar un valor del dato en este caso, ya que es irrelevante.]

Figura 12 Diseño de CP Autenticar usuario (Elaboración propia).

3.2.3 Pruebas de Usabilidad

Este tipo de prueba se refiere a asegurar de que la interfaz de usuario sea intuitiva, amigable y funcione correctamente. Las Listas de Chequeo de Usabilidad (LCU) es una técnica común que ayuda a controlar la ejecución de una actividad de manera rigurosa y que además permite identificar resultados con base en variables relacionadas a la calidad de la ejecución, el cumplimiento, los riesgos o los factores que se quieran medir. Con el seguimiento de una actividad a través de una LCU, se puede obtener la información para tomar medidas correctivas que ayuden a reducir el riesgo de fracaso del proyecto (canzion23, 2016).

Para la realización de esta prueba se empleará la Lista de Chequeo de Usabilidad presente en el E Lista de Chequeo de Usabilidad creada para probar el sistema implementadocreada específicamente para probar los requisitos no funcionales planteados en el capítulo anterior.

3.2.4 Apache Jmeter

En JMeter, un plan de pruebas es una jerarquía de componentes en forma de árbol. Cada nodo del árbol es un componente. A su vez, un componente es una instancia de un tipo de componente en la que quizás se han configurado algunas de sus propiedades.
Los diferentes **componentes** de los que puede constar un plan de pruebas son(Arsys, 2018):

- **Plan de pruebas:** es el tipo de componente que representa la raíz del árbol.
- **Grupo de hilos:** representa un grupo de usuarios. En JMeter cada hilo es un usuario virtual.
- **Controladores (muestras y controladores lógicos):** las muestras realizan peticiones contra la aplicación y los controladores lógicos establecen el orden en que se ejecutan éstos.
- **Elementos de configuración:** establecen propiedades de configuración que se aplican a las muestras que afectan.
- **Aserciones:** comprueban condiciones que aplican a las peticiones que realizan contra la aplicación las muestras que afectan.
- **Oyentes:** recopilan datos de las peticiones que realizan las muestras que afectan.
- **Temporalizador:** añaden tiempo extra a la ejecución de las peticiones que realizan contra la aplicación las muestras que afectan.
- **Elemento preprocesador:** realizan acciones o establecen configuraciones previas a la ejecución de las muestras que afectan.
- **Elemento post-procesado**: realizan acciones o establecen configuraciones posteriormente a la ejecución de las muestras que afectan.

JMeter fue la herramienta elegida para realizar las pruebas de software, específicamente para realizar pruebas de rendimiento, ya que es relativamente ligero y de código abierto, puede escribir un script de prueba más flexible y tiene una alta aceptación por parte de la comunidad de global. Proporciona además capacidades de extensión más avanzadas, lo que le permite definir y ampliar el nuevo soporte de protocolos.

3.2.5 Resultado de las pruebas realizadas

Después de realizadas las pruebas a la 1era iteración se detectaron un total de 7 no conformidades, distribuidas de la siguiente manera:

Prueba funcional:

-Validación:

- Los datos de los usuarios que se registra no se guardan en base de datos.
- Se registraron 2 usuarios con las mismas credenciales.

-Funcionalidad:

- Los datos que el usuario intenta modificar no se guardaban correctamente.
- No se elimina la cuenta del usuario al intentar activar dicha acción.

Prueba no funcional:

-Protección ante errores del usuario:

- Al dejar vacío alguno de los campos del formulario de autenticar usuario, no se mostraba un mensaje indicándole a este que debía llenar los campos de manera obligatoria.
- A introducir mal alguna de las credenciales, no se mostraba mensaje indicándole al usuario que estaba cometiendo un error al realizar dicha acción.

-Accesibilidad:

- En la opción de eliminar cuenta del usuario no se realizaba el mensaje de confirmación de la acción.

De las 7 no conformidades detectadas se les dio solución a todas, de manera que la 1era iteración quedo libre de no conformidades. Además, se concluyó que la HU Gestionar usuario estaba implementada de manera correcta.

Conclusiones parciales

En este capítulo, se tuvieron en cuenta varios aspectos como son:

- Se definieron los estándares de codificación que debe llevarse a cabo para elaborar el sistema.
- Se elaboró las estrategias de pruebas que debería llevar a cabo el sistema.
- Se definieron que pruebas deberían llevarse a cabo para probar el correcto funcionamiento y calidad del sistema.

CONCLUSIONES

Una vez realizada la presente investigación, se concluye que:

- A partir del estudio realizado de los conceptos asociados a los referentes teóricos de la investigación se logró una mayor comprensión de la propuesta de solución y se determinó la necesidad de crear un sistema de recopilación automática de información en los sitios web de navegación nacional.
- La metodología AUP en su variación UCI, el uso de tecnologías y herramientas seleccionadas, permitieron analizar y describir las funcionalidades que se debían de ejecutar, concretando así, en concordancia con las especificaciones del cliente y las características que debería de tener el subsistema a desarrollar.
- Con el desarrollo de la presente investigación se obtuvo una parte del sistema de recopilación automática de información en los sitios web de navegación nacional.
- Se definieron las pruebas de software a realizar, para verificar que los requisitos definidos se implementaron en el sistema.

RECOMENDACIONES

- Integrar esta investigación a otra que conlleve el análisis y clasificación de la información recopilada.

REFERENCIAS BIBLIOGRÁFICAS

ALEXANDER SHVETS, 2020. Design Patterns and Refactoring. [en línea]. [Consulta: 16 octubre 2021]. Disponible en: https://sourcemaking.com.

ANGLADA MARTÍNEZ, R.A. y GARÓFALO HERNÁNDEZ, A.A., 2013. Marco de trabajo para el desarrollo de herramientas orientadas a la gestión e integración de servicios telemáticos de infraestructura en GNU/Linux. *Revista Cubana de Ciencias Informáticas*, vol. 7, no. 2, pp. 157-168. ISSN 2227-1899.

ARKAITZ GARRO, 2014. HTML5. [en línea]. [Consulta: 30 junio 2021]. Disponible en: https://www.arkaitzgarro.com/html5/index.html.

ARSYS, 2018. Por qué elegir PostgreSQL y llevarlo a Cloud. *Blog de arsys.es* [en línea]. [Consulta: 1 julio 2021]. Disponible en: https://www.arsys.es/blog/soluciones/postgresql-servidores/.

BUSTELO RUESTA, C. y AMARILLA IGLESIAS, R., 2001. Gestión del conocimiento y gestión de la información. *revista PH*, pp. 226. ISSN 2340-7565. DOI 10.33349/2001.34.1153.

CABELLO, M.V.N., 2010. *Introducción a las Bases de Datos relacionales*. S.l.: Vision Libros. ISBN 978-84-9983-617-1.

CANZION23, 2016. Estrategias de pruebas del software | Dataprix TI. *Dataprix* [en línea]. [Consulta: 14 noviembre 2021]. Disponible en: https://www.dataprix.com/es/blog-it/canzion23/estrategias-pruebas-del-software.

CASTELLANOS, R.A.G., LAVÍN, M.Y. y LORENZO, L.D.C., 2003. Metodología de la Investigación Científica para las Ciencias Técnicas. , pp. 59.

COSMOS, 2012. Catch Notes, una completa aplicación para tomar y compartir notas, listas y recordatorios en Android. *Xataka Android* [en línea]. [Consulta: 28 junio 2021]. Disponible en: https://www.xatakandroid.com/productividad-herramientas/catch-notes-una-completa-aplicacion-para-tomar-y-compartir-notas-listas-y-recordatorios-en-android.

DELGADO, Y.H., 2020. “Subsistema de recopilación automática de noticias de los medios informativos de la Universidad de las Ciencias Informáticas”. , pp. 80.

DESARROLLOWEB, 2015. Laravel. [en línea]. [Consulta: 30 junio 2021]. Disponible en: https://desarrolloweb.com/home/laravel.

DIAGRAMASUML, 2021. ▷ Diagrama de despliegue. Teoria y ejemplos. *DiagramasUML.com* [en línea]. [Consulta: 16 octubre 2021]. Disponible en: https://diagramasuml.com/despliegue/.

E.KENDALL, K. y E.KENDALL, J., 2011. (PDF) Analisis y.Diseno de Sistemas 8ed Kendall PDF | Ricardo Rubio - Academia.edu. [en línea]. [Consulta: 14 octubre 2021]. Disponible en: https://www.academia.edu/7102592/Analisis_y.Diseno_de_Sistemas_8ed_Kendall_PDF.

ELIZALDE, O., 2019. Triangulación de datos. *lamalditatesis* [en línea]. [Consulta: 17 noviembre 2021]. Disponible en: https://www.lamalditatesis.org/post/triangulacion-de-datos.

FORMATIVA, A., 2017. Definición, usos y ventajas del lenguaje CSS3. *Aula Formativa* [en línea]. [Consulta: 16 noviembre 2021]. Disponible en: https://blog.aulaformativa.com/definicion-usos-ventajas-lenguaje-css3/.

GARCÍA PEÑALVO, F.J., GARCÍA HOLGADO, A. y VÁZQUEZ INGELMO, A., 2019. REQUISITOS INGENIERÍA DE SOFTWARE I - PDF Free Download. [en línea]. [Consulta: 14 octubre 2021]. Disponible en: http://docplayer.es/215941840-Requisitos-ingenieria-de-software-i.html.

GONZALEZ-LONGATT, F.M., 2012. Introducción a los Sistemas de Información: Fundamentos. , pp. 7.

GRISHMAN, R. y SUNDHEIM, B., 2010. Message Understanding Conference- 6: A Brief History. *COLING 2010 Volume 1: The 16th International Conference on Computational Linguistics* [en línea]. S.l.: s.n., [Consulta: 15 noviembre 2021]. Disponible en: https://aclanthology.org/C96-1079.

GUSTAVO TERRERA, 2017. Pruebas de Caja Negra y un enfoque práctico. *Testing Baires* [en línea]. [Consulta: 14 noviembre 2021]. Disponible en: https://testingbaires.com/pruebas-caja-negra-enfoque-practico/.

HENRÍQUEZ MIRANDA, C., 2014. Modelo de extracción de información desde recursos web para aplicaciones de la planificación automática. *Prospectiva*, vol. 10, no. 2, pp. 74. ISSN 22161368, 16928261. DOI 10.15665/rp.v10i2.236.

HERNÁNDEZ SAMPIERI, R., FERNÁNDEZ COLLADO, C. y BAPTISTA LUCIO, P., 2014. *Metodología de la investigación*. México: McGraw Hill Interamericana. ISBN 978-1-4562-2396-0.

IONOS, 2019. Diagramas de secuencia: mostrar interacciones con UML. *IONOS Digitalguide* [en línea]. [Consulta: 16 octubre 2021]. Disponible en: https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagramas-de-secuencia/.

IONOS DIGITALGUIDE, 2020. PHP 8: todo lo que debes saber sobre la nueva actualización. *IONOS Digitalguide* [en línea]. [Consulta: 28 junio 2021]. Disponible en: https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/php-8/.

JETBRAINS, 2021. PhpStorm: el IDE rápido e inteligente para programación en PHP de JetBrains. *JetBrains* [en línea]. [Consulta: 30 junio 2021]. Disponible en: https://www.jetbrains.com/es-es/phpstorm/.

KINSTA, 2021. El Framework PHP Laravel - Construcción de Aplicaciones Web para Todos. *Kinsta* [en línea]. [Consulta: 30 junio 2021]. Disponible en: https://kinsta.com/es/base-de-conocimiento/que-es-laravel/.

LARAVEL TIPS, 2021. Aplica mejores prácticas siguiendo los patrones GRAPS. *Laravel Tip* [en línea]. [Consulta: 16 octubre 2021]. Disponible en: https://www.laraveltip.com/aplica-mejores-practicas-siguiendo-los-patrones-graps/.

LINARES-PONS, N., VERDECIA-MARTÍNEZ, E.Y. y ÁLVAREZ-SÁNCHEZ, E.A., 2014. Tendencias en el desarrollo de las TIC y su impacto en el campo de la enseñanza. *Revista Cubana de Ciencias Informáticas*, vol. 8, no. 1, pp. 71-78. ISSN 2227-1899.

MARTÍNEZ, M.B.B. y HERNÁNDEZ, M.M.R., 2013. DIAGRAMA DE CLASE. , pp. 24.

MDN, 2021. CSS - Aprende sobre desarrollo web | MDN. [en línea]. [Consulta: 30 junio 2021]. Disponible en: https://developer.mozilla.org/es/docs/Learn/CSS.

MENESES, Y.P., 2018. El proceso de informatización de la sociedad cubana es un hecho. , pp. 3.

MERKURY, 2017. Estándares de codificación - ¡Mejora tu código! *Ohmyroot!* [en línea]. [Consulta: 13 noviembre 2021]. Disponible en: https://www.ohmyroot.com/buenas-practicas-legibilidad-del-codigo/.

MIRANDA, CARLOS HENRIQUE y GUZMÁN, JAIME ALBERTO, 2012. Modelo de extracción de información desde recursos web para aplicaciones de la planificación automática. *IONOS Digitalguide* [en línea]. [Consulta: 27 junio 2021]. Disponible en: https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-el-web-scraping/.

MORALES MATURANA, F.I., 2019. Estimación de esfuerzo en proyectos de software a partir de historias de usuario. , pp. 124.

OCTOPARSE, 2020. Los 30 Mejores Software Gratuitos de Web Scraping en 2021. *Octoparse* [en línea]. [Consulta: 16 noviembre 2021]. Disponible en: https://www.octoparse.es/blog/30-mejores-software-gratuitos-de-web-scraping.

ORALLO, E.H., 2012. El Lenguaje Unificado de Modelado (UML). , pp. 6.

OREA, V., 2010. ESTIMACIÓN DE PROYECTOS DE SOFTWARE CON PUNTOS DE CASOS DE USO. , pp. 10.

ORTÍ, C.B., 2017. LAS TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN (T.I.C.). , pp. 7.

PEÑA, D.M., HERNÁNDEZ, L.R.B., CORNELIO, O.M. y OSVIEL RODRIGUEZ VALDÉS, 2016. Extensión de la herramienta Visual Paradigm for UML para la evaluación y corrección de Diagramas de Casos de Uso. [en línea], [Consulta: 28 junio 2021]. DOI 10.13140/RG.2.1.3813.5287. Disponible en: http://rgdoi.net/10.13140/RG.2.1.3813.5287.

PÉREZ, A.V., 2017. Herramienta para recopilar la información en las noticias publicadas en los sitios web de internet. *Serie Científica de la Universidad de las Ciencias Informáticas* [en línea], vol. 10, no. 5. [Consulta: 15 noviembre 2021]. ISSN 2306-2495. Disponible en: https://publicaciones.uci.cu/index.php/serie/article/view/118.

PIMENTEL, S., 2015. 6 beneficios de HTML5. *Blog Netcommerce* [en línea]. [Consulta: 16 noviembre 2021]. Disponible en: https://info.netcommerce.mx/6-beneficios-de-html5/.

PRESSMAN, R.S., 2013. *Ingeniería del software: un enfoque práctico* [en línea]. S.l.: s.n. [Consulta: 14 noviembre 2021]. ISBN 978-1-4562-1836-2. Disponible en: http://www.ingebook.com/ib/NPcd/IB_BooksVis?cod_primaria=1000187&codigo_libro=4272.

RAIOLA, 2020. Bootstrap 4: Qué es, cómo instalarlo en tu web y cómo se utiliza. [en línea]. [Consulta: 30 junio 2021]. Disponible en: https://raiolanetworks.es/blog/bootstrap/.

RIQUELME, J.C., RUIZ, R. y GILBERT, K., 2006. Minería de Datos: Conceptos y Tendencias. , vol. 10, no. 29, pp. 9.

RIVERA, D., 2019. Patrón MVC en laravel. [en línea]. [Consulta: 14 octubre 2021]. Disponible en: https://blog.pleets.org/article/mvc-en-laravel.

RODRÍGUEZ, T., 2015. Metodología de desarrollo para la Actividad productiva de la UCI. , pp. 16.

SÁNCHEZ, JUAN FRANCISCO, 2018. Pruebas de rendimiento con JMeter. Ejemplos básicos. *SDOS* [en línea]. [Consulta: 30 junio 2021]. Disponible en: https://www.sdos.es/blog/pruebas-de-rendimiento-con-jmeter-ejemplos-basicos.

SANTANDER, U., 2020. Metodologías de desarrollo de software: ¿qué son? *Becas Santander* [en línea]. [Consulta: 16 noviembre 2021]. Disponible en: https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html.

SCRUM MÉXICO, 2018. Historias de Usuario, Escritura, Definición, Contexto y Ejemplos — SCRUM MÉXICO. [en línea]. [Consulta: 14 octubre 2021]. Disponible en: https://scrum.mx/informate/historias-de-usuario.

TAPIA, N., 2018. Ventajas y desventajas del lenguaje PHP » BaulPHP. *BaulPHP* [en línea]. [Consulta: 16 noviembre 2021]. Disponible en: https://www.baulphp.com/ventajas-y-desventajas-del-lenguaje-php/.

TESTER, 2019. Pruebas funcionales / No funcionales: ¿Qué son y para qué sirven? *Tester House* [en línea]. [Consulta: 14 noviembre 2021]. Disponible en: https://testerhouse.com/teoria-testing/pruebas-funcionales/.

UCI, 2021a. Centro de Innovación y Desarrollo para Internet (CIDI) | Universidad de las Ciencias Informáticas. *Potal Web de la Universidad de las Ciencias Informáticas* [en línea].

[Consulta: 15 noviembre 2021]. Disponible en: https://www.uci.cu/investigacion-y-desarrollo/centros-de-desarrollo/centro-de-innovacion-y-desarrollo-para-internet.

UCI, 2021b. Misión | Universidad de las Ciencias Informáticas. [en línea]. [Consulta: 29 junio 2021]. Disponible en: https://www.uci.cu/universidad/mision.

WILL HILLIER, 2021. What Is Web Scraping? [A Complete Step-by-Step Guide]. [en línea]. [Consulta: 15 noviembre 2021]. Disponible en: https://careerfoundry.com/en/blog/data-analytics/web-scraping-guide/.

ANEXOS

A Entrevista realizada a especialista del grupo de trabajo en las redes sociales para obtener información sobre el proceso de recopilación de la información por el grupo de trabajo de las redes sociales del centro CIDI.

1- ¿Cuál es la función del grupo de trabajo en las redes sociales?

2- ¿Qué tipo de información es de interés para este grupo?

3- ¿Cómo se realiza el proceso de recopilación de datos en el grupo?

4- ¿De qué sitios de navegación nacional el grupo recopila información?

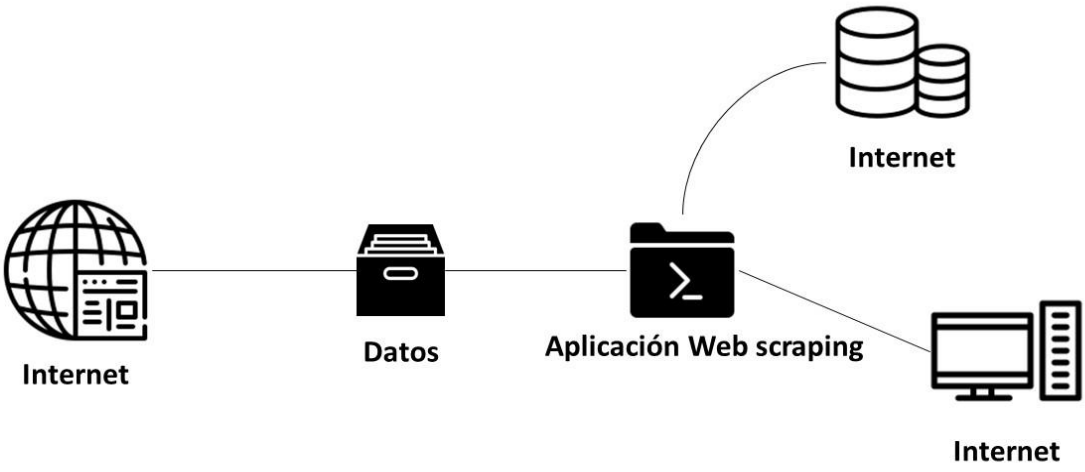
5- ¿Cómo se clasifica la información que es recopilada?

6- ¿Cuál es el estado actual de este proceso? ¿Se realiza de manera manual o de forma automática?

7- ¿Cuánto tarda un especialista del grupo en realizar el proceso de recopilación de información?

8- ¿Cómo consideraría usted que deba funcionar un sistema de recopilación automática de la información para satisfacer las necesidades del grupo?

B Representación del funcionamiento del webscraping



C Uso de los distintos lenguajes de programación orientados a *backend.*

D Estimación de HU del Sistema

Historia de Usuario	Iteración	Mejor Caso	Peor Caso	Resultado
Gestionar Usuario	1	4 días	8 días	6 días
Gestionar Tarea	2	6 días	10 días	8 días
Recopilación automática de la información	3	10 días	15 días	13 días

E Lista de Chequeo de Usabilidad creada para probar el sistema implementado

Elementos definidos por la metodología				
No	**Indicador a evaluar**	**Evaluación**	**NP**	**Observación**
Visibilidad del sistema				
1	¿La página refleja la identidad de la empresa logos, compañía...)?			
2	¿Cada pantalla empieza con un título que describe su contenido?			
3	¿Los enlaces del menú se resaltan cuando se seleccionan?			
4	¿Los iconos que aparecen se identifican claramente con lo que representan?			
5	¿El nombre de los enlaces es el mismo que el título de la página a la que dirige?			
6	¿Los títulos de las páginas, tablas e imágenes son descriptivos y distintivos?			
Lenguaje común entre sistema y usuario				
7	¿El lenguaje es simple, con un tono adecuado?			
8	¿La información que se presenta en la aplicación es fácil de entender y memorizar?			

9	¿Utiliza los conceptos establecidos para las funciones estándar? (“buscar" para las búsquedas, etc.)			
10	¿Evita el lenguaje técnico: términos informáticos o propios de Internet?			
Libertad y control por parte del usuario				
11	¿Existe una manera lógica de acceder a páginas relacionadas o a otras secciones?			
12	¿Tras una acción relevante hay una opción de vuelta atrás?			
13	¿Si una acción tiene consecuencias, el sistema proporciona información y pide confirmación antes de continuar?			
14	¿En las páginas internas hay un acceso a la página de inicio en una zona visible y reconocible?			
15	¿El sitio evita que los usuarios se registren de manera innecesaria?			
16	¿La interfaz de búsqueda está ubicada donde los usuarios esperan encontrarla (en la parte superior derecha de la página)?			
Estética y diseño minimalista				
17	¿Cumple el sitio con el principio de usabilidad de			

	realizar las operaciones con un máximo de tres click?			
18	¿Existe suficiente contraste entre el color del fondo y el del texto?			
19	¿Los tipos y tamaños de letra son legibles y distinguibles?			
20	¿Añade color de fondo a los div que llevan imagen de fondo? (Para los usuarios que desactivan las imágenes, desaparece el contraste entre texto y fondo, convirtiéndose en texto ilegible.)			
21	¿El uso de los colores es moderado?			
22	¿Se usan frases breves y concisas: que resuman los puntos clave y vayan al grano?			
Prevención de errores				
23	¿Existe suficiente espacio entre los elementos de acción (links, botones, etc.) para prevenir que el usuario haga click en el elemento incorrecto?			
24	¿Se dan indicaciones para completar campos problemáticos?			
25	¿Los botones de acción, (tales como “Enviar”) siempre son invocados por el usuario y no automáticamente invocados por el sistema			

	cuando el último campo de un formulario ha sido lleno?			
26	¿El espacio entre los campos del formulario es suficiente como para distinguirlos unos de otros?			
Ayuda y documentación				
27	¿El mensaje de error permite volver a la situación anterior?			
28	¿La política de privacidad del sitio es fácil de encontrar, especialmente esas páginas que piden información personal?			
29	¿Cuándo existen múltiples pasos en una tarea, el sitio muestra todos los pasos que deben ser completados y provee una retroalimentación al usuario indicándole la posición actual en toda la ruta de la tarea?			
30	¿La funcionalidad de los controles para nuevos dispositivos es exactamente la misma que para los otros dispositivos?			
Flexibilidad y eficiencia de uso				
31	Se defina de manera correcta gráficos y tablas utilizando atributos (leyendas, unidades de medida, etc.)			
32	¿Las partes o secciones más importantes de los sitios son accesibles desde la página de inicio?			

34	¿Existen aceleradores, accesos rápidos a operaciones frecuentes?			
34	¿Se implementen validaciones antes de que el usuario envíe información?			

F Interfaces del sistema de recopilación automático de la información en los sitios web de navegación nacional

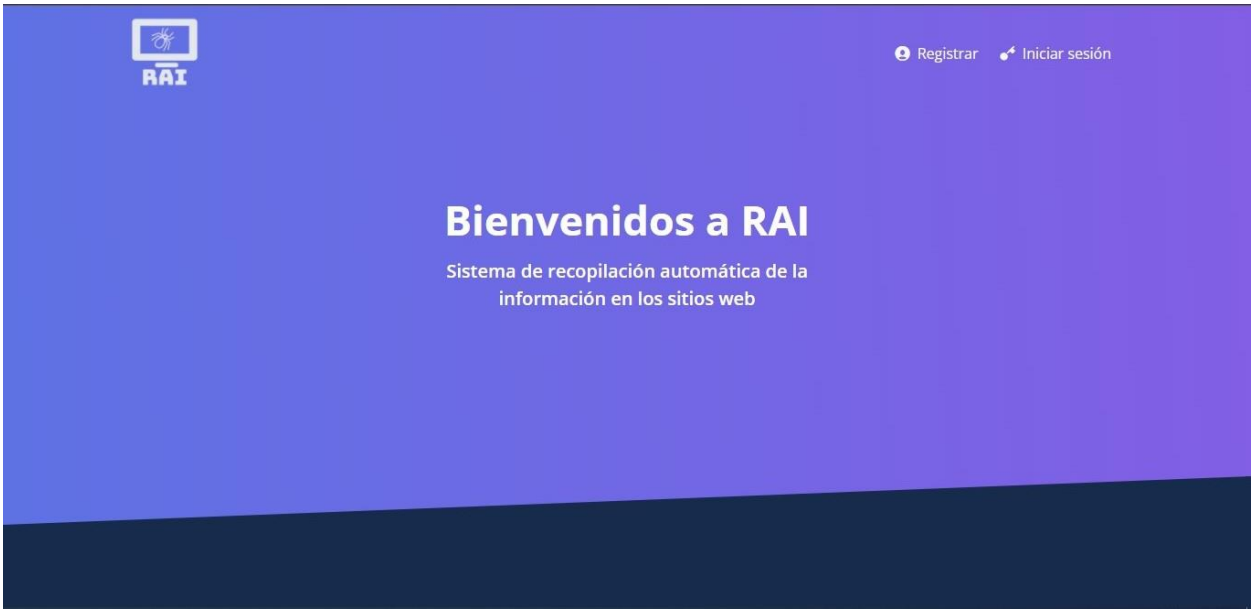


Figura 13 Interfaz de bienvenida del sistema.

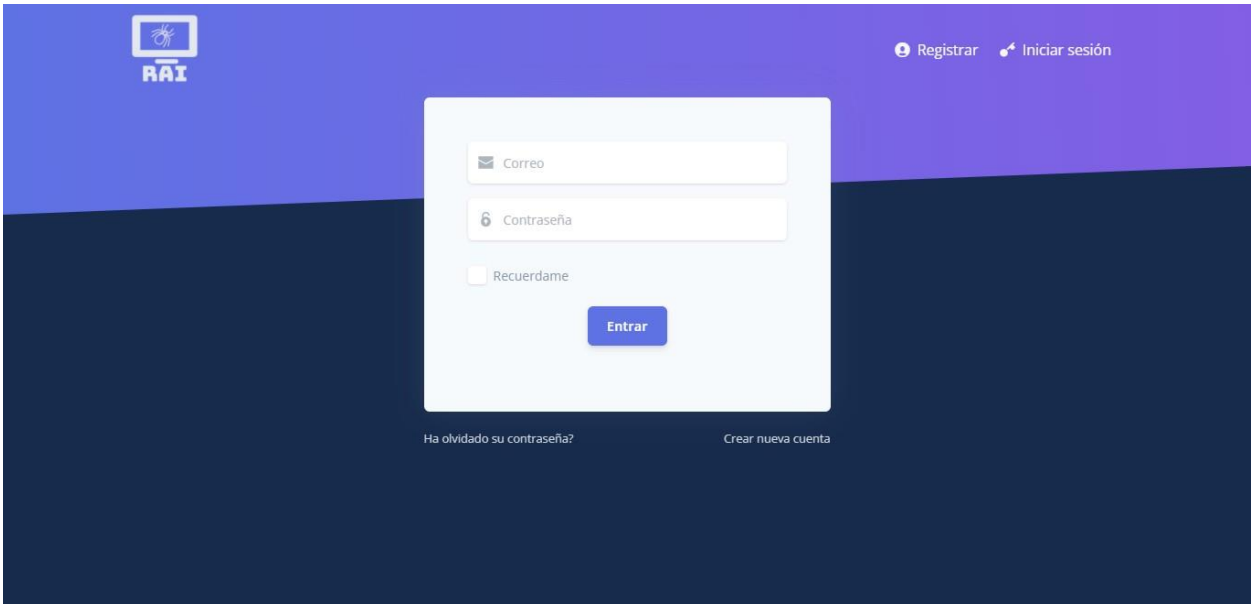


Figura 14 Interfaz de inicio de sesión.

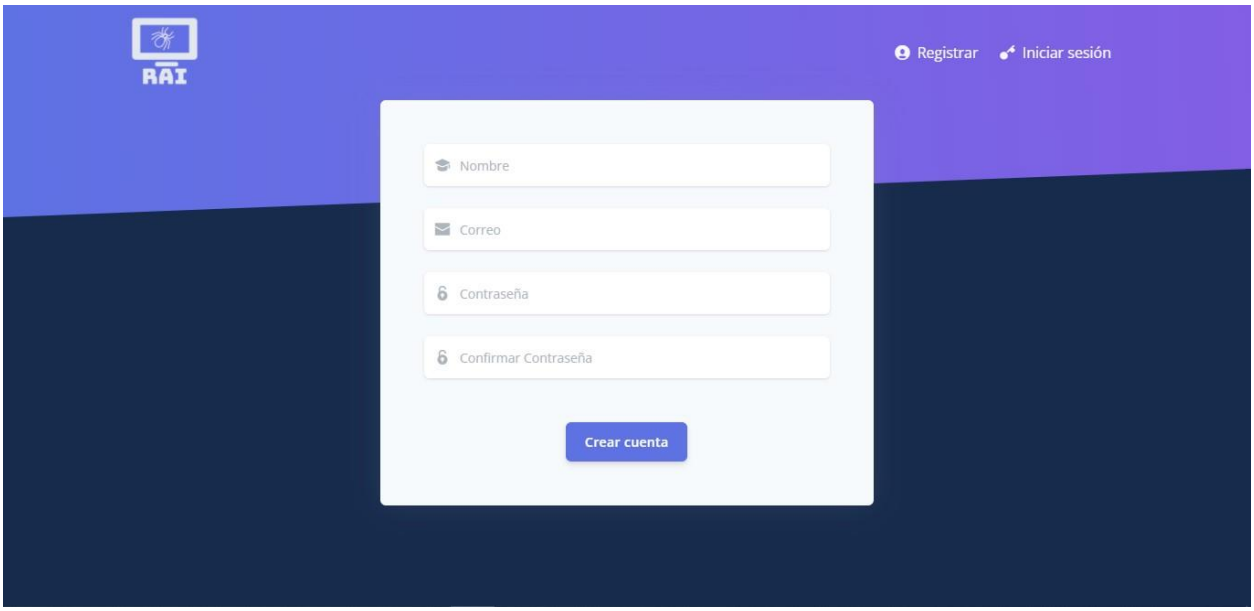


Figura 15 Interfaz de registro.

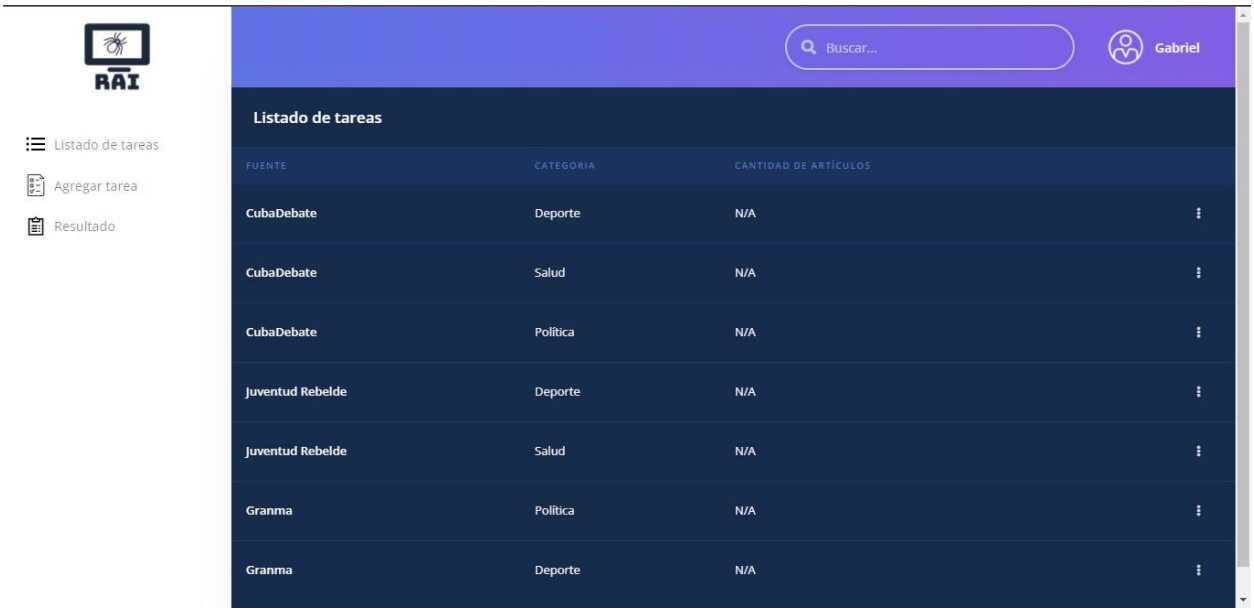


Figura 16 Interfaz de listado de tareas.

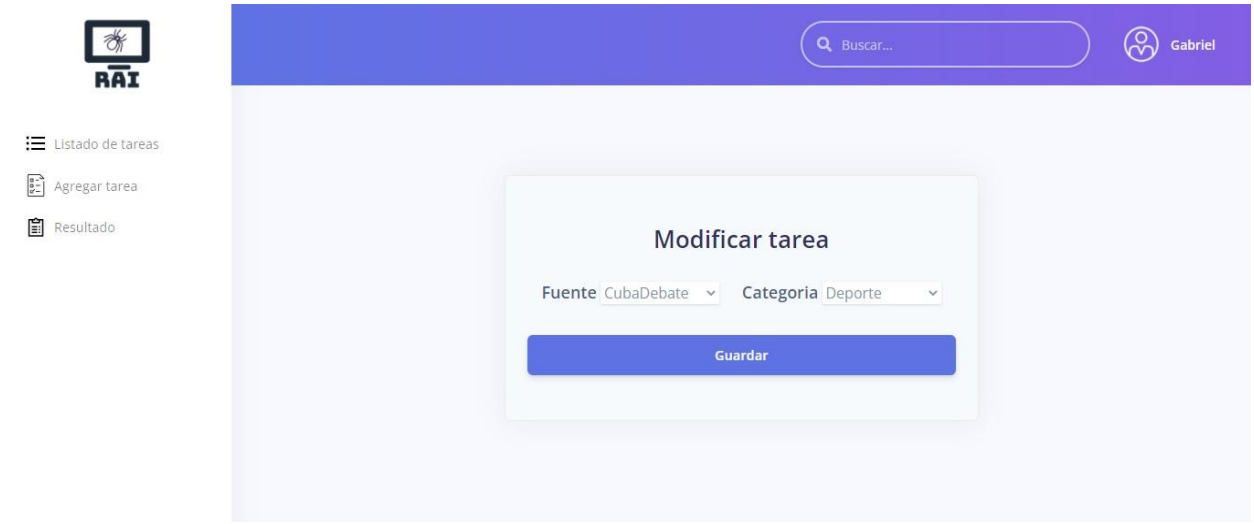


Figura 17 Formulario de modificar tarea.

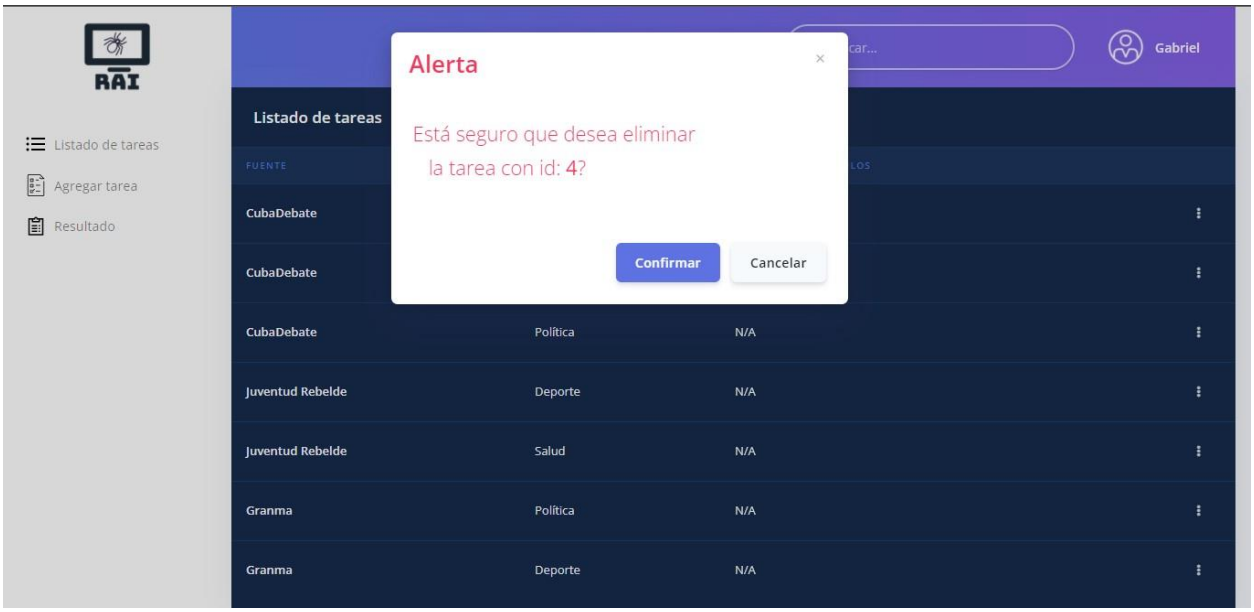


Figura 18 Opción de eliminar tarea.

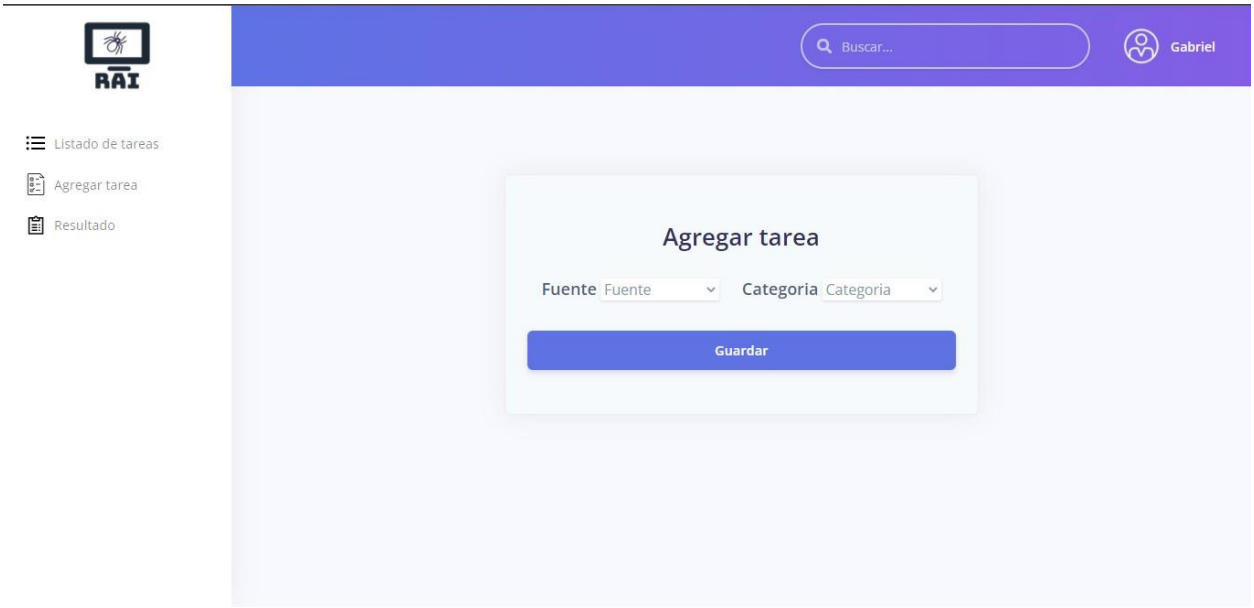


Figura 19 Formulario de agregar tarea.

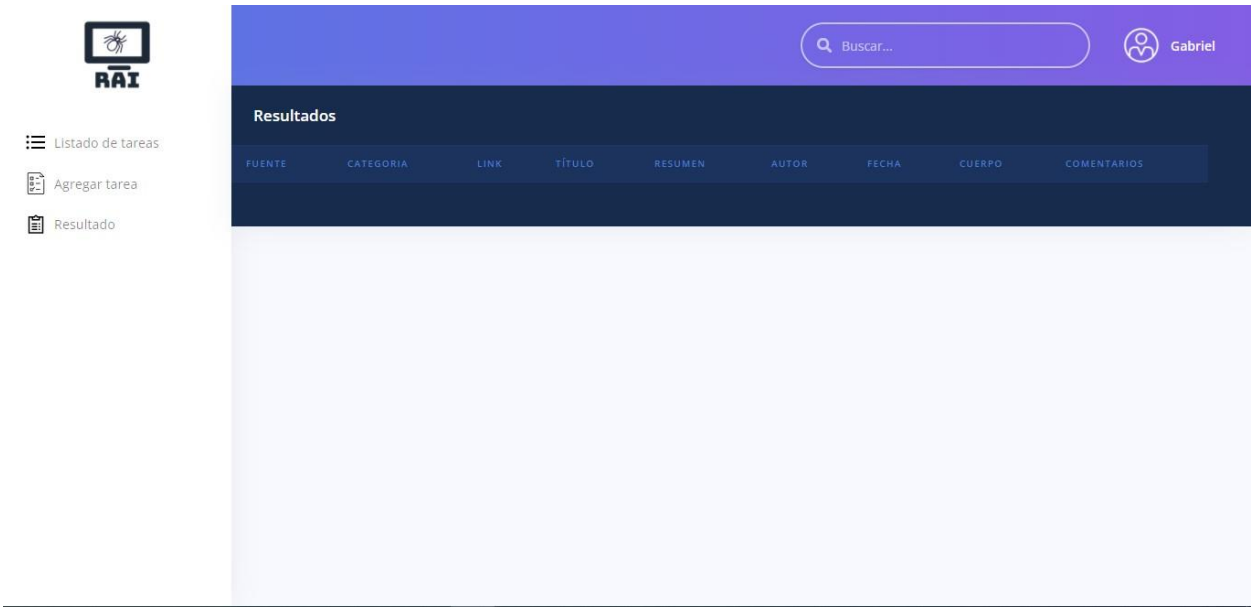


Figura 20 Interfaz de resultados de tareas en ejecución.

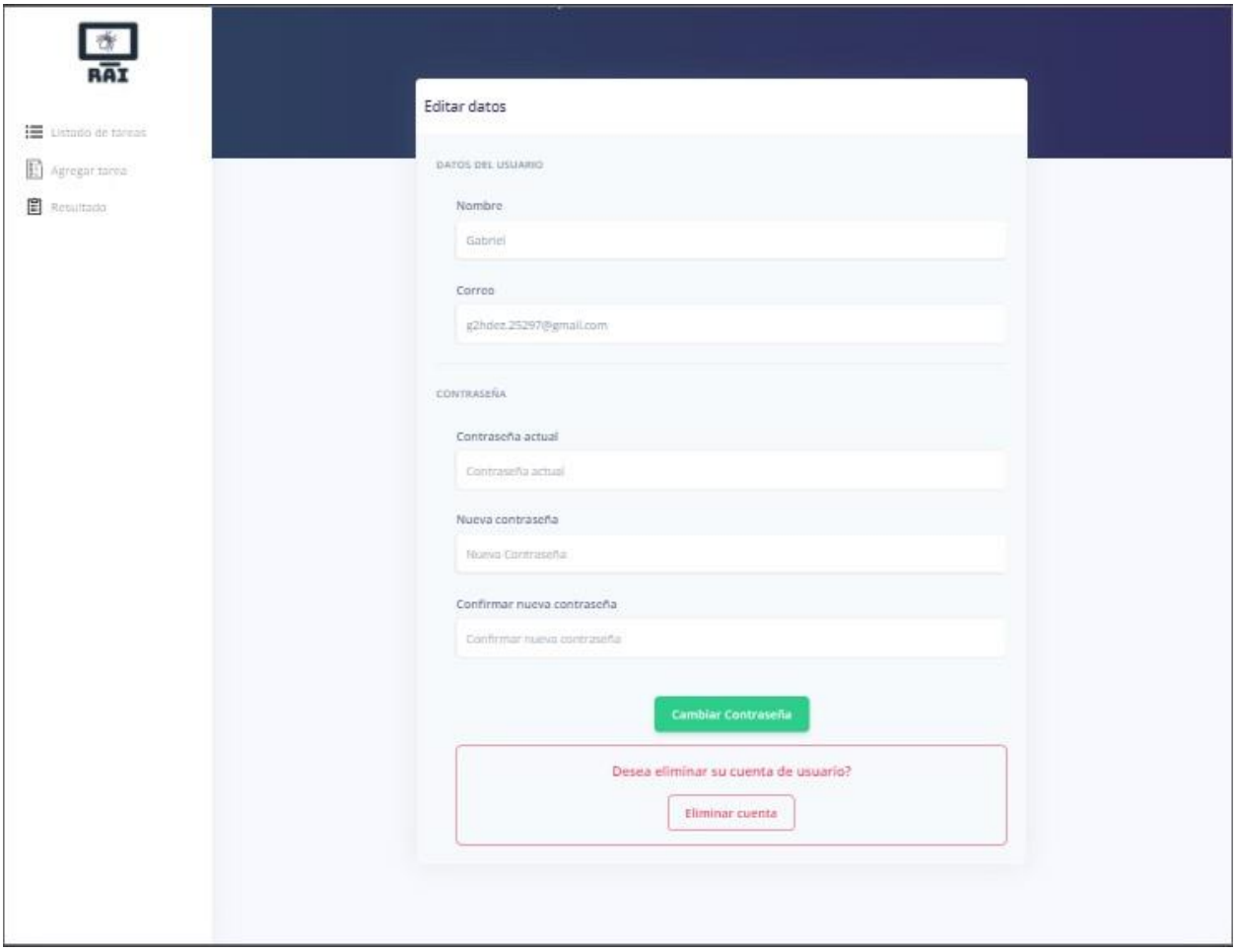


Figura 21 Formulario de editar datos de usuario y eliminar cuenta del usuario.