



Universidad de las Ciencias Informáticas

Facultad 1

Herramienta para la clasificación y análisis de información recopilada en los medios de difusión web

Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas

Autor: Víctor José Remedios Pérez

Tutores: MSc. Sahilyn Delgado Pimentel

Ing. Claudio Fernández Cabrera

Ing. Yadelis Velázquez Godoy

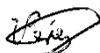
La Habana, 2021

Año 63 de la Revolución

Declaración de autoría

Declaro por este medio que yo, Victor José Remedios Pérez, con carné de identidad 97020811866, soy el autor principal del trabajo titulado Herramienta para la clasificación y análisis de información recopilada 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.



Victor José Remedios Pérez

Firma del Autor



MSc. Sahilyn Delgado Pimentel

Firma del Tutor



Ing. Claudio Fernández Cabrera

Firma del Tutor



Ing. Yadelis Velázquez Godoy

Firma del Tutor

Agradecimientos

El camino ha sido largo y difícil pero nunca he estado solo, la presente investigación es en parte gracias al esfuerzo de mi madre, mi mayor guerrera y ejemplo en la vida, junto a mi abuelo Victor que está y estará siempre en mi corazón. Gracias a mi papá Tony, a mi tía Dania, a mi abuela Mimí y a mi familia en general, a mis amistades de toda una vida y a los que se sumaron durante el transcurso de esta travesía. Muchas son las personas que han aportado su granito de arena en mi formación profesional y mi crecimiento personal, gracias de todo corazón.

Gracias a mis tutores por su paciencia y ayuda, en especial a Claudio, por su dedicación, sus chistes y todas las enseñanzas que me ha brindado, por convertirse en un amigo y ejemplo para mí. Gracias al tribunal y demás trabajadores de esta gran universidad, en especial a los de la Facultad 1 por sus consejos y por guiar a tantas personas de forma espiritual y profesional.

Muchos son los recuerdos que asaltan mi cabeza mientras nacen estas líneas, espero que tanto mi familia, amistades, profesores y muchas personas más sientan lo agradecido que estoy con ellos, incluso aquellas que no leerán estas palabras. ¡De todo corazón, gracias por todo!

Resumen

El desarrollo de las Tecnología de la Información y las Comunicaciones (TIC) han provocado gran auge de los datos en el internet. En la actualidad la clasificación y análisis de esa información es vital para las diferentes tomas de decisiones en varios sectores de la sociedad. Existen múltiples herramientas para ello, pero la mayoría son de difícil acceso para nuestros profesionales. La presente investigación consiste en el diseño e implementación de una herramienta capaz de clasificar y analizar la información recopilada en los medios de difusión web, además, generar reportes que sean de utilidad en la toma de decisiones. Para ello se realizó un estudio de sistemas homólogos, se definen las tecnologías y herramientas a utilizar en el proceso de desarrollo. Se realiza un levantamiento de los requisitos funcionales y no funcionales, se definen la arquitectura y los patrones de diseño empleados. Para la verificación de la propuesta de solución se realizan la técnica de caja negra y la validación por el método de ladov.

Palabras claves: Reporte, clasificación, análisis, información, web.

Índice

RESUMEN.....	- 4 -
INTRODUCCIÓN.....	- 7 -
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA HERRAMIENTA PARA EL ANÁLISIS Y CLASIFICACIÓN DE INFORMACIÓN RECOPIADA EN MEDIOS DE DIFUSIÓN WEB.....	- 13 -
1.1 ANÁLISIS DE LAS TECNOLOGÍAS, HERRAMIENTAS Y LENGUAJES A UTILIZAR	- 15 -
1.1.1 <i>Herramientas de Desarrollo</i>	- 16 -
1.2 ENTORNOS DE DESARROLLO INTEGRADO.....	- 18 -
1.2.1 <i>Lenguajes utilizados</i>	- 19 -
1.3 METODOLOGÍA DE DESARROLLO DE SOFTWARE	- 21 -
CONCLUSIONES PARCIALES.....	- 24 -
CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA HERRAMIENTA PARA EL ANÁLISIS Y CLASIFICACIÓN DE INFORMACIÓN RECOPIADA EN MEDIOS DE DIFUSIÓN WEB	- 25 -
2.1 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN.....	- 25 -
2.2 RECOPIACIÓN DE LOS REQUISITOS	- 26 -
2.2.1 <i>Requisitos funcionales</i>	- 27 -
2.2.2 <i>Requisitos no funcionales</i>	- 29 -
2.2.3 <i>Especificación de requisitos mediante HU</i>	- 31 -
2.3 MODELO DE DISEÑO	- 35 -
2.3.1 <i>Patrón arquitectónico</i>	- 36 -
2.3.2 <i>Estilo arquitectónico</i>	- 37 -

2.4 PATRONES GRASP	- 37 -
2.5 MODELO DE DATOS	- 38 -
2.6 DIAGRAMA DE CLASE	- 39 -
2.7 DIAGRAMA DE SECUENCIA	- 40 -
CONCLUSIONES PARCIALES.....	- 42 -
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA DE LA HERRAMIENTA PARA EL ANÁLISIS Y CLASIFICACIÓN DE INFORMACIÓN RECOPIADA EN MEDIOS DE DIFUSIÓN.....	- 43 -
3.1 DIAGRAMA DE DESPLIEGUE.....	- 43 -
3.2 DIAGRAMA DE COMPONENTES.....	- 44 -
3.3 ESTÁNDARES DE CODIFICACIÓN	- 45 -
3.4 PRUEBAS DE SOFTWARE	- 47 -
3.5 ESTRATEGIA DE PRUEBAS	- 48 -
3.5.1 Pruebas de caja negra	- 50 -
3.5.2 Prueba de partición equivalente	- 50 -
3.5.3 Casos de prueba	- 52 -
3.5.4 Clasificación de las no conformidades	- 56 -
3.5.5 Resultados de la prueba de Caja Negra	- 57 -
3.6 TÉCNICA DE IADOV	- 57 -
CONCLUSIONES PARCIALES.....	- 61 -
CONCLUSIONES GENERALES	- 62 -
REFERENCIAS BIBLIOGRÁFICAS.....	- 63 -
ANEXOS	- 67 -

Introducción

La evolución de los seres humanos se ha caracterizado por la constante interacción entre sí y con el medio que los rodea. El motor impulsor del desarrollo alcanzado ha sido la necesidad de solucionar las diversas problemáticas a las que se ha enfrentado. Durante este período de tiempo se han alcanzado múltiples hitos biológicos y tecnológicos, algunos ejemplos de estos últimos lo constituyen: la comunicación hablada, la escritura y la imprenta, catalogadas como las revoluciones de la comunicación humana. Estas han sentado los cimientos de algo realmente sorprendente y abarcador: el internet, que ha revolucionado el modo de interactuar entre los sujetos.

El siglo XX, estuvo marcado por diferentes avances tecnológicos en la Electrónica, la Informática y las Telecomunicaciones de forma independiente. La confluencia alcanzada provocó la generalización de conceptos nuevos como: "Tecnologías de la Información", "Sociedad del Conocimiento", "Era de la Información" o "Telemática". La historia de las Tecnología de la Información y la Comunicaciones (TIC) en la época moderna empieza en la década de los 70 del siglo pasado, al estallar la revolución digital (Díaz, Morillo, & Navarro, 2020). Existen múltiples ejemplos de TIC como el teléfono, los celulares, la imprenta, el correo y las computadoras.

En su afán por eliminar los males que acechan al mundo, las personas han apostado por ellas como la solución a todos los problemas y se realizan grandes inversiones cada día en torno a un desarrollo sostenible. La comunicación mediada por el ordenador, la computadora digital y las redes informáticas, ha logrado una ramificación en la vida social y en el modo de relacionarnos con el entorno que nos rodea, generando millones de datos continuamente a un ritmo exponencial. Todos esos datos son guardados y almacenados en servidores distribuidos por todo el planeta. Esta información en muchos casos es utilizada con fines políticos, comerciales y otros.

Sin duda alguna el que ha causado más impacto en el desarrollo de las sociedades es el Internet. No fue hasta finales de la centuria pasada donde se aceleró de forma vertiginosa su crecimiento, siendo así el mayor exponente de la convergencia entre estas ramas. El internet es más que una plataforma para el intercambio de datos: es la red mundial que permite el acceso a la información desde cualquier parte del mundo y hace posible la comunicación desde distintos lugares sin la necesidad de estar frente a frente.

Todo esto sucede en un lapso casi instantáneo, lo que facilita el intercambio y la obtención de información prácticamente de inmediato (Martin, Canchola, & Mayén, 2017).

La humanidad se adentra a la denominada cuarta revolución de la comunicación. La potencia y capacidad de cómputo de los ordenadores han avanzado a un ritmo acelerado debido al desarrollo. Cada día existen programas y aplicaciones accesibles a una mayor cantidad de personas mediante interfaces más amigables para los usuarios. Estos son varios de los factores que han posibilitado la reducción de los costos de las TIC y el internet, ampliándose así su uso en diversos sectores, como el empresarial, el ocio, la educación, la salud y en muchos más.

No obstante, mientras para algunas regiones geográficas y países la acelerada expansión y uso de las TIC no se ha detenido, para América Latina y otras pocas regiones en desarrollo su difusión se ha mostrado lenta (Quiroga-Parra, Torrent-Sellens, & Zorrilla, 2017). Cuba no se encuentra ajena a esta realidad y pese al bloqueo impuesto por el gobierno de Estados Unidos de Norteamérica realiza numerosos esfuerzos para ampliar el uso de las TIC en la sociedad cubana. La inauguración de la Universidad de Ciencias Informáticas (UCI), en el marco del “Proyecto Futuro implementado por el Estado cubano con el fin del desarrollo de la informatización de la sociedad le ha convertido en un referente en el auge del desarrollo del software. Su participación en la creación de aulas inteligentes, así como la acogida y promoción de eventos nacionales e internacionales ha evidenciado la calidad de la formación de sus docentes y estudiantes. Lo anterior responde a dos objetivos fundamentales del gobierno: informatizar el país y potenciar la industria del software para contribuir al desarrollo económico del mismo.

La Universidad de Ciencias Informáticas avanza cada día a una mayor interconexión entre sus estudiantes y trabajadores encabezando el progreso de la informatización de la sociedad. El centro de alto estudios marcha a la vanguardia, junto a otros sectores del país en la lucha contra campañas subversivas en las redes sociales que realizan los enemigos de Cuba, con el fin de denigrar al país, manchar la reputación de Cuba, propalar medias verdades y mentiras, confundir a la población, generar caos, miedo e inseguridad, entre otras.

En la Facultad 1 de la UCI, se encuentra el grupo para seguimiento, análisis y generación de contenidos del Centro de Desarrollo e Innovación para Internet (CIDI), encargado de monitorear la información que se genera en los medios. La no existencia de una herramienta informática ocasiona que actualmente el proceso

se realice de forma manual. Sus especialistas acceden a diferentes *links* de uno en uno, leen y clasifican el contenido según ciertos parámetros. Posteriormente copian los datos necesarios, realizan un análisis de los mismos y confeccionan un documento en formato pdf. El documento generado es enviado a directivos de la Universidad para su estudio; con el objetivo de elaborar un conjunto de reportes y hacer comparativas que son importantes para la posterior utilización de la misma en la toma de decisiones.

Lo explicado anteriormente provoca que el trabajo se torne engorroso, lento y propenso a errores humanos. Consumiendo tiempo y recursos que pueden ser destinados a otras valiosas tareas. Comprometiéndose la calidad del resultado y dilatando la toma de decisiones; afectando de esta forma los procesos que se realizan en la Universidad.

Sobre la base de los elementos expuestos anteriormente se formula el siguiente problema de investigación:

¿Cómo contribuir a la clasificación y análisis de información recopilada en los medios de difusión web?

Para la realización de la investigación se define como objeto de investigación: la clasificación y análisis de la información recopilada en los medios de difusión web.

Para dar solución al problema planteado, se define como objetivo general: Desarrollar una herramienta web que permita la clasificación y el análisis de información publicada en los medios de difusión web para ayudar a la toma de decisiones.

Para cumplir con el objetivo general antes mencionado, se les dará cumplimiento a los siguientes objetivos específicos:

1. Fundamentar los conceptos, características y antecedentes de los sistemas que se utilizan para la clasificación de información y análisis estadísticos.
2. Diseñar e implementar una herramienta que permita la clasificación y análisis de información publicada en los medios de difusión web.
3. Validar la herramienta implementada a través de prueba de software.

Para dar cumplimiento a los objetivos específicos se definieron las siguientes tareas de investigación:

1. Realizar un estudio sobre los métodos de la clasificación y el análisis de información publicada en los medios de difusión web.
2. Seleccionar las herramientas, estándares, metodología y tecnologías que se necesitan para implementar la propuesta de solución.
3. Elaborar los artefactos requeridos por la metodología de desarrollo seleccionada e implementar la herramienta para la clasificación y el análisis de información publicada en los medios de difusión web.
4. Validar la herramienta propuesta.

Después de haber tratado los elementos fundamentales del área de la ciencia a incidir y los objetivos fundamentales, se formula la siguiente *Idea a defender*, la creación de una herramienta que permita la clasificación y el análisis de información publicada en los medios de difusión web, contribuirá a una acertada toma de decisiones.

Para el desarrollo de las tareas científicas se han combinado diferentes métodos teóricos y empíricos de la investigación en la búsqueda y procesamiento de la información. Estos son:

Métodos teóricos:

- **Histórico - Lógico:** se empleó con el propósito de constatar teóricamente cómo ha evolucionado en el tiempo la clasificación y el análisis de información publicada en los sitios web, así como las herramientas y tecnologías utilizadas en el desarrollo de aplicaciones similares.
- **Analítico-Sintético:** se empleó para el análisis de los elementos esenciales referentes a la información relacionada con las herramientas de clasificación y análisis de Información en los medios de difusión web.

- **Inductivo-Deductivo:** se empleó para arribar a razonamientos que puedan ser aplicables al problema a resolver luego de adquirir una serie de elementos referentes a la clasificación y el análisis de información en los medios de difusión web.
- **Modelación:** se empleó a través de los diagramas del lenguaje de modelado UML, se utilizó en la representación de las características y las relaciones entre los objetos de la solución.

Métodos empíricos:

- **Observación:** se emplea para comprender cómo se desarrolla la clasificación y el análisis de información en los sitios web a través del análisis de sistemas que lo implementan.
- **Entrevista:** se emplea para obtener los requisitos funcionales de la solución propuesta y para interactuar con personal familiarizado con la clasificación y el análisis de la información en los medios de difusión web.

El presente documento está estructurado en tres capítulos:

Capítulo 1: Fundamentación teórica de la herramienta para el análisis y clasificación de información recopilada en medios de difusión.

En este capítulo se abordan diferentes herramientas líderes en la clasificación y análisis de la información. Se describen las principales herramientas, tecnología y lenguaje de programación empleada. Se definió la tecnología AUP versión UCI para utilizarla en el desarrollo de la propuesta de solución.

Capítulo 2: Análisis y diseño de la herramienta para el análisis y clasificación de información recopilada en medios de difusión.

A lo largo de este capítulo se especifican los requisitos funcionales y no funcionales necesarios para lograr que la herramienta para la clasificación y análisis de información recopilada en los medios de difusión web de la UCI ayude de forma satisfactoria al grupo de trabajo en redes sociales del Centro de Innovación y Desarrollo para Internet(CIDI), los elementos fundamentales del diseño y de la arquitectura a tener en cuenta para la implementación del sistema, auxiliado por el modelado de diagramas.

Capítulo 3: Implementación y prueba de la herramienta para el análisis y clasificación de información recopilada en medios de difusión.

En este capítulo se muestra el modelo de implementación como resultado del diseño anteriormente realizado. Se describen las pruebas a utilizar y los resultados obtenidos de estas, con el objetivo de comprobar el correcto funcionamiento de la herramienta para la clasificación y análisis de información recopilada en los medios de difusión web.

Como posible resultado se pretende que, al concluir la investigación se cuente con una herramienta que permitan la clasificación y el análisis de información recopilada en los medios de difusión web. Facilitando y automatizando el trabajo en el grupo de redes sociales de CIDI, apoyando la toma de decisiones de la universidad.

Capítulo 1: Fundamentación teórica de la herramienta para el análisis y clasificación de información recopilada en medios de difusión web

Introducción

En el presente capítulo se estudian herramientas eficaces y reconocidas a nivel internacional para la clasificación y análisis de datos. Además, se describen las herramientas y tecnologías factibles para el desarrollo de la solución informática del proyecto. Se selecciona el lenguaje de programación y la metodología AUP en su versión UCI para su uso en la propuesta de solución.

El gran volumen de datos y metadatos generados cada día va aumentando de forma vertiginosa, haciéndose necesario realizar un estudio de ellos con una alta calidad, se hace inevitable una imperiosa utilización de herramientas de análisis de datos con softwares especializados y capaces de ofrecer los mejores resultados. La analítica es uno de los recursos más utilizados en la actualidad, muchas organizaciones invierten cada día en métodos efectivos para recopilar información precisa para ampliar su eficacia.

Debido a la variedad de herramientas de clasificación y análisis de información existente en la actualidad, los expertos han de seleccionar las que le ayudan a optimizar su actividad analítica. Seguidamente se explican algunas de ellas.

Power BI

Es una herramienta de inteligencia de negocios creada por Microsoft, es el nombre colectivo de una variedad de aplicaciones, ofrece a las organizaciones recopilar, administrar y analizar datos de una variedad de fuentes de datos, desde hojas de cálculo de Excel hasta base de datos, pueden ser almacenadas en la nube o localmente, además permite compartir información, se actualiza en tiempo real y está disponible para todos los dispositivos. Power BI ayuda a los usuarios a ver no solo lo que sucedió en el pasado y lo que está sucediendo en el presente, sino también lo que podría suceder en el futuro, a través de predicciones por medio de técnicas de Series Temporales utilizando el algoritmo de Suavizado exponencial. Permitiendo a los usuarios generar pronósticos y prepararse para satisfacer la demanda futura y otras medidas claves. Power Query es uno de los componentes de Power BI es una herramienta de conexión de datos que le permite transformar, combinar y mejorar datos de varias fuentes (Bermeo-Pérez & Campoverde-Molina, 2020).

Tableau

Tableau es una herramienta de visualización de datos interactivos principalmente enfocada al Business Intelligence, conjunto de estrategias y herramientas que facilitan la toma de decisiones en una empresa basándose en sus datos, ayudando a convertir los datos en información útil mediante representaciones visuales.

En general, Tableau permite trabajar con los datos, pieza fundamental en los negocios, aportando flexibilidad, seguridad y facilidad de uso tanto para trabajar con datos locales como remotos y permite ser ejecutado en múltiples sistemas operativos como Windows, Linux y Mac OS (Fenollar, 2019).

SAP BW (Business Warehouse)

SAP Business Warehouse es una herramienta de Inteligencia de Negocios basado en la plataforma de SAP NetWeaver. Recopila, transforma y almacena los datos generados en aplicaciones SAP y No SAP, y los hace accesibles a través de informes integrados, inteligencia empresarial y herramientas de análisis, así como software de terceros (Sanguino, 2019).

Geckoboard

Es un software para empresas utilizado para crear tableros de datos claros y efectivos que mantengan a los equipos informados, motivados y alineados con múltiples facetas del negocio y sus indicadores clave de rendimiento (o los llamados "KPI" en inglés, Key Performance Indicators).

Se centra en plasmar las matrices y datos más importantes en una sola pantalla de manera comprensible, fácil y rápida. Ayuda a los miembros de un equipo a visualizar y a entender los datos más importantes de la empresa. En estos tableros se plasma el progreso en comparación con las formas de análisis tradicionales que nos proporcionan datos que realmente no son siempre útiles (Frías, 2018).

Tabla 1: Comparación de Sistemas Homólogos [Elaboración propia]

Herramientas	Multiplataforma	Privativo
Power BI	Sí	Sí
Tableau	Sí	Sí
SAP BW (Business Warehouse)	Sí	Sí
Geckoboard	Sí	Sí

Lo expresado anteriormente reafirma que pese a la existencia de múltiples y eficaces herramientas para la clasificación y el análisis de información estas no le dan solución a la problemática planteada. Varias son de pago o privativas lo que dificulta el acceso de nuestros profesionales a ellas, no responden a la soberanía tecnológica que aspira la dirección del país. Sin embargo, las herramientas antes mencionadas poseen elementos positivos a tener en cuenta en la propuesta de solución, por ejemplo, accesibilidad de forma remota y facilidad de interpretación de los informes generados. Son de fácil manejo para los usuarios.

1.1 Análisis de las tecnologías, herramientas y lenguajes a utilizar

En este epígrafe se explican las tecnologías, herramientas y lenguaje, utilizadas en el desarrollo de este proyecto.

1.1.1 Herramientas de Desarrollo

El **UML** (del inglés Unified Modeling Language, en español Lenguaje de Modelado Unificado) es un lenguaje de modelado visual estándar destinado a ser utilizado para el modelado de procesos de negocio; y para el análisis, diseño e implementación de sistemas basados en software. UML es un lenguaje común para los analistas, arquitectos y desarrolladores de software, utilizado para describir, especificar, diseñar nuevos procesos de negocio. UML proporciona valiosa ayuda a los profesionales visualizando, comunicando y aplicando sus diseños (Brito, Bravo & Armas, 2019). El UML permite una comunicación simple y veloz entre programador y clientes, posibilitando que estos últimos puedan expresar adecuadamente sus opiniones respecto al producto, así como la conformidad con el mismo.

Visual Paradigm es una herramienta CASE: (Ingeniería de Software Asistida por Computación). La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación.

La herramienta está diseñada para un amplio rango de usuarios, incluyendo ingenieros de software, analistas de sistema, analistas de negocio, arquitectos de sistemas y cualquiera que esté interesado en construir un sistema confiable a gran escala a través del uso del enfoque orientado a objetos. VP-UML da soporte a los últimos estándares de Java y notación UML y provee a la industria una gran variedad de código autogenerado y soporte para Java para realizar ingeniería inversa. Además, VP-UML permite una integración completa con herramientas de desarrollo como Eclipse, Borland, Netbeans para dar soporte en la etapa de implementación del software. La transición del análisis al diseño y luego a la implementación se realiza sin percances con este tipo de herramienta CASE, por lo que disminuye significativamente el esfuerzo empleado en todas las etapas del ciclo de vida del desarrollo (Galván & Zamorano, 2020).

PostgreSQL, también conocido como Postgres es la base de datos de software libre más avanzada del mundo. Es un manejador de base de datos de alta disponibilidad, lo que significa que puede funcionar de manera muy estable en el servidor en momentos de alta demanda. Esta es una de sus principales características y por lo que es muy usado entre las grandes empresas. Además de poseer una gran resistencia a los fallos, también cuenta con las siguientes características:

- ✓ Es compatible con el modelo relacional, lo que garantiza la integridad de los datos.

- ✓ Alta Concurrencia, lo que le permite manejar gran cantidad de transacciones a la vez sin sufrir fallos o bloqueos.
- ✓ Soporta múltiples tipos de datos de manera nativa, lo que representa una ventaja sobre sus competidores. Entre estos tipos de datos tenemos: JSON, Array, IP, MAC, Figuras Geométricas, etc.
- ✓ Es de tipo OBJETO-RELACIONAL, lo que permite trabajar con sus datos como si fueran objetos y ofrece mecanismos de la orientación a objetos, como herencia de tablas.
- ✓ Soporte a TRIGGERS, lo que permite crear disparadores de ejecución partiendo de eventos como guardar, actualizar, borrar, etc.
- ✓ Vistas, las vistas permiten consultar los datos de una o varias tablas de manera customizada haciendo más rápida la respuesta del manejador.
- ✓ Soporte a Bases de Datos Distribuidas. Esto asegura que las transacciones sean realizadas con éxito entre todos los sistemas involucrados.
- ✓ Soporte a múltiples lenguajes de programación. Postgres es capaz de trabajar con funciones escritas en diversos lenguajes como C, C++, Java, PHP o Python. Además, ofrece interfaces para ODBC y JDBC y otras interfaces de programación (Soto, 2021).

Al ser multiplataforma, gratuito, poseer potencia y robustez, así como su escalabilidad, estabilidad y confiabilidad hacen de él una poderosa herramienta gestora de bases de datos objeto-relacionales (ORDBMS).

Un servidor web es un software que usa protocolo de transferencia de hipertexto http (por sus siglas en inglés Hipertext Transfer Protocol) para servir los archivos que componen una página web a los usuarios como respuestas a las solicitudes de los navegadores web. **Apache** es un programa diseñado para funcionar como servidor web HTTP, es de código abierto y multiplataforma. Consta de una sección principal que es el núcleo y se le puede agregar diversos módulos que agregan distintas funcionalidades. Es extensible permite utilizar diversos lenguajes de programación (Cruz, 2017). Su bajo costo, amplia documentación, fiabilidad y soporte, además, de su escalabilidad y flexibilidad, hacen de él una herramienta adecuada para la propuesta de solución.

Bootstrap es un framework para el desarrollo web, de código abierto, desarrollado por Twitter y utilizado para diseñar páginas web con HTML, CSS y JavaScript. Está enfocado al diseño web adaptable o diseño responsable, es decir, al utilizar Bootstrap para diseñar nuestro sitio web podremos visualizarlo

correctamente en cualquier dispositivo, móvil o escritorio. Facilita el proceso de maquetado de un sitio web gracias a su sistema de columnas (Guerra, 2018). Su compatibilidad con diferentes dispositivos y con las principales librerías de JavaScript, su fácil entendimiento, su constante desarrollo, amplia documentación y las diversas plantillas gratis en internet son algunas de las principales ventajas para la utilización de este framework.

Laravel es uno de los framework que manejan PHP y de código abierto más populares, gracias a su sintaxis (fácilmente comprensible) y su facilidad para desarrollar proyectos de forma rápida y eficaz. Su popularidad radica en que este framework busca desarrollar código PHP de una forma más sencilla y presentable (Rodríguez, 2021). Este autor describe las siguientes características del frameworks antes mencionado:

- ✓ Su arquitectura es MVC (Modelo-Vista-Controlador), con la que puedes relacionar entre sí las partes de una aplicación.
- ✓ Es capaz de ejecutar tareas en segundo plano, con la intención de mejorar el rendimiento.
- ✓ Puedes integrar Laravel con plataformas de terceros o bibliotecas.
- ✓ Tienes a la mano el uso de extensiones para aumentar la funcionalidad principal de Laravel.

Además de las características antes mencionadas también cuenta con abundante documentación, una comunidad amplia y constantes actualizaciones, todo lo anterior lo hacen una excelente opción para desarrollar proyectos de forma rápida y eficaces.

1.2 Entornos de Desarrollo Integrado

PHP Storm es un Entorno de Desarrollo Integrado (IDE) con numerosas funcionalidades y se puede utilizar el FTP, Git y la Base de Datos y el terminal desde el mismo.

Las características principales de PHPStorm son:

- ✓ Asistencia de código inteligente: se realizan inspecciones para verificar que el código es correcto y analizar el proyecto en su conjunto.
- ✓ Navegación de código inteligente: el IDE entiende donde queremos ir y nos lleva allí instantáneamente.

- ✓ Refactorización rápida y segura: podemos refactorizar nuestro código de manera segura para renombrar, mover, eliminar, extraer un método, etc.
- ✓ Fácil debug y test.
- ✓ Ambiente de desarrollo: control de versiones con Git, SVN, Mercurial, Bases de Datos y SQL con soporte para Vagrant, Docker, Composer y mucho más (Cabana, 2017).

Lo anterior demuestra que es un IDE potente para proyectos que usen el lenguaje PHP y el framework Laravel.

1.2.1 Lenguajes utilizados

Los lenguajes de programación del lado del cliente se utilizan para integrarse en sitios web; un determinado código escrito en un lenguaje de Script se incorpora de forma directa dentro de un código HTML y se ejecuta interpretado. La programación del lado del cliente permite validar algunos datos en la máquina de este antes de que sean enviados al servidor; de esta manera, los usuarios pueden tener un informe de error de forma inmediata.

El lenguaje **HTML** (hipertexto markup language) se utiliza para el desarrollo y creación de páginas web. Se compone de una serie de etiquetas que el navegador interpreta. Entre las etiquetas que se incluyen dentro del lenguaje HTML se encuentran: hipervínculos, etiquetas para imágenes, saltos de página, entre otras (Peiró, 2017). Cada cierto tiempo aparecen versiones nuevas con novedosas funcionalidades. Es uno de los lenguajes que más se usan en la actualidad, especialmente en el desarrollo de proyectos web. Su posibilidad de describir hipertextos, su rápido despliegue, su facilidad de ser reconocido y admitido por cualquier tipo de explorador y su reconocimiento de archivos pequeños propicia su uso en la propuesta de solución planteada.

JavaScript es un lenguaje de programación que permite incorporar dinamismo al sitio web. Por ejemplo, puedes agregar elementos multimedia, formularios, encuestas, juegos, galerías de imágenes, botones interactivos, para captar la atención de los usuarios.

JavaScript fue creado por Brendan Eich, y aparece a mediados del año 1995 con el nombre de LiveScript. Así mismo, nació como un lenguaje práctico con el fin de añadir algunas características interactivas a las páginas web. Luego creció de manera acelerada, y hoy es el lenguaje de programación base que se utiliza

en casi todos los sitios web en el mundo. Es un lenguaje multiparadigma, es decir, soporta la programación funcional, orientada a objetos e imperativa. Por otro lado, puedes crear archivos JavaScript con la extensión .js, así como escribir scripts dentro de las etiquetas HTML, CSS, XML, etc. (Herrera, 2020). Este lenguaje tiene numerosas bondades, es de fácil aprendizaje, además, tiene parecido con varios lenguajes de desarrollo como Java y C. Su versatilidad le permite integrarse a otras tecnologías. Posibilita la creación de interfaces dinámicas, es multiplataforma y compatible con diversos CMS. Es actualizado frecuentemente, lo expresado anteriormente justifica su elección para la propuesta de solución.

jQuery es una librería desarrollada en 2006 por John Resig que permite añadir una capa de interacción AJAX entre la web y las aplicaciones que desarrollemos controlando eventos, creando animaciones y diferentes efectos para enriquecer la experiencia de usuario. Es un software libre y de código abierto (posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2). Cuenta con un diseño que facilita la navegación por un documento y seleccionar elementos DOM proporcionando a los desarrolladores de aplicaciones web complementos que agilizan el desarrollo de proyectos. Esto permite a los desarrolladores centrarse en lo importante y crear abstracciones para interacción y animación de bajo nivel, efectos avanzados y widgets temáticos de alto nivel sin invertir tiempo en desarrollar complejos algoritmos y métodos que los controlen desde cero y generando menos código que las aplicaciones hechas con JS puro (Parada, 2019). Con menos líneas de código se pueden obtener los mismos resultados que se logran programando con JavaScript. Esta ventaja contribuye a la disminución de errores y al ahorro de tiempo, ambos necesarios en el desarrollo de la propuesta de solución.

XQuery es un lenguaje que puede consultar datos XML estructurados o semiestructurados. Con el soporte de tipo de datos XML provisto en el Motor de base de datos, los documentos se pueden almacenar en una base de datos y luego consultar mediante XQuery. XQuery se basa en el lenguaje de consulta XPath existente, con soporte agregado para una mejor iteración, mejores resultados de clasificación y la capacidad de construir el XML necesario. XQuery opera en el modelo de datos XQuery. Esta es una abstracción de los documentos XML y los resultados de XQuery que se pueden escribir o no. La información de tipo se basa en los tipos proporcionados por el lenguaje de esquema XML del W3C (Microsoft, 2017).

CSS son las siglas en inglés para «hojas de estilo en cascada» (*cascading style sheets*). Básicamente, es un lenguaje que maneja el diseño y presentación de las páginas web, es decir, cómo lucen cuando un visitante las visita. Funciona junto al lenguaje HTML, que se encarga del contenido básico de las páginas.

Se les denomina hojas de estilo «en cascada» porque puedes tener varias hojas y una de ellas con las propiedades heredadas (o «en cascada») de otras. Con CSS puedes crear reglas para decirle a tu sitio web cómo quieres mostrar la información y puedes guardar los comandos para elementos de estilo (como fuentes, colores, tamaños, etc.) separados de los que configuran el contenido (Santos, 2021). Este lenguaje permite más opciones de estilo, reduce el exceso de código repetido por lo que reduce el tamaño de los archivos acelerando así el tiempo de carga de la página. Es de fácil comprensión para cualquier programador o diseñador por la uniformidad que posee.

PHP es un lenguaje de programación destinado a desarrollar aplicaciones para la web y crear páginas web, favoreciendo la conexión entre los servidores y la interfaz de usuario. Es de código abierto y posee gran dinamismo. Generalmente PHP es definido como un lenguaje del lado del servidor. Esto significa que se aplica en la programación que tiene lugar en el servidor web responsable de ejecutar la aplicación o, más a menudo, en un sitio web (Souza, 2020). Su vasta documentación, amplia comunidad y su bajo nivel en la curva de aprendizaje hacen de PHP, uno de los lenguajes más populares en el desarrollo web. También admite gran cantidad de datos y posee gran compatibilidad con las principales bases de datos, ratificándolo como uno de los lenguajes principales en el desarrollo web en la actualidad.

1.3 Metodología de desarrollo de software

Según Peñalvo, Holgado & Ingelmo (2020) la metodología la metodología es definida como un proceso para producir software de forma organizada, empleando una colección de técnicas y convenciones de notación predefinidas. Las metodologías se pueden clasificar en robustas o ágiles.

El autor Trejo (2017) plantea que en las metodologías tradicionales se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada.

Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar.

Los autores Ortega & Camacho (2019) definen las metodologías ágiles como procesos ligeros que utilizan ciclos iterativos cortos. Involucra a los usuarios con entusiasmo para establecer, priorizar y garantizar que se atiendan las necesidades”, esto quiere decir que estas metodologías son aplicadas en el desarrollo de software con el fin de brindar resultados óptimos en corto plazo y así mejorar la satisfacción del cliente.

Las metodologías mencionadas anteriormente no se aplican en su totalidad en la mayoría de los proyectos y en muchas ocasiones pueden afectar los diferentes procesos de los mismos. En la UCI se elaboró una metodología híbrida, esta tiene como base fundamental la metodología ágil AUP. A continuación, se explicarán sus características fundamentales. La metodología AUP en su versión UCI tiene 3 fases, estas son:

Fase 1 – 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.

Fase 2 – 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. (Esta fase unifica las fases Elaboración, Construcción y Transición de la metodología AUP).

Fase 3 – Cierre: En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto (Sánchez, 2015).

La autora Sánchez (2015) describe cuatro posibles escenarios para la metodología AUP en su versión UCI, a continuación, se mencionan sus características:

Escenario No. 1: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que puedan modelar una serie de interacciones entre los trabajadores del negocio/actores del sistema (usuario), similar a una llamada y respuesta respectivamente, donde la atención se centra en cómo el usuario va a utilizar el sistema. Es necesario que se tenga claro por el proyecto que los Casos de Uso del

Negocio (CUN) muestran cómo los procesos son llevados a cabo por personas y los activos de la organización.

Escenario No. 2: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelarían exclusivamente los conceptos fundamentales del negocio. Se recomienda este escenario para proyectos donde el objetivo primario es la gestión y presentación de información.

Escenario No. 3: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio con procesos muy complejos, independientes de las personas que los manejan y ejecutan, proporcionando objetividad, solidez, y su continuidad. Se debe tener presente que este escenario es muy conveniente si se desea representar una gran cantidad de niveles de detalles y la relaciones entre los procesos identificados.

Escenario No. 4: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos, ya que una Historias de Usuario (HU) no debe poseer demasiada información.

De acuerdo con lo expresado anteriormente se decide utilizar la metodología AUP en su versión UCI porque esta responde con gran flexibilidad a los cambios que puedan surgir. Garantiza el cumplimiento de las normas de desarrollo de software de la Universidad. Además, se decide encapsular los requisitos en el escenario número cuatro debido a su dinamismo y flexibilidad, ahorrando así tiempo y esfuerzos.

Conclusiones parciales

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

- ✓ Con el estudio realizado a los sistemas internacionales, queda acentuada la necesidad del desarrollo de una herramienta que permita la clasificación y análisis de información recopilada.
- ✓ 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 escogieron PHP, HTML, CSS, JavaScript 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.

Capítulo 2: Análisis y diseño de la herramienta para el análisis y clasificación de información recopilada en medios de difusión web

Introducción

En el presente capítulo se describe la propuesta de solución y los roles. Además, se hace un análisis detallado de los principales requerimientos del sistema. Estos son encapsulados mediante historias de usuarios (HU) como plantea la metodología AUP-UCI en su escenario 4. Se realiza una explicación del patrón y el estilo arquitectónico. Se muestran los diagramas de entidad-relación, de clase y de secuencia.

2.1 Descripción de la propuesta de solución

La propuesta de solución planteada responde a la problemática expresada en el capítulo anterior. En el desarrollo de esta se hace uso de las herramientas y tecnologías expuestas previamente. El sistema será el encargado de automatizar el análisis y clasificación de la información recopilada en los sitios web. Además, gestionará reportes y agilizará el proceso de tomas de decisiones.

El sistema contará con una interfaz en la cual se podrán gestionar usuarios con diferentes roles. Una vez autenticados tendrán acceso a una nueva interfaz con las funcionalidades específicas de su usuario. Culminada la tarea realizada cerrarán la sesión, evitando así que personas no autorizadas accedan al sistema.

La tabla que se muestra a continuación ilustra los roles definidos para una mayor comprensión de la propuesta de solución.

Tabla 2: Roles [Elaboración propia]

Rol	Descripción
Especialista	Usuario que posee una cuenta en el sistema que le permite crear, modificar y eliminar los reportes que han sido clasificados previamente. Además puede ejecutar otras funcionalidades no administrativas del sistema.
Administrador	Usuario que posee una cuenta en el sistema con todos los permisos requeridos para trabajar en el mismo, será el encargado de su configuración y control.

2.2 Recopilación de los requisitos

Un requisito es una condición necesaria para tener acceso a algo, o para que una cosa suceda. La palabra requisito proviene del latín requisitum, que significa pretender o requerir alguna cosa ("Requisito," 2019).

En la ingeniería de sistemas, un requisito es un dato necesario en la fase de diseño de un producto. En este caso, los requisitos dictan la norma sobre lo que debe ejecutar el producto a crear.

Un requisito, en este caso, también se refiere a una condición que debe cumplir el usuario para poder ejecutar algo en el sistema ("Requisito," 2019).

La ingeniería de requerimientos proporciona el mecanismo apropiado para entender lo que desea el cliente, analizar las necesidades, evaluar la factibilidad, negociar una solución razonable, especificar la solución sin ambigüedades, validar la especificación y administrar los requerimientos a medida que se transforman en un sistema funcional. Incluye siete tareas diferentes: concepción, indagación, elaboración, negociación, especificación, validación y administración. Es importante notar estas ocho tareas ocurren en paralelo y que todas se adaptan a las necesidades del proyecto (Pressman, 2010).

2.2.1 Requisitos funcionales

Especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física. Especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto. Indican características y restricciones sobre la funcionalidad del software. Definen el comportamiento interno del sistema (Niño, 2018).

La tabla a continuación refleja los requisitos funcionales (RF) definidos en la propuesta de solución. Ver Anexo 1.

Tabla 3: Requisitos funcionales [Elaboración propia]

No	Nombre	Prioridad	Complejidad	Descripción
RF1	Registrar usuario	Alta	Alta	El sistema permite a los administradores que registren nuevos usuarios.
RF2	Autenticar usuario	Alta	Media	El sistema permite que los usuarios puedan autenticarse.
RF3	Cerrar cesión	Baja	Baja	El sistema permite que los usuarios puedan cerrar sesión.
RF4	Modificar usuario	Media	Media	Luego de autenticado un usuario con rol administrador el sistema permite que este modifique los datos de los usuarios registrados.
RF5	Listar usuario	Media	Baja	Los administradores pueden listar todos los usuarios que están registrados en el sistema.

RF6	Eliminar usuario	Media	Baja	Los administradores pueden eliminar todos los usuarios que están registrados en el sistema.
RF7	Asignar rol	Alta	Media	Luego de añadido un usuario, el sistema permite que el usuario con rol de administrador pueda asignar roles a los usuarios añadidos.
RF8	Eliminar rol	Baja	Media	Los administradores son los encargados de eliminar los privilegios a cada uno de los usuarios registrados en el sistema.
RF9	Modificar rol	Media	Media	El sistema permite a los administradores modificar los distintos roles de los usuarios registrados.
RF10	Listar rol	Media	Baja	El sistema permite a los administradores listar todos los roles que están creados en él.
RF11	Extraer información	Media	Media	El sistema permite a los usuarios autenticados extraer la información de una base de datos a la que se encuentra conectado.
RF12	Clasificar información	Alta	Media	El sistema permite a los usuarios autenticados clasificar la información analizada previamente.
RF13	Analizar la información	Alta	Media	El sistema permite a los usuarios autenticados analizar la información de la

				base de datos a la que se encuentra conectado.
RF14	Mostrar información clasificada	Media	Media	El sistema permite mostrar la información clasificada.
RF15	Crear reporte	Alta	Alta	Todos los usuarios registrados en el sistema pueden crear reportes.
RF16	Mostrar reporte	Alta	Alta	El sistema muestra los reportes creados.
RF17	Exportar reporte	Alta	Alta	El sistema permite exportar los reportes creados en formato pdf.
RF18	Salvar reporte	Alta	Alta	Todos los usuarios registrados en el sistema podrán guardar los reportes.
RF19	Eliminar reporte	Alta	Alta	El sistema permite eliminar los reportes creados.

2.2.2 Requisitos no funcionales

Son propiedades o cualidades que el producto debe tener, también son conocidos como atributos de calidad. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable; normalmente están vinculados a RF (Niño, 2018).

La tabla que se ilustra a continuación refleja los requisitos no funcionales (RnF) definidos en la propuesta de solución.

Tabla 4: Requisitos no funcionales [Elaboración propia]

Requisitos no funcionales	
Seguridad	<p>RnF1 El sistema permite acceder a las funcionalidades según el rol designado a cada usuario.</p> <p>RnF2 Existirán tres administradores del sistema.</p> <p>RnF3 Para acceder al sistema hay que realizar un proceso de autenticación.</p> <p>RnF4 La contraseña correspondiente a cada usuario debe tener un mínimo de ocho caracteres, combinación de letras, números y caracteres especiales.</p> <p>RnF5 Cada usuario podrá acceder a la información correspondiente a su rol.</p>
Rendimiento/Eficiencia	<p>RnF6 El tiempo de respuesta del sistema al realizar cualquier operación no debe exceder los 5 segundos.</p>
Confiabilidad	<p>RnF7 El sistema garantiza la integridad de la información.</p>
Soporte	<p>RnF8 El sistema permite añadir nuevas funcionalidades de ser necesario.</p>
Software	<p>RnF9 Uso de lenguaje de programación PHP en su versión 8.</p> <p>RnF10 Uso de lenguaje de programación CSS en su versión 3.</p> <p>RnF11 Uso de lenguaje de programación HTML en su versión 5.</p> <p>RnF12 Uso de framework de desarrollo Laravel en su versión 8.</p>

	RnF13 Uso de servidor web Apache en su versión 2.4. RnF14 Uso de PostgreSQL en su versión 13.1.
Portabilidad	RnF15 Se puede acceder al sistema mediante los navegadores web Mozilla Firefox, Ópera o Google Chrome.
De hardware	RnF16 Para la interfaz web: 4 GB RAM, CPU de 2 núcleos y al menos 1000 GB de Disco Duro.
De interfaz	RnF17 La interfaz es agradable a la vista del usuario.

2.2.3 Especificación de requisitos mediante HU

La metodología AUP-UCI en el escenario 4 propone encapsular los requisitos funcionales mediante las Historias de Usuario (HU) para una mayor especificación de los mismos. El empleo de esta técnica describió qué se espera a la salida y los beneficios obtenidos por el usuario final. A continuación, se muestran las HU correspondientes a los requisitos funcionales Registrar usuario, Autenticar usuario, Crear reporte y Salvar reporte.

Historia de usuario

Número: 1

Nombre: Registrar usuario

Usuario: Cliente

Prioridad en negocio: Alta

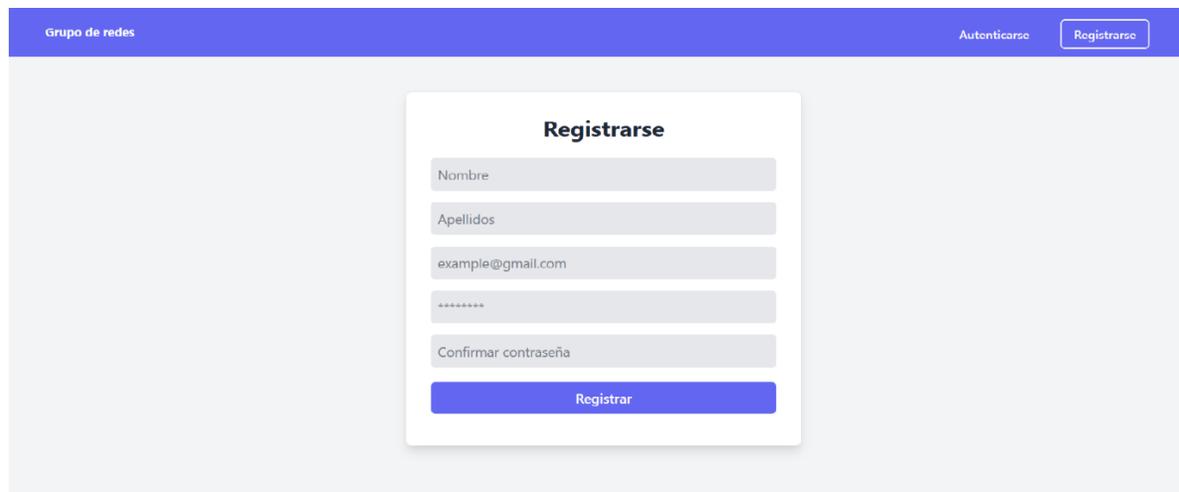
Riesgo en desarrollo: Alta

Programador responsable: Victor José Remedios Pérez

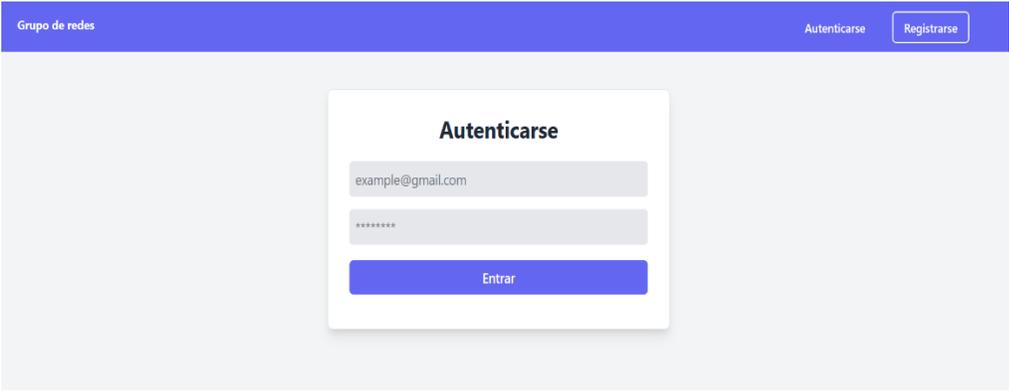
Descripción: El sistema debe permitir al administrador registrar nuevos usuarios. (Datos de registro: Nombre, Apellidos, Correo)

Observaciones: Muestra a los usuarios registrados en el sistema.

Interfaz:



The screenshot shows a web interface for user registration. At the top, there is a blue header bar with the text "Grupo de redes" on the left, "Autenticarse" in the center, and a "Registrarse" button on the right. Below the header, the main content area is light gray. In the center, there is a white registration form titled "Registrarse". The form contains five input fields: "Nombre", "Apellidos", "example@gmail.com", "*****", and "Confirmar contraseña". At the bottom of the form is a blue "Registrar" button.

Historia de usuario	
Número: 2	Nombre: Autenticar usuario
Usuario: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Programador responsable: Victor José Remedios Pérez	
Descripción: Cuando el usuario entra al sistema se muestra la vista de autenticación, en la cual debe introducir su usuario y su contraseña y luego pulsar en el botón entrar.	
Observaciones: Muestra la vista principal dentro de la herramienta en la cual aparece toda la información del usuario en cuestión.	
Interfaz: 	

Historia de usuario	
Número: 15	Nombre: Crear reportes
Usuario: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Programador responsable: Victor José Remedios Pérez	
Descripción: El sistema debe permitir a los usuarios autenticados crear reportes (Semanal, Quincenal, Mensual, Trimestral). Seleccionada una opción das clic en el botón Aceptar y se crea el reporte.	
Observaciones: Muestra la lista de reportes creados en el sistema.	
Interfaz: 	

Historia de usuario	
Número: 18	Nombre: Salvar reporte
Usuario: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Programador responsable: Víctor José Remedios Pérez	
Descripción: El sistema debe permitir al usuario guardar el reporte una vez creado.	
Observaciones: El sistema muestra la lista de las últimas 3 salvas guardadas.	
Interfaz: <div style="text-align: center;">  </div>	

2.3 Modelo de diseño

La metodología de desarrollo de software AUP-UCI brinda la posibilidad de realizar diseños sencillos y simples para una mayor comprensión del desarrollador en la fase de implementación, lo cual le permite realizar la tarea en un menor tiempo.

2.3.1 Patrón arquitectónico

En la propuesta de solución se utiliza Laravel el cual nos ofrece varios tipos de patrones. Uno de ellos es el Modelo Vista Controlador (MVC), el cual goza de gran popularidad entre los desarrolladores gracias a que promueve la organización de las líneas de código y la reutilización de estas.

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo ("Servicio de informática ASP.NET MVC 3 Framework," 2012).

La Vista, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.

El Controlador, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

A continuación, se muestra la representación del Modelo Vista Controlador (MVC).

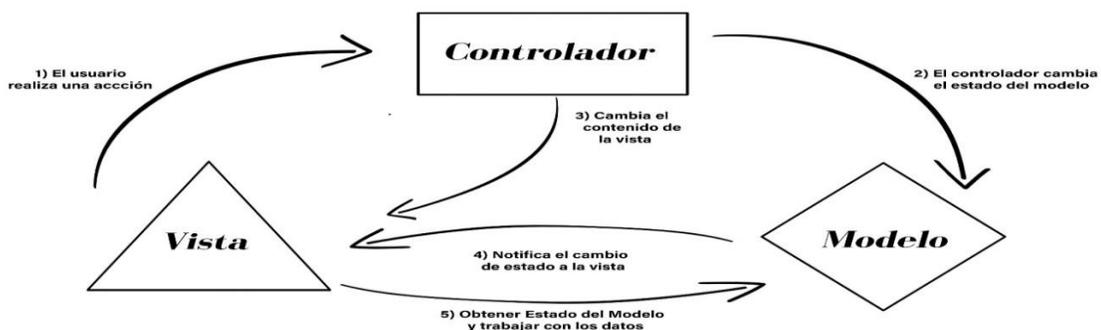


Ilustración 1: Representación del MVC [Elaboración propia]

2.3.2 Estilo arquitectónico

Los estilos arquitectónicos son una generalización y abstracción de los patrones de diseño. Caracteriza una familia de sistemas que están relacionados por compartir propiedades estructurales y funcionales. También puede definirse como la descripción de los tipos componente y de los patrones de interacción entre ellos (Cristiá, 2021).

El estilo Llamada-Retorno usa una estructura implícita de subsistemas. Los datos son pasados como parámetros y el manejador principal proporciona un ciclo de control sobre las subrutinas. Permite al diseñador del software construir una estructura de programa relativamente fácil de modificar y ajustar a escala. Se basa en la bien conocida abstracción de procedimientos/funciones/métodos. Es utilizado en grandes sistemas de software. La descomposición en módulos disminuye la complejidad ("Estilos arquitectónicos," 2021).

Este estilo guarda estrecha relación con el patrón de diseño MVC, el cual fue utilizado en la propuesta de solución. Todo lo expresado anteriormente contribuyó a que se seleccionara como arquitectura el estilo Llamada-Retorno.

2.4 Patrones GRASP

Según el autor Cuello (2018) un patrón es un par problema/solución con nombre que se puede aplicar en nuevos contextos con consejos acerca de cómo aplicarlo en nuevas situaciones y discusiones sobre sus compromisos. Un patrón de diseño es una solución repetible a un problema recurrente en el diseño de software. Esta solución no es un diseño terminado que puede traducirse directamente a código, sino más bien una descripción sobre cómo resolver el problema.

En la propuesta de solución fueron necesarios utilizar los siguientes patrones GRASP:

Creador: El Patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que conecte con el objeto producido en cualquier evento. La nueva instancia del objeto deberá ser creada por la clase que: tiene la información necesaria para realizar la creación del objeto, usa directamente las instancias creadas del objeto, o almacena o maneja varias instancias de la clase. Este patrón brinda soporte de bajo acoplamiento, lo cual supone menos dependencias entre clases y

posibilidades. Este patrón se usa en la propuesta de solución mediante una clase controladora que se encarga de solicitarle a las clases entidades la información pedida por las clases interfaces.

Experto: Asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.

Controlador: Es la clase encargada de asignar la responsabilidad de manejar los eventos de un sistema a una clase que represente un sistema global.

Bajo acoplamiento: Este patrón asigna la responsabilidad de controlar el flujo de eventos del sistema a clases específicas. Esto facilita la centralización de actividades. La clase controladora no realiza las actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Las clases que se encuentran en la capa del modelo no tienen mucha dependencia con las demás clases presentes en la Vista o el Controlador, garantizando bajo acoplamiento.

Alta cohesión: El patrón alta cohesión describe cuan fuertemente los contenidos internos de una rutina están relacionados entre sí. Este patrón propone asignar la responsabilidad de manera que la complejidad se mantenga dentro de límites manejables asumiendo solamente las responsabilidades que deben manejar, evadiendo un trabajo excesivo.

2.5 Modelo de datos

Un diagrama entidad-relación, también conocido como modelo entidad relación o ERD, es un tipo de diagrama de flujo que ilustra cómo las "entidades", como personas, objetos o conceptos, se relacionan entre sí dentro de un sistema. Los diagramas ER se usan a menudo para diseñar o depurar bases de datos relacionales en los campos de ingeniería de software, sistemas de información empresarial, educación e investigación. También conocidos como los ERD o modelos ER, emplean un conjunto definido de símbolos, tales como rectángulos, diamantes, óvalos y líneas de conexión para representar la interconexión de entidades, relaciones y sus atributos. Son un reflejo de la estructura gramatical y emplean entidades como sustantivos y relaciones como verbos ("¿Qué es un diagrama entidad relación?," 2021).

Los diagramas de ER se relacionan con los diagramas de estructura de datos (DSD), que se centran en las relaciones de los elementos dentro de las entidades, en lugar de las relaciones entre las entidades mismas.

Los diagramas ER a menudo se combinan con los diagramas de flujo de datos (DFD), que trazan el flujo de la información para procesos o sistemas ("¿Qué es un diagrama entidad relación?," 2021).

A continuación, se muestra el modelo de datos:

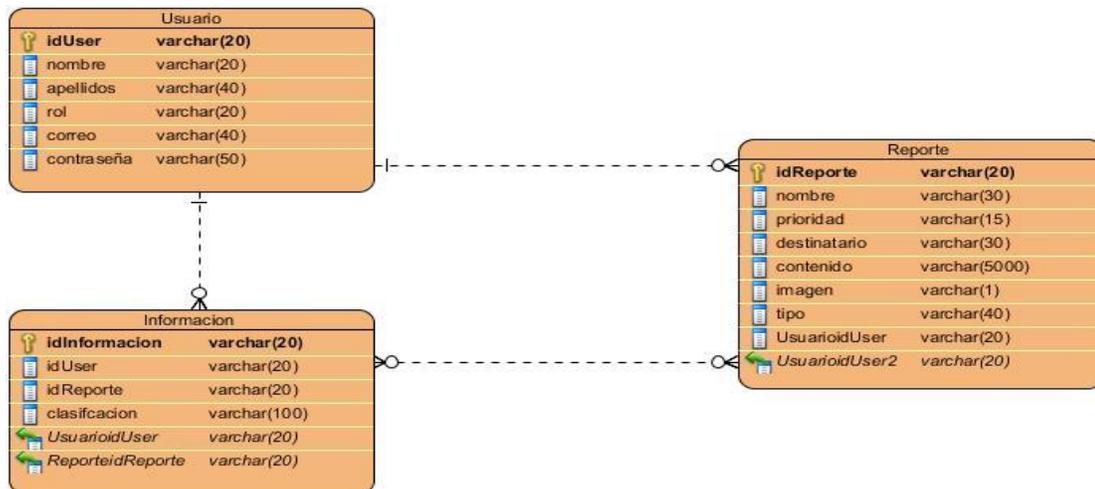


Ilustración 2: Representación del Modelo de Datos [Elaboración propia]

2.6 Diagrama de clase

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés) puede ayudarte a modelar sistemas de diversas formas. Uno de los tipos más populares en el UML es el diagrama de clases. Popular entre los ingenieros de software para documentar arquitectura de software, los diagramas de clases son un tipo de diagrama de estructura porque describen lo que debe estar presente en el sistema que se está modelando. Sin importar tu nivel de familiaridad con diagramas UML o diagramas de clases, nuestro software UML está diseñado para ser simple y fácil de usar ("¿Qué es el lenguaje unificado de modelado?," 2021).

El UML se estableció como un modelo estandarizado para describir un enfoque de Programación Orientada a Objetos (POO). Como las clases son los componentes básicos de los objetos, los diagramas de clases son los componentes básicos del UML. Los diversos componentes en un diagrama de clases pueden representar las clases que se programarán en realidad, los objetos principales o la interacción entre clases y objetos ("¿Qué es el lenguaje unificado de modelado?," 2021).

La figura de clase en sí misma consiste en un rectángulo de tres filas. La fila superior contiene el nombre de la clase, la fila del centro contiene los atributos de la clase y la última expresa los métodos o las operaciones que la clase puede utilizar. Las clases y las subclasses se agrupan para mostrar la relación estática entre cada objeto ("¿Qué es el lenguaje unificado de modelado?," 2021).

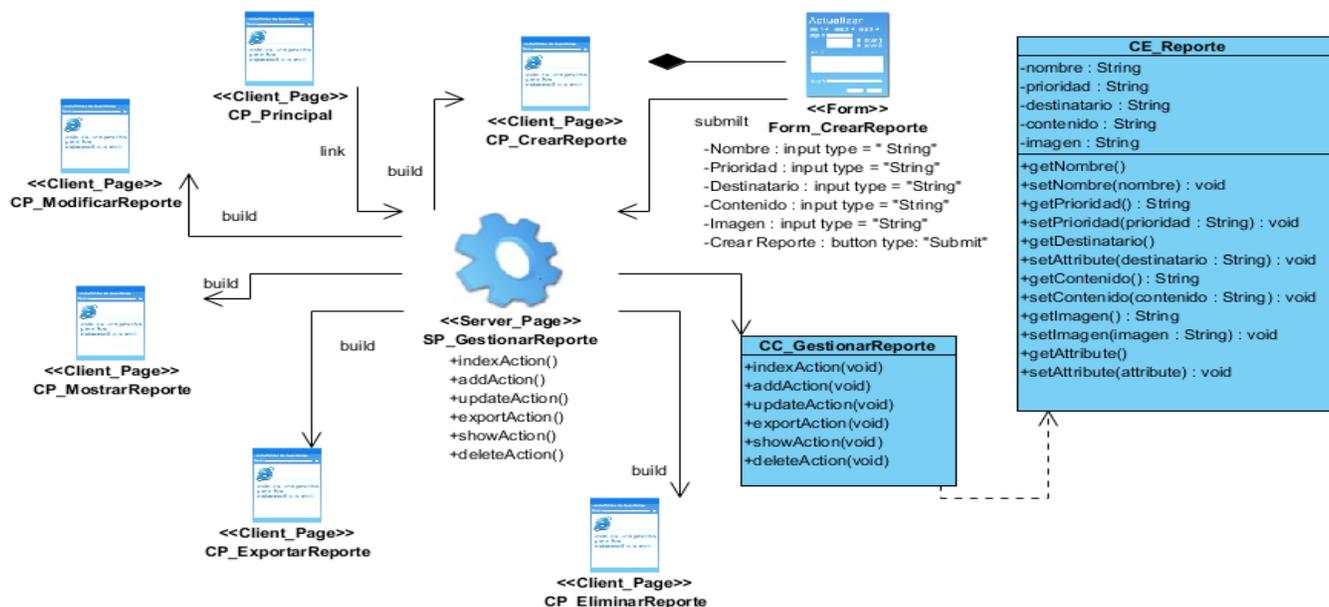


Ilustración 3: Representación del Diagrama de Clases [Elaboración propia]

2.7 Diagrama de secuencia

Los diagramas de secuencia se pueden modelar a partir de los casos de uso pues al tener identificados los actores y las actividades que realiza, en este diagrama se plasma el intercambio de los mensajes, entradas y salidas entre los elementos de nuestro sistema. Se encuentra dentro de la vista de interacción pues proporciona una vista integral del comportamiento de un sistema. Un diagrama de secuencia es una forma de diagrama de interacción que muestra los objetos como líneas de vida a lo largo de la página y con sus interacciones en el tiempo representadas como mensajes dibujados como flechas desde la línea de vida origen hasta la línea de vida destino. Los diagramas de secuencia son buenos para mostrar qué objetos y qué mensajes disparan esas comunicaciones. Los diagramas de secuencia no están pensados para mostrar lógicas de procedimientos complejos. A continuación, se muestra el diagrama de secuencia del requisito Crear reporte.

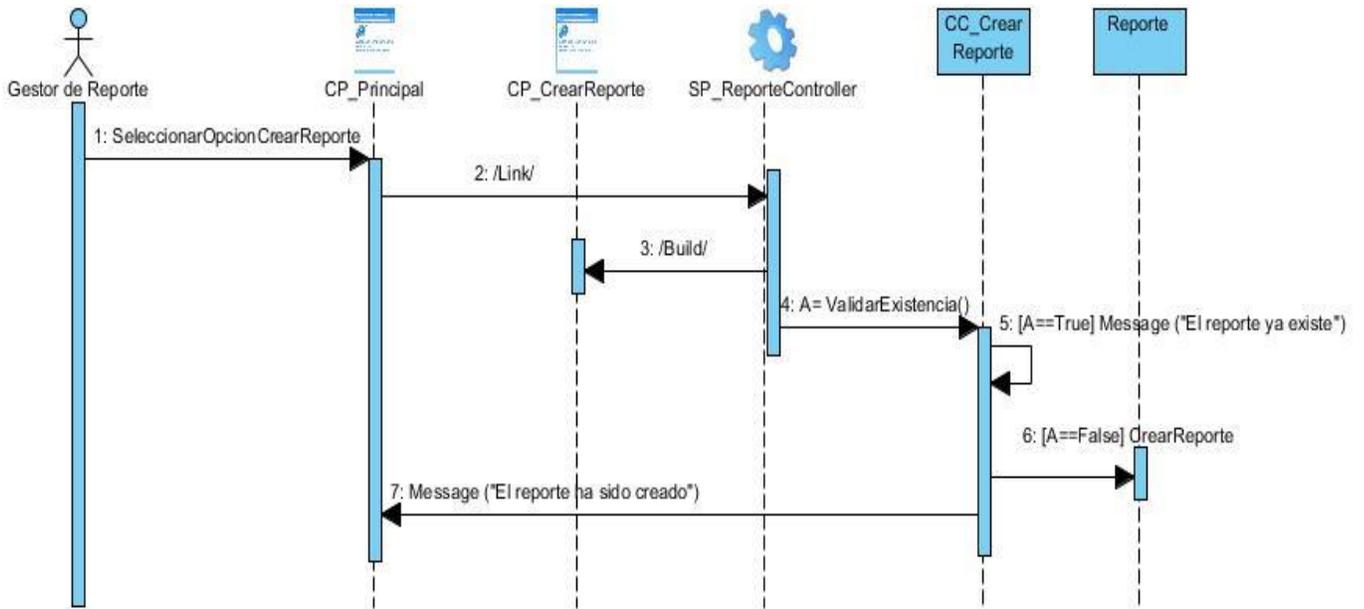


Ilustración 4: Representación del Diagrama de Secuencia [Elaboración propia]

Conclusiones Parciales

En el presente capítulo se realizó un estudio más detallado sobre la propuesta de solución mediante:

- ✓ El encapsulamiento de los requerimientos funcionales y no funcionales a través de las historias de usuario permitió una mayor comprensión de la propuesta de solución.
- ✓ La definición del patrón arquitectónico MVC, el estilo arquitectónico Llamada-Retorno y los patrones GRASP permiten que el código fuente de la propuesta de solución sufra la menor cantidad de impactos negativos en el futuro.
- ✓ Los diagramas de ER, de clases y de secuencia contribuyeron a entender de manera más eficiente la lógica del negocio, lo cual es muy importante para el desarrollo de la propuesta de solución.

Capítulo 3: Implementación y prueba de la herramienta para el análisis y clasificación de información recopilada en medios de difusión

Introducción

En el presente capítulo se muestran los diagramas de despliegue y de componentes. Se abordan los estándares de codificación y se plantea una estrategia de pruebas con el objetivo de comprobar la calidad de la propuesta de solución.

3.1 Diagrama de despliegue

Un diagrama de despliegue es un tipo de diagrama UML que muestra la arquitectura de ejecución de un sistema, incluyendo nodos como entornos de ejecución de hardware o software, y el middleware que los conecta (Siriwardhana, 2021).

Los diagramas de despliegue se utilizan normalmente para visualizar el hardware y el software físico de un sistema. Usándolo puedes entender cómo el sistema se desplegará físicamente en el hardware (Siriwardhana, 2021).

Los diagramas de despliegue ayudan a modelar la topología de hardware de un sistema en comparación con otros tipos de diagramas UML, que en su mayoría esbozan los componentes lógicos de un sistema (Siriwardhana, 2021).

Un nodo, representado como un cubo, es una entidad física que ejecuta uno o más componentes, subsistemas o ejecutables. Un nodo podría ser un elemento de hardware o software, está representado por una línea sólida entre dos nodos. Muestra el camino de la comunicación entre los nodos (Siriwardhana, 2021).

A continuación, se muestra el diagrama de despliegue de la propuesta de solución.

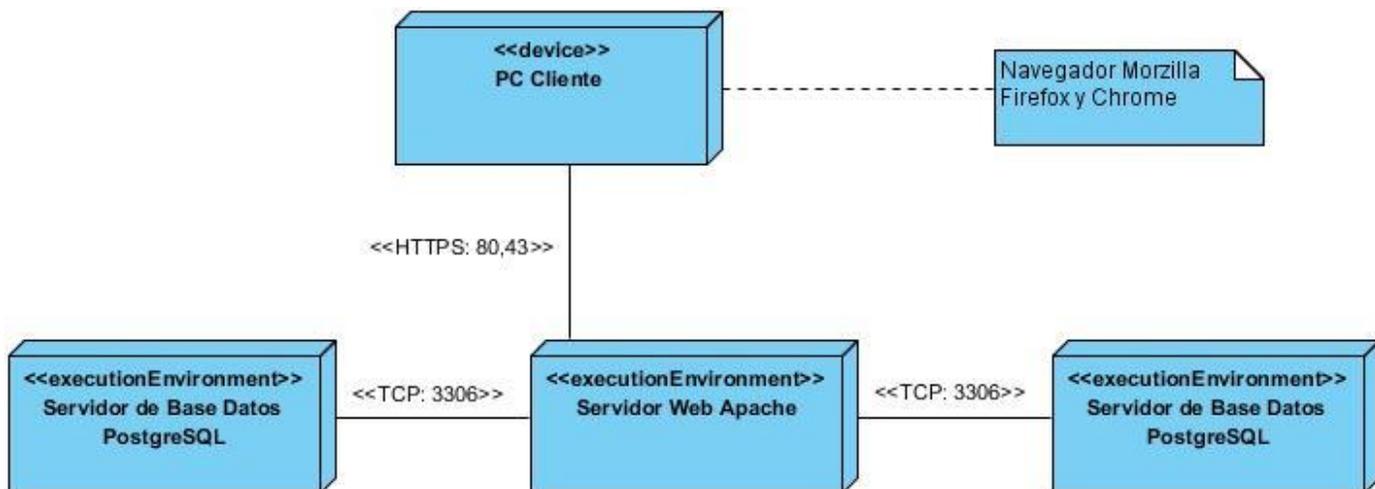


Ilustración 5: Representación del Diagrama de Despliegue [Elaboración propia]

3.2 Diagrama de componentes

Los diagramas de componentes UML representan las relaciones entre los componentes individuales del sistema mediante una vista de diseño estática. Pueden ilustrar aspectos de modelado lógico y físico.

En el contexto del UML, los componentes son partes modulares de un sistema independientes entre sí, que pueden reemplazarse con componentes equivalentes. Son autocontenidos y encapsulan estructuras de cualquier grado de complejidad. Los elementos encapsulados solo se comunican con los otros a través de interfaces. Los componentes no solo pueden proporcionar sus propias interfaces, sino que también pueden utilizar las interfaces de otros componentes, por ejemplo, para acceder a sus funciones y servicios. A su vez, las interfaces de un diagrama de componentes documentan las relaciones y dependencias en una arquitectura de software ("Diagrama de componentes: modelado eficiente de sistemas con módulos de software," 2020).

En la siguiente ilustración, se muestra el diagrama de componentes.

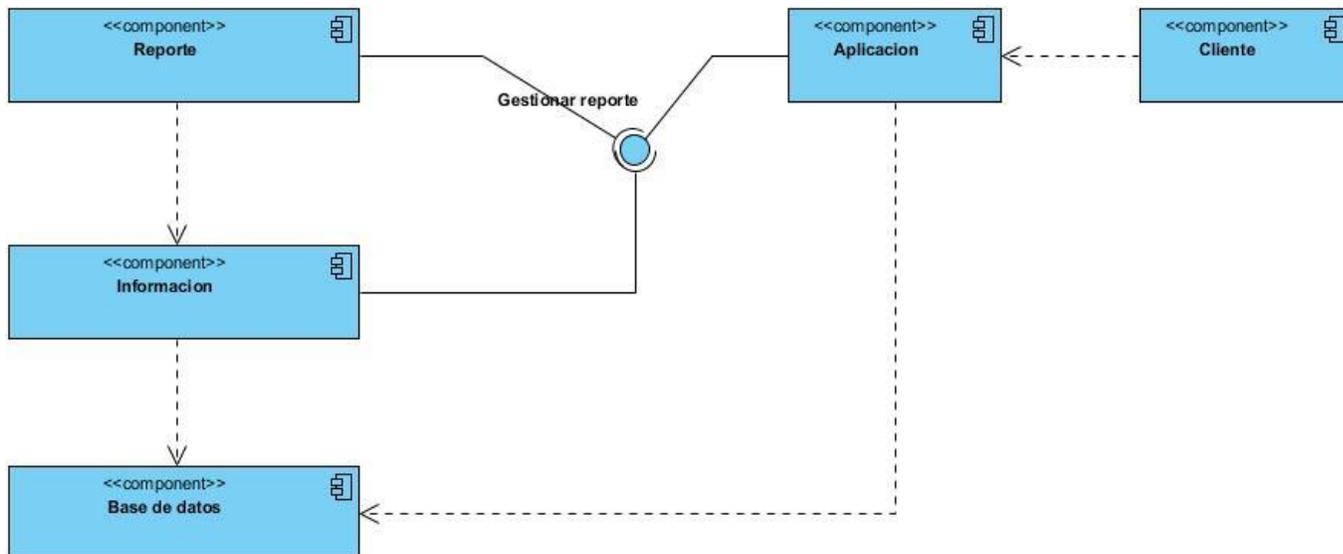


Ilustración 6: Representación del Diagrama de Componente [Elaboración propia]

3.3 Estándares de codificación

Laravel sigue los estándares PSR-1 y PSR-4. Y además tiene algunas recomendaciones propias. Lo que en algunos entornos llaman el 'Laravel «flavor» of PSR-2'. Estos son los estándares extraídos de su guía de colaboración.

La declaración del namespace debe estar en la misma línea que `<?php`:

```
<?php namespace Curso\Http\Controllers;
```

No existe en los estándares PSR una regla acerca de esto.

Las llaves de apertura de las clases deben ir en la misma línea que el nombre de la clase.

```
class SQLiteConnection extends Connection {  
  
    protected function getDefaultQueryGrammar()  
  
    {
```

```
    return $this->withTablePrefix(new QueryGrammar);  
  
}  
  
(...)
```

En PSR-2 recomiendan que la llave vaya en la línea siguiente.

Las funciones y estructuras de control deben seguir el estilo de llaves Allman. El estilo Allman define que la llave de apertura de las estructuras de control debe 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.

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

En PSR-2 recomiendan que la llave de apertura vaya en la misma línea separada por un espacio del paréntesis de cierre de la declaración.

Indentar con tabuladores y alinear con espacios.

En PSR-2 recomiendan indentar con espacios.

Si usas PHP-Sniffer, este paquete te permite comprobar si tu código sigue los estándares de Laravel y reformatea tu código para que sea compatible con ellos: [Laravel PHP_CodeSniffer](#)

Este es otro paquete que te ayuda a formatear tu código para que siga los estándares PSR-1 y PSR-2 y GrahamCampbell está entre uno de sus colaboradores: [FriendsOfPHP/PHP-CS-Fixer](#)

Nombres de las rutas. Usa nombres en línea con las convenciones internas de Laravel:

users.index

users.create

users.store

users.show

users.edit

users.update

users.destroy

Nombres de los modelos Eloquent. Laravel usa como convención para los modelos la notación StudlyCaps y para el nombre de la tabla relacionada con tu modelo, el nombre del modelo en minúsculas y plural. Por ejemplo, para class User extends Eloquent (González; & Cabrera, 2015), Laravel buscará una tabla llamada users. Y como convención, usará como clave primaria un campo llamado id. Para los nombres de las columnas en la tabla, usa la notación snake_case. Y para los métodos de los modelos, usa camelCase. Por ejemplo, para una columna llamada nombre_completo, el método sería getNombreCompletoAttribute(\$name). Para las tablas pivots, el convenio es usar el singular de ambas tablas unidas por un guion bajo. Por ejemplo: user_product. Ver **Anexo 3**.

3.4 Pruebas de software

El ISTQB (International Software Testing Qualifications Board), define las pruebas como: “El proceso que consiste en todas las actividades del ciclo de vida, tanto estáticas como dinámicas relacionadas con la planificación, preparación y evaluación de productos de software y productos relacionados con el trabajo para determinar que cumplen los requisitos especificados, para demostrar que son aptos para el propósito y para detectar defectos”.

Según el autor Peño (2015) las define como: “Las pruebas de software son la investigación empírica y técnica realizada para facilitar a los interesados información sobre la calidad del producto o servicio bajo pruebas”.

Las pruebas no funcionales se enfocan en validar un sistema o aplicación por medio de sus requerimientos no funcionales, es decir, la forma en que el sistema funciona y no por medio de comportamientos específicos.

A continuación, se muestran las características de calidad y tipos de pruebas del sistema según Godoy (2020) :

Funcionalidad: Consisten en la revisión de los requisitos aceptados por el cliente a partir de las funcionalidades presentes en la aplicación.

Usabilidad: Prueba enfocada a factores humanos, estéticos, consistencia en la interfaz de usuario.

Seguridad: Asegurar que los datos o el sistema solamente son accedidos por los actores definidos según niveles de acceso.

Portabilidad: Enfocada a asegurar que funciona en diferentes configuraciones de hardware y software.

Confiabilidad: Verificar que los procesos de recuperación llevan a un estado conocido o deseado, informando al usuario a través de mensajes.

Mantenibilidad: Se verifica la facilidad con la que el producto puede ser modificado para corregir defectos o cumplir con nuevos requisitos.

3.5 Estrategia de pruebas

Para verificar el cumplimiento de los RnF descritos fue diseñada una lista de chequeo [Anexo A], identificándose indicadores a chequear para la evaluación de cada requisito. En el caso de los requisitos de eficiencia las pruebas se realizarán en el entorno real de la aplicación por lo que no se mostrarán resultados de estas como parte de la investigación.

En el desarrollo de las pruebas intervinieron los siguientes roles:

- ✓ Coordinador de la prueba
- ✓ Probador
- ✓ Jefe de equipo de desarrollo
- ✓ Desarrollador

En la tabla 5 puede observarse el cronograma planificado en el que se recogen las especificaciones de cada actividad del proceso de prueba.

Tabla 5: Cronograma de planificación para el proceso de pruebas: [Elaboración propia]

Actividad	Fecha	Responsable	Participantes
Elaboración de la estrategia de prueba	Día 1	Coordinador de la prueba	Coordinador de la prueba
Diseñar la prueba	Día 1	Coordinador de la prueba	Coordinador de la prueba
Montaje del entorno de prueba	Día 2	Coordinador de la prueba	Coordinador de la prueba
Primera iteración de las pruebas	Día 2	Coordinador de la prueba	Probador
Solución de los defectos en los artefactos en prueba y actualización de los CP y artefactos de apoyo	Día 3	Jefe de equipo de desarrollo	Desarrollador
Segunda iteración de las pruebas	Día 4	Coordinador de la prueba	Probador
Evaluación de los resultados de las pruebas	Día 5	Coordinador de la prueba	Probador

3.5.1 Pruebas de caja negra

Las Pruebas de Caja Negra, constituyen 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. Las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa (Pressman, 2010).

Las pruebas de caja negra intentan encontrar errores en las categorías siguientes según Pressman (2010):

- ✓ Funciones incorrectas o faltantes.
- ✓ Errores de interfaz.
- ✓ Errores en las estructuras de datos o en el acceso a bases de datos externas.
- ✓ Errores de comportamiento o rendimiento.
- ✓ Errores de inicialización y terminación.

3.5.2 Prueba de partición equivalente

Este método de prueba de caja negra divide el dominio de entrada de un programa en clases de datos, a partir de las cuales deriva los casos de prueba. Cada una de las clases de equivalencia representa a un conjunto de estados válidos o inválidos para las condiciones de entrada (González & Fernández, 2015).

Implementación

Se plantearon dos iteraciones. Al terminar cada iteración se realiza la revisión del plan de iteraciones y se ajusta a partir de los cambios identificados y aprobados. Las HU se realizan en correspondencia con la asignación de tareas de desarrollo a un equipo o persona. Las tareas se escriben en lenguaje técnico debido a que en esta actividad participa de manera directa el equipo de desarrollo.

Iteración 1:

Se implementaron 5 de las 5 HU planificadas para esta iteración.

Iteración 2:

Se implementaron 14 de las 14 HU planificadas para esta iteración.

En la siguiente tabla puede observarse el tiempo de implementación real de algunas de las HU implementadas en ambas iteraciones.

Tabla 6: CP asociado a HU: [Elaboración propia]

Historia de usuario	Tiempo estimado	Tiempo real	Iteración
Registrar usuario	18	14	1
Autenticar usuario	15	12	1
Modificar usuario	19	15	1
Listar usuario	12	9	1
Eliminar usuario	10	8	1

Tabla 7: Descripción de las variables: [Elaboración propia]

No.	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	campo de texto	No	Cadena de caracteres alfabéticos, sin incluir números.
2	Apellidos	campo de texto	No	Cadena de caracteres alfabéticos, sin incluir números.
3	Correo	campo de texto	No	Cadena de caracteres alfabéticos, sin incluir números.

4	Contraseña	campo de texto	No	Cadena de caracteres alfanuméricos.
5	Confirmar Contraseña	campo texto	No	Cadena de caracteres alfanuméricos.

3.5.3 Casos de prueba

Las descripciones de requisitos brindaron la posibilidad de diseñar los casos de prueba correspondientes, constituyendo la guía principal del probador. En este sub-epígrafe se muestran los casos de prueba correspondientes a los requisitos Registrar, Autenticar, Eliminar y Mostrar patrón.

Descripción de caso de prueba (DCP) del requisito Registrar Usuario

Descripción general

El requisito comienza cuando el usuario selecciona la opción “Registrar Usuario”, y culmina cuando se ha registrado un nuevo usuario.

Tabla 8: Caso de prueba del requisito Registrar Usuario

Escenario	Descripción	Nombre	Apellidos	Correo	Contraseña	Confirmar contraseña	Respuesta del sistema	Flujo central
EC 1.1 Selecciona la opción Registrar usuario	Muestra un formulario Registrar usuario	NA	NA	NA	NA	NA	Muestra los datos que se deben introducir para registrar un nuevo usuario y la opción de Registrar.	Inicio/Registrar
EC 1.2 Selecciona el botón Entrar	Introduce los datos y selecciona la opción Registrar	V	V	V	V	V	Valida los datos introducidos.	Inicio/Registrar/ Aceptar
EC 1.3 Existen campos vacíos o datos incorrectos	Existen campos con datos incorrectos	I	V	V	V	V	No deja registrar el usuario	Inicio/Registrar
		V	I	V	V	V		
		V	V	I	V	V		
		V	V	V	I	V		
		V	V	V	V	I		
		I	I	I	I	I		

Descripción de caso de prueba (DCP) del requisito Autenticar Usuario

Descripción general

El requisito comienza cuando el usuario selecciona la opción “Autenticar Usuario”, y culmina cuando se ha autenticado un nuevo usuario.

Tabla 9: Caso de prueba del requisito Autenticar Usuario

Escenario	Descripción	Correo	Contraseña	Respuesta del sistema	Flujo central
EC 1.1 Selecciona la opción Autenticar usuario	Muestra un formulario Autenticar usuario	NA	NA	Muestra los datos que se deben introducir para autenticar un usuario y la opción de Entrar.	Inicio/Autenticar
EC 1.2 Selecciona el botón Entrar	Introduce los datos y selecciona la opción Entrar	V	V	Valida los datos introducidos.	Inicio/Autenticar
EC 1.3 Existen campos vacíos o datos incorrectos	Existen campos con datos incorrectos	I	V	No deja autenticar el usuario y muestra un mensaje de error.	Inicio/Autenticar
		V	I		
		I	I		

Descripción de caso de prueba (DCP) del requisito Editar Usuario

Descripción general

El requisito comienza cuando el usuario selecciona la opción “Editar Usuario”, y culmina cuando se ha editado el usuario.

Tabla10: Caso de prueba del requisito Editar Usuario

Escenario	Descripción	Nombre	Apellidos	Correo	Contraseña	Confirmar contraseña	Respuesta del sistema	Flujo central
EC 1.1 Selecciona la opción Editar usuario	Muestra un formulario Editar usuario	NA	NA	NA	NA	NA	Muestra los datos que se deben introducir para editar un nuevo usuario y la opción de Entrar.	Inicio/ListarUsuario/ Editar
EC 1.2 Selecciona el botón Entrar	Introduce los datos y selecciona la opción Entrar	V	V	V	V	V	Valida los datos introducidos.	Inicio/ListarUsuario/ Editar
EC 1.3 Existen campos vacíos o datos incorrectos	Existen campos con datos incorrectos	I	V	V	V	V	No deja modificar el usuario	Inicio/ListarUsuario/ Editar
		V	I	V	V	V		
		V	V	I	V	V		
		V	V	V	I	V		
		V	V	V	V	I		
		I	I	I	I	I		

DCP del requisito Eliminar usuario

Descripción general

El requisito comienza cuando el usuario selecciona la opción “Eliminar”, y culmina cuando se ha eliminado el usuario.

Condiciones de ejecución

El usuario debe estar autenticado en el sistema y el usuario a eliminar previamente creado.

Tabla 11: Caso de prueba del requisito Eliminar Usuario

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Selecciona la opción “Listar usuario”.	Muestra un listado con los usuarios.	Muestra un listado de usuarios almacenados en el sistema.	Inicio/ListarUsuario/ Eliminar
EC 1.2 Selecciona la opción Eliminar.	Selecciona el botón que dice eliminar.	Elimina el usuario.	Inicio/ListarUsuario/ Eliminar

3.5.4 Clasificación de las no conformidades

Una no conformidad es un fallo en el sistema de gestión de la calidad que puede producirse por varias razones: no alcanzar el nivel de aceptación establecido en un determinado indicador y errores en la documentación del sistema. Se trata de una desviación entre lo que hay escrito y lo que ha ocurrido. Este fallo queda registrado en un informe y se establecen las acciones preventivas y correctivas necesarias para arreglar lo que no funcione y evitar que vuelva a ocurrir. Las mismas se clasifican de acuerdo al nivel de importancia en (González; & Cabrera, 2015):

1. **Significativas:** Son aquellas que afectan la calidad del producto o servicio de manera visible, impidiendo o no el cumplimiento de algún requisito.

2. **No significativas:** Son aquellas que resultan menos visibles, que no atentan contra el cumplimiento de algún requisito.
3. **Recomendaciones:** Son aquellas que quedan en función de la apreciación del probador para oportunidades de mejoras del producto o servicio.

3.5.5 Resultados de la prueba de Caja Negra

Se efectuaron dos iteraciones de pruebas donde fueron detectados diferentes tipos de errores y se le dieron solución. Durante las pruebas funcionales se detectaron doce no conformidades, seis significativas, cuatro no significativas y dos recomendaciones. Durante la primera iteración de detectaron tres no conformidades significativas, dos no significativas y dos recomendaciones. En la segunda iteración se detectaron 2 no conformidades significativas, cero no significativas y una recomendación.

3.6 Técnica de ladov

La técnica de V.A. ladov en su versión original fue creada por su autor para el estudio de la satisfacción por la profesión en carreras pedagógicas. Constituye una vía indirecta para el estudio de la satisfacción, ya que los criterios que se utilizan se fundamentan en las relaciones que se establecen entre tres preguntas cerradas que se intercalan dentro de un cuestionario y cuya relación el sujeto desconoce. Estas tres preguntas se relacionan a través de lo que se denomina el cuadro lógico de ladov (González; & Cabrera, 2015).

A continuación, se muestra el cuadro lógico de ladov con las tres preguntas del cuestionario referentes a la satisfacción con la propuesta de solución:

Tabla 12: Cuadro lógico de ladov para usuarios potenciales [Elaboración propia]

3- ¿Le gusta la solución propuesta para generar reportes?	1- ¿Considera usted que es necesario una herramienta para la clasificación y el análisis de la información recopilada en los medios de difusión web?		
	No	No sé	Si
	2- Si usted va a crear un reporte de los datos clasificados y analizados recopilados en los medios de difusión web, ¿utilizaría la herramienta desarrollada?		

	Si	No sé	No	Si	No sé	No	Si	No sé	No
Me gusta mucho	1	2	6	2	2	6	6	6	6
Me gusta más de lo que me disgusta	2	2	3	2	3	3	6	3	6
Me es indiferente	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta en lo absoluto	6	6	6	6	4	4	6	4	5
No sé	2	3	6	3	3	3	6	3	4

La escala de satisfacción es la siguiente:

1. Clara satisfacción
2. Más satisfecho que insatisfecho
3. No definida
4. Más insatisfecho que satisfecho
5. Clara insatisfacción
6. Contradictoria

Para medir el grado de satisfacción se tomó una muestra de 6 usuarios, que mantuvieron acceso continuo al portal web. El resultado obtenido se muestra a continuación:

Tabla 13: Satisfacción de usuarios potenciales [Elaboración propia]

Resultado	Cantidad
Máximo de satisfacción	2
Más satisfecho que insatisfecho	3
No definido y contradictorio	0

Más insatisfecho que satisfecho	1
Máxima insatisfacción	0

Con el objetivo de obtener el Índice de Satisfacción Grupal (ISG) se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y - 1 de la siguiente forma:

Tabla 14: Escala numérica para el ISG

Escala	Nivel de satisfacción
1	Máximo de satisfacción
0,5	Más satisfecho que insatisfecho
0	No definido y contradictorio
-0,5	Más insatisfecho que satisfecho
-1	Máxima insatisfacción

Luego es posible calcular el ISG a partir de la siguiente ecuación:

$$ISG = \frac{A(+1) + B(+0,5) + C(0) + D(-0,5) + E(-1)}{N} \quad (1)$$

Donde:

A, B, C, D, E, representan el número de sujetos con su índice individual.

N representa el número total de sujetos del grupo.

El ISG obtenido para los usuarios potenciales fue el siguiente:

$$ISG = \frac{2(+1) + 3(+0,5) + 0(0) + 1(-0,5) + 0(-1)}{6} = 0.5 \quad (2)$$

Como se aprecia, el Índice de Satisfacción Grupal es de 0.5 lo que significa que los usuarios están más satisfechos que insatisfechos con la propuesta de solución, y reconocen su utilidad en el mejoramiento de lo que plantea el marco teórico de esta investigación.

Conclusiones parciales

Tras definir las características de implementación y pruebas de la propuesta de solución, se plantean las siguientes conclusiones:

- ✓ Se elaboraron los diagramas de despliegue y de componentes para mayor entendimiento de la propuesta de solución.
- ✓ Se planteó una estrategia de pruebas y validación para ejecutar una vez implementada la propuesta de solución.
- ✓ La técnica de partición equivalente del método de caja negra arrojó varias no conformidades que fueron corregidas.
- ✓ La técnica de ladov permitió conocer el nivel de satisfacción con respecto a la herramienta a los usuarios.

Conclusiones Generales

1. El análisis y el estudio de la fundamentación teórica de los principales conceptos asociados a la investigación sobre los métodos de clasificación y análisis de la información, permitió comprender como se realiza este proceso en los medios de difusión web.
2. El estudio del entorno de desarrollo permitió la selección de las tecnologías y la metodología más adecuada para el diseño de la herramienta.
3. El estudio de las distintas áreas del conocimiento como son la programación, ingeniería y gestión de software, bases de datos, entre otras, permitieron la elaboración de los artefactos ingenieriles y la implementación de la propuesta de solución.
4. La implementación de una estrategia de pruebas adecuada permitió validar la herramienta una vez desarrollada.

Referencias Bibliográficas

1. Bermeo-Pérez, S. K., & Campoverde-Molina, M. A. (2020). Implementación de inteligencia de negocios, en el inventario de la Cooperativa GranSol, con la herramienta Power BI *FIPCAEC*, 5, 240-266. doi:<https://doi.org/10.23857/fipcaec.v5i14.169>
2. Cabana, J. (2017). Editores de código: Sublime Text y PHPStorm. In.
3. Cristiá, M. (2021). Una Teoría para el Diseño de Software.
4. Cruz, M. E. A. (2017). *Desarrollo de un aplicativo web que permita el análisis de los registros de auditoría del sistema de gestión de base de datos MYSQL utilizando herramientas de Softwre Libre*
5. UNIVERSIDAD DE GUAYAQUIL,
6. Cuello, T. T. (2018). Entorno Integral de Desarrollo para Prácticas de Control Automática en un Sistema de Laboratorios Remotos. 11.
7. Daynelis Brito Morales, Johan Bravo Borrell, & Armas, L. J. (2019). Aplicación móvil para el análisis de la información captada en SIGEv3.0 *Serie Científica de la Universidad de las Ciencias Informáticas* 12, 55-71
8. Diagrama de componentes: modelado eficiente de sistemas con módulos de software. (2020). Retrieved from <https://www.ionos.es/digitalguide/paginas-web/>
9. Díaz, I. V. R., Morillo, Z. M., & Navarro, D. C. R. (2020). Las Tecnologías de la Información, la Educación Superior y el enfrentamiento a la Covid 19.
10. Estilos arquitectónicos. (2021). Retrieved from https://www.ecured.cu/Estilos_arquitect%C3%B3nicos#Arquitectura_de_Llamada_y_retorno
11. Fenollar, N. C. (2019). *Automatización de informes de actividad en redes sociales con Tableau*
12. (Maestría), Escola Tècnica Superior d'Enginyeria Informàtica Universitat Politècnica de València

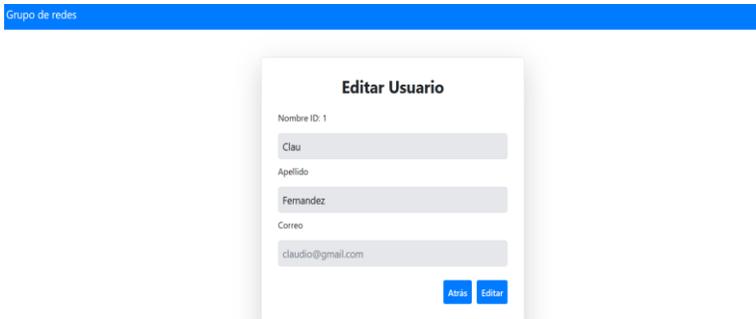
13. Frías, E. R. (2018). Geckoboard – Un software para comprender los datos clave de la empresa. Retrieved from <http://www.geckoboard.com/> (<http://www.geckoboard.com/>)
14. Galván, P., & Zamorano, J. (2020). Herramientas CASE. Retrieved from https://es.wikiversity.org/w/index.php?title=Herramientas_CASE&oldid=156870
15. Godoy, Y. V. (2020). Herramienta para el diseño y gestión de casos de pruebas funcionales. Universidad de las Ciencias Informáticas,
16. González, P. R. P., & Cabrera, C. F. (2015). Módulo para la gestión de patrones de diseño de Recursos Educativos Abiertos. Universidad de las Ciencias Informáticas
17. Guerra, P. N. R. (2018). ¿Que es Bootstrap? Y sus ventajas en el diseño de páginas web. In.
18. Herrera, C. (2020). Ventajas y desventajas de usar JavaScript en la programación web. In.
19. loadview-testing.com. (2021). Pruebas de carga frente a pruebas de estrés. Retrieved from <https://www.loadview-testing.com/es/pruebas-de-carga-vs-pruebas-de-estres/>
20. Martín, G. H., Canchola, V. H. O., & Mayén, J. a. V. A. (2017). Uso de las tecnologías de la información y comunicación (TIC) en las residencias médicas en México. *ACTA MÉDICA GRUPO ÁNGELES*, 15.
21. Microsoft. (2017). XQuery Language Reference (SQL Server). Retrieved from <https://docs.microsoft.com/en-us/sql/xquery/xquery-language-reference-sql-server?view=sql-server-ver15>
22. Ortega, M. A., & Camacho, E. D. (2019). Uso de los modelos tradicionales y las metodologías ágiles aplicadas en la industria de software colombiano
23. Parada, M. (2019). Qué es jQuery. In.
24. Peiró, R. (2017). Lenguaje HTML - Qué es, definición y concepto In.

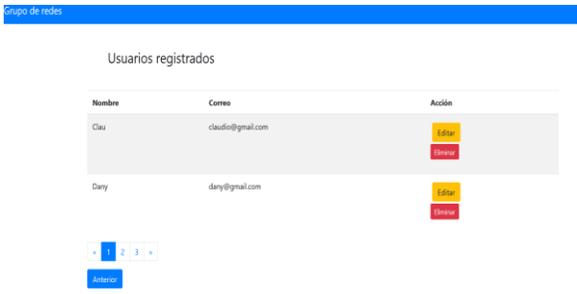
25. Peñalvo, F. J. G., Holgado, A. G., & Ingelmo, A. V. (2020). METODOLOGÍAS DE INGENIERÍA DE SOFTWARE.
26. Peño, J. M. S. (2015). *Pruebas de Software. Fundamentos y Técnicas*, UNIVERSIDAD POLITÉCNICA DE MADRID
27. Pressman, R. S. (2010). *Software Engineering a Practitioner's Approach* (7ma ed.).
28. ¿Qué es el lenguaje unificado de modelado? (2021). Retrieved from <https://www.lucidchart.com/>
29. ¿Qué es un diagrama entidad relación? (2021). Retrieved from <https://www.lucidchart.com>
30. Quiroga-Parra, D. J., Torrent-Sellens, J., & Zorrilla, C. P. M. (2017). Usos de las TIC en América Latina: una caracterización. *Ingeniare. Revista chilena de ingeniería*, 25, 289-305.
31. Requisito. (2019). *Requisito*. Retrieved from <https://www.significados.com/requisito/>
32. Rodríguez, H. (2021). ¿Qué es Laravel?: ¡la mejor opción para desarrollar tus aplicaciones web! In.
33. Sánchez, T. R. (2015). Metodología de desarrollo para la Actividad productiva de la UCI., 15.
34. Sanguino, M. F. V. (2019). *VISUALIZACIÓN DE DATOS DEL SAP BUSINESS WAREHOUSE CON HERRAMIENTAS NO SAP* Universidad de Antioquia
35. Santos, D. (2021). Introducción al CSS: qué es, para qué sirve y otras 10 preguntas frecuentes. In.
36. Servicio de informática ASP.NET MVC 3 Framework. (2012). Retrieved from <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>
37. Siriwardhana, S. (2021). Tutorial de diagrama de despliegue. Retrieved from <https://www.google.com/amp/s/creatly.com/blog/es/diagramas/tutorial-de-diagrama-de-despliegue/amp/>
38. Soto, L. (2021). ¿Qué es PostgreSQL? ¿Y para qué sirve? Retrieved from <https://luissoto.website/que-es-postgresql-y-para-que-sirve/>

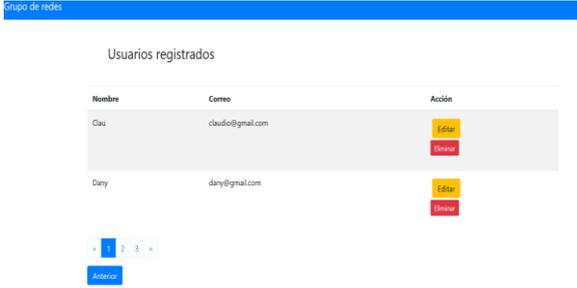
39. Souza, I. (2020). Descubre qué es el lenguaje de programación PHP y en qué situaciones se hace útil. In.
40. Trejo, L. M. (2017). *“Implementación de diseño web bajo metodología ágil para laboratorio químico industrial”* UNIVERSIDAD POLITÉCNICA DE SINALOA PROGRAMA ACADÉMICO DE INGENIERÍA EN INFORMÁTICA
41. Valencia, M. M. A. (2000). *La triangulación metodológica: sus principios, alcances y limitaciones.*, Universidad de Antioquia,
42. Yisel Niño Benitez, N. S. M. (2018). Requisitos de Seguridad para aplicaciones web., 12.

Anexos

1: Historias de usuario

Historia de usuario	
Número: 4	Nombre: Modificar usuario
Usuario: Cliente	
Prioridad en negocio: Media	Prioridad en negocio: Media
Programador responsable: Victor José Remedios Pérez	
Descripción: El sistema debe permitir a los usuarios modificar sus datos una vez que se hayan autenticado.	
Observaciones: Se muestran los campos a modificar del usuario.	
Interfaz: 	

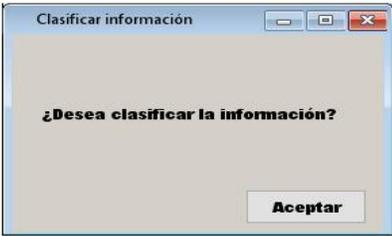
Historia de usuario	
Número: 5	Nombre: Listar usuario
Usuario: Cliente	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Programador responsable: Victor José Remedios Pérez	
Descripción: El sistema debe permitir listar los usuarios registrados en el sistema.	
Observaciones: Se muestran todos los usuarios registrados.	
Interfaz: 	

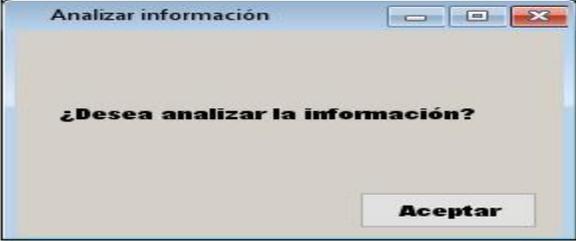
Historia de usuario	
Número: 6	Nombre: Eliminar usuario
Usuario: Cliente	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Programador responsable: Victor José Remedios Pérez	
Descripción: El sistema debe permitir a los usuarios con el rol de administrador eliminar cualquier usuario registrado en el mismo.	
Observaciones: Se muestra un botón para eliminar a cada usuario.	
Interfaz: 	

Historia de usuario	
Número: 7	Nombre: Asignar rol
Usuario: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Programador responsable: Victor José Remedios Pérez	
Descripción: El sistema debe permitir a los usuarios con el rol de administrador asignarles los roles a los usuarios registrados en el mismo.	
Observaciones: Se muestra un botón para asignar rol.	
Interfaz:	
	

Historia de usuario	
Número: 8	Número: 8
Usuario: Cliente	
Prioridad en negocio: Baja	Prioridad en negocio: Baja
Programador responsable: Victor José Remedios Pérez	
Descripción: El sistema debe permitir a los usuarios con el rol de administrador eliminar los roles a los usuarios registrados en el mismo.	
Observaciones: Se muestra un botón para eliminar el rol.	
Interfaz:  <p>The screenshot shows a table with a header row containing the text "Eliminar rol". Below the header is a single row with two columns. The first column contains the text "Usuario" and the second column contains a button labeled "Eliminar rol".</p>	

Historia de usuario	
Número: 9	Nombre: Modificar rol
Usuario: Cliente	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Programador responsable: Victor José Remedios Pérez	
Descripción: El sistema debe permitir a los usuarios con el rol de administrador editar los roles a los usuarios registrados en el mismo.	
Observaciones: Se muestra los botones para editar el rol.	
Interfaz: 	

Historia de usuario	
Número: 12	Nombre: Clasificar información
Usuario: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Programador responsable: Victor José Remedios Pérez	
Descripción: El sistema debe permitir a todos los usuarios autenticados en el mismo clasificar la información extraída de la base de datos.	
Observaciones: Se muestra un cartel para clasificar la información y un botón para aceptar.	
Interfaz:	
	

Historia de usuario	
Número: 13	Nombre: Analizar información
Usuario: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Programador responsable: Victor José Remedios Pérez	
Descripción: El sistema debe permitir a todos los usuarios autenticados en el mismo analizar la información previamente clasificada.	
Observaciones: Se muestra un cartel para analizar la información y un botón para aceptar.	
Interfaz:	
	

2: Estándares de codificación de Laravel

Otras convenciones de código recomendadas:

Usa siempre el tag de apertura largo de **PHP <?php**.

No dejes espacios antes del tag de apertura.

No uses el tag de cierre de PHP **?>** al final de los ficheros.

Mantener el límite de tus líneas en 80 caracteres.

Declara las propiedades de las clases antes de los métodos.

Declara los métodos en este orden: **public, protected y private**.

A excepción de los constructores y los métodos setUp y tearDown de los tests PHPUnit que deben ir primero para mejorar la lectura.

Siempre que sean aplicables, usa el prefijo Abstract y los sufijos Interface, Trait y Exception.

Usa sólo una 'instrucción' por línea. (PSR-2) No uses los «;» para unir varias instrucciones. Incluso los if() más sencillos escríbelos en varias líneas.

// Haz esto:

```
if (strlen($path) === 0) {  
  
    return $this->getPathPrefix() ?: "  
  
}
```

Una excepción es el 'operador condicional ternario' a ? b : c que sí se puede usar en una sola línea o dividir en varias en caso de que las expresiones sean muy complejas, para facilitar su lectura.

```
$this->pathPrefix = $is_empty ? null : $prefix;
```

Los comentarios de una sola línea colócalos al mismo nivel que la línea a la que se refieren. (Como en el ejemplo anterior del 'if')

Elimina los espacios en blanco al final de las líneas y en las líneas en blanco. (PSR-2) Especialmente si usas algún sistema de versión de código, lo agradecerás.

Utiliza espacios para mejorar la lectura de los operadores:

Después del signo de negación !.

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

Antes y después del signo igual «=», de los operadores matemáticos y lógicos y de los signos del operador condicional ternario.

```
public function setHost($host)
```

```
{  
  
    $this->host = $host;  
  
    return $this;  
  
}
```

Usa comillas sencillas habitualmente y comillas dobles cuando quieras expandir una variable.

Usa los operadores lógicos && y || en lugar de AND y OR.

Usa una línea en blanco antes de los returns a menos que sea la única línea dentro de una estructura. (Como un if)

Coloca los returns al comienzo de las funciones. (Return early) Especialmente en los casos en los que hagas validaciones con un return, hazlos al comienzo de la función. De ese modo es más fácil leer lo que hace la función y cuáles son sus condiciones de salida.

No uses else para salir al final de las funciones.

```
// Haz esto
```

```
public function getMimeType($path)
```

```
{  
  
    $path = Util::normalizePath($path);  
  
    $this->assertPresent($path);  
  
    if (! $object = $this->adapter->getMimeType($path)) {  
  
        return false;  
  
    }
```

```

    }

    return $object['mimetype'];
}

// Haz esto

public function getMimetype($path)
{
    $path = Util::normalizePath($path);

    $this->assertPresent($path);

    if (! $object = $this->adapter->getMimetype($path)) {

        return false;

    }

    return $object['mimetype'];
}

```

Para los nombres de los métodos: Cuando un objeto tiene una relación principal con varias 'cosas', como objetos, parámetros, etc... se usan estos nombres:

set()

has()

all()

get()

replace()

remove()

clear()

isEmpty()

add()

register()

count()

keys()

Para relaciones a varios dónde la relación no sea principal, se aplican estos otros nombres:

Relación Principal	Otras Relaciones
--------------------	------------------

get()	getXXX()
-------	----------

set()	setXXX()
-------	----------

n/a	replaceXXX()
-----	--------------

has()	hasXXX()
-------	----------

all()	getXXXs()
-------	-----------

replace()	setXXXs()
-----------	-----------

remove()	removeXXX()
----------	-------------

clear()	clearXXX()
---------	------------

isEmpty y()	isEmptyXXX()
-------------	--------------

add()addXXX()

register() registerXXX()

count()countXXX()

keys() n/a

Aunque setXXX y replaceXXX se parecen, la diferencia es que mientras setXXX puede modificar o añadir elementos a la relación, replaceXXX no puede añadir elementos. Y si recibe una clave desconocida, debe lanzar una excepción.

3: Lista de chequeo para requisitos no funcionales

Peso	Indicadores a evaluar	NP	RE	RF
Usabilidad				
crítico	¿Cada campo posee un identificador intuitivo?			
crítico	¿Existe un correcto contraste entre los colores de la aplicación?			
Seguridad				
crítico	¿El sistema cuenta con roles definidos y permisos asociados en correspondencia de estos roles?			
crítico	¿Existe un solo administrador del sistema?			
crítico	¿El acceso al sistema se realiza mediante un proceso de autenticación?			
crítico	¿El sistema valida que la contraseña del usuario tenga un mínimo de 8 caracteres, combinación de letras, números y caracteres especiales?			

crítico	¿Los usuarios tienen acceso a las funcionalidades que corresponden al rol asignado?			
Portabilidad				
	¿El sistema permite acceder y ejecutar las funcionalidades desde el navegador Chrome?			
	¿El sistema permite acceder y ejecutar las funcionalidades desde el navegador Firefox?			
Confiabilidad				
	¿El sistema muestra al usuario mensajes de información como ayuda a las acciones que este debe realizar?			
crítico	¿El sistema muestra mensajes de error explicando los errores que se presentan?			
Mantenibilidad				
crítico	¿El sistema permite la incorporación de nuevas funcionalidades?			