

**UNIVERSIDAD DE MATANZAS**  
**FACULTAD DE CIENCIAS TÉCNICAS**  
**DEPARTAMENTO DE INFORMÁTICA**



**Sistema de gestión de la información del Centro de Estudios  
Biotecnológicos (CEBIO) de la Universidad de Matanzas.**  
**Ejercicio Profesional en opción al título de Ingeniero Informático.**

**Autor:** Rolando Baez Naranjo.

**Tutor:** Liz Pérez Martínez.

**Matanzas, 2023**

## ***Pensamiento***

*“El conocimiento no es una vasija que se llena sino un fuego que se enciende.”*

*Plutarco*

### **Declaración de autoría**

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de Matanzas, especialmente a la Facultad de Ciencias Técnicas a que hagan el uso que estimen pertinente de él. Y para que así conste, firmo la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del 2022.

---

**Firma de Autor**

---

**Firma de Tutor**

## Opinión del Tutor

## **Agradecimientos**

*Nunca es tarea sencilla dar las gracias a todas las personas que han contribuido en mi beneficio; dedicar este trabajo a ellos es un honor y un privilegio, han sido compañeros de viaje en esta larga travesía y considero meritorio que pueda mostrar mi agradecimiento a través de estas sencillas pero sinceras palabras.*

*Primeramente mostrar mi gratitud a la profesora Liz Pérez Martínez, Tutora de este trabajo, por brindarme sus conocimientos y apoyarme cuando lo he pedido, nada de esto hubiera sido posible sin su ayuda y sin el apoyo de esta entidad y de este maravilloso claustro de profesores que a pesar de la adversidades y contratiempos que no han sido pocos en el transcurso de la carrera, por factores externos que todos conocemos, siempre estuvieron ayudándonos y exhortándonos cada día para que no desmayáramos*

*A mis padres por su amor y apoyo incondicional, por creer en mí, incluso cuando yo mismo no lo hacía, ellos son los que me han brindado el soporte material y económico para concentrarme en los estudios y nunca abandonarlos, impulsándome siempre a perseguir mis metas y nunca flaquear frente a las adversidades.*

*Agradecerle también a toda mi familia y a mis compañeros de estudios los cuales muchos de ellos se han convertido en mis amigos...gracias les doy por las historias compartidas, los trabajos realizados en conjunto y las historias vividas.*

*En conclusión quiero expresar mi profunda gratitud a todos en general y reafirmar el compromiso de seguir buscando el conocimiento y de contribuir al progreso de nuestra sociedad.*

## **Resumen**

En el Centro de Estudio Biotecnológico de la Universidad de Matanzas (CEBIO) se realizan investigaciones sobre la obtención y evaluación de aditivos nutricionales para la salud animal, el mejoramiento genético y micro propagación de especies vegetales de interés agropecuario, así como la estabilización y utilización de enzimas con aplicaciones agrícolas, industriales y médicas. Actualmente, el proceso de la gestión de la información de los procesos investigativos del centro resulta complejo y difícil de realizar, la información se almacena de forma manual y en formato Word, por lo que es complicado hacer búsquedas y actualizaciones. La información no tiene manera de llegar al público. En el presente trabajo se describe la investigación realizada en el proceso de gestión de la información del Centro de Estudio Biotecnológico, con el objetivo de mejorar este y lograr que sea de una manera más eficiente. Con el desarrollo del Sistema para la gestión de la información de las investigaciones en el centro, se logró un medio que unificara los datos y garantizara la integridad y consistencia de los mismos. Es posible acceder a la información fácilmente y gestionarla de forma óptima.

# Índice

Introducción .....	1
Capítulo 1. Marco Teórico Referencial.....	4
1.1 Descripción de la organización Objeto de Estudio.....	4
1.1.2 Descripción del negocio .....	5
1.2 Gestión de la información .....	6
1.3 Sistemas de Gestión de la Información.....	7
1.4 Metodologías de desarrollo de software .....	7
1.4.1 Metodologías tradicionales.....	8
1.4.2 Metodologías ágiles .....	8
1.4.3 Comparación entre las metodologías ágiles y las tradicionales.....	9
1.4.4 Análisis comparativo entre metodologías de desarrollo ágiles .....	14
1.4.5 Por qué seleccionar XP?: .....	14
1.5 Lenguaje de programación.....	14
1.6 Framework de desarrollo web .....	17
1.6.1 Framework Laravel.....	17
1.6.2 Framework Vue.js.....	17
1.6.3 Framework Nuxt.js .....	18
1.6.4 Framework Vuetify.....	18
1.7 Aplicación web .....	18
1.8 Tendencias tecnológicas a considerar.....	19
1.9 Gestor de base de datos.....	20
1.10 Servidor web .....	21
1.11 Visual Studio Code.....	22
1.12 Conclusiones Parciales del Capítulo .....	22
Capítulo 2: Análisis, diseño y desarrollo de la solución propuesta .....	23
2.1 Descripción de la solución.....	23
2.2 Roles del Sistema.....	23
2.3 Materiales y herramientas utilizadas .....	24
2.4 Etapa de planificación .....	24
2.4.1 Equipo de trabajo y roles .....	25
2.4.2 Historias de usuario.....	25
2.4.3 Plan de Iteraciones .....	29
2.4.4 Reuniones.....	30
2.5 Etapa de diseño .....	30
2.5.1 Prototipo de interfaz de usuario .....	31

2.6 Arquitectura propuesta.....	31
2.7 Patrones de diseño.....	32
2.7.1 Patrones GRASP.....	32
2.8 Tareas de Ingeniería.....	32
2.9 Base de Datos.....	37
2.10 Estudio De Factibilidad.....	38
2.10.1 Método Punto de Función.....	38
2.10.2 Cálculo de los Puntos de Función Ajustados (PFA).....	39
2.10.3 Puntos de Función Ajustados (PFA).....	40
2.10.4 Cálculo del Esfuerzo.....	40
2.10.5 Esfuerzo en horas / persona (E).....	40
2.10.6 Cálculo del Presupuesto del Proyecto.....	41
2.10.7 Análisis de los beneficios.....	41
2.11 Conclusiones Parciales del Capitulo.....	41
Capítulo 3: Validación de la solución propuesta.....	42
3.1 Pruebas al software.....	42
3.1.1 Pruebas de Aceptación.....	44
3.1.2- Vista Crear Personal.....	45
3.1.3 Vista Crear Grupo de Investigación.....	50
3.2 Pruebas de Seguridad.....	56
3.3 Conclusiones parciales del Capítulo 3.....	59
Conclusiones Generales.....	60
Recomendaciones.....	61
Bibliografía.....	62
Anexos.....	65



## **Introducción**

La información ha existido siempre, desde la misma existencia de la sociedad humana, apareció primero de forma oral, después de forma escrita y, por último, en formato digital. En el siglo XX aparece como un fenómeno de gradual importancia en todos los terrenos. La información tributa al conocimiento, constituye el eje central del cambio en la sociedad moderna, es un recurso esencial para la transformación de cualquier infraestructura técnica y organizativa. Es, a la vez, la causa y el efecto del cambio y la forma en que se agrupan, clasifican y formatean los datos con determinado propósito.

Actualmente el internet se ha convertido en un medio de comunicación importante para la humanidad debido a la facilidad que otorga a los usuarios tanto como a las pequeñas y grandes organizaciones para buscar, compartir o publicar la información que requieran en una variedad de sitios web alojados en servidores distribuidos. Además, permite publicar servicios, productos y procesos automatizados en línea disponibles para todos los usuarios con acceso a la red global (Molina Ríos, Zea Ordóñez, & Contenido Segarra, 2017). Internet también concede una ventaja competitiva tanto en el sector público como privado poniendo a su alcance las tecnologías necesarias que faciliten el crecimiento, revelación o expansión de sus actividades sociales y comerciales (Molina Ríos, 2018). Por estas y otra gran cantidad de ventajas muchas instituciones en el país han abierto sus puertas al proceso de informatización para lograr eficiencia en la gestión de la información.

Una de estas instituciones donde se ha iniciado dichos procesos es el Centro de Estudios Biotecnológicos (CEBIO) de la Universidad de Matanzas en el cual se realizan las investigaciones sobre la obtención y evaluación de aditivos nutricionales para la salud animal, el mejoramiento genético y micro propagación de especies vegetales de interés agropecuario, así como la estabilización y utilización de enzimas con aplicaciones agrícolas, industriales y médicas. Todo ello contribuye al desarrollo científico, económico y social del territorio y del país, así como la formación integral de los estudiantes de las carreras de Agronomía, IPAI, LEA, Medicina Veterinaria y Química. EL Centro de Estudios Biotecnológicos (CEBIO) de la Universidad de Matanzas en estos momentos carece de alguna plataforma o sitio donde poder darse a conocer como institución y poder mostrar y explicar que es como tal el CEBIO, así como los servicios, líneas de investigación y proyectos que tienen y en los que se trabajan en estos momentos. También surge la necesidad de informar al público en general de como contactar con dicha institución. A partir de la problemática anterior surge el **problema de investigación** que da origen al presente trabajo de diploma:

¿Cómo incrementar la eficiencia en la gestión de la información en el Centro de Estudios Biotecnológicos de la Universidad de Matanzas (CEBIO)?

Este problema se enmarca en el **Objeto de estudio:** La gestión de la información en el Centro de Estudios Biotecnológicos (CEBIO) en la Universidad de Matanzas.

Delimitándose como **campo de acción:** Informatización de la gestión de la información en el Centro de Estudios Biotecnológicos (CEBIO) en la Universidad de Matanzas.

Se tiene como **Hipótesis:**

Si se desarrolla una aplicación web, entonces se incrementará la eficiencia en la gestión de la información de los procesos investigativos en el Centro de Estudios Biotecnológicos de la Universidad de Matanzas (CEBIO).

**Objetivo general:**

Desarrollar una aplicación web para la informatización y el incremento de la eficiencia de la gestión de la información la cual ayudaría a visualizar el trabajo investigativo que se realiza en el Centro de Estudios Biotecnológicos (CEBIO) en la Universidad de Matanzas.

**Objetivos Específicos:**

1. Determinar el marco teórico referencial para incrementar la eficiencia.
2. Seleccionar los métodos y técnicas que permitan obtener los fundamentos teóricos y la comprensión necesaria de los distintos elementos del tema que aborda la investigación.
3. Definir los requisitos del sistema, así como diseñar el modelo de datos e interfaz de usuario que sirvan como punto de partida en la adecuada determinación de las herramientas y metodología de desarrollo a utilizar.
4. Desarrollar la aplicación web que contribuya a la gestión de la información de los procesos investigativos en el Centro de Estudios Biotecnológicos (CEBIO) en la Universidad de Matanzas.
5. Implementar la aplicación web que contribuya en la gestión de la información de los procesos investigativos en el Centro de Estudios Biotecnológicos (CEBIO) en la Universidad de Matanzas.
6. Validar el funcionamiento correcto de la aplicación web desarrollada.

Para el desarrollo de la investigación se emplean los siguientes **métodos y técnicas:**

- **Métodos teóricos:**

**Análisis Histórico-lógico:** se empleó para comprender sus causas históricas, así como las condiciones y antecedentes que favorecieron su surgimiento y posterior auge como objeto de estudio, basándose en una detallada revisión bibliográfica.

**Analítico-sintético:** se aplicó para realizar un estudio general y análisis de toda la información que tuviese relación con el tema, así como la interpretación de los elementos necesarios para darle solución, basándose en investigaciones anteriores, a la problemática actual.

**Hipotético-deductivo:** Implicó la correcta deducción de los primeros juicios acerca de la eficiencia al introducir la aplicación web. A partir de este supuesto se pueden inferir conclusiones y establecer predicciones, de las cuales se deducen las valoraciones de que la aplicación necesita un mantenimiento para incluir nuevas funcionalidades y corregir sus cambios.

**Enfoque de sistema:** Proporcionó la orientación general de la aplicación web como un sistema de clases que se relacionan entre sí a partir de la comunicación que establecen entre ellas.

- **Métodos empíricos** abordados a través de las técnicas siguientes:

**Observación:** Acompañó la investigación desde los primeros momentos, se interpretó directamente en la gestión y ocupación de los actores en su vínculo con el sistema, facilitando la comprensión del proceso y las razones que dificultan su mayor desarrollo.

**Entrevista:** Se realizó para recopilar información a través de una planificada conversación junto a los especialistas de diversos temas del objeto de estudio, cuyo análisis posterior posibilita el avance en la investigación, incluyendo el adecuado levantamiento de los requisitos del negocio.

**Análisis de documentos:** Incluye la revisión de documentos como son las resoluciones, decretos y leyes vigentes relacionados con el tema. También se consultaron libros y diversos artículos de bancos de información de Internet relacionados con el tema a investigar.

En correspondencia con lo tratado anteriormente, el documento queda estructurado como sigue a continuación:

### **Capítulo I: Marco teórico referencial**

Recoge los conceptos fundamentales asociados al dominio del problema. Realiza un estudio de las tendencias y tecnologías actuales sobre las que se apoya la propuesta y analiza las soluciones existentes.

## **Capítulo II: Análisis, diseño y desarrollo de la solución propuesta**

Se expone la propuesta de solución al problema de investigación, la organización y disposición del proyecto mediante la presentación de una planificación inicial haciendo uso del marco de trabajo de desarrollo de software XP. Es analizado y estudiado los beneficios obtenidos mediante la implementación del proyecto de software.

## **Capítulo III: Validación de la solución propuesta**

Se hace el análisis de los resultados obtenidos. También se realizan las pruebas al software con el objetivo de entregarle al cliente un producto totalmente funcional, cumpliendo con todos los requisitos demandados por el mismo y satisfaciendo sus necesidades.

Finalmente, son presentadas las **Conclusiones y Recomendaciones** de la investigación para dejar el camino abierto a futuros estudios relacionados con la temática abordada.

De igual forma, quedan recogidas las **Bibliografías** utilizadas y **Anexos** que fueron necesarios para el desarrollo del trabajo de investigación y mejor entendimiento del mismo.

## **Capítulo 1. Marco Teórico Referencial**

En el presente capítulo se aborda los fundamentos teóricos, antecedentes y métodos que posee la investigación. Son reflejados los principales conceptos que soportan el objeto de estudio para brindar una mejor comprensión del mismo. Es analizada la metodología de desarrollo escogida para garantizar la adecuada realización de la aplicación en correspondencia con las características del proyecto a implementar. También, se hace referencia a las distintas tecnologías y herramientas seleccionadas para el desarrollo del software.

### **1.1 Descripción de la organización Objeto de Estudio**

El Centro de Estudios Biotecnológico de la Universidad de Matanzas (CEBIO) es el encargado de realizar investigaciones sobre la obtención y evaluación de aditivos nutricionales para la salud animal, el mejoramiento genético y micro propagación de especies vegetales de interés agropecuario, así como la estabilización y utilización de enzimas con aplicaciones agrícolas, industriales y médicas. Todo ello contribuye al desarrollo científico, económico y social del territorio y del país, así como la formación integral de los estudiantes de las carreras de Agronomía, IPAI, LEA, Medicina Veterinaria y Química. Tiene tres líneas principales de investigación las cuales son

Biotecnología Vegetal, Aditivos Nutricionales y Tecnología Enzimática. El centro tiene como objetivo:

- Contribuir a la formación integral de los estudiantes de las carreras de Agronomía, IPAI, LEA, Medicina Veterinaria y Química.
- Desarrollar tecnologías para el mejoramiento genético y propagación de especies vegetales de interés agrícola para el territorio matancero y el país.
- Caracterizar y evaluar aditivos nutricionales y sus productos metabólicos obtenidos a partir de cultivos de microorganismos para mejorar indicadores fisiológicos, inmunológicos y de respuesta productiva en animales de interés económico.
- Inmovilizar enzimas con aplicaciones en las ramas agrícolas, industriales y médicas.
- Generar y coordinar actividades de posgrado que contribuyan a incrementar los conocimientos de los profesionales y técnicos del territorio en correspondencia con las tres líneas de investigación del CEBIO.
- Introducir los resultados obtenidos en la práctica social.

### **1.1.2 Descripción del negocio**

En el Centro de Estudios Biotecnológicos de la Universidad de Matanzas (CEBIO) se realizan investigaciones sobre la obtención y evaluación de aditivos nutricionales para la salud animal, el mejoramiento genético y micro propagación de especies vegetales de interés agropecuario, así como la estabilización y utilización de enzimas con aplicaciones agrícolas, industriales y médicas, todas estas investigaciones no tienen ninguna plataforma donde poder plasmar los resultados obtenidos. Por tanto actualmente el estudiante, profesor o alguna otra persona interesada en este tipo de investigaciones no tendría manera de informarse con respecto a esto. Es por eso que tendrían que visitar el centro de manera presencial y allí solicitar una reunión con la directora del centro o algún jefe de las líneas de investigación a las que dedican su estudio, en el cual selecciona la información que desea y el personal especializado le da la información necesaria, lo mismo sucede con los proyectos de investigación asociados a estas líneas de investigación, por lo que gestionar la información necesaria sería un poco lento ya que el interesado tendría que solicitar una reunión, esperar a poder ser atendido, la información puede que no esté bien organizada y sería bastante lento el proceso. Lo mismo sucede con las actividades de posgrado que contribuyen a incrementar los conocimientos de los profesionales y técnicos del territorio en

correspondencia con las tres líneas de investigación del Centro de Estudios Biotecnológicos, no hay manera de informar de dichas actividades. Si la persona desea saber cuáles son las actividades de posgrado que otorga el centro, o de estos conocer alguna de sus especificaciones, debe realizar una llamada a la sede del centro o presentarse en el mismo para obtener la información.

Por otra parte, los directivos no poseen una herramienta que les permita tomar decisiones y llegar a conclusiones de manera rápida y eficaz, pues la información se almacena en softwares de terceros que no satisfacen sus necesidades, ya que la información no llega de primera mano y no cuentan con funcionalidades necesarias para el modelo de negocio del Centro de Estudios Biotecnológico.

## **1.2 Gestión de la información**

La gestión de información es un proceso que ha alcanzado un gran auge e importancia a nivel mundial. Debido a esta repercusión diversos autores e instituciones han dedicado un espacio dentro de sus investigaciones para dejar reflejado definiciones como las que se muestran a continuación:

- Según Ponjuán (2003), la gestión de la información es el proceso mediante el cual se obtiene, despliegan o utilizan recursos básicos (económicos, físicos, humanos, materiales) para manejar información dentro y para la sociedad a la que sirve (Suárez Alfonso, Cruz Rodríguez, & Pérez Macias, 2015).
- Para Lynda Woodman (1985), la gestión de información se define como: Todo lo que tiene que ver con obtener la información correcta, en la forma adecuada, para la persona indicada, al costo correcto, en el momento oportuno, en el lugar indicado para tomar la acción precisa (Barrios, 2018).
- Gestión de información: Se trataría de la explotación de la información para la consecución de los objetivos de la entidad. Su creación, adquisición, procesamiento y difusión (Arévalo, 2019).
- Según Fairer Wessels, la gestión de información: Es vista como la planificación, organización, dirección y control de la información dentro de un sistema abierto (Suárez Alfonso, Cruz Rodríguez, & Pérez Macias, 2015).
- Se define gestión de información como: El proceso que se encarga de gestionar la información necesaria para la toma de decisiones y un mejor funcionamiento de los procesos, productos y servicios de la organización (Infante, s.f).

Teniendo en cuenta las definiciones anteriores podemos decir que la gestión de información es todo lo que tiene que ver con obtener la información correcta, en la forma adecuada, para la persona indicada, al costo correcto, en el momento oportuno, en el

lugar indicado para tomar la acción precisa. Se selecciona este concepto porque está estrechamente relacionada con la gestión de datos, sistemas, tecnología y procesos.

### **1.3 Sistemas de Gestión de la Información**

Los sistemas de gestión de la información actualmente ocupan un lugar dentro de los principales sistemas informáticos de diversas instituciones. A lo largo del desarrollo y uso de las tecnologías de la información diversos autores han definidos algunos conceptos como los enunciados a continuación:

- **Según Davis y Olsón (1985) los Sistemas de gestión de la información son:** Sistemas integrados y automatizados para proveer la información que sostenga las funciones de operatividad, gestión y toma de decisiones en una organización (Lara, s.f).
- **Para Moreiro González (1998) define los Sistemas de Gestión de la Información como:**

El conjunto de políticas y normas relacionadas entre sí que se establecen para el acceso y tratamiento de los recursos de información, incluye los registros administrativos y los archivos, el soporte tecnológico de los recursos y el público a que se destina (González, 1998).

El sistema de gestión de la información (MIS) es un conjunto de herramientas organizativas, técnicas, tecnológicas y de información que se integran en un único sistema para recoger, almacenar, procesar y producir información destinada a realizar funciones de gestión. El sistema de información acumula y procesa la información normativa, de planificación y contable entrante para convertirla en información analítica que sirve de base para prever el desarrollo del sistema de gestión, ajustar los objetivos y planificar un nuevo ciclo de reproducción (Pérez, 2021).

### **1.4 Metodologías de desarrollo de software**

En la década de los noventa surgieron metodologías de desarrollo de software ligeras, más conocidas como metodologías ágiles, que buscaban reducir la probabilidad de fracaso por subestimación de costos, tiempos y funcionalidades en los proyectos de desarrollo de software. Estas metodologías nacieron como reacción a las metodologías existentes con el propósito de disminuir la burocracia que implica la aplicación de las metodologías tradicionales en los proyectos de pequeña y mediana escala. Las metodologías tradicionales buscan imponer disciplina al proceso de desarrollo de software y de esa forma volverlo predecible y eficiente. Para conseguirlo se soportan en un proceso detallado con énfasis en planeación propio de otras ingenierías. El principal problema de este enfoque es que hay muchas actividades que hacer para seguir la

metodología y esto retrasa la etapa de desarrollo (Navarro Cadavid, Fernández Martínez, & Morales Vélez, 2013).

#### **1.4.1 Metodologías tradicionales**

Llevan un control estricto del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Centran su atención en llevar una documentación exhaustiva de todo el proyecto y en cumplir con un plan de proyecto.

Las metodologías tradicionales de desarrollo de software son orientadas por planeación. Inician el desarrollo de un proyecto con un riguroso proceso de elicitación de requerimientos, previo a etapas de análisis y diseño. Con esto tratan de asegurar resultados con alta calidad circunscritos a un calendario.

En las metodologías tradicionales se concibe un solo proyecto, de grandes dimensiones y estructura definida; se sigue un proceso secuencial en una sola dirección y sin marcha atrás; el proceso es rígido y no cambia; los requerimientos son acordados de una vez y para todo el proyecto, demandando grandes plazos de planeación previa y poca comunicación con el cliente una vez ha terminado ésta (Khurana & Sohal, 2011).

#### **1.4.2 Metodologías ágiles**

Dan mayor importancia a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Se basa en la filosofía de que es más importante desarrollar software que funcione, que conseguir una buena documentación y es más importante responder ante un cambio, que seguir estrictamente un plan.

El enfoque de las metodologías ágiles está teniendo una amplia efectividad en proyectos donde los requisitos son muy cambiantes, ya que en este tipo de proyectos la comunicación con el cliente debe ser fundamental y precisamente ese es uno de los principios básicos de las metodologías ágiles.

En marzo de 2001, 17 críticos de los modelos de producción basados en procesos resumieron en cuatro postulados lo que hoy en día se conoce como el Manifiesto Ágil (Menzinsky, López, & Palacio, 2019). Los cuatro postulados mencionados en aquella reunión mencionaban los siguientes criterios: Individuos e interacciones sobre procesos y herramientas, Software que funciona sobre documentación exhaustiva, Colaboración de clientes sobre la negociación del contrato y Respuestas a cambios sobre seguir un plan. Estos cuatro postulados son considerados como la definición canónica del desarrollo ágil.



### 1.4.3 Comparación entre las metodologías ágiles y las tradicionales

El objetivo final siempre será tener metodologías diferentes para aplicar de acuerdo con el proyecto que se desarrolle. Estas pueden involucrar prácticas tanto de metodologías ágiles como de metodologías tradicionales. De esta manera se pudiera tener una metodología para cada proyecto, la problemática sería definir cada una de las prácticas y en el momento preciso definir parámetros para saber cuál usar.

Metodologías ágiles	Metodologías tradicionales
No existe un contrato tradicional, debe ser bastante flexible.	Existe un contrato prefijado.
Están orientadas hacia las necesidades del cliente.	Están orientados hacia el proceso del software.
Pocos artefactos.	Más artefactos.
La arquitectura se va definiendo y mejorando a lo largo del proyecto.	Se promueve que la arquitectura se defina tempranamente en el proyecto.
Basadas en heurísticas o estadísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Grupos pequeños (< 10 integrantes) y trabajando en el mismo sitio.	Aplicables a proyectos de cualquier tamaño, pero suelen ser especialmente efectivas/usadas en proyectos grandes y con equipos posiblemente dispersos.
Proceso menos controlado, con pocas políticas para el desarrollo.	Procesos mucha más controlados, con numerosas políticas o normas.
Se hace énfasis en los aspectos humanos: el individuo y el trabajo en equipo.	Se hace énfasis en la definición del proceso: roles, actividades y artefactos.
Especialmente preparadas para cambios durante el proyecto.	Cierta resistencia a los cambios

Tabla 1. Comparación entre la metodología ágil y la tradicional. Fuente: elaboración del autor.

Después de haber realizado una comparación entre las metodologías ágiles y las tradicionales se observa que las ágiles se adecuan más para guiar el desarrollo del software propuesto, principalmente porque el proyecto no es altamente complejo, se cuenta con pocos roles y no se dispone de mucho tiempo para su puesta en marcha.

En la actualidad existen un sin número de metodologías ágiles, unas más populares que otras, cada una aportando al desarrollo ágil distintos métodos que ayudan a mejorar de una manera eficaz la calidad del software. Entre las metodologías ágiles más populares se encuentran las siguientes:

#### **1.4.3.1 Programación Extrema (XP)**

La Programación Extrema o XP (Extreme Programming) es un enfoque de desarrollo de sistemas que acepta lo que se conoce como buenas prácticas en esta área y las lleva al extremo (Rosado Gómez, Quintero Duarte, & Meneses Guevara, 2012). En esta metodología cabe resaltar la importancia del cliente, las pruebas, la refactorización, la simplicidad, la propiedad colectiva del código que se ven reflejadas en las cuatro prácticas esenciales de XP (Pressman R. S., 2010)

- Entregas limitadas o pequeñas: Consiste en realizar entregas parciales de módulos del sistema.
- Semana de trabajo de 40 horas: Los equipos de desarrollo de XP trabajan de manera intensa durante una semana típica de 40 horas.
- Cliente en el sitio: Esta práctica insiste en que el cliente debe hacer parte fundamental y activa del grupo de trabajo y debe estar presente durante todo el proceso de desarrollo.

La programación extrema engloba un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades estructurales: planeación, diseño, codificación y pruebas.

**Planeación:** Esta actividad comienza con una entrevista a los clientes, para entender el contexto del negocio y definir las características principales y funcionalidad que se requiere, estas características se transforman en requerimientos del negocio que se especifican mediante Historias de Usuario; las cuales recogen la interacción hablada entre desarrolladores y usuarios. Una vez hechas las Historias de Usuario, el equipo de desarrollo las divide en tareas, estima el esfuerzo, recursos requeridos para su implementación, se genera el plan de entregas, las iteraciones, la rotación de parejas y las reuniones diarias.

**Diseño:** Es la etapa en donde son evaluadas las historias de usuario por el equipo del proyecto para dividir las en tareas, cada tarea representa una característica distinta del sistema y se puede diseñar una prueba de unidad que verifique cada tarea, estas tareas se representan por medio de las tarjetas CRC (Clase-Responsabilidad-Colaborador). Las tarjetas CRC identifican y organizan las clases bajo el paradigma orientado a objetos (lo que incluye asignación de responsabilidades), cada tarjeta contiene el nombre de la

clase (que representa una o más historias de usuario), una descripción de las responsabilidades o métodos asociados con la clase, así como la lista de las clases con que se relaciona o que colaboran con ella. Las tarjetas CRC son el único trabajo de diseño que se genera como parte del proceso de XP.

**Codificación:** Se lleva a cabo la programación en pareja, la unidad de pruebas y la integración del código. Durante esta etapa se espera la disponibilidad del cliente para que éste pueda resolver cualquier duda que se presente durante una jornada de trabajo.

**Prueba:** Cada tarea que se identificó con las historias de usuario, representa una característica distinta del sistema y se realiza una prueba de unidad por cada una de ellas, existen pruebas unitarias las cuales son diseñadas para probar cada uno de los métodos y clases, dichas pruebas son realizadas por los programadores.

#### **1.4.3.2 Scrum**

Su nombre no corresponde a una sigla, sino a un concepto deportivo, propio del rugby, relacionado con la formación requerida para la recuperación rápida del juego ante una infracción menor (Navarro Cadavid, Fernández Martínez, & Morales Vélez, 2013); (Faniran, Badru, & Ajayi, 2017).

Es un proceso ágil y liviano que sirve para administrar y controlar el desarrollo de software. El desarrollo se realiza en forma iterativa e incremental (una iteración es un ciclo corto de construcción repetitivo). Cada ciclo o iteración termina con una pieza de software ejecutable que incorpora nuevas funcionalidades. Las iteraciones en general tienen una duración entre 2 y 4 semanas.

- Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.
- Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos auto organizados, que en la calidad de los procesos empleados.
- Solapamiento de las diferentes fases del desarrollo, en lugar de realizarlas una tras otra.

Los principios Scrum son congruentes con el manifiesto ágil y se utilizan para guiar actividades de desarrollo dentro de un proceso de análisis que incorpora las siguientes actividades estructurales: requerimientos, análisis, diseño, evolución y entrega (Pressman R. S., 2010); (Srivastava, Bhardwaj, & Saraswat, 2017).

Las características del desarrollo basado en Scrum son (Ahmed, 2010):

**Colaboración:** El desarrollo basado en Scrum promueve la colaboración ya que está impulsado por equipos multifuncionales donde cada persona con sus habilidades y

experiencia contribuye a la mejor solución de diseño. Un equipo multifuncional incluye una combinación de programadores, arquitectos de software, analistas de software y expertos en control de calidad.

**Reuniones diarias:** La metodología Scrum está marcada por reuniones diarias de corta duración en las que el equipo de desarrollo de productos se comunica y evalúa el estado de progreso del desarrollo de software, lo que aumenta la productividad de los miembros del equipo.

**Product Backlog:** El product backlog captura los requisitos para que un producto de software se entregue con éxito. Mantiene una lista ordenada de características, correcciones de errores, requisitos no funcionales.

**Sprint Backlog:** El sprint backlog registra la lista de tareas que realizará el equipo de desarrollo durante el próximo sprint. Esta lista se elabora recogiendo las tareas desde la parte superior de la cartera de productos hasta que se realice el trabajo suficiente para el próximo sprint, teniendo en cuenta la capacidad de trabajo y los resultados pasados del equipo de desarrollo.

**Roles:** El desarrollo basado en Scrum se rige por 3 funciones principales (Mahalakshmi & Sundararajan, 2015); (Faniran, Badru, & Ajayi, 2017):

- Product Owner: responsable de definir, priorizar y comunicar los requisitos del producto y guía el proceso de desarrollo del producto.
- Equipo de desarrollo: responsable de ejecutar las tareas asignadas por el propietario del producto dentro del plazo del sprint. Por lo general, un equipo multifuncional de 3 a 9 individuos implementa las tareas de desarrollo del producto previstas por el propietario del producto.
- Scrum Master: responsable de hacer cumplir las reglas y los principios del desarrollo basado en Scrum. El Scrum Master elimina los impedimentos para el desarrollo y ayuda a mejorar el proceso, el equipo de desarrollo y el producto de software que se está desarrollando.

#### **1.4.3.3 Crystal Methodologies**

El método de desarrollo Crystal, es una colección de enfoques de desarrollo de software ágil, se centra principalmente en las personas y la interacción entre ellas mientras trabajan en un proyecto de desarrollo de software. También hay un enfoque en la criticidad comercial y la prioridad comercial del sistema en desarrollo.

A diferencia de los métodos de desarrollo tradicionales, Crystal no repara las herramientas y técnicas de desarrollo, sino que mantiene a las personas y los procesos en el centro del proceso de desarrollo. Sin embargo, lo importante no son solo las personas o los procesos, sino la interacción entre los dos.

Una de sus características principales es la vital importancia que se les da a los desarrolladores que componen el grupo de trabajo, por lo cual sus puntos de estudio están destinados a:

- Aspecto humano del equipo.
- Tamaño de un equipo.
- Comunicación entre los desarrolladores.
- Políticas a seguir.
- Espacio físico de trabajo.

Algunas de las Fases de la metodología Crystal son las siguientes:

**Puesta en escena:** Consiste en la planificación del siguiente incremento, el equipo selecciona los requerimientos que serán implementados en el incremento y planifican lo que harán.

**Revisiones:** Cada incremento tiene varias iteraciones y cada iteración incluye las actividades de construcción, demostración y resumen de objetivos del incremento.

**Monitoreo:** Los progresos son monitoreados a partir de las diferentes entregas. El proceso se mide con los hitos clave y la estabilidad de las fases.

**Paralelismo y flujo:** Cuando el monitoreo nos brinda un estado suficientemente estable es hora de pasar a la próxima etapa.

**Estrategia de diversidad holística:** Se utiliza para dividir grandes equipos funcionales en equipos multifuncionales.

**Técnica de puesta a punto de la metodología:** Se basa en entrevistas y talleres para elaborar una metodología específica para el proyecto: Sirve para modificar o fijar el proceso de desarrollo.

**Puntos de vista de usuario:** Dependiendo del color de la metodología Crystal que se elija se recomienda la opinión de dos usuarios por cada versión del producto (CC); también tres revisiones por parte del cliente en cada iteración (CO).

El Método Crystal es ampliable. Puede ser utilizado por equipos pequeños o grandes para trabajar en objetos simples o complejos. Da importancia a las habilidades de desarrollo y las interacciones que a su vez fomentan el intercambio de ideas. También es beneficioso para los clientes, ya que entrega primero los componentes más importantes del producto. Pero por otro lado, el Método Crystal no planifica en base a los requerimientos de los proyectos.

#### **1.4.4 Análisis comparativo entre metodologías de desarrollo ágiles**

Aunque existen varias metodologías de desarrollo ágiles disponibles, cada una se aplica a un conjunto específico de proyectos. Un proyecto de desarrollo de software tiene varios factores asociados, como el tamaño del proyecto, la complejidad, el tiempo asignado, el presupuesto, etc. La selección de la metodología adecuada para el desarrollo de software depende de tales factores. Por lo tanto, un análisis comparativo de metodologías ágiles ayudará a decidir cuál se puede utilizar en una situación dada (Kirmani, 2017)

#### **1.4.5 Por qué seleccionar XP?:**

Luego de analizar estas metodologías se escoge XP por las siguientes razones:

✓ Teniendo en cuenta las características del equipo de desarrollo, el cual está conformado por dos individuos, resulta apropiado la implementación de prácticas correspondientes a la metodología XP, tales como la programación en pares, la permanente interacción con el cliente, la refactorización constante, entre otras ventajas que brinda dicha metodología.

✓ A diferencia de XP, Scrum no recomienda ninguna buena práctica de desarrollo.

✓ La aplicación que se desarrolla es muy práctica y guarda mucha relación con los procesos que realiza el cliente en su quehacer diario, por lo que no requiere demasiada documentación sino muy buena calidad para su puesta en práctica.

✓ XP reduce el número de cambios necesarios para el proyecto lo que permite mejorar el tiempo de implementación. Por otra parte, existe propiedad de código compartido y simplicidad en el código lo que permite que este sea más entendible y modificable por los desarrolladores.

✓ XP aceptan los cambios con más facilidad en sus iteraciones.

A raíz del análisis anteriormente realizado y teniendo en cuenta que el cliente es también tutor del proyecto, se puede afirmar que el sistema a desarrollar puede culminarse con resultados exitosos aplicando la metodología XP.

#### **1.5 Lenguaje de programación**

Se conoce como lenguaje de programación a un programa destinado a la construcción de otros programas informáticos. Su nombre se debe a que comprende un lenguaje formal que está diseñado para organizar algoritmos y procesos lógicos que serán luego llevados a cabo por un ordenador o sistema informático, permitiendo controlar así su comportamiento físico, lógico y su comunicación con el usuario humano.

Dicho lenguaje está compuesto por símbolos y reglas sintácticas y semánticas, expresadas en forma de instrucciones y relaciones lógicas, mediante las cuales se construye el código fuente de una aplicación o pieza de software determinado. Así, puede llamarse también lenguaje de programación al resultado final de estos procesos creativos.

En la actualidad existen distintos lenguajes de programación para la Web, principalmente se destacan dos grupos que se diferencian entre sí por el lugar que ocupan en la arquitectura Cliente - Servidor. El primer grupo está formado por los lenguajes que se ejecutan en el lado cliente: HTML, JavaScript y CSS, todos encargados de aportar dinamismo a la aplicación en los navegadores. También existe un segundo grupo de lenguajes que se ejecutan en el lado servidor entre ellos se encuentran PHP, Java y Python. Estos se caracterizan por desarrollar la lógica de negocio, además de ser los encargados del acceso a Bases de Datos y el tratamiento de la información. Las características de los lenguajes empleados se muestran a continuación.

#### **Del lado del cliente:**

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento, en el proyecto se emplearon las siguientes tecnologías:

#### **HTML (*Hypertext Markup Language*)**

Es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces (hyperlinks) que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos, sonido). Está compuesto por una serie de etiquetas que el navegador interpreta y da forma en la pantalla. La utilización de HTML como uno de los lenguajes para el desarrollo, responde a las necesidades de manipulación y maquetación de los elementos visuales de la aplicación. La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, entre otras), así como los diferentes efectos que se quieren dar (cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado (Gauchat, 2012).

Este lenguaje es parte fundamental de las normas de la web, ya que es la base para construir páginas en internet, y el uso de sus etiquetas semánticas ayudan a definir la estructura del documento y permiten que las páginas web sean mejor indexadas por los navegadores.

#### **CSS (*Cascading Style Sheets*)**

Básicamente, es un lenguaje que maneja el diseño y presentación de las páginas web, es decir, como lucen cuando un usuario las visita. Funciona junto con el lenguaje de HTML que se encarga del contenido básico de las páginas. Se les denomina hojas en cascada porque puedes tener varias hojas y una de ellas con las propiedades heredadas (o en cascada) de otras (Santos, 2022).

Con CSS puedes crear reglas para decirle a tu sitio web como quieres mostrar la información y guardar los comandos para elementos de estilo (como fuentes, colores, tamaño, etc.) separadas de los que configuran el contenido. Además puedes crear formatos específicos útiles para comunicar tus ideas y producir experiencias más agradables visualmente para los usuarios del sitio web (Santos, 2022).

### **JavaScript:**

Es un lenguaje de programación de computadoras dinámico utilizado habitualmente en navegadores web para controlar el comportamiento de páginas web e interactuar con los usuarios. Permite comunicación asincrónica y puede actualizar partes de una página web o incluso reemplazar completamente su contenido. Es utilizado para mostrar información de fechas y hora, ejecutar animaciones en un sitio web, validar formularios, sugerir resultados mientras el usuario escribe en un cuadro de búsqueda y más (Dimes, 2015).

JavaScript es un lenguaje de programación interpretada, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. A pesar de su nombre, legalmente *JavaScript* es una marca registrada de la empresa Sun Microsystems.

### **Del lado del servidor:**

#### **Lenguaje de programación PHP**

PHP por sus siglas en inglés (Personal Home Pages) es una herramienta con tecnología popular basado en fuente de desarrollo libre. Es un lenguaje considerado de alto nivel combinado con páginas HTML (HyperText Markup Language), además es uno de los lenguajes de programación web más populares. Este tipo de programación permite que su código sea autodocumentado y fácil de rastrear. Permite la conexión a diferentes tipos de servidores de base de datos tales como: MySQL, Postgre, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird, SQLite; lo cual facilita la creación de aplicaciones Web muy robustas. Tiene la capacidad de ser ejecutado en la mayoría de los sistemas



operativos tales como UNIX, Linux, Windows y Mac OS X, y puede interactuar con los servidores de Web más populares (Wall, 2020)

### **¿Por qué usar PHP en el proyecto?**

El lenguaje de programación PHP se determinó para el desarrollo de la solución propuesta porque en primer lugar de todos los Lenguajes de Programación estudiados en la carrera, es en uno de los que más se ha trabajado y se ha adquirido experiencia, además PHP es un lenguaje fácil de aprender y utilizar, es compatible con la mayoría de los sistemas operativos y servidores web, lo que lo hace muy versátil y adaptable a diferentes entornos de desarrollo. Además, se integra fácilmente con otros lenguajes de programación y tecnologías web como HTML, CSS, JavaScript y bases de datos. Cuenta con una amplia librería de funciones, que permiten hacer cualquier tipo de operación, como trabajo con archivos y carpetas, procesamiento de formularios, paginación de resultados y muchas otras opciones. Esta amplia librería concede la posibilidad de realizar un producto de acuerdo a las necesidades del usuario.

### **1.6 Framework de desarrollo web**

Es un conjunto de herramientas, bibliotecas y convenciones de programación que se utilizan para facilitar el desarrollo de aplicaciones web. Estos frameworks proporcionan una estructura predefinida y una serie de componentes reutilizables que permiten a los desarrolladores crear aplicaciones web de manera eficiente y con menos código (Chacon, Straub, & Ramírez, 2019).

#### **1.6.1 Framework Laravel**

Es conocido como el framework de los artesanos de la Web. Se trata de una tecnología que permite desarrollar programas escalables, con mucho menos códigos y de forma elegante, al evitar el código desprolijo e interminable. Está basado en el patrón MVC, que implica la modularización de los componentes y partes del software. Esto permite separar elementos de la vista y el diseño, de la estructura de datos, y el código orientado a la lógica (Aguirre, 2021).

#### **1.6.2 Framework Vue.js**

Es un framework de desarrollo web progresivo y de código abierto que se utiliza para construir interfaces de usuario interactivas y sofisticadas. Utiliza componentes reutilizables para construir interfaces de usuarios. Estos componentes son bloques de construcción que se pueden combinar para crear aplicaciones más grandes. Proporciona una amplia gama de transiciones y animaciones que se pueden utilizar para mejorar la experiencia del usuario (Chauhan, 2019).

Es altamente escalable y se puede utilizar para construir aplicaciones pequeñas y grandes. La documentación de Vue.js es clara y completa, lo que facilita su uso y aprendizaje.

### **1.6.3 Framework Nuxt.js**

Es un framework de desarrollo web basado en Vue.js que utiliza para construir aplicaciones web modernas, escalables y de alta calidad. Proporciona una estructura predefinida y una serie de características útiles que permiten a los desarrolladores crear aplicaciones web de manera más eficiente y de menos código. Entre sus características destacadas se encuentra la generación de sitios estáticos, la renderización del lado del servidor y la integración con herramientas de construcción y pruebas (Chauhan, 2020).

### **1.6.4 Framework Vuetify**

Es un Framework de desarrollo web de código abierto para Vue.js que se enfoca en la creación de interfaces de usuario atractiva y responsivas. Ofrece una estética moderna y limpia para las interfaces de usuarios. Además de ofrecer una amplia gama de componentes predefinidos que se pueden personalizar según las necesidades del proyecto, se enfoca en crear interfaces de usuarios responsivas que se adapten a diferentes tamaños de pantalla y dispositivos. Los desarrolladores pueden personalizar fácilmente el tema de la aplicación utilizando las herramientas proporcionadas por Vuetify (Díaz J. , 2020).

## **1.7 Aplicación web**

Una aplicación web es un sistema informático que los usuarios usan accediendo a un servidor web a través de los protocolos de Internet. Las aplicaciones web son populares debido a la practicidad del navegador web como cliente ligero. La facilidad para actualizar y mantener las aplicaciones web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su creciente popularidad (Urgellés, 2012). La característica fundamental es que permite interactuar con el usuario a través de consultas.

Las aplicaciones web son un desarrollo poco costoso, sencillo y rápido. Presenta acceso ubicuo, sin necesidad de distribución e idealmente, con pocos requerimientos técnicos. Con datos centralizados y fácil integración de datos múltiples fuentes. Por sus ventajas es utilizada en este trabajo para llevar a cabo la realización del software (Urgellés, 2012).

### **Los tres niveles típicos de las aplicaciones web son:**

- **El nivel de interfaz de usuario:** Está compuesto por las páginas HTML que el usuario solicita a un servidor web y que visualiza en un cliente web (normalmente, un navegador web).

- **El nivel de lógica de negocio:** Está compuesto por los módulos que implementan la lógica de la aplicación y que se ejecutan en un servidor de aplicaciones.
- **El nivel de datos:** Está compuesto por los datos, normalmente gestionados por un sistema de gestión de bases de datos (servidor de datos), que maneja la aplicación web (Anon, s.f).

## **1.8 Tendencias tecnológicas a considerar**

### **Software libre**

El término software libre se refiere a un programa de ordenador con libertad para su utilización, distribución, modificación y estudio. Desde el punto de vista técnico-legal, se considera software libre a los programas licenciados en términos que garantizan a sus usuarios el derecho de ejecutarlos, copiarlos, distribuirlos, estudiarlos, cambiarlos y mejorarlos (Zanotti, 2016).

### **Arquitectura Cliente-Servidor**

La arquitectura cliente-servidor se define como un modelo de procesos cooperantes necesarios para la realización de una tarea (Polini, 2010). En esta arquitectura el cliente realiza una petición bajo el protocolo HTTP, utilizando generalmente un navegador web y el servidor se encarga de procesar y responder a las solicitudes del cliente.

### **Modelo-vista-controlador (MVC)**

Es un patrón de arquitectura de software que separa los datos y la lógica del negocio de una aplicación, de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado, define componentes para la representación de la información, y por otro lado para la interacción del usuario.

Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento (Tedeschi, 2013).

De manera genérica, los componentes de MVC se podrían definir como sigue:

El **Modelo:** Es la representación de la información con la cual el sistema opera. Gestiona todos los accesos a dicha información, tanto consultas como actualizaciones. Implementa también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la vista aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario).

Las peticiones de acceso o manipulación de información llegan al modelo a través del controlador.

El **Controlador**: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al modelo cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). Se podría decir que el controlador hace de intermediario entre la vista y el modelo.

La **Vista**: Presenta el modelo (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho modelo la información que debe representar como salida.

En este proyecto se decide utilizar el patrón de diseño Modelo-Vista-Controlador debido a las grandes ventajas que destaca:

- ✓ Tiene una **fácil organización**, puesto que solo cuenta con tres componentes.
- ✓ Es un patrón que se puede **adaptar a diferentes frameworks**.
- ✓ Se puede **escalar fácilmente**.
- ✓ **Facilita el trabajo en equipo**.

### 1.9 Gestor de base de datos

Un sistema gestor de base de datos (SGBD) o Database Management System (DBMS) es un conjunto de programas invisibles para el usuario final con el que se administra y gestiona la información que incluye una base de datos.

Los gestores de datos o gestores de base de datos permiten administrar todo acceso a la base de datos, pues tienen el objetivo de servir de interfaz entre esta, el usuario y las aplicaciones.

Entre sus funciones se encuentran la de permitir a los usuarios de negocio almacenar la información, modificar datos y acceder a los activos de conocimiento de la organización. Asimismo, el gestor de base de datos también se ocupa de realizar consultas y hacer análisis para generar informes.

Además, los sistemas de gestión de base de datos pueden entenderse como una colección de datos interrelacionados, estructurados y organizados en el ecosistema formado por dicho conjunto de programas que acceden a ellos y facilitan su gestión (Darias Perez, 2021).

**MySQL** ha sido considerado como el sistema de gestión de base de datos relacionales más común entre los desarrolladores de aplicaciones y de código abierto, creada bajo el respaldo de Oracle Corporation. Tiene sus códigos fuente bajo términos de licencia GNU, es considerada también la mejor opción si se dispone a crear una aplicación open-source (Taran & Silnov, 2017).

Fue seleccionado este gestor de Base de Datos MySQL, porque es uno de los servidores de bases de datos de código abierto más populares y conocidos del mundo, un sistema de manejo de bases de datos con un gran nivel de estabilidad y facilidad de desarrollo que se integra fácilmente con el lenguaje de programación PHP. Dispone, además, de una arquitectura que lo hace extremadamente rápido y fácil de personalizar. Sumándole a todos estos beneficios, que es un servidor que se adecúa perfectamente a las exigencias del cliente.

### 1.10 Servidor web

Un servidor web o servidor HTTP es un programa que, utilizando el modelo Cliente-Servidor y el protocolo HTTP, procesa una aplicación del lado del servidor, a través de conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente y genera o cede una respuesta en cualquier lenguaje o Aplicación del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo (Márquez Díaz, 2011).

**Xampp** es un servidor web local multiplataforma que permite la creación y prueba de páginas web u otros elementos de programación. Sin embargo, Xampp integra una serie de herramientas que potencian y facilitan la experiencia al desarrollador. Es decir, Xampp en sí mismo no es un programa, sino un paquete de programas o software que contiene herramientas de gestión de base de datos (Dongee, 2021).

- **X:** Hace referencia a los diferentes **sistemas operativos** en los que se puede instalar el programa, como Linux, Windows, Mac, Ubuntu, etc.
- **A:** Por el servidor web Apache que permitirá trabajar con las diferentes herramientas del paquete y que será la principal interfaz a usar. Este servidor web de código abierto será fácil de instalar gracias a Xampp.
- **M:** La letra M hace referencia a la incorporación del sistema de gestión de bases de datos conocido como **MySQL**; en algunas versiones de Xampp, sobre todo en las más recientes, incorpora **MariaDB**.
- **P:** Como todo servidor web de código abierto, Xampp utiliza un lenguaje de programación, siendo en este caso **PHP**; este lenguaje es bastante conocido y soporta diferentes sistemas de bases de datos.
- **P: Perl** es otro lenguaje de programación de Xampp, pero esta vez enfocado en la administración del sistema y programación de red.

### Node.js

Es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor pero no limitándose a ello, basándose en el lenguaje de programación JavaScript, asíncrono, con entrada/salida de datos en una arquitectura orientada a

eventos. Fue creada con el enfoque de ser útil en la creación de programas de red altamente escalables, como por ejemplo, servidores web (Finley, 2011).

Fue creado por los desarrolladores originales de JavaScript. Lo transformaron de algo que solo podía ejecutarse en los navegador en algo que se podría ejecutar en los ordenadores como si de aplicaciones independientes se tratara. Gracias a Node.js se puede ir un paso más allá en la programación con JavaScript no solo creando sitios web interactivos, sino teniendo la capacidad de hacer cosas que otros lenguajes de secuencia de comando como Python pueden crear (Lucas, 2019).

### **1.11 Visual Studio Code**

Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft. Es software libre y multiplataforma, está disponible para Windows, GNU/Linux y macOS. VS Code tiene una buena integración con Git, cuenta con soporte para depuración de código, y dispone de un sinnúmero de extensiones, que básicamente te da la posibilidad de escribir y ejecutar código en cualquier lenguaje de programación.

Para tener una idea de la popularidad de Visual Studio Code y la aceptación que ha tenido en el mundo de desarrollo, según una encuesta realizada por Stack Overflow a más de 80,000 desarrolladores en mayo del 2021, Visual Studio Code es el entorno de desarrollo más usado y con mucha diferencia, un 71.06% (Flores, 2022).

La gran customización y la amplia gama de funcionalidades que se le pueden añadir, permiten al desarrollador crear un entorno de trabajo agradable y útil, mejorando la velocidad de desarrollo y la comodidad del programador.

### **1.12 Conclusiones Parciales del Capítulo**

En el desarrollo del primer capítulo se dio a conocer las bases teóricas sobre las cuales se sustenta la propuesta de trabajo. Se analiza cómo se efectúa el proceso y se corroboró la necesidad de diseñar esta aplicación para solucionar los problemas existentes. Quedaron reflejados todos los conceptos relacionados con la presente investigación y se seleccionan las tecnologías y técnicas que se aplicarán al diseño e implementación de la aplicación informática. Se justifica la utilización de una metodología ágil para el desarrollo de la aplicación, específicamente la metodología de programación extrema XP. Se fundamenta la utilización de los lenguajes JavaScript, Html y CSS en el lado del cliente, siendo Nuxt.js, Vue.js y Vuetify escogidos como framework y el lenguaje PHP con Laravel como framework en el lado del servidor, además del uso de Visual Studio Code como entorno de desarrollo integrado y gestor de base de datos MySQL.

## Capítulo 2: Análisis, diseño y desarrollo de la solución propuesta

En el presente capítulo se dan a conocer cuáles son los procesos que serán automatizados. Se exponen los elementos necesarios para la descripción de la solución propuesta a través de las Historias de Usuarios (HU) que acumulan la necesidad existente definida por el cliente, es llevado a cabo el análisis de los requerimientos. Se aplica la Metodología XP, Extreme Programming (Programación Extrema), metodología ágil de desarrollo, con el objetivo de garantizar el diseño de un programa lo más ajustado posible y se logra como ventaja la incorporación del cliente como un miembro del equipo de desarrollo.

### 2.1 Descripción de la solución

El sistema a implementar facilitará a los usuarios la gestión de la información de los procesos investigativos del Centro de Estudios Biotecnológicos de la Universidad de Matanzas, ya que se mantendrá en línea y tendrá mayor información visual sobre los datos de este proceso de forma organizada. Los usuarios podrán acceder al sistema sin estar autenticados, aquí recibirán información como los eventos, publicaciones científicas, premios obtenidos por la institución y demás. Los administradores del sistema realizarán inserciones, búsquedas, eliminaciones y modificaciones en la información. También van a estar los Jefes de Grupos de investigaciones que se van a encargar de gestionar sus respectivos grupos de investigación, así como también van a estar los jefes de proyectos de investigación que se van a encargar de gestionar sus proyectos.

Además el sistema, debe contar con las siguientes características:

1. Rapidez para acceder a la información de la aplicación, ya que el sistema será solamente para gestionar la información de las investigaciones en el centro.
2. Los usuarios contarán con una aplicación sencilla y de fácil uso que estará disponible en todo momento.
3. La información estará más centralizada lo que permite una mejor búsqueda o consulta de la misma.

### 2.2 Roles del Sistema

Administrador del sistema	Es el usuario registrado que interactúa mayormente con el sistema, gestionar la
---------------------------	---

	aplicación, añadir o eliminar el personal, crear usuarios, asignarles permisos, mantener el control de la información de las visitas.
Jefe de Grupo de Investigación	Es el usuario asignado por el administrador que se encarga de gestionar su Grupo de investigación.
Jefe de Proyecto de Investigación	Es el usuario asignado por el administrador que se encarga de gestionar su proyecto de investigación.
Visitante	Es aquel usuario no registrado que tiene solo acceso a percibir toda la información del centro.

Tabla 2: Roles del Sistema. Fuente: Elaboración del autor.

### 2.3 Materiales y herramientas utilizadas

Descripción	Herramienta
Metodología de desarrollo	Programación Extrema XP
Lenguaje de programación	PHP 8.1.6
Servidor Web	XAMPP
Gestor de base de datos	MySQL 10.4.24
Framework web	Laravel 9.0

Tabla 3: Herramientas Utilizadas. Fuente: Elaboración del autor.

### 2.4 Etapa de planificación

La etapa de planificación es la etapa inicial de todo el proyecto. Se realiza con el objetivo de lograr una eficiente organización del prototipo inicial del problema y proporcionar así un buen comienzo a una solución eficaz. Con este objetivo y según las ideas del cliente sobre el software se desarrollarán las Historias de Usuarios, mediante la cual se obtendrá un punto de partida para el resto de la planificación del proyecto. Igualmente



se realizará un estimado de cada una de las entregas del proyecto y del tiempo, basándose en que la planificación inicial se podría afectar debido a cambios que pudiesen sufrir estos aspectos durante el desarrollo del proyecto.

#### 2.4.1 Equipo de trabajo y roles

La metodología XP define roles de trabajo asociando a cada uno con diversas actividades. A continuación, se definen los roles, quedando designado el programador que sería el encargado de producir el código del sistema, el jefe de proyecto y el cliente que no es más que el que escribe las historias de usuario, les asigna la prioridad y diseña las pruebas funcionales para validar su implementación. A continuación, se muestra la asignación de estos roles a las personas responsables.

Miembro	Roles
Ing. Eduardo Berrio Turiño	Jefe de proyecto
Dra. C. Madyu Matos Trujillo	Cliente
Rolando Baez Naranjo	Programador, Tester

Tabla 4: Roles del equipo. Fuente: Elaboración del autor.

#### 2.4.2 Historias de usuario

Las historias de usuario son artefactos utilizados en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento las historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas (Joskowicz, 2008).

Se delimitaron un grupo de historias de usuario que corresponden con las funcionalidades que debe contener el sistema, así como con los requisitos no funcionales que delimitan herramientas fundamentales en el desarrollo de la aplicación y características que debe cumplir la misma.

Escalas equivalentes a la prioridad en el negocio:

- **Alta:** Asignada a las Historias de Usuario que corresponden a funcionalidades esenciales en el desarrollo del proyecto, a las que el cliente define como primordiales.
- **Media:** Dada a las Historias de Usuario que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación directa sobre el proyecto que se esté desarrollando.
- **Baja:** Se le otorga a las Historias de Usuario que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no fueron pedidas expresamente por el cliente.

Escala nominal de riesgo de desarrollo:

- **Alta:** Cuando para la implementación de la Historia de Usuario se considera la posible existencia de errores que lleven a inoperatividad del código.
- **Media:** Cuando pueden aparecer errores en la implementación de la Historia de Usuario que puedan retrasar la entrega de la versión.
- **Baja:** Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

En la tabla siguiente se muestran las historias de usuarios que surgieron a partir de las entrevistas con el cliente cubriendo todas las funcionalidades de la aplicación, aunque la planificación es flexible ante los cambios que puedan ocurrir durante el desarrollo del proyecto.

Queda definido el nivel de prioridad (P) con el que deben darle solución a las HU que sirve como guía para el desarrollo de las mismas. Se determina el riesgo (R). El número de la iteración (I) donde se realizará. Se definen también los puntos estimados (E) requeridos por historia, que no son más que los tiempos en los que se concibió inicialmente el desarrollo de cada HU. Un punto se considera como una semana ideal de trabajo, donde se trabaje el tiempo planeado sin ningún tipo de interrupción.

No	Nombre	P	R	I	E	Entrega
1	Diseño y creación de Base de Datos.	Alta	Alta	1	2	1
2	Diseño de la interfaz de usuario.	Medio	Medio	1	1	1

3	Autenticación.	Alta	Medio	1	1	1
4	Gestionar Usuarios.	Alta	Medio	1	2	1
5	Gestionar Personal Registrado.	Medio	Medio	2	3	2
6	Asignar Roles.	Alta	Medio	2	1	2
7	Gestionar Grupos de Investigación.	Alta	Medio	2	2	2
8	Gestionar Proyectos de Investigación.	Alta	Medio	2	2	2
9	Gestionar Publicaciones Científicas.	Alta	Medio	3	2	3
10	Gestionar Premios.	Medio	Baja	3	1	3
11	Gestionar Actividades de Postgrado.	Medio	Baja	3	1	3
12	Gestionar Revistas.	Baja	Medio	3	1	3
13	Gestionar Libros.	Baja	Medio	4	1	4
14	Generar Grafica con la información de las visitas a la página por la variación de la fecha.	Alta	Alta	4	3	4
15	Gestionar Contenido	Medio	Medio	4	1	4

Tabla 5: Historias de Usuarios Planificadas. Fuente: Elaboración del autor.

A continuación, en las tablas siguientes se muestran una descripción detallada de las Historias de Usuario más importantes en el desarrollo del proyecto.

<b>Historia de usuario</b>	
<b>Número:</b> 1	<b>Usuario:</b> Administrador
<b>Nombre:</b> Diseño y creación de la base de datos	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Rolando Baez Naranjo	

<b>Descripción:</b> Se creará el esquema de base de datos para la carga de información teniendo en cuenta las relaciones existentes entre las tablas además de validar la carga de la información y la recuperación de la misma para las transacciones que se realizaran entre la misma y el sistema web.
<b>Observaciones:</b> Crear los roles del sistema en el gestor de base de datos para una mayor seguridad.

Tabla 5.1: Historia de usuario No1: Diseño y creación de la Base de Datos.

Historia de usuario	
<b>Número:</b> 2	<b>Usuario:</b> Administrador
<b>Nombre:</b> Diseño de la interfaz de usuario	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Rolando Baez Naranjo	
<b>Descripción:</b> Se diseñan las interfaces de la aplicación.	
<b>Observaciones:</b> Resaltar los colores de la empresa y potenciar la experiencia de usuario.	

Tabla 5.2: Historia de usuario No2: Diseño de la interfaz de usuario.

Historia de usuario	
<b>Número:</b> 3	<b>Usuario:</b> Administrador
<b>Nombre:</b> Autenticación	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Rolando Baez Naranjo	
<b>Descripción:</b> El administrador podrá crear un nuevo usuario con toda la información requerida como nombre de usuario y contraseña. Para que los usuarios puedan acceder al sistema deben autenticarse insertando su correo y contraseña.	
<b>Observaciones:</b> Se debe verificar si son correctos los datos para acceder al sistema y si el usuario está registrado en la BD.	

Tabla 5.3: Historia de usuario No3: Autenticación.

Historia de usuario	
<b>Número:</b> 4	<b>Usuario:</b> Administrador
<b>Nombre:</b> Gestionar Usuarios	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Rolando Baez Naranjo	
<b>Descripción:</b> El Administrador puede obtener, asignar un usuario a partir de una persona registrada, actualizar y eliminar los usuarios del sistema. También puede asignarlo como Jefe de Grupo de Investigación o como jefe de Proyecto de Investigación.	
<b>Observaciones:</b> Crear los roles del sistema en el gestor de base de datos para una mayor seguridad.	

Tabla 5.4: Historia de usuario No4: Gestionar Usuarios.

Historia de usuario	
<b>Número:</b> 8	<b>Usuario:</b> Administrador, Jefe de Proyecto de Investigación.
<b>Nombre:</b> Gestionar Grupos de Investigación.	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Rolando Baez Naranjo	
<b>Descripción:</b> El Administrador o el Jefe de su respectivo grupo de investigación puede editar el grupo de investigación y eliminarlo si es preciso. El Administrador también puede crear un nuevo grupo de investigación y asignarle un Jefe de Grupo de Investigación.	
<b>Observaciones:</b> El Jefe de Grupo de Investigación solo puede gestionar su respectivo grupo de investigación.	

Tabla 5.5: Historia de usuario No8: Gestionar Grupos de Investigación.

### 2.4.3 Plan de Iteraciones

El proyecto está conformado por cuatro iteraciones, por lo que se obtuvo un total de cuatro entregas para las cuales se desarrollaron partes de la aplicación completamente funcionales. Para la determinación de cada una de las iteraciones en cada ciclo, se

implementan y prueban historias de usuario seleccionadas según su prioridad en el sistema, además del criterio y necesidad del cliente. Cada ciclo comienza con una reunión donde se planifica dicha iteración y una entrevista con el cliente, para captar indicaciones. Una vez concluida la iteración que va a estar conformada por un conjunto de historias de usuarios, teniendo en cuenta los requisitos exigidos por el cliente, la aplicación poseerá mayor número de funcionalidades.



Tabla 6: Plan de Iteraciones. Fuente: Elaboración Propia.

#### 2.4.4 Reuniones

El planeamiento es esencial para cualquier tipo de metodología, es por ello que XP requiere de una revisión continua del plan de trabajo. A pesar de no llevar a cabo una documentación extensa, es muy estricta a la hora de la organización del trabajo. Quincenalmente se realizará una reunión general con el equipo de trabajo con el objetivo de ver el progreso de cada iteración y en caso de algún inconveniente se deberá planificar una reunión aparte que deberá ser dentro de dicha quincena.

#### 2.5 Etapa de diseño

En XP solo se diseñan aquellas historias de usuario que el cliente ha seleccionado para la iteración actual por dos motivos: por un lado, se considera que no es posible tener un diseño completo del sistema y sin errores desde el principio. El segundo motivo es que, dada la naturaleza cambiante del proyecto, el hacer un diseño muy extenso en las fases iniciales del proyecto para luego modificarlo, se considera un desperdicio de tiempo. Es

importante resaltar que esta tarea es permanente durante la vida del proyecto partiendo de un diseño inicial que va siendo corregido y mejorado en el transcurso del proyecto.

### 2.5.1 Prototipo de interfaz de usuario

En la **Imagen 6** se muestra la página principal de la aplicación web.



Imagen No. 1: Página principal de la aplicación web

### 2.6 Arquitectura propuesta

Los patrones arquitectónicos son plantillas que describen los principios estructurales globales que construyen las distintas arquitecturas de software viables. Plantean una organización estructural fundamental para un sistema de software, expresando un conjunto de subsistemas predefinidos, especificando responsabilidades y organizando las relaciones entre ellos (Wilson, 2018).

#### 2.6.1-Patrón Arquitectónico tres capas

En el caso del desarrollo del sistema de la presente investigación se ha definido el diseño de una arquitectura distribuida en tres capas, el dividir una aplicación en capas separadas que desempeñan diferentes roles y funcionalidades, ayuda a mejorar el mantenimiento del código, permite también diferentes tipos de despliegue y contribuye a organizar, reutilizar y rediseñar mejor los componentes que integran el sistema.

#### 2.6.2-Diseño de capas:

- **Presentación:** Es la capa que se encarga de presentar información e interactuar con la capa externa (usuario). Se define en general como Interfaz Gráfica de Usuario (GUI).
- **Lógica de aplicación:** Es la capa que se encarga de implementar las operaciones solicitadas por los usuarios a través de la capa de presentación.
- **Acceso a datos:** Es la que permite el acceso a la base de datos del sistema.

## 2.7 Patrones de diseño

Un patrón de diseño provee un esquema para refinar componentes de un sistema de software y la forma en que se relacionan entre sí. Describe una estructura generalmente recurrente de comunicación de componentes que resuelve un problema de diseño general dentro de un contexto particular (Rojas, s.f).

A continuación, se definen algunos patrones a utilizar:

### 2.7.1 Patrones GRASP

**Experto:** Este patrón se utiliza para asignar la responsabilidad que tendrá cada clase, en dependencia de la información que maneja. Se evidencia en la definición de las clases de acuerdo a las funcionalidades que deben realizar a partir de la información manejada dentro del sistema, como por ejemplo las clases controladoras.

**Creador:** Este patrón se utiliza para asignar a determinadas clases la responsabilidad de crear instancias de otras. En las clases controladoras se encuentran definidas las acciones del sistema y se ejecutan cada una de ellas. En las acciones se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase controladora es "creador" de dichas entidades.

**Bajo Acoplamiento:** Este patrón es evidenciado en la distribución del sistema, cada clase controladora maneja la información de las entidades que están relacionadas con las funcionalidades que se especifican en dicha clase controladora para lograr un bajo acoplamiento de clases.

## 2.8 Tareas de Ingeniería

Para esta fase se dividen las funcionalidades operativas que conforman las HU en tareas más pequeñas, las que toman el nombre de tareas de ingeniería. Estas tienen el objetivo de realizar un análisis más detallado de cada funcionalidad y lograr una estimación más real de su tiempo de desarrollo. Lo principal de las tareas de ingeniería es que son importantes para guiar la implementación por lo que no necesariamente deben ser entendibles para el cliente por ser descritas en un lenguaje técnico.

No.	Nombre HU	Tarea de Ingeniería
1	Diseño y creación de la base de datos	<ol style="list-style-type: none"><li>1. Diseñar la Base de Datos.</li><li>2. Crear la Base de Datos.</li></ol>



<b>2</b>	Diseño de la interfaz de usuario	3. Diseñar la interfaz de usuario.
<b>3</b>	Autenticación	4. Autenticar usuario existente.
<b>4</b>	Gestionar Usuarios	5. Listar Usuarios. 6. Mostrar Usuario. 7. Asignar Usuario. 8. Editar Usuario. 9. Eliminar Usuario.
<b>5</b>	Gestionar Personal Registrado	10. Listar Personal. 11. Mostrar Personal. 12. Crear Personal. 13. Editar Personal. 14. Eliminar Personal.
<b>6</b>	Asignar Roles	15. Obtener Roles. 16. Mostrar Roles. 17. Asignar Roles.
<b>7</b>	Gestionar Grupos de Investigación.	18. Listar Grupo de Investigación. 19. Mostrar Grupo de Investigación. 20. Crear Grupo de Investigación. 21. Editar Grupo de Investigación. 22. Eliminar Grupo de Investigación. 23. Asignar Jefe de Grupo de Investigación.
<b>8</b>	Gestionar Proyectos de Investigación	24. Listar Proyectos de Investigación. 25. Mostrar Proyecto de Investigación. 26. Crear Proyecto de Investigación.

		<p>27. Editar Proyecto de Investigación.</p> <p>28. Eliminar Proyecto de Investigación.</p> <p>29. Asignar Jefe del Proyecto de Investigación.</p> <p>30. Asignar Grupo de Investigación al que pertenece.</p>
<b>9</b>	Gestionar Publicaciones Científicas	<p>31. Listar Publicaciones.</p> <p>32. Mostrar Publicación.</p> <p>33. Crear Publicación.</p> <p>34. Editar Publicación.</p> <p>35. Eliminar Publicación.</p>
<b>10</b>	Gestionar Premios	<p>36. Listar Premios.</p> <p>37. Mostrar Premios.</p> <p>38. Crear Premio.</p> <p>39. Editar Premio.</p> <p>40. Eliminar Premio.</p>
<b>11</b>	Gestionar Actividades de Postgrado	<p>41. Listar Actividades de Postgrado.</p> <p>42. Crear nueva Actividad.</p> <p>43. Mostrar Actividades de Postgrado.</p> <p>44. Eliminar Actividad de Postgrado.</p> <p>45. Editar Actividad de Postgrado.</p>
<b>12</b>	Gestionar Revistas	<p>46. Crear Revista.</p> <p>47. Editar Revista.</p>

		48. Eliminar Revista.
13	Gestionar Libros	49. Crear Libro. 50. Editar Libro. 51. Eliminar Libro.
14	Generar Grafica con la información de las visitas a la página por la variación de la fecha	52. Mostrar grafica Lineal de las visitas a la página.
15	Gestionar Contenido	53. Editar Contenido. 54. Agregar Imagen. 55. Eliminar Imagen.

Tabla 7: Tareas de ingenierías por historias de usuario. Fuente: Elaboración del autor.

En las tablas que se muestran a continuación se relacionan algunas tareas de ingeniería que tenían mayor peso en el desarrollo de esta investigación.

Tarea de Ingeniería	
<b>Número de tarea:</b> 1	<b>Número de historia de usuario:</b> 1
<b>Nombre:</b> Diseño de la base de datos	
<b>Tipo de tarea:</b> Diseño	<b>Puntos estimados:</b> 0.5
<b>Programador responsable:</b> Rolando Baez Naranjo	
<b>Descripción:</b> Analizar profundamente el negocio para crear la base de datos que permita almacenar dicha información.	

Tabla 7.1: Tarea de ingeniería 1. Fuente: Elaboración del autor.

Tarea de Ingeniería	
<b>Número de tarea:</b> 2	<b>Número de historia de usuario:</b> 1
<b>Nombre:</b> Creación de la base de datos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.5
<b>Programador responsable:</b> Rolando Baez Naranjo	
<b>Descripción:</b> Crear la base de datos con sus respectivas relaciones y la integridad correspondiente entre las tablas.	

Tabla 7.2: Tarea de ingeniería 2. Fuente: Elaboración del autor.

Tarea de Ingeniería	
<b>Número de tarea: 3</b>	<b>Número de historia de usuario: 2</b>
<b>Nombre:</b> Diseñar la interfaz de usuario.	
<b>Tipo de tarea:</b> Diseño	<b>Puntos estimados:</b> 1.5
<b>Programador responsable:</b> Rolando Baez Naranjo	
<b>Descripción:</b> Al realizar el levantamiento de requisitos el cliente brinda información de los requisitos no funcionales que desea que posea el sistema. El cliente pide que la interfaz del usuario sea amigable y de fácil uso, que tenga una paleta de color relacionada al logo de la institución. Pide que la página posea un carrusel de imágenes, cada rol tendrá una interfaz parecida, pero con sus privilegios.	

Tabla 7.3: Tarea de ingeniería 3. Fuente: Elaboración del autor.

Tarea de Ingeniería	
<b>Número de tarea: 7</b>	<b>Número de historia de usuario: 4</b>
<b>Nombre:</b> Asignar Usuario.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1.5
<b>Programador responsable:</b> Rolando Baez Naranjo	
<b>Descripción:</b> Los Usuarios no se pueden crear, son asignados a partir del personal registrado que ya existe en el sistema.	

Tabla 7.4: Tarea de ingeniería 7. Fuente: Elaboración del autor.

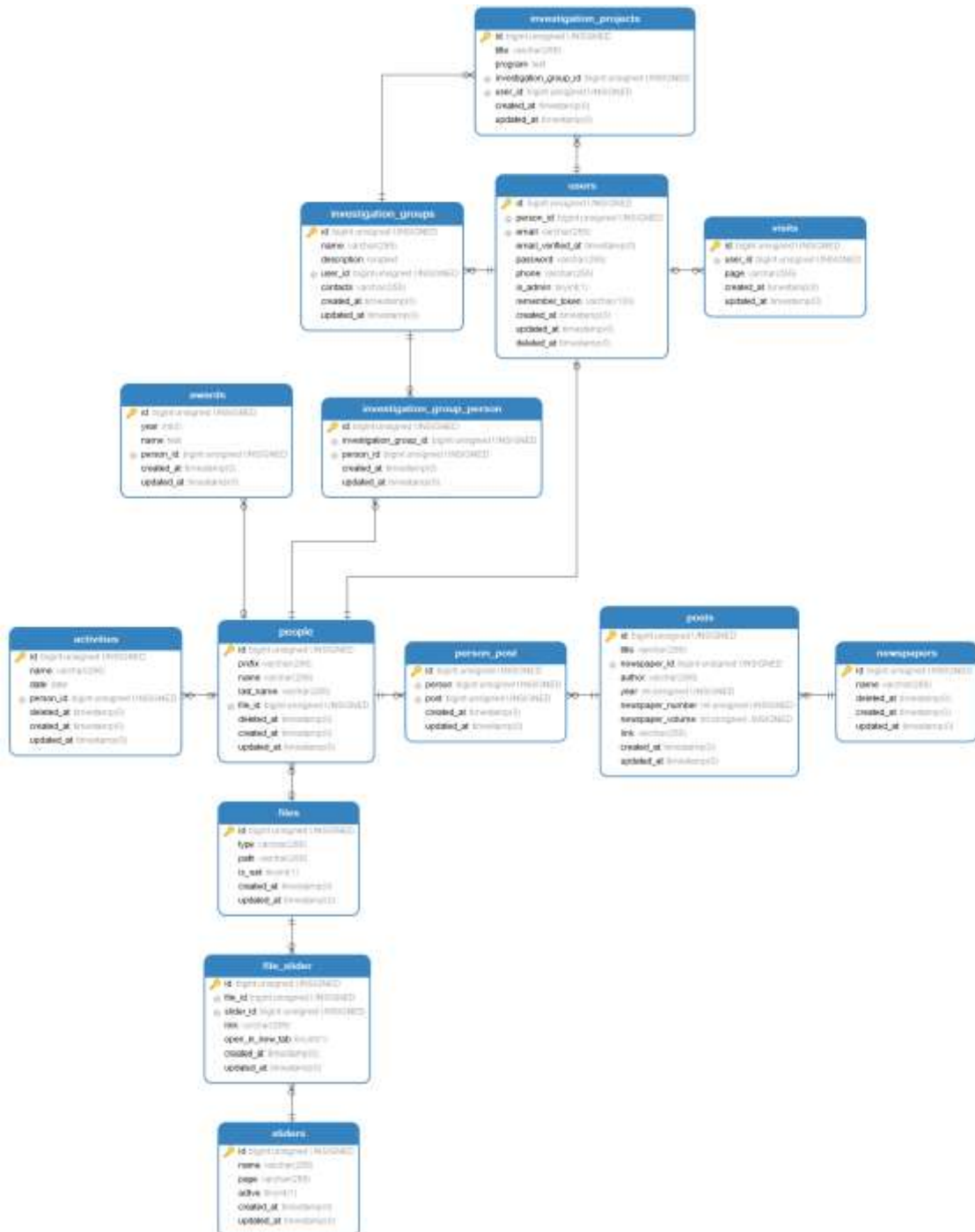
Tarea de Ingeniería	
<b>Número de tarea:20</b>	<b>Número de historia de usuario: 7</b>
<b>Nombre:</b> Crear Grupo de Investigación.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1.5
<b>Programador responsable:</b> Rolando Baez Naranjo	

**Descripción:** El Administrador puede crear un nuevo grupo de investigación, llenando los campos validos que pide el sistema.

Tabla 7.5: Tarea de ingeniería 20. Fuente: Elaboración del autor.

## 2.9 Base de Datos

En la ilustración se observa el modelo físico de la base de datos de la aplicación web



## Imagen No. 2: Modelo Físico de la Base de Datos

### 2.10 Estudio De Factibilidad

Para determinar la factibilidad del producto que se está desarrollando es necesario realizar un proceso de estimación, teniendo en cuenta aspectos que puedan afectarla: la estructuración, los recursos involucrados, la complejidad del proyecto, el tamaño y posibles riesgos (Pressman, 2014).

Existen numerosos modelos para realizar una estimación del costo de un software en desarrollo: análisis de Puntos de Función, COCOMO I, COCOMO II, entre otros.

#### 2.10.1 Método Punto de Función

Para este proyecto usaremos Análisis de Puntos de Función, ya que permite solucionar algunos problemas existentes en el cálculo del tamaño del software por líneas de código (existentes en modelos como COCOMO I y COCOMO II) y medidas de productividad, sobre todo, en sus estimaciones al usar diferentes niveles de lenguajes de programación. Como alternativa, se mide la funcionalidad del software a través del punto de vista del usuario e ignora lenguaje de programación y tecnologías (por ejemplo: frameworks) utilizados en la implementación. Está enfocado en la funcionalidad y complejidad de cualquier tipo de software (Web, Servidores, Mobiles, entre otros). Para medir el tamaño de la aplicación desde el punto de vista funcional o del usuario se utilizan los Puntos de Función, las cuales, se desglosan en aspectos externos de las aplicaciones de software: consultas, interfaces, archivos lógicos, entradas y salidas (Busquelle, 2010).

Para obtener una correcta estimación del esfuerzo realizado, así como el tamaño del software, se utilizó la técnica de IFPUG con un enfoque micro.

#### **Componentes:**

**EO:** son los procesos que envían datos hacia el exterior de la aplicación.

**EI:** son los procesos que envían datos hacia el interior de la aplicación y actualiza cualquier archivo interno.

**EQ:** son procesos donde se involucra entrada y salida de datos. En el caso de la entrada de datos no puede producir modificaciones en ningún archivo, en caso de la salida de datos no puede contener información derivada de la original.

**EIF:** son grupos de datos mantenidos externamente.

**ILF:** son grupos de datos internos en el sistema y vinculados entre sí.

Para obtener los Puntos de Función Sin Ajustar, es necesario la información obtenida de los archivos EIF e ILF, las transacciones EO, EI y EQ a través del anexo 1 para aplicar el peso indicado a cada complejidad de cada archivo.

Componente	Bajo	Medio	Alto	Total
EI	$14 * 3 = 42$	$0 * 4 = 0$	$0 * 6 = 0$	42
EO	$0 * 4 = 0$	$0 * 5 = 0$	$0 * 7 = 0$	0
EQ	$11 * 3 = 33$	$0 * 4 = 20$	$0 * 6 = 0$	33
ILF	$6 * 7 = 42$	$0 * 10 = 0$	$0 * 15 = 0$	42
EIF	$0 * 5 = 0$	$0 * 7 = 0$	$0 * 10 = 0$	0
				<b>117</b>

Tabla 8: Cantidad de componentes por su peso. Fuente: Elaboración del autor.

### Calculo de Puntos de Función Sin Ajustar (PFSA)

Los PFSA se calculan como la suma de los productos de cada componente por su peso determinado en la tabla correspondiente.

$$\text{PFSA} = 42+0+33+42+0$$

$$\text{PFSA} = 117$$

Con el valor de los PFSA calculado, es necesario ajustarlo auxiliándose de 14 características generales. Esto permite obtener una estimación de las condiciones de trabajo dependiendo del sistema que se va a desarrollar. A cada característica se le asigna un peso que fluctúa entre 0 y 5 indicando la importancia de estas sobre el sistema, donde cero indica que el factor no presenta influencia en la aplicación y cinco cuando el factor presenta gran influencia sobre la misma. La siguiente tabla están ilustradas las 14 características generales a tener en cuenta. Finalmente se calcula el Grado Total de Influencia (GTI) sumando los aportes de cada característica, para luego obtener el Factor de Ajuste (Portillo, 2004).

### 2.10.2 Cálculo de los Puntos de Función Ajustados (PFA).

Nº	Nombre del Factor	Valor
----	-------------------	-------

1	Comunicación de datos	3
2	Procesamiento distribuido	1
3	Rendimiento	4
4	Configuración de Explotación Compartida	1
5	Tasa de transacciones	1
6	Entrada de Datos en Línea	4
7	Eficiencia con el Usuario Final	4
8	Actualizaciones en Línea	1
9	Procesamiento complejo	1
10	Reusabilidad del Código	1
11	Facilidad de implementación	1
12	Facilidad de Operación	4
13	Instalaciones Múltiples	2
14	Facilidad de Cambios	3
	<b>Ajuste de Complejidad Técnica (ACT)</b>	<b>31</b>

Tabla 9: Cálculo de los puntos de función y ajustados. Fuente: Elaboración del autor.

### 2.10.3 Puntos de Función Ajustados (PFA)

$$PFA = PFSA * [0.65 + [0.01 * ACT]]$$

$$PFA = 117 * (0.65 + (0.01 * 31))$$

$$PFA = 112.32$$

### 2.10.4 Cálculo del Esfuerzo

Líneas de Código (LC)

$$LC = PFA * (\text{Líneas por PF})$$

Para calcular las Líneas por PF o Líneas por Puntos de Función se utilizó el **Anexo 4**.

$$LC = 112.32 * 20$$

$$LC = 2247$$

### 2.10.5 Esfuerzo en horas / persona (E)

$$E = \frac{PFA}{\left(\frac{1 \text{ persona}}{8 \text{ horas}}\right)}$$



$$E = \frac{112.32}{(8)} = 898.56 \text{ horas}$$

Tomando 26 días laborables en el mes y 8 horas productivas al día, se obtiene 208 horas laborables al mes.

$$\text{Duración al mes (DM)} = \frac{E}{\left(208 \frac{\text{horas}}{\text{mes}}\right)}$$

$$\text{Duración al mes (DM)} = \frac{898.56 \text{ horas}}{\left(208 \frac{\text{horas}}{\text{mes}}\right)} = 5 \text{ meses}$$

### 2.10.6 Cálculo del Presupuesto del Proyecto

Costo Total del proyecto = sueldo de 1 participante \* cantidad de participantes \* DM

Costo Total del proyecto = 4000 \* 1 \* 5 = 20000

### 2.10.7 Análisis de los beneficios

Realizando un análisis del costo respecto a los beneficios descritos a lo largo de este trabajo se puede concluir que los aspectos positivos son superiores a los costos. Además, debe señalarse que para esta institución este software resultó gratuito puesto que es el resultado del trabajo de diploma del autor.

### 2.11 Conclusiones Parciales del Capítulo

Las principales conclusiones a las que se pudieron llegar en este capítulo son:

1. Mediante la metodología XP, en este capítulo se reflejaron las etapas imprescindibles para desarrollar el software, exceptuando las pruebas del mismo. Se planteó la composición del equipo de desarrollo, las diferentes historias de usuarios y las respectivas interacciones necesarios para acometer la labor de construcción de la aplicación. Se realizó la planificación del proyecto y análisis de costo de este aplicando el modelo de Puntos de Función.
2. Los aspectos abordados anteriormente permitieron reflejar el diseño, estructura y funcionalidad del software a desarrollar, proporcionando una visión clara del proyecto en su conjunto para que las distintas partes involucradas en el mismo llegasen a un acuerdo.

## Capítulo 3: Validación de la solución propuesta

En este capítulo se realizan las pruebas al software que permiten comprobar la calidad de este producto, lo que constituye uno de los pasos más importantes en el desarrollo de un sistema. No debe existir ninguna característica en el programa que no haya sido probada con la intención de mostrar un error no descubierto hasta entonces y con el fin de verificar la fiabilidad y calidad de la aplicación como un todo.

### 3.1 Pruebas al software

Para la metodología XP el uso de pruebas es fundamental puesto que es la manera de comprobar que las funcionalidades que se van implementando funcionan correctamente y cumplen con lo requerido por el cliente. XP anima a probar constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. Con el proceso de pruebas se determina el status de la calidad de un producto. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos o si es el software que se quería desarrollar. En las pruebas se usan casos de prueba, especificados de forma estructurada mediante Técnicas de Prueba.

Los niveles de trabajo en los cuales se pueden realizar las pruebas son:

- Prueba de Unidad.
- Prueba de Integración.
- Prueba de Sistema.
- Prueba de Aceptación.
- Prueba de Seguridad.

No.	Nombre HU	Pruebas a realizar
1	Diseño y creación de la base de datos	Test base de datos
2	Diseño de la interfaz de usuario	Test diseño de la interfaz de usuario
3	Autenticación	Test autenticar usuarios.
4	Gestionar Usuarios	Test Listar Usuarios. Test Mostrar Usuario. Test Asignar Usuario. Test Editar Usuario. Test Eliminar Usuario.

5	Gestionar Personal Registrado	Test Listar Personal. Test Mostrar Personal. Test Crear Personal. Test Editar Personal. Test Eliminar Personal.
6	Asignar Roles	Test Obtener Roles. Test Mostrar Roles. Test Asignar Roles.
7	Gestionar Grupos de Investigación.	Test Listar Grupo de Investigación. Test Mostrar Grupo de Investigación. Test Crear Grupo de Investigación. Test Editar Grupo de Investigación. Test Eliminar Grupo de Investigación. Test Asignar Jefe de Grupo de Investigación.
8	Gestionar Proyectos de Investigación	Test Listar Proyectos de Investigación. Test Mostrar Proyecto de Investigación. Test Crear Proyecto de Investigación. Test Editar Proyecto de Investigación. Test Eliminar Proyecto de Investigación. Test Asignar Jefe del Proyecto de Investigación. Test Asignar Grupo de Investigación al que pertenece.
9	Gestionar Publicaciones Científicas	Test Listar Publicaciones.

		Test Mostrar Publicación. Test Crear Publicación. Test Editar Publicación. Test Eliminar Publicación.
10	Gestionar Premios	Test Listar Premios. Test Mostrar Premios. Test Crear Premio. Test Editar Premio. Test Eliminar Premio.
11	Gestionar Actividades de Postgrado	Test Listar Actividades de Postgrado. Test Crear nueva Actividad. Test Mostrar Actividades de Postgrado. Test Eliminar Actividad de Postgrado. Test Editar Actividad de Postgrado.
12	Gestionar Revistas	Test Crear Revista. Test Editar Revista. Test Eliminar Revista.
13	Gestionar Libros	Test Crear Libro. Test Editar Libro. Test Eliminar Libro.
14	Generar Grafica con la información de las visitas a la página por la variación de la fecha	Test Mostrar grafica Lineal de las visitas a la página.
15	Gestionar Contenido	Test Editar Contenido. Test Agregar Imagen. Test Eliminar Imagen.

Tabla 10. Plan de Pruebas por Historias de Usuario. Fuente: Elaboración del autor.

### 3.1.1 Pruebas de Aceptación

Las pruebas de aceptación (PA) se hacen con el propósito de demostrar al cliente el cumplimiento de un requisito del software. Se caracterizan por:

- Describir un escenario (secuencia de pasos) de ejecución o uso del sistema desde la perspectiva del cliente.

- Estar asociada a un requisito funcional o requisito no funcional.
- Cubrir desde escenarios típicos/frecuentes hasta los más excepcionales.
- Tener la mayor cantidad de instanciaciones (ejecuciones con valores concretos) posibles.
- Ser el tester quien realice el trabajo de diseñar las instanciaciones y su aplicación.

Uno de los pilares de la metodología XP es el uso de pruebas para comprobar el funcionamiento de los códigos que se vayan implementando. Las pruebas de aceptación son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. Dado que la responsabilidad es grupal, es recomendable publicar los resultados de las pruebas de aceptación, de manera que todo el equipo esté al tanto de esta información.

### 3.1.2- Vista Crear Personal

Condición	Clases Válidas	Clases Inválidas de Entrada
Prefijo	1- En el campo escribir el prefijo. 2- Dejar el campo vacío.	-
Nombre	3- En el campo escribir el nombre	4- Dejar el campo vacío. 5- No puede escribir números ni caracteres especiales
Apellidos	6- En el campo escribir el apellido	7- Dejar el campo vacío. 8- No puede escribir números ni

		caracteres especiales
--	--	-----------------------

Tabla 11. Prueba de aceptación. Fuente: Elaboración del autor.

No.	Clases de equivalencias	Prefijo	Nombre	Apellidos	Resultados Esperados
1	2, 4, 7				<ul style="list-style-type: none"> <li>Nombre no puede estar vacío</li> <li>Apellidos no puede estar vacío.</li> </ul>
2	5, 8		·\$%&/8	“·!4%	<ul style="list-style-type: none"> <li>El formato del campo nombre es invalido.</li> <li>El formato del campo apellidos es invalido.</li> </ul>
3	1, 3, 6				<ul style="list-style-type: none"> <li>Personal Creado con éxito.</li> </ul>

Tablas 11.1 Clases de equivalencia. Fuente: Elaboración del autor.

En la **Tabla11.2** se muestra la PA 1: Test Crear Personal

Tabla de Pruebas	
<b>No</b>	1
<b>Requerimiento</b>	Estar conectado a la base de datos.
<b>Objetivo</b>	Probar la acción de crear personal (para cubrir las clases inválidas2, 4, 7).
<b>Tipo de Prueba</b>	Funcional.
<b>Hardware</b>	Sistema de cómputo con un Procesador Intel CORE i3 - Disco Duro de 1tB - Memoria RAM de 8GB.

<b>Software</b>	Sistema Operativo Windows 10 - Base de Datos MySQL - Navegador de Internet Mozilla Firefox 38.0, Google Chrome 47.0 e Internet Explorer 11.0.	
<b>Personal</b>	Ingeniero de Pruebas.	
<b>Caso de Prueba</b>	1	
<b>Datos de Entrada</b>	Equipos:	
	Prefijo: Nombre: Apellidos:	
<b>Resultados Esperados</b>	<b>Mensaje</b> Prefijo, Nombre, Apellidos <ul style="list-style-type: none"> <li>• “Nombre no puede estar vacío.”</li> <li>• “Apellidos no puede estar vacío.”</li> </ul>	
<b>Resultados Obtenidos</b>	SI(X) No()	
<b>Casos de Excepción</b>		
<b>Aprobado Por:</b> Rolando Baez Naranjo	<b>Cargo:</b> Jefe de Proyecto	<b>Líder:</b> Rolando Baez Naranjo

Tabla 11.2. PA 1 Fuente: Elaboración Propia

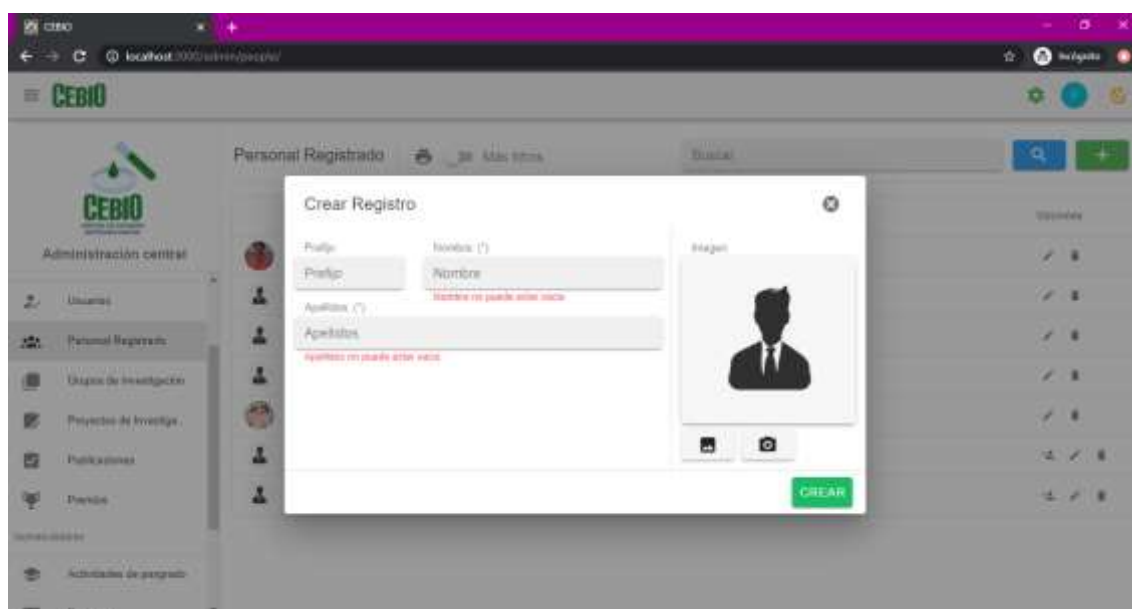


Imagen No. 3 Vista Crear Personal.

En la **Tabla11.3** se muestra la PA 2: Test Crear Personal

### Tabla de Pruebas

<b>No</b>	2	
<b>Requerimiento</b>	Estar conectado a la base de datos.	
<b>Objetivo</b>	Probar la acción de crear personal (para cubrir las clases inválidas 5, 8).	
<b>Tipo de Prueba</b>	Funcional.	
<b>Hardware</b>	Sistema de cómputo con un Procesador Intel CORE i3 - Disco Duro de 1tB - Memoria RAM de 8GB.	
<b>Software</b>	Sistema Operativo Windows 10 - Base de Datos MySQL - Navegador de Internet Mozilla Firefox 38.0, Google Chrome 47.0 e Internet Explorer 11.0.	
<b>Personal</b>	Ingeniero de Pruebas.	
<b>Caso de Prueba</b>	1	
<b>Datos de Entrada</b>	Equipos:	
	Nombre:	
	Apellidos:	
<b>Resultados Esperados</b>	<b>Mensaje</b> Nombre, Apellidos <ul style="list-style-type: none"> <li>• “El formato del campo nombre es invalido.”</li> <li>• “El formato del campo apellidos es invalido.”</li> </ul>	
<b>Resultados Obtenidos</b>	SI(X) No()	
<b>Casos de Excepción</b>		
<b>Aprobado Por:</b> Rolando Baez Naranjo	<b>Cargo:</b> Jefe de Proyecto	<b>Líder:</b> Rolando Baez Naranjo

Tabla 11.3 PA 2 Fuente: Elaboración Propia



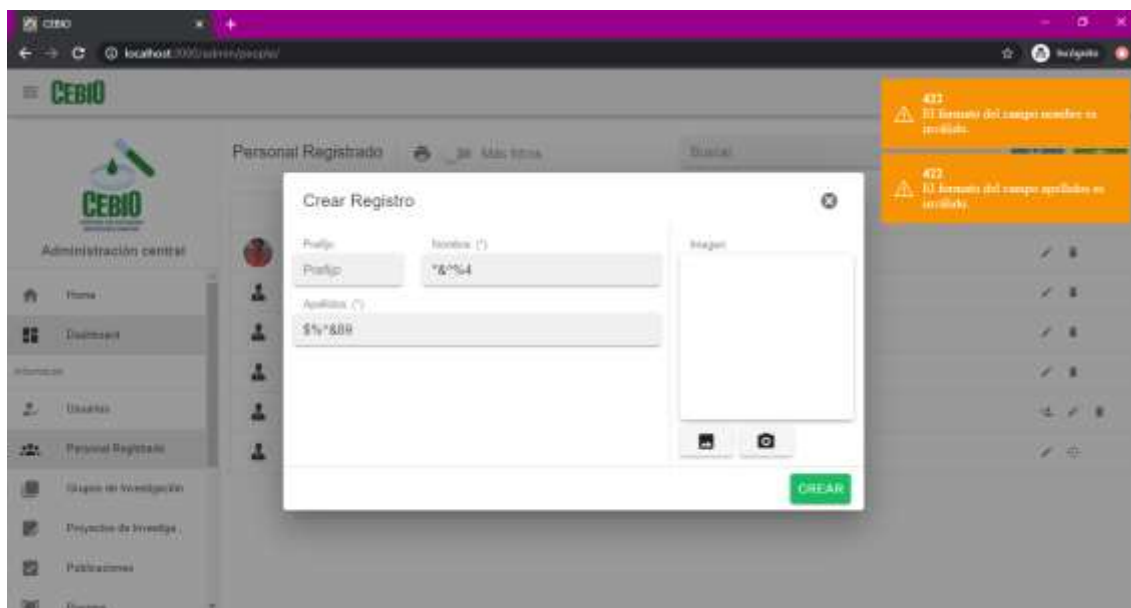


Imagen No. 4 Vista Crear Personal.

En la **Tabla11.4** se muestra la PA 3: Test Crear Personal

Tabla de Pruebas	
<b>No</b>	3
<b>Requerimiento</b>	Estar conectado a la base de datos.
<b>Objetivo</b>	Probar la acción de crear personal (para cubrir las clases validas 1, 3, 6).
<b>Tipo de Prueba</b>	Funcional.
<b>Hardware</b>	Sistema de cómputo con un Procesador Intel CORE i3 - Disco Duro de 1tB - Memoria RAM de 8GB.
<b>Software</b>	Sistema Operativo Windows 10 - Base de Datos MySQL - Navegador de Internet Mozilla Firefox 38.0, Google Chrome 47.0 e Internet Explorer 11.0.
<b>Personal</b>	Ingeniero de Pruebas.
<b>Caso de Prueba</b>	1
<b>Datos de Entrada</b>	Equipos:

	Prefijo: Nombre: Apellidos:	
<b>Resultados Esperados</b>	<b>Mensaje</b> Prefijo, Nombre, Apellidos "Personal creado con éxito."	
<b>Resultados Obtenidos</b>	SI(X) No()	
<b>Casos de Excepción</b>		
<b>Aprobado Por:</b> Rolando Baez Naranjo	<b>Cargo:</b> Jefe de Proyecto	<b>Líder:</b> Rolando Baez Naranjo

Tabla 11.4 PA 3 Fuente: Elaboración Propia

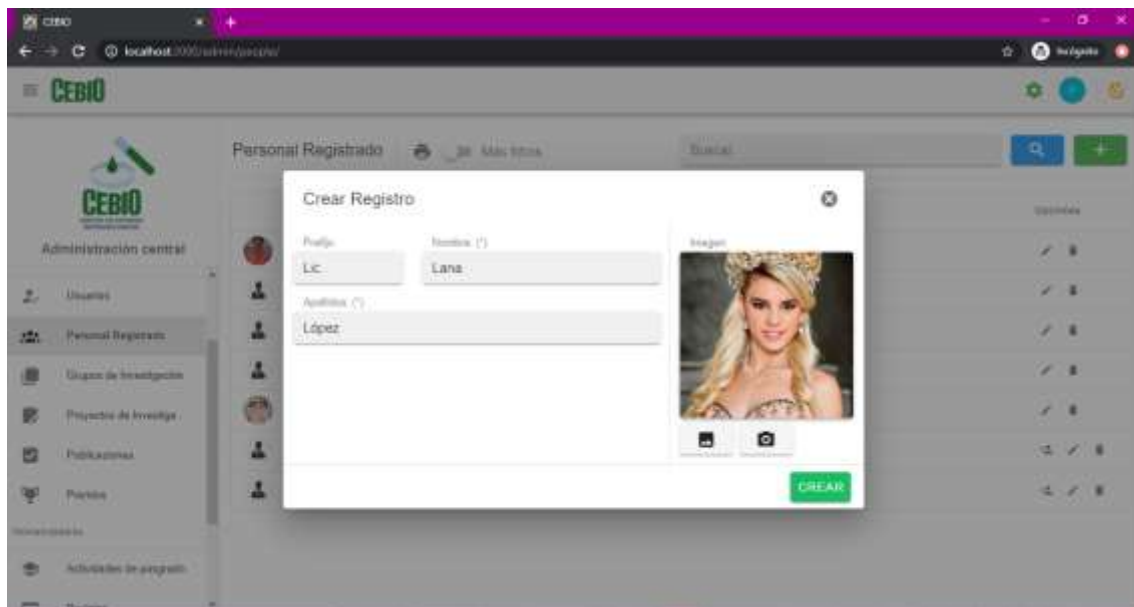


Imagen No. 5 Vista Crear Personal.

### 3.1.3 Vista Crear Grupo de Investigación

Condición	Clases Válidas	Clases Inválidas de Entrada
Nombre (N)	1- En el campo escribir un nombre.	2- Dejar el campo vacío.

Descripción (D)	3- En el campo escribir o añadir la descripción.	4- Dejar el campo vacío.
Contacto (C)	5- En el campo rellenar el contacto.	6- Dejar el campo vacío.
Jefe (J)	7- En el campo seleccionar el Jefe	8- En el Select, no seleccionar nada.
Profesores (P)	9- En el campo seleccionar los profesores	10- En el Select, no seleccionar nada.

Tabla 12. Prueba de aceptación. Fuente: Elaboración del autor.

No.	Clases de equivalencias	N	D	C	J	P	Resultados Esperados
1	2,4,6						<ul style="list-style-type: none"> <li>Nombre no puede estar vacío.</li> <li>Contacto no puede estar vacío.</li> <li>Descripción no puede estar vacío.</li> </ul>
2	8,10						<ul style="list-style-type: none"> <li>El campo user id es</li> </ul>

							obligatorio. <ul style="list-style-type: none"> <li>El campo people debe ser un array</li> </ul>
3	1, 3, 5, 7, 9	Aditivos Tecnicos	Es una prueba	anaflores@nautacu	Ana Flores	Tomas Acosta	<ul style="list-style-type: none"> <li>Grupo de Investigación creado con éxito.</li> </ul>

Tablas 12.1 Clases de equivalencia. Fuente: Elaboración del autor.

En la **Tabla 12.2** se muestra la PA 1: Test Crear Grupo de Investigación

Tabla de Pruebas	
<b>No</b>	1
<b>Requerimiento</b>	Estar conectado a la base de datos.
<b>Objetivo</b>	Probar la acción de crear Grupo de Investigación (para cubrir las clases inválidas 2, 4, 6).
<b>Tipo de Prueba</b>	Funcional.
<b>Hardware</b>	Sistema de cómputo con un Procesador Intel CORE i3 - Disco Duro de 1tB - Memoria RAM de 8GB.
<b>Software</b>	Sistema Operativo Windows 10 - Base de Datos MySQL - Navegador de Internet Mozilla Firefox 38.0, Google Chrome 47.0 e Internet Explorer 11.0.
<b>Personal</b>	Ingeniero de Pruebas.
<b>Caso de Prueba</b>	1
<b>Datos de Entrada</b>	Equipos:
	Nombre:
	Descripción:
	Contacto:

<b>Resultados Esperados</b>	<b>Mensaje</b> Nombre, Descripción, Contacto <ul style="list-style-type: none"> <li>• “Contacto no puede estar vacío.”</li> <li>• “Descripción no puede estar vacía.”</li> <li>• “Contacto no puede estar vacío.”</li> </ul>	
<b>Resultados Obtenidos</b>	SI(X) No()	
<b>Casos de Excepción</b>		
<b>Aprobado Por:</b> Rolando Baez Naranjo	<b>Cargo:</b> Jefe de Proyecto	<b>Líder:</b> Rolando Baez Naranjo

Tabla 12.2. PA 1 Fuente: Elaboración Propia

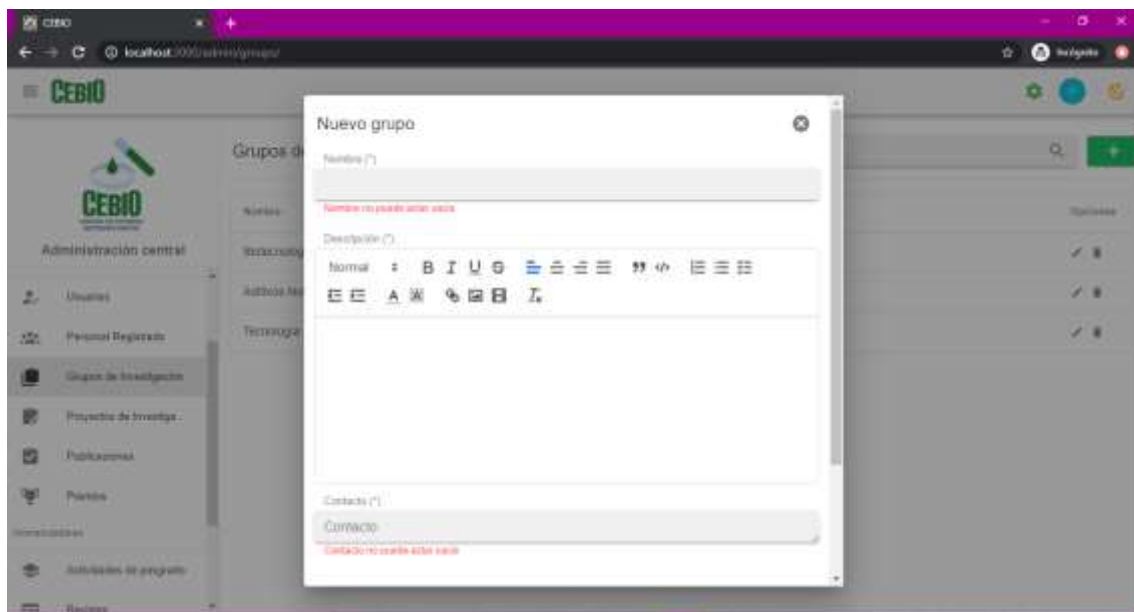


Imagen No. 6 Vista Crear Grupo de Investigación

En la **Tabla12.3** se muestra la PA 2: Test Crear Grupo de Investigación

<b>Tabla de Pruebas</b>	
<b>No</b>	2
<b>Requerimiento</b>	Estar conectado a la base de datos.
<b>Objetivo</b>	Probar la acción de crear Grupo de Investigación (para cubrir las clases inválidas 8, 10).
<b>Tipo de Prueba</b>	Funcional.
<b>Hardware</b>	Sistema de cómputo con un Procesador Intel CORE i3 - Disco Duro de 1tB - Memoria RAM de 8GB.

<b>Software</b>	Sistema Operativo Windows 10 - Base de Datos MySQL - Navegador de Internet Mozilla Firefox 38.0, Google Chrome 47.0 e Internet Explorer 11.0.	
<b>Personal</b>	Ingeniero de Pruebas.	
<b>Caso de Prueba</b>	1	
<b>Datos de Entrada</b>	Equipos:	
	Jefe: Profesores:	
<b>Resultados Esperados</b>	<b>Mensaje</b> Jefe, Profesores <ul style="list-style-type: none"> <li>• “El campo user id es obligatorio.”</li> <li>• “El campo people debe ser un array.”</li> </ul>	
<b>Resultados Obtenidos</b>	SI(X) No()	
<b>Casos de Excepción</b>		
<b>Aprobado Por:</b> Rolando Baez Naranjo	<b>Cargo:</b> Jefe de Proyecto	<b>Líder:</b> Rolando Baez Naranjo

Tabla 12.3 PA 2 Fuente: Elaboración Propia

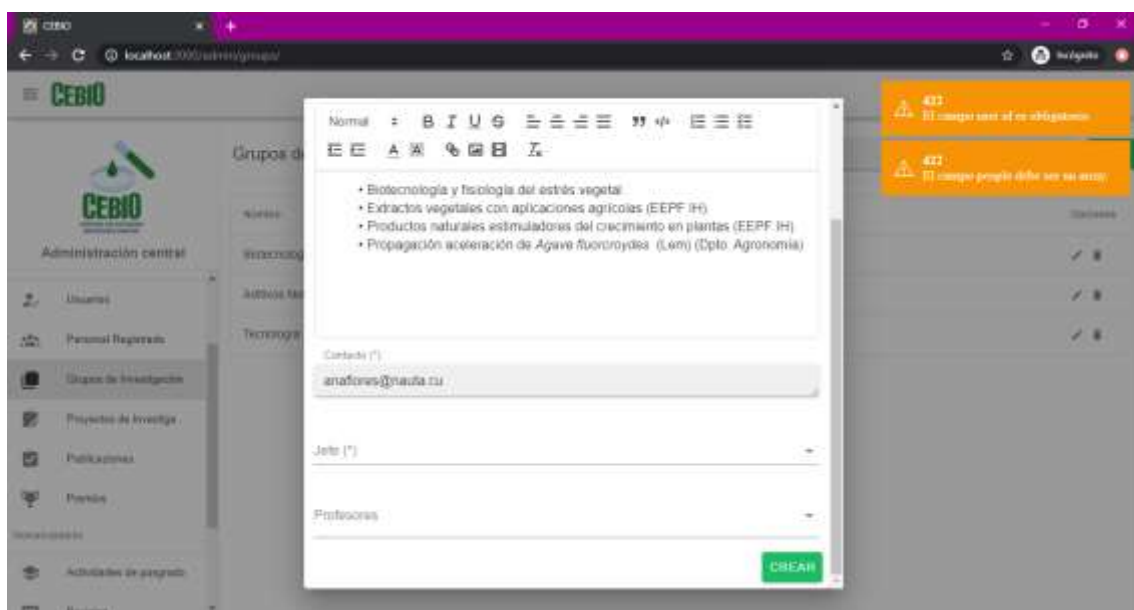


Imagen No. 7 Vista Crear Grupo de Investigación.

En la **Tabla 12.4** se muestra la PA 3: Test Crear Grupo de Investigación

<b>Tabla de Pruebas</b>	
<b>No</b>	3
<b>Requerimiento</b>	Estar conectado a la base de datos.

<b>Objetivo</b>	Probar la acción de crear Grupo de Investigación (para cubrir las clases validas 1, 3, 5, 7, 9).	
<b>Tipo de Prueba</b>	Funcional.	
<b>Hardware</b>	Sistema de cómputo con un Procesador Intel CORE i3 - Disco Duro de 1tB - Memoria RAM de 8GB.	
<b>Software</b>	Sistema Operativo Windows 10 - Base de Datos MySQL - Navegador de Internet Mozilla Firefox 38.0, Google Chrome 47.0 e Internet Explorer 11.0.	
<b>Personal</b>	Ingeniero de Pruebas.	
<b>Caso de Prueba</b>	1	
<b>Datos de Entrada</b>	Equipos:	
	Nombre: Descripción: Contacto: Jefe: Profesores:	
<b>Resultados Esperados</b>	<b>Mensaje</b> Nombre, Descripción, Contacto, Jefe, Profesores <ul style="list-style-type: none"> <li>• “Grupo de investigación creado con éxito.”</li> </ul>	
<b>Resultados Obtenidos</b>	SI(X) No()	
<b>Casos de Excepción</b>		
<b>Aprobado Por:</b> Rolando Baez Naranjo	<b>Cargo:</b> Jefe de Proyecto	<b>Líder:</b> Rolando Baez Naranjo

Tabla 12.4 PA 3 Fuente: Elaboración Propia

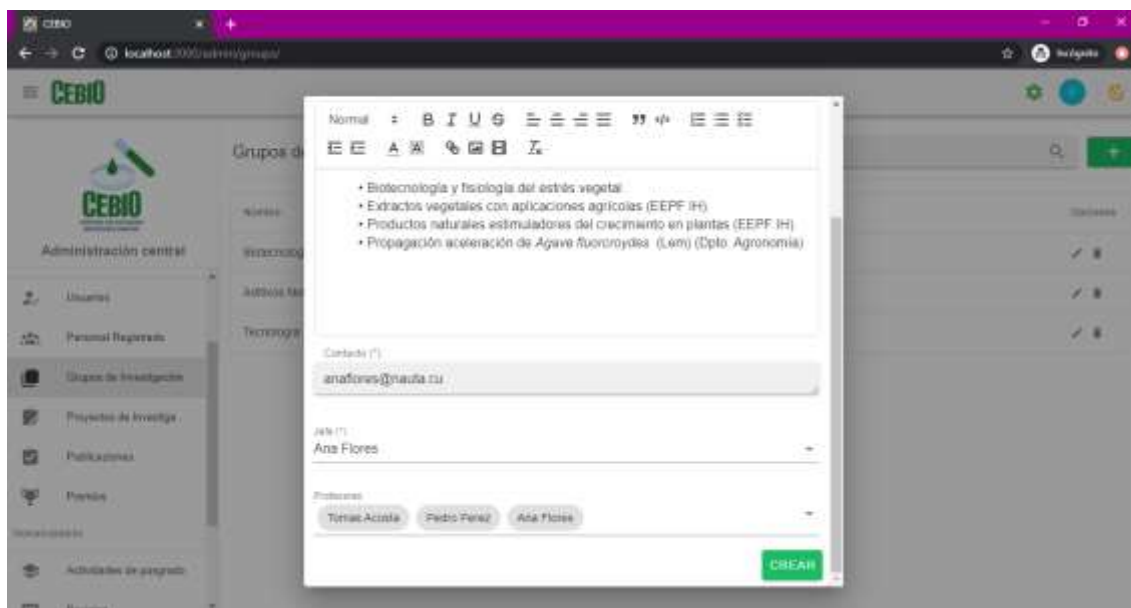


Imagen No. 8 Vista Crear Grupo de Investigación.

### 3.2 Pruebas de Seguridad

Las aplicaciones web están expuestas a un gran número de amenazas, tales como el mal uso por parte de los usuarios, ataques de seguridad, mala calidad del software o falta de un proceso de desarrollo estructurado. Para evitar que estos ataques se aprovechen de las vulnerabilidades comunes al software y causen un daño, es necesario que desde el principio en el ciclo de desarrollo se tomen medidas para disminuir el impacto, lo cual comienza desde la fase de levantamiento de requerimientos hasta la de entrega y soporte al cliente, pasando por las pruebas funcionales (Díaz S. , 2018).

De acuerdo como se plantea VEGA es una plataforma de software libre diseñada para realizar y ejecutar pruebas sobre la seguridad de las páginas web.

Las principales características son:

- Capacidad de realización de Análisis de Vulnerabilidades y Crawler automático.
- Ejecución de pruebas de Accesibilidad UI.
- Capacidad de realizar un Crawler (descargar copia) del website al completo.
- Función de manipulación manual de paquetes HTTP mediante interceptación mediante Proxy (similar a Paros Proxy, ZAP, etc.)
- Análisis del contenido
- Mensajes de notificación personalizables
- Modelo de datos propio
- API en JavaScript personalizable para el desarrollo de complementos y extensiones personalizadas.



Para realizar las pruebas de seguridad se utilizó VEGA en su versión 1.0, la cual nos permite realizar un conjunto de pruebas que se dividen en 2 grandes módulos que se observan en las siguientes imágenes.

La **Figura** muestra Injection Modules

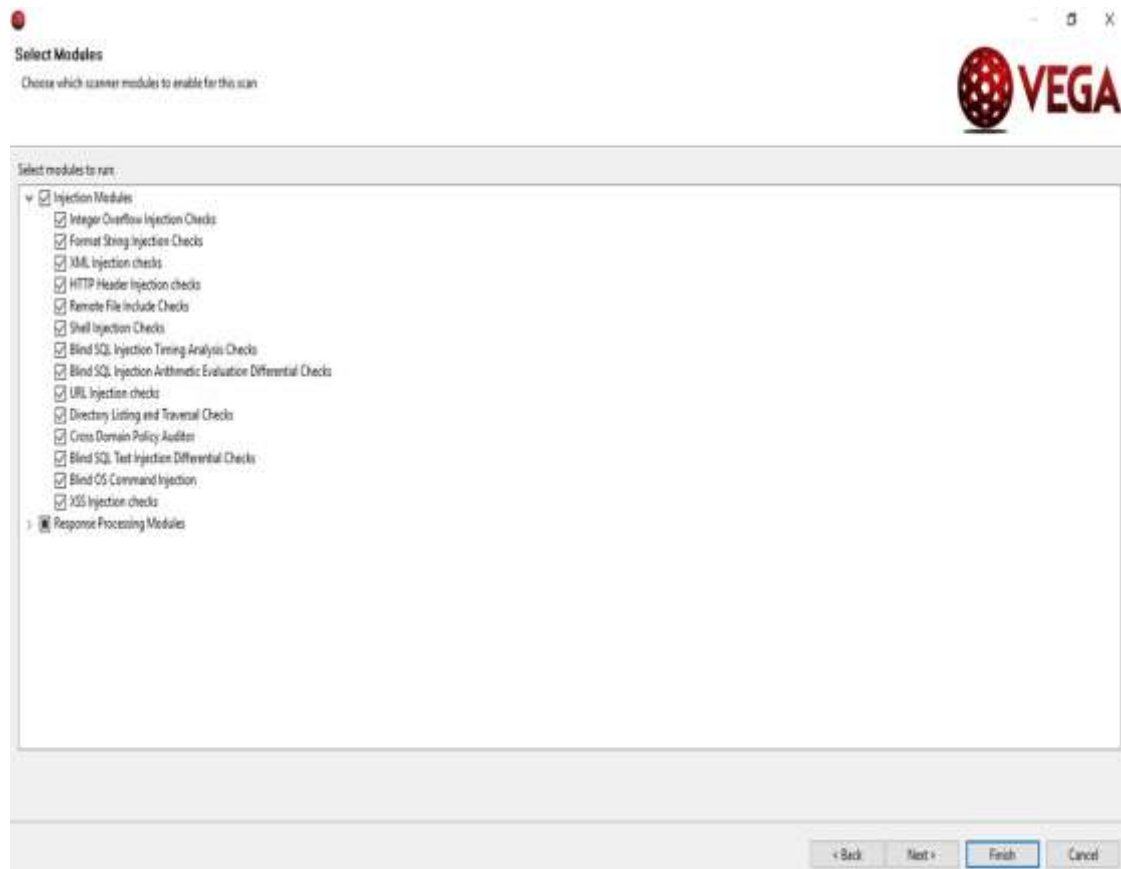


Imagen No. 9 Injection Modules

La **Figura** muestra Response Processing Modules.

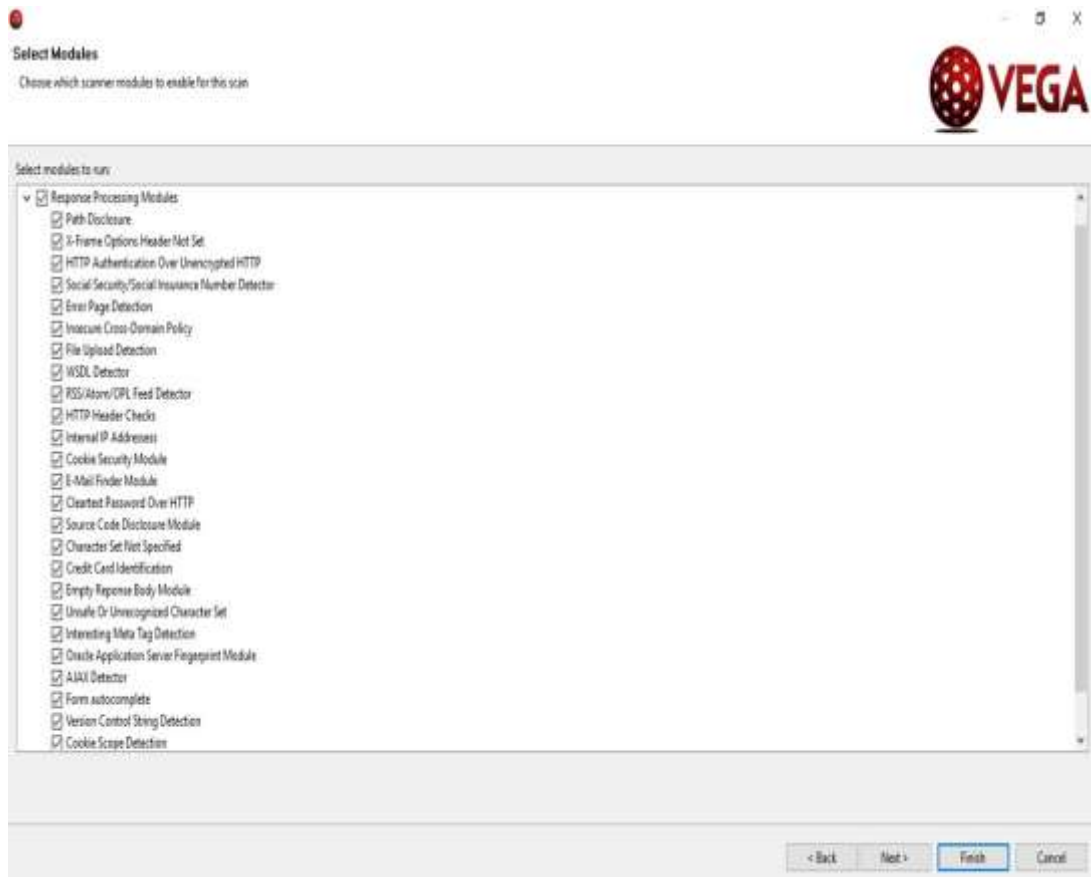


Imagen No. 10 Processing Modules.

Seleccionando ambos módulos se le realizó al software una prueba en su totalidad obteniendo como resultado:

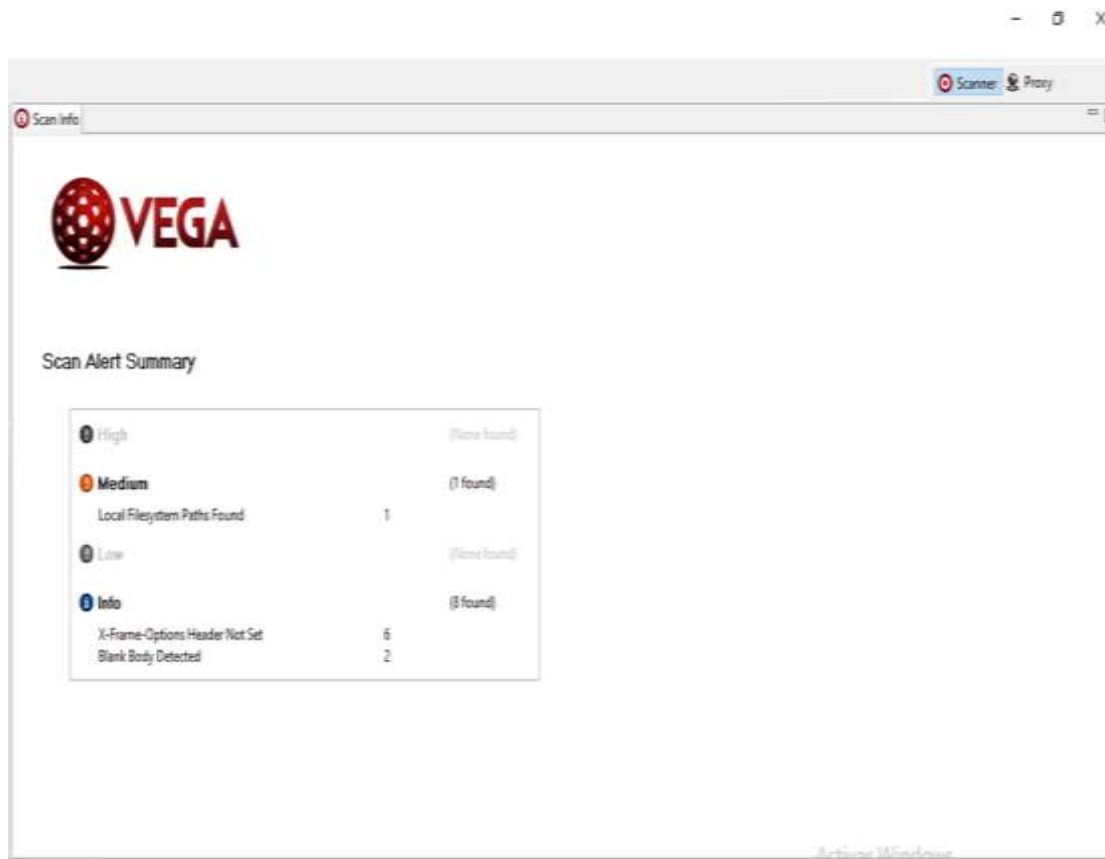


Imagen No. 11 Prueba Vega

Esta alerta detectada por el Vega es genera por causa de que la página web está en localhost, debido a esto las rutas son estáticas.

### 3.3 Conclusiones parciales del Capítulo 3

En este capítulo se detallaron elementos técnicos de la implementación y prueba del sistema de forma tal que se evidenciaran los resultados obtenidos y asegurara la calidad del producto a través de las diferentes pruebas realizadas. En un primer momento se desglosaron y resolvieron las historias de usuarios en pequeñas funcionalidades que constituyeron tareas de ingeniería mejorando así la estimación realizada en el acápite anterior. Las pruebas de aceptación propuestas por la metodología seleccionada, permitieron verificar si los resultados obtenidos fueron los esperados, documentando todos los resultados arrojados, quedando el producto listo para usarse en el Centro de Estudios Biotecnológico de la Universidad de Matanzas.

## Conclusiones Generales

A lo largo de la investigación se describen los pasos para darle cumplimiento al objetivo de desarrollar una herramienta que permita gestionar información en el Centro de Estudios Biotecnológico de la Universidad de Matanzas.

Una vez finalizado el trabajo se plantean las siguientes conclusiones

1. El estudio realizado sobre los antecedentes, la bibliografía y documentos relacionados con el objeto de estudio, permitió confeccionar un marco teórico referencial que en conjunto al estudio de las herramientas y tecnologías a utilizar, proporcionó los elementos necesarios para dar solución a la problemática abordada.
2. El análisis realizado permitió la selección de las herramientas (framework Laravel), metodología de desarrollo (Programación extrema XP), lenguajes de programación (PHP) y técnicas más apropiadas para dar solución al objetivo general fueron resultado del análisis realizado.
3. Se utilizó la metodología XP para el desarrollo de la aplicación Web que se presenta, así como tecnologías de avanzada para el diseño de la aplicación.
4. Mediante la aplicación de pruebas a la aplicación Web, resultó posible obtener resultados favorables, con el consiguiente análisis de errores detectados que fueron subsanados como parte del desarrollo de esta aplicación.

De manera general, se concluye que el software desarrollado posee la funcionalidad y calidad demandada, presentándose como una herramienta atractiva, cómoda, segura e intuitiva a la hora de trabajar con la información del Centro de Estudios Biotecnológicos de la Universidad de Matanzas.

## Recomendaciones

A partir de los resultados o beneficios que proporciona este trabajo de diploma, se proponen las siguientes recomendaciones.

- Continuar la ampliación de la aplicación mediante la incorporación de nuevas funcionalidades que brinden un mayor control y análisis de los datos almacenados.
- Proporcionar un número superior de notificaciones, reportes y gráficos que posibiliten una interpretación superior del flujo de información.

## Bibliografía

- Aguirre, S. (2021). *Framework total Vol I: crea apps desde cero con Laravel + Bootstrap + MySQL*. Plandos.
- Ahmed, A. (2010). *Agile software development: Impact on productivity and quality Management of Innovation and Technology (ICMIT)*.
- Anon. (s.f). *Introducción a las Aplicaciones Web.Universidad de Sevilla*. Obtenido de <http://www.josedomingo.org/web/mod/resource/view.php?id=2337>
- Arévalo, A. J. (2019). *Gestión de la Información, gestión de contenidos y conocimientos*. Obtenido de Jornadas de trabajo del Grupo SIOU, Salamanca: Gestión de la Información, gestión de contenidos y conocimientos, Arévalo, Alonso: <http://eprints.rclis.org/11273/1/Jornadas GRUPO SIOU.pdf>
- Barrios, N. (2018). *La gestión de información y sus recursos*. Obtenido de <http://revistas.mes.edu.cu/greenstone/collect/repo/import/repo/20090606/0006176X010305.pdf>
- Busquelle, J. E. (2010). Análisis de puntos de función. *Lámpsakos*(4). Obtenido de <https://funlam.edu.co/revistas/index.php/lampsakos/article/view/806>
- Chacon, S., Straub, B., & Ramírez, A. (2019). Editorial O'Reilly.
- Chauhan, A. (2019). *Learning Vue.js 2: Learn how to build amazing and complex reactive web applications easily with Vue.js*. Packt Publishing Ltd.
- Chauhan, A. (2020). *Nuxt.js: What is it and why you should use it ?* Obtenido de <https://www.simform.com/nuxt-js-guide/>
- Darias Perez, S. (2021). *Gestor de Base de Datos: Que es, funcionalidades y ejemplos*. Obtenido de Intelequia: <https://intelequia.com/blog/post/gestor-de-base-de-datos-qu%C3%A9-es-funcionalidades-y-ejemplos>
- Díaz, J. (2020). *Aprender a desarrollar aplicaciones web con Vue.js*. Editorial Rama.
- Díaz, S. (2018). *Pruebas de seguridad en aplicaciones web como imperativo en la calidad de*. s.l: s.n.
- Dimes, T. (2015). *JavaScript Una Guía de Aprendizaje para el Lenguaje de Programación JavaScript*. Estados Unidos: Babelcube Incorporated.
- Dongee. (2021). Obtenido de <https://www.dongee.com/tutoriales/que-es-xampp/>
- Faniran, V., Badru, A., & Ajayi, N. A. (2017). *Adopting Scrum as an Agile Approach in Distributed Software Development: A Review of Literature 1st International Conference on Next Generation Computing Applications (NextComp)*. Obtenido de [https://www.researchgate.net/publication/309120635\\_Adopting\\_Scrum\\_as\\_an\\_Agile\\_Approach\\_in\\_Distributed\\_Software\\_Development\\_A\\_Review\\_of\\_Literature?enrichId=rgreq-cc384e356398f9caa934d02d22c7a066-xxx&enrichSource=Y292ZXJQYWdlOzMwOTYzNTtBUoxMTI1ODY1OTU4](https://www.researchgate.net/publication/309120635_Adopting_Scrum_as_an_Agile_Approach_in_Distributed_Software_Development_A_Review_of_Literature?enrichId=rgreq-cc384e356398f9caa934d02d22c7a066-xxx&enrichSource=Y292ZXJQYWdlOzMwOTYzNTtBUoxMTI1ODY1OTU4)
- Finley, K. (2011). *Wait, What's Node.js Good for Again ?* Obtenido de Red Write Web.

- Flores, F. (2022). *Que es VisualStudio Code y que ventajas ofrece*. Obtenido de <https://openwebinar.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>
- Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 Y JAVAScript* MARCOMBO. S.A.
- González, M. J. (1998). Introducción al estudio de la información y la documentación. Obtenido de <http://www.uahurtado.cl/estudiantes/formacion-integral/cursos-optativos-de-formaciongeneral->
- Infante, C. M. (s.f). *Sistema de Gestión de Información para el Centro de Tecnologías para la Formación*. Obtenido de <http://fec.uh.cu/CUGIO/1%20acciones/ProyectosProtocolos/15/MIC%20Mailin%20Carballos>
- Joskowicz. (2008). *Reglas y Prácticas en Extreme Programming*. s.l: s.n.
- Khurana, H., & Sohal, J. S. (2011). Agile: The necessitate of contemporary software developers. *International Journal of Engineering*.
- Kirmani, M. M. (2017). Agile methods for mobile application development: Acomparative analysis. *International Journal of Advanced Research in Computer Science* .
- Lara, P. L. (s.f). *Los Sistemas de Gestión de Información, piedra angular de la estrategia integral de*. Obtenido de <http://www.bibliociencias.cu/gsd/collect/eventos/index/assoc/HASH01f2.dir/doc.pdf>
- Lucas, J. (2019). *Que es Node.js y para qué sirve*. Obtenido de <https://openwebinars.net/blog/que-es-nodejs/>
- Mahalakshmi, M., & Sundararajan, M. (2015). *Tracking the student's performance in Web-based education using Scrum methodology* 015 *International Conference on Computing and Communications Technologies (ICCT)*.
- Márquez Díaz, J. (2011). *Instalación y configuración de Apache, un servidor web*. s.l: s.n.
- Menzinsky, A., López, G., & Palacio, J. (2019). *Scrum Manager: Temario Troncal I* (Vol. Vol. 2.6.1). In. S.Master (Ed).
- Molina Ríos, J. R. (2018). *Comparación de metodologías en aplicación web*. s.l: s.n.
- Molina Ríos, J. R., Zea Ordóñez, M. P., & Contenido Segarra, M. J. (2017). *Metodología de desarrollo en aplicaciones web*. s.l: s.n.
- Navarro Cadavid, A., Fernández Martínez, J. D., & Morales Vélez, J. (2013). Revisión de metodologías ágiles para el desarrollo de software. *PROSPECTIVA*, 30-39. Obtenido de <http://www.redalyc.org/articulo.oa?id=496250736004>
- Pérez, D. C. (2021). *ceupe*. Obtenido de <https://www.ceupe.com/blog/que-es-un-sistema-de-gestion-de-informacion.html>
- Polini, D. M. (2010). *Implementación de un servidor FTP utilizando el modelo cliente/servidor*. s.l: s.n.
- Portillo, A. P. (2004). Estudio de Tecnicas basadas en Puntos de Funci on para la Estimacion del Esfuerzo en Proyectos Software. *Recuperado el, 2.*

Pressman2014*Software Engineering: A Practitioner's Approach*

Pressman, R. S. (2010). *Ingeniería del software: Un enfoque práctico (Séptima edición ed.)*. McGRAW-HILL INTERAMERICANA EDITORES.

Rojas, O. J. (s.f). *Patrones de Diseño*. Obtenido de [http://eva.uci.cu/file.php/158/Documentos/Recursos\\_bibliograficos/Libros\\_y\\_articulos\\_UD\\_1/Diseno de software/Patrones de Diseno Art. 2.pdf](http://eva.uci.cu/file.php/158/Documentos/Recursos_bibliograficos/Libros_y_articulos_UD_1/Diseno_de_software/Patrones_de_Diseno_Art.2.pdf)

Rosado Gómez, A., Quintero Duarte, A., & Meneses Guevara, C. D. (2012). Desarrollo ágil de software aplicando programación extrema. *Revista Ingenio*, 24-29. Obtenido de <https://doi.org/https://doi.org/10.22463/2011642X.2003>

Santos, D. (2022). *Introducción al CSS: qué es, para qué sirve y otras 10 preguntas frecuentes*. Obtenido de Lectura De 13 Min: [blog.hubspot.es](https://blog.hubspot.es/que-escss) [https://blog.hubspot.es >que-escss](https://blog.hubspot.es/que-escss) Introducción al CSS: qué es, para qué sirve y otras 10preguntas frecuentes

Srivastava, A., Bhardwaj, S., & Saraswat, S. (2017). *SCRUM model for agile methodology 2017 International Conference on Computing, Communication and Automation (ICCCA)*.

Suárez Alfonso, A., Cruz Rodríguez, I., & Pérez Macias, Y. (2015). La gestión de la información: Herramienta esencial para el desarrollo de habilidades en la comunidad estudiantil universitaria. *Universidad y Sociedad*.

Taran, A., & Silnov, D. S. (2017). *Research of attacks on MySQL servers using HoneyPot technology*. s.l: s.n.

Tedeschi. (2013). Obtenido de <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controladormvc>.

Urgellés, L. (2012). *Intranet Corporativa EMPLENI. Módulo Gestión de Capacitación*. Edgar Núñez Torres (Tutor). *Tesis de diploma*. Instituto Superior Minero Metalúrgico de Moa.

Wall, D. (2020). *Multi-Tier Application Programming with PHP*. s.l: s.n.

Wilson, O. (2018). *PATRONES ARQUITECTURALES PARA APLICACIONES WEB*. Obtenido de <http://eva.uci.cu/mod/resource/view.php?id=8978>

Zanotti. (2016). Obtenido de <https://www.gnu.org/philosophy/free-sw.es.html>



## Anexos

**Anexo 1** Tabla de ponderaciones para EI, EQ, EO, ILF, EIF. **Fuente:** Elaboración Propia.

<b>Clasificación de entradas y consultas</b>	<b>1 – 4 Atributos</b>	<b>5 – 15 Atributos</b>	<b>Más de 15 Atributos</b>
0 o 1 ficheros accedidos	BAJA 3	BAJA 3	MEDIA 4
2 ficheros accedidos	BAJA 3	MEDIA 4	ALTA 6
Más de 2 ficheros accedidos	MEDIA 4	ALTA 6	ALTA 6

<b>Clasificación de entradas y consultas</b>	<b>1 – 5 Atributos</b>	<b>5 – 19 Atributos</b>	<b>Más de 19 Atributos</b>
0 o 1 ficheros accedidos	BAJA 4	BAJA 4	MEDIA 5
2 a 3 ficheros accedidos	BAJA 4	MEDIA 5	ALTA 7
Más de 5 Entidades o registros lógicos	MEDIA 5	ALTA 7	ALTA 7

<b>Clasificación de entradas y consultas</b>	<b>1 – 19 Atributos</b>	<b>20 – 50 Atributos</b>	<b>Más de 50 Atributos</b>
1 entidad o registro lógico	BAJA 7	BAJA 7	MEDIA 10
2 - 5 Entidades o registros lógicos	BAJA 7	MEDIA 10	ALTA 15
Más de 2 ficheros accedidos	MEDIA 10	ALTA 15	ALTA 15

<b>Clasificación de entradas y consultas</b>	<b>1 – 4 Atributos</b>	<b>5 – 15 Atributos</b>	<b>Más de 15 Atributos</b>
0 o 1 ficheros accedidos	BAJA 5	BAJA 5	MEDIA 7
2 ficheros accedidos	BAJA 5	MEDIA 7	ALTA 10
Más de 2 ficheros accedidos	MEDIA 7	ALTA 10	ALTA 10

**Anexo 2** Tabla para obtener las Líneas por Puntos de Función. Fuente: Elaboración Propia.

Entorno y Lenguaje	Líneas de Código por PF	Horas por PF
Lenguajes 2GL: Ensamblador, C, ...	300	20 a 30
Lenguajes 3GL: Cobol	100	10 a 20
Lenguaje 4GL: Visual XX	20	5 a 10

**Anexo 3** Vista de la página de administración del sistema.



## Anexo 4 Interfaz del Sitio Web.

