**UNIVERSIDAD NACIONAL MICAELA BASTIDAS DE APURIMAC**

**Facultad de Ingeniería**

**ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA INFORMÁTICA Y SISTEMAS**

**“INFORME DE PRÁCTICAS PRE PROFESIONALES”**

* Desarrollo de una Aplicación web para la gestión eficiente de quejas en la Defensoría Universitaria
* Desarrollo de un Módulo de Atención y Formulario en Línea para Denuncias en la Página Web de la Defensoría Universitaria de la UNAMBA

**Presentado por:**

**KAREN SHEILA CASTAÑEDA PARI**

Informe de prácticas pre profesionales realizado en la:

**Dirección de Defensoría Universitaria-UNAMBA**

**Abancay – Perú, febrero del 2024**

**UNIVERSIDAD NACIONAL MICAELA BASTIDAS DE APURÍMAC FACULTAD DE INGENIERÍA**

**ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA INFORMÁTICA Y SISTEMAS**



**“INFORME DE PRÁCTICAS PRE PROFESIONALES”**

**REALIZADO EN:**

**DEFENSORÍA UNIVERSITARIA – UNAMBA**

El jurado calificador hace constancia del presente informe de prácticas pre- profesionales, presentado por el estudiante Karen Sheila Castañeda Pari.

**Primer Miembro: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Dr. ffff**

**Segundo Miembro: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Ing. fff**

**AGRADECIMIENTO**

Agradecer a la Universidad Nacional Micaela Bastidas de Apurímac por ser mi Alma Máter de darme la oportunidad de pertenecer a esta casa de estudios y de apostar en mi formación profesional y del servicio a la comunidad.

A la Escuela Académico Profesional de Ingeniería Informática y Sistemas por brindarme todos los conocimientos y experiencias a través de los docentes y maestros, la cual estaré eternamente agradecido a cada uno de ellos.

Al defensor universitario MVZ. Víctor Raúl Cano Fuentes y a la Lic. Charo Mendoza Portillo. Agradecer a ellos de ser parte de mi formación profesional durante mis prácticas pre profesionales y por inculcarme conocimientos y brindarme su amistad.

**Dedicatoria**

A Dios por darme la oportunidad en esta vida para poder aprender de ello. A mis padres, mis hermanos y familiares, que me brindan su amor y su apoyo incondicional. Como así también a todos mis amigos y amigas, que me enseñaron el valor de la amistad, la cual siempre estaré agradecido con cada uno de ellos. Gracias por todo.

**Índice**

**INDICE DE TABLAS**

[**Tabla 1** Infraestructura tecnológica (Hardware) 13](#_Toc157619437)

[**Tabla 2** Infraestructura tecnológica (Software) 13](#_Toc157619438)

[**Tabla 3** Herramientas Utilizadas 24](#_Toc157619439)

**INDICE DE FIGURAS**

[**Figura 1** Frontis de la UNAMBA 8](#_Toc152106454)

[**Figura 2** Plano de ubicación de la UNAMBA 11](#_Toc152106455)

[**Figura 3** Organigrama de la UNAMBA 11](#_Toc152106456)

1. **PRESENTACION**
   1. **Objetivo del Informe**

El presente informe tiene por objetivo dar a conocer y evidenciar las actividades realizadas durante la etapa de Practicas Preprofesionales en la Oficina de Defensoría Universitaria en la UNAMBA. El informe busca destacar mi aporte al progreso y la optimización de los procedimientos internos de la Oficina de Defensoría Universitaria.

* 1. **Periodo de Prácticas**
* **Fecha de Inicio:** 05 de octubre del 2023.
* **Fecha de Culminación:** 19 de febrero del 2023.
* **Horario de prácticas:** Lunes a viernes de 8:00 am a 1:00 pm.
* **Total, de horas:** 480
* **Cumplimiento de horas:**

Hasta la fecha de culminación, se han cumplido satisfactoriamente las 480 horas requeridas, según lo establecido en el Reglamento de Prácticas Pre-Profesionales de la Escuela Profesional de Ingeniería Informática y Sistemas.

* 1. **Institución y área** 
     1. **Institución**

Las prácticas preprofesionales se realizaron en la Universidad Nacional Micaela Bastidas de Apurímac.

**Figura 1**  
Frontis de la UNAMBA

****

*Nota*. Fuente:Universidad Nacional Micaela Bastidas de Apurímac. (2023) (https://www.unamba.edu.pe/).

* + 1. **Área**

Oficina de Defensoría Universitaria

* 1. **Funciones del área**

La Defensoría Universitaria de la UNAMBA, es un órgano autónomo en el ejercicio de sus funciones e independiente de los órganos de gobierno de la Universidad, está encargada de:

* Velar por el respeto de los derechos de docentes, estudiantes y trabajadores administrativos de la Comunidad Universitaria, frente a actos u omisiones de las autoridades o funcionarios de la Universidad que los vulneren.
* Es competente para Conocer denuncias y reclamaciones que formulen los miembros de la comunidad universitaria vinculadas con la infracción de sus derechos individuales.
* Proponer normas, políticas y acciones que permitan mejorar la defensa de los derechos de las personas en diferentes servicios que la Universidad brinda.
* Los pronunciamientos, las recomendaciones y las propuestas de la Defensoría Universitaria no tienen carácter vinculante; por consiguiente, no pueden modificar acuerdos o resoluciones emanadas de los órganos de gobierno de la Universidad, que corresponden a un procedimiento de la Ley y normas conexas.
* Emite recomendaciones y sugerencias a los responsables de los órganos de gobierno y demás instancias administrativas para prevenir vulneraciones, reservándose el derecho de trasladar las infracciones a las instancias correspondientes.
* Orienta acerca de procedimientos y vías correspondientes para canalizar quejas, denuncias o reclamaciones que no sean de su competencia.
* Solicita medios probatorios de denuncias interpuestas a autoridades, docentes, estudiantes o personal administrativo.

1. **ASPECTOS GENERALES DE LA EMPRESA**
   1. **Razón Social**

**Nombre de la institución:** Universidad Nacional Micaela Bastidas de Apurímac

**RUC:** 20527056579

**Tipo de Institución**: Institución pública del estado.

1. **Misión**

Formar profesionales humanistas, científicos y tecnológicos logrando calidad en el estudiante, con valores, identidad cultural, responsabilidad social, liderazgo y comprometidos con el desarrollo sostenible.

1. **Visión**

Los peruanos acceden a una educación que les permite desarrollar su potencial y convertirse en ciudadanos que valoran su cultura, conocen sus derechos y responsabilidades, desarrollan sus talentos y participan dentro de la comunidad académica de manera innovadora, competitiva y comprometida en las dinámicas sociales, contribuyendo al desarrollo de sus comunidades y del país en su conjunto.

* 1. **Actividades que realiza**

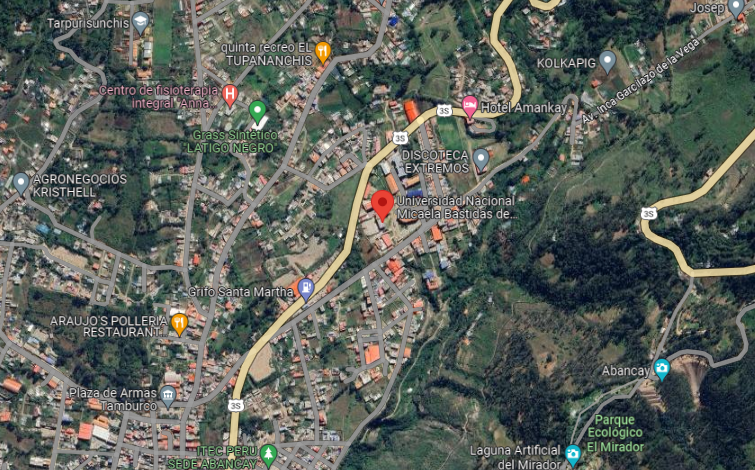
Las funciones de la Universidad Nacionalidad Micaela Bastidas de Apurímac son la docencia, la investigación, la extensión, la formación de recursos humanos, el desarrollo tecnológico, la innovación productiva y la promoción de la cultura.

* 1. **Aspectos técnicos:**
     1. **Ubicación Geográfica**

La Universidad Nacional Micaela Bastidas de Apurímac se encuentra ubicado en:

* **Departamento:** Apurímac
* **Provincia:** Abancay
* **Distrito:** Tamburco
* **Dirección legal:** Av. Inca Garcilazo de la Vega S/N Tamburco - Abancay – Apurímac.
* **Página web:** [www.unamba.edu.pe](http://www.unamba.edu.pe)
  + 1. **Plano de Ubicación**

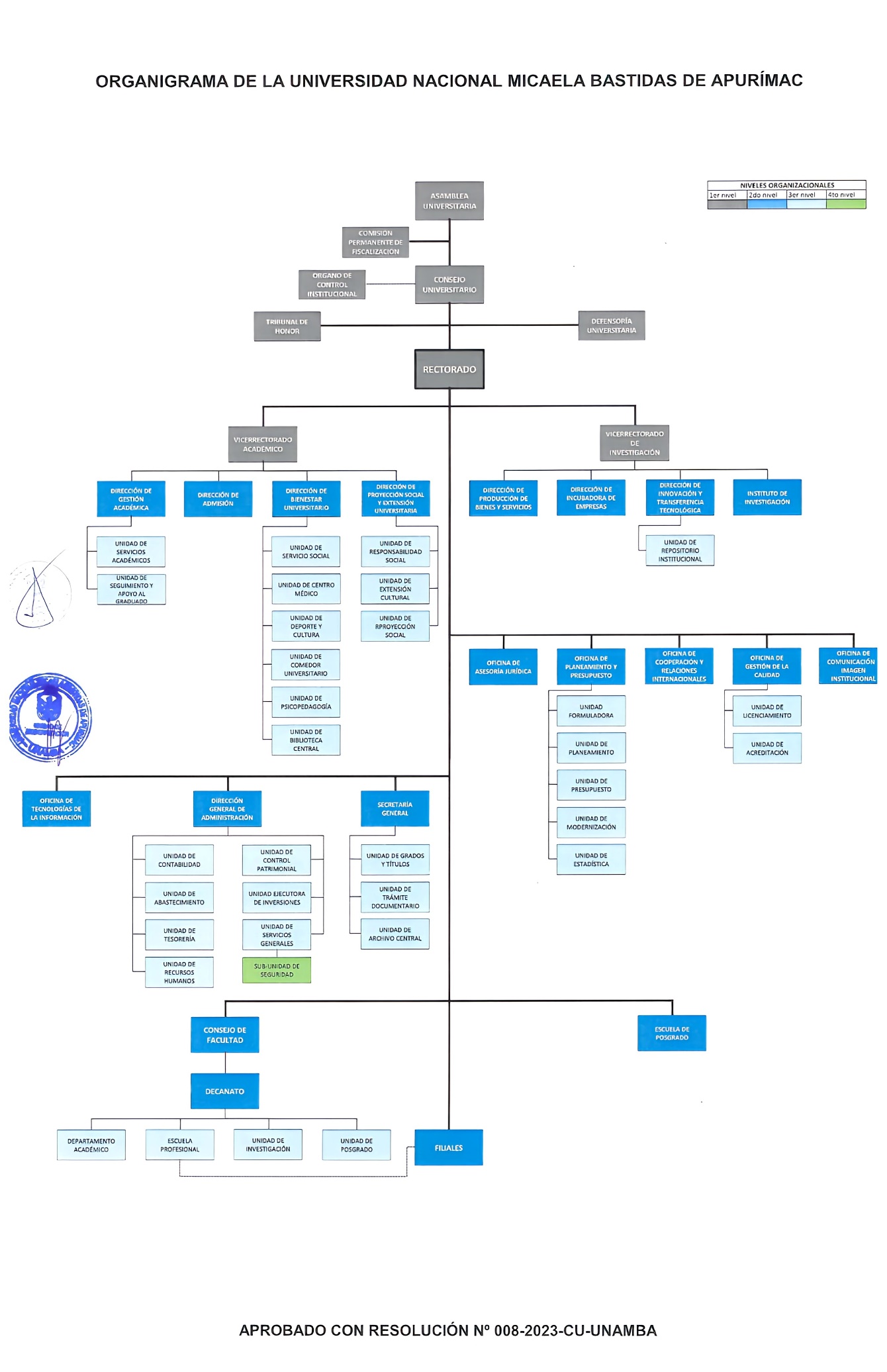
**Figura 2**  
Plano de ubicación de la UNAMBA



*Nota*. Extraído de Google Maps

* + 1. **Organización**

**Figura 3**  
Organigrama de la Universidad Nacional Micaela Bastidas de Apurímac

****

* + 1. **Infraestructura Tecnológica: Hardware y Software (equipos y accesorios)**
       1. **Hardware**

**Tabla 1**  
Infraestructura tecnológica (Hardware)

|  |  |  |
| --- | --- | --- |
| Cantidad | Hardware | Descripción |
| 2 | Computadora de escritorio | Memoria RAM 8GB.  Procesador: Intel ® Core tm i7 8 Gen  Procesador: x64  Almacenamiento: 1TB HDD |
| 1 | Impresora | Marca Kyocera Ecosys m2640idw  Tipo de impresión en blanco y negro (toner) |
| 1 | Impresora | Marca Epson L5190  Tipo de impresión a color y blanco y negro |

* + - 1. **Software**

**Tabla 2**  
Infraestructura tecnológica (Software)

|  |  |
| --- | --- |
| Nombre de Software | Descripción |
| **Sistema Operativo** | Windows 11 pro |
| **Programas** | Word, Power Point, Excel, Winrar, Google Chrome, Adobe Reader, SIGA (Sistema integrado de Gestión Administrativa) |
| **Antivirus** | ESET NOD32 Antivirus |

*Nota***.** Elaboración propia

1. **ACTIVIDADES REALIZADAS**

* Desarrollo de una Aplicación web para la gestión eficiente de quejas en la Defensoría Universitaria
* Desarrollo de un Módulo de Atención y Formulario en Línea para Denuncias en la Página Web de la Defensoría Universitaria de la UNAMBA

1. **DESCRIPCIÓN DE ACTIVIDADES REALIZADAS**
   1. **Actividad I**

DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN EFICIENTE DE QUEJAS EN LA DEFENSORÍA UNIVERSITARIA.

* + 1. **Planteamiento del problema**

La oficina de Defensoría Universitaria de la UNAMBA es encarga de la tutela de los derechos de los miembros de la comunidad. La defensoría atiende denuncias por infracción a derechos individuales y pueden acudir a ella los estudiantes, docentes, egresados y personal administrativo. Sin embargo, actualmente enfrenta ciertas dificultades que afectan su eficacia en la gestión de quejas, debido a que la defensoría universitaria lleva el control de las quejas de manera tradicional. Donde se usan archivadores físicos en el cual se tiene registrado cada queja. A cada queja presentada y aceptada se le asigna un número de expediente. Cada expediente incluye documentos como (formato de defensoría universitaria presentada por el denunciante, carta solicitando descargo al quejado, descargo del quejado y entre otros documentos referentes a la queja).

En tal sentido se percibe que al llevarse todo de forma manual, existen ciertas deficiencias a la hora de brindar información sobre el estado y detalles sobre una queja en específico. Estas dificultade se expresan a continuación:

* Retrasos en la entrega de documentos solicitados: Debido al proceso manual de búsqueda y recuperación de documentos, los interesados experimentan demoras significativas para obtener la información necesaria sobre el estado y los detalles de una queja en particular.
* Errores de duplicidad de expedientes: La gestión manual de los casos aumenta el riesgo de errores humanos, como la creación inadvertida de expedientes duplicados. Estos errores causan confusión y retrasos en el proceso de resolución de quejas.
* Riesgo de pérdida o deterioro de documentos: La manipulación constante de documentos físicos aumenta el riesgo de pérdida, daño o deterioro de la información importante. Esto puede tener consecuencias graves, como la incapacidad de proporcionar evidencia crucial en la resolución de las quejas.
  + 1. **Objetivos**
       1. **Objetivo General**

Desarrollar una aplicación web que permita gestionar de manera eficiente las quejas recibidas por la Defensoría Universitaria de la UNAMBA, con el fin de mejorar la atención y resolución de quejas.

* + - 1. **Objetivos Específicos**
* Implementar un sistema de control de expedientes que facilite el seguimiento y la organización de las quejas.
* Minimizar los errores de duplicidad de expedientes mediante un sistema de gestión automatizado.
* Reducir el tiempo de búsqueda de documentos solicitados de una queja en particular.
* Garantizar la integridad y seguridad de los documentos digitales para reducir el riesgo de pérdida o deterioro.
  + 1. **Justificación**

La implementación de una aplicación web para la gestión de quejas en la Defensoría Universitaria de la UNAMBA responde a la necesidad de modernizar sus procesos y mejorar la calidad del servicio ofrecido a la comunidad universitaria. Esta solución permitirá superar las deficiencias asociadas con el método tradicional de gestión de quejas, agilizando los tiempos de respuesta, reduciendo errores humanos y garantizando la integridad de la información. Además, proporcionará una plataforma más accesible y transparente para los usuarios, mejorando así la experiencia general de atención y fortaleciendo la confianza en la institución.

* + 1. **Marco Teórico**

**Aplicación web**

Las aplicaciones web, también conocidas como webapps, son programas alojados en un servidor web que se pueden acceder a través de internet o intranet mediante un navegador. Estas aplicaciones están evolucionando hacia entornos de cómputo avanzados, integrándose no solo con características aisladas, funciones y contenido para el usuario final, sino también con bases de datos corporativas y aplicaciones empresariales (Pressman, 2010).

**Base de Datos**

Una base de datos es esencialmente una colección de datos almacenados y organizados, diseñada para que un programa de computadora pueda recuperarlos de manera eficiente y realizar operaciones como actualizaciones, inserciones y eliminaciones. Representa aspectos relevantes del mundo real que interesan al usuario y puede variar en su estructura de organización. En el contexto de un Sistema de Gestión de Bases de Datos (DBMS), la base de datos se concibe como un sistema de archivos electrónico, integrando datos de diferentes modos de organización y capturando hechos conocidos, como números telefónicos, direcciones y nombres (Gutiérrez Díaz, n.d.)

**MySQL**

MySQL es un sistema de gestión de bases de datos relacionales de código abierto que almacena datos en tablas con filas y columnas. A través del lenguaje de consulta estructurada (SQL), los usuarios pueden definir, manipular y consultar datos. Su naturaleza de código abierto permite la modificación del código fuente sin costos adicionales, y ha evolucionado con la colaboración de usuarios durante más de 25 años. MySQL pertenece a la categoría de sistemas de gestión de bases de datos relacionales (RDBMS), organizando datos en relaciones predefinidas mediante tablas, y ha dado lugar a variantes como MariaDB y Percona Server for MySQL. Su disponibilidad bajo la Licencia Pública General de GNU permite su uso gratuito, pero también se ofrece con otras licencias para aplicaciones comerciales (*MySQL | Google Cloud*, 2023).

**XAMPP**

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 (Jesús, 2022).

XAMPP es un servidor web multiplataforma gratuito y de código abierto. XAMPP es una forma abreviada de multiplataforma, Apache, MySQL, PHP y Perl. XAMPP es un popular servidor web multiplataforma que permite a los programadores escribir y probar su código en un servidor web local. Fue creado por Apache Friends y el público puede revisar o modificar su código fuente nativo. Incluye MariaDB, Apache HTTP Server e intérpretes para PHP y Perl, entre otros lenguajes informáticos. Debido a la simplicidad de implementación de XAMPP, un desarrollador puede instalar rápida y fácilmente una pila WAMP o LAMP en un sistema operativo, con el beneficio adicional de que también se pueden cargar aplicaciones complementarias comunes como WordPress y Joomla (Educba, 2021).

**Servidor**

(Sierra, 2008) define un servidor, como un ordenador o máquina informática que está al “servicio” de otras máquinas, ordenadores o personas llamadas clientes y que le suministran a estos, todo tipo de información.

Según (Sierra, 2008) existen los siguientes tipos de servidores:

**Servidor de correo**, es el servidor que almacena, envía, recibe y realiza todas las operaciones relacionadas con el e-mail de sus clientes.

**Servidor proxy**, es el servidor que actúa de intermediario de forma que el servidor que recibe una petición no conoce quién es el cliente que verdaderamente está detrás de esa petición.

**Servidor web,** un servidor web es un ordenador que almacena, procesa y entrega archivos de sitios web a los usuarios desde un navegador en términos de hardware un servidor, se conecta a Internet para intercambiar datos con otros dispositivos. Estos datos, como archivos HTML, imágenes, JavaScript y CSS, son gestionados y almacenados en el hardware del servidor web. Además, el software del servidor web controla el acceso de los usuarios a los archivos alojados. (Betania, 2022).

Servidor de base de datos:

**Servidores clúster**, Son servidores especializados en el almacenamiento de la información teniendo grandes capacidades de almacenamiento y permitiendo evitar la pérdida de la información por problemas en otros servidores.

**Servidores dedicados**, son exclusivos para una persona o una empresa.

**Servidores de imágenes**,permiten alojar fotos sin usar recursos del servidor web principal.

**Framework**

El Framework es una especie de plantilla, un esquema conceptual, que simplifica la elaboración de una tarea, ya que solo es necesario complementarlo de acuerdo a lo que se quiere realizar (Muente, 2020).

En el ámbito del desarrollo de software, un framework se define como una estructura de soporte que facilita la organización y desarrollo de otros proyectos de software. En programación, se trata de un conjunto de funciones o código genérico diseñado para realizar tareas comunes en diversas aplicaciones, como la creación de objetos o la conexión a bases de datos. Este enfoque proporciona una base sólida para la construcción de aplicaciones específicas, permitiendo a los diseñadores y programadores dedicar más tiempo a la identificación de requisitos de software al evitar la implementación de componentes más triviales y genéricos (Biancha Gutiérrez et al., 2010).

**Laravel**

Laravel es un marco web que se utiliza principalmente para crear aplicaciones web personalizadas mediante el uso de PHP. Este marco web maneja muchas tareas que pueden resultar tediosas de crear por uno mismo, como el enrutamiento, la creación de plantillas HTML y la autenticación. Laravel se ejecuta completamente del lado del servidor, ya que se basa en PHP, y se enfoca en gran medida en la manipulación de datos y en seguir un patrón de diseño Modelo-Vista-Controlador (Heddings, 2020).

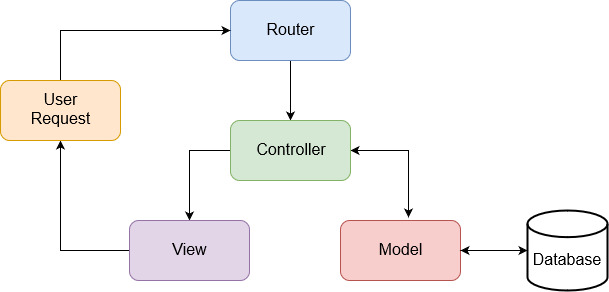
Según (Altube, 2021), las características principales de laravel son:

* Su motor de plantillas llamado **Blade** ofrece numerosas posibilidades para crear páginas visualmente muy potentes y efectivas, capaces de utilizar tus propias variables y reutilizarlas.
* Su arquitectura es conocida como **MVC (Modelo-Vista-Controlador)** que da muchas facilidades para relacionar de manera clara y sencilla todas las partes de una aplicación.
* **Eloquent** es un **ORM** (Mapeo Objeto-Relacional) incluido en el framework Laravel, que facilita la interacción con la base de datos a través de objetos en lugar de consultas SQL directas.
* En **seguridad**, ofrece un nivel bastante sólido de mecanismos hash y salt para cifrar utilizando bibliotecas como BCrypt.
* **Artisan**, es el sistema de comandos de laravel y brinda a los desarrolladores la facilidad de controladores, entidades y actualizar la base de datos entre otras funcionalidades.
* **Librerías y modularidad**, laravel se beneficia de la colaboración de Symfony en varios aspectos, como modularidad y comunidad.
* **Base de datos y migraciones**, actualizar y migrar la base de datos sin necesidad de borrarla y volverla a crear, gracias a su sistema de migraciones. Schema Builder hace que no se requiera usar SQL, lo que lo hace más fácil e intuitivo para los desarrolladores

Según (Vadillo Romero, 2020), es importante conocer bien los actores principales del ecosistema Laravel:

* **Router:** recibe todas las peticiones y las envía al controlador adecuado (también puede ejecutar algún middleware específico antes de llamar al controlador).
* **Controladores (Controllers):** contienen toda la lógica para reaccionar a las peticiones entrantes.
* **Vistas (Views):** contienen el código HTML y separan la presentación de la lógica de la aplicación (controlador).
* **Modelos (Models):** se utilizan para interactuar con la base de datos y aplicar la lógica de negocio.

**Figura 4**Diagrama de Laravel

****

* + - 1. **Control de versiones**

El control de versiones es un sistema que almacena y organiza los cambios realizados a uno o varios archivos para permitir acceder a distintas versiones de los mismos posteriormente (Scillone, 2016).

(Scillone, 2016) considera las siguientes ventajas:

* Permite revertir archivos o el proyecto entero a versiones anteriores.
* Permite comparar versiones, ver cambios. Permite
* Permite trabajar de forma colectiva en grupos de desarrollo.
* Permite recuperar los archivos en caso de pérdida o de algún error catastrófico
  + - 1. **Git**

Según (Martinez Sanchez) Git es un sistema de control de versiones distribuido que se originó en el desarrollo del kernel de Linux y es ampliamente utilizado en proyectos Open Source como Android y Eclipse, así como en proyectos comerciales. La principal diferencia de Git con otros sistemas de control de versiones radica en cómo modela sus datos. Mientras que la mayoría de los sistemas almacenan la información como una lista de cambios en los archivos a lo largo del tiempo, Git la modela como un conjunto de instantáneas de un mini sistema de archivos. Cada vez que se confirma un cambio o se guarda el estado del proyecto en Git, se toma una foto del estado de todos los archivos en ese momento, y se guarda una referencia a esa instantánea. Git es eficiente al no almacenar nuevamente los archivos que no han sido modificados, sino solo un enlace al archivo anterior idéntico que ya tiene almacenado.

* + - 1. **GitHub**

(Chacon & Straub, 2021)describe que GitHub destaca como el principal proveedor de alojamiento para repositorios Git, sirviendo como el punto central donde millones de desarrolladores colaboran en sus proyectos. Una gran proporción de repositorios Git se encuentra en GitHub, utilizado por muchos proyectos de código abierto para alojar su Git, gestionar seguimientos de fallos, llevar a cabo revisiones de código, entre otras funciones. Aunque GitHub no forme parte directa del proyecto de código abierto de Git, es altamente probable que, en el uso profesional de Git, se necesite interactuar con GitHub en algún momento.

* + - 1. **Node JS**

Es un entorno de tiempo de ejecución JavaScript de código abierto y multiplataforma, centrado en aplicaciones de red y servidor. Permite desarrollar aplicaciones rápidas y escalables con un código claro. Compatible con diversos sistemas operativos, fue concebido por Ryan Dahl en 2009 como un experimento que ganó impulso rápidamente entre ingenieros en busca de alto rendimiento sin las complejidades de lenguajes tradicionales (Jesus Lucas, 2019).

* + - 1. **Composer**

Composer es una herramienta que facilita la gestión de las librerías de terceros que se utilizan en los proyectos PHP. Esta herramienta, inspirada en npm y Bundler, permite declarar las librerías que se necesitan en un proyecto, así como sus versiones. Composer se encarga de descargar e instalar estas librerías, así como sus dependencias, de forma automática. Esto simplifica el proceso de instalación y mantenimiento de las librerías, y permite estar al día de las últimas versiones (Alvarado, 2014).

* + - 1. **NPM**

npm, parte esencial de Node.js, facilita el intercambio de paquetes entre desarrolladores y contribuye al éxito de Node. Con más de 1.4 millones de paquetes publicados y 27 mil millones de descargas, npm es el manejador de paquetes predeterminado para JavaScript. Se compone de un repositorio online para software libre y una herramienta de terminal que facilita la instalación, manejo de dependencias y publicación de paquetes en proyectos Node.js. Su utilidad radica en la capacidad de consumir paquetes desde el repositorio y gestionar dependencias mediante la línea de comandos, siendo fundamental para proyectos basados en Node (Hernandez, 2021).

* + - 1. **Metodología XP**

La metodología XP es un enfoque ágil para el desarrollo de software que se centra en la satisfacción del cliente, la comunicación, la simplicidad y la flexibilidad. Se basa en la idea de que el software se debe desarrollar en ciclos cortos e iterativos, con una estrecha colaboración entre el cliente y el equipo de desarrollo.

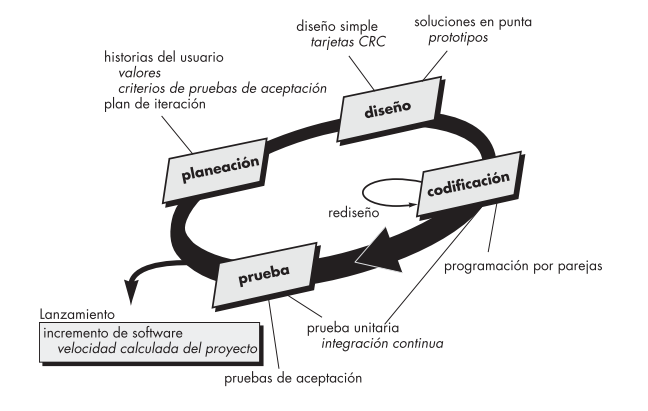
Algunos de los principios clave de XP son:

* **Comunicación constante**: El cliente y el equipo de desarrollo deben comunicarse de forma regular para garantizar que el software se está desarrollando de acuerdo con los requisitos.
* **Simplificación**: Las soluciones deben ser lo más simples posible, para facilitar su comprensión y mantenimiento.
* **Flexibilidad**: El software debe ser flexible para adaptarse a los cambios en los requisitos.

XP se utiliza a menudo en proyectos con requisitos cambiantes o con un alto riesgo técnico. Es una metodología eficaz para crear software de alta calidad que satisfaga las necesidades del cliente (Gimson, 2012).

1. **Fases de la Metodología XP**

**Figura 5**Fases de la metodología XP



*Nota*: La imagen representa las fases de la metodología XP, empezando por la planeación, el diseño, codificación, pruebas y finalmente el lanzamiento. Obtenido de (Pressman, 2010).

**Planeación:** La actividad de planeación, también conocida como juego de planeación, comienza escuchando la actividad para recopilar requerimientos, lo que permite a los miembros técnicos del equipo XP comprender el contexto comercial del software y adquirir la sensibilidad de la salida, así como las características principales y funcionalidades requeridas. El cliente puede agregar historias, cambiar el valor de una ya existente, descomponerlas o eliminarlas a medida que avanza el trabajo (Pressman, 2010).

Por lo tanto, el equipo XP revisa todas las entregas que faltan y modifica sus planes de acuerdo con eso.

**Diseño:** El diseño en XP sigue el principio de mantenerlo sencillo y se enfoca en la implementación de una historia, sin agregar funcionalidad adicional. Se recomienda el uso de tarjetas CRC para pensar en el software orientado a objetos. Si se encuentra un problema de diseño difícil, se crea un prototipo llamado solución en punta para mitigar el riesgo y validar las estimaciones originales. XP fomenta el rediseño como una forma de mejorar la estructura interna del código sin cambiar su comportamiento externo. El diseño en XP es flexible y se modifica continuamente durante la construcción. El rediseño se realiza de manera continua para mejorar el diseño a medida que se construye el sistema (Pressman, 2010).

**Codificación:** En la metodología XP, las pruebas unitarias se desarrollan antes de la codificación. Esto ayuda a los desarrolladores a centrarse en lo que es necesario para pasar la prueba, evitando la creación de código innecesario. La programación por parejas también es una práctica importante en XP, ya que permite la resolución de problemas en tiempo real y el aseguramiento de la calidad. Finalmente, la integración continua ayuda a evitar los problemas de compatibilidad e interfaces (Pressman, 2010).

**Prueba:** En la metodología XP, las pruebas son un elemento clave para garantizar la calidad del software. Las pruebas unitarias se crean antes de la codificación y se automatizan para facilitar su ejecución y repetición. Las pruebas de integración y validación del sistema se realizan a diario para detectar problemas de forma temprana. Las pruebas de aceptación, especificadas por el cliente, se centran en la funcionalidad general del sistema (Pressman, 2010).

1. **Roles en la Metodología XP**

* **Programador:** es el encargado de escribir las pruebas unitarias y producir el código del sistema. Es responsable de las decisiones técnicas y de construir el sistema. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.
* **Cliente:** es responsable de escribir historias de usuario, pruebas funcionales y asignar prioridades a las historias de usuario. También decide qué historias se implementan en cada iteración para aportar mayor valor al negocio. Puede representar a varias personas afectadas por el sistema.
* **Encargado de pruebas (Tester):** ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.
* **Encargado de seguimiento (Tracker):** proporciona retroalimentación al equipo sobre las estimaciones y el tiempo real dedicado, con el fin de mejorar futuras estimaciones. También se encarga de seguir el progreso de cada iteración, evaluar la alcanzabilidad de los objetivos y determinar si es necesario realizar cambios para lograr dichos objetivos.
* **Entrenador (Coach):** es responsable del proceso global, lidera el equipo, toma decisiones importantes y guía a los miembros del equipo para aplicar las prácticas XP y seguir el proceso correctamente. A medida que el equipo madura, tiende a estar en segundo plano.
* **Consultor:** es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.
* **Gestor (Big boss):** es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.
  + 1. **Metodología**

Para realizar la aplicación web se llegó a utilizar la metodología de desarrollo la programación extrema (XP) debido a que es muy eficiente durante el proceso de pruebas y planificación, facilita los cambios, origina una programación muy organizada y la satisfacción del programador, además de fomentar la comunicación entre el desarrollador y el cliente.

* + - 1. **Análisis de Requerimientos:**

**Tabla 3**Herramientas Utilizadas

|  |  |  |
| --- | --- | --- |
| **Herramientas de Software** | | |
| **Logo** | **Herramienta** | **Descripción** |
|  | Windows 10 | Sistema Operativo |
|  | Visual Studio Code | Editor de código |
|  | MySQL Workbench | Gestor de base de datos relacional |
|  | PHP | Lenguaje de Programación |
|  | Laravel | Framework de código abierto para PHP |
|  | LiveWire | Framework de Laravel |
|  | Alpine Js | Framework de JavaScript |
|  | Tailwindcss | Framework de css |
|  | XAMPP | Herramienta para levantar el servidor web local multiplataforma |
|  | Git | Sistema control de versiones |
|  | GitHub | Plataforma para alojar código y controlar sus versiones |

* + - 1. **Fase I: Planificación**

1. **Roles asignados en XP**

Los roles en XP asignados para el desarrollo del sistema se detallan en la siguiente tabla:

**Tabla 4**  
Asignación de roles

|  |  |  |
| --- | --- | --- |
| **Roles en XP** | **Responsable** | **Cargo** |
| Jefe del proyecto | Karen Sheila Castañeda Pari | Practicante |
| Programador | Karen Sheila Castañeda Pari | Practicante |
| Cliente | Defensoría Universitaria-UNAMBA |  |
| Tester | Karen Sheila Castañeda Pari | Practicante |

1. **Historia de Usuario**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Historias de usuario** | **Tareas** | **Horas** |
| 1 | Diseño de Base de Datos | * Definir el diseño conceptual de la base de datos * Definir la estructura de la base de datos * Diseño del modelo lógico | 20 |
| 2 | Implementación de la Base de datos |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |

1. **Detalles de Historia de Usuario**

Se muestra la información detallada de cada historia de usuario, tal y como fueron definidas:

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **ID: 1** | **Nombre de Historia:** | |
| **Usuario:** | | |
| **Prioridad en negocio:** | | **Tiempo estimado (Horas):** |
| **Riesgo en desarrollo:** | | **Iteración asignada:** |
| **Programador responsable:** | | |
| **Descripción:** | | |

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **ID: 1** | **Nombre de Historia:** | |
| **Usuario:** | | |
| **Prioridad en negocio:** | | **Tiempo estimado (Horas):** |
| **Riesgo en desarrollo:** | | **Iteración asignada:** |
| **Programador responsable:** | | |
| **Descripción:** | | |

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **ID: 1** | **Nombre de Historia:** | |
| **Usuario:** | | |
| **Prioridad en negocio:** | | **Tiempo estimado (Horas):** |
| **Riesgo en desarrollo:** | | **Iteración asignada:** |
| **Programador responsable:** | | |
| **Descripción:** | | |

1. **Plan de entrega**

A lo largo del proceso de desarrollo del sistema, se completaron cuatro iteraciones. Al inicio del proyecto, se llevó a cabo una reunión para implementar cada historia de usuario de acuerdo con el plan de entrega inicialmente acordado, priorizar cada historia de usuario y estimar el tiempo de finalización de cada iteración

* + - 1. **Fase II: Diseño**
      2. **Fase III: Codificación**
      3. **Fase IV: Pruebas**
    1. **Diagrama de Actividades**
    2. **Análisis de Requerimientos**
  1. **Actividad II**

DESARROLLO DE UN MÓDULO DE ATENCIÓN Y FORMULARIO EN LÍNEA PARA DENUNCIAS EN LA PÁGINA WEB DE LA DEFENSORÍA UNIVERSITARIA DE LA UNAMBA.

* + 1. Planteamiento del problema

El problema radica en la falta de un sistema en línea para presentar quejas y dar seguimiento a las mismas desde la página web de la Defensoría Universitaria (https://defensoria.unamba.edu.pe/). Esto obliga a los usuarios a acudir personalmente a la oficina para presentar una queja o para conocer el estado de la queja, además de tener que estar pendientes del correo electrónico para recibir notificaciones de la defensoría. La solución propuesta consiste en la creación de un módulo en línea en el sitio web de la Defensoría Universitaria, que permita a los usuarios completar el formato único de solicitud de queja de defensoría universitaria y enviar virtualmente sus quejas, además de poder adjuntar pruebas como videos, fotos, documentos, etc. y dar seguimiento al estado de sus quejas.

* + 1. Objetivos
       1. Objetivos Generales
       2. Objetivos específicos
    2. Justificación
    3. Marco Teórico
       1. Librería php Word
       2. Defensoría Universitaria
       3. Formato de solicitud de queja
    4. Metodología