



Universidad
Nacional
de Loja

Facultad de Energía, las Industrias y los Recursos Naturales no Renovables

CARRERA DE COMPUTACIÓN

Desarrollo de un módulo para la gestión de prácticas pre profesionales de la Universidad Nacional de Loja

Línea de investigación: Desarrollo de Software

PROPUESTA ANTEPROYECTO.

Autores:

- ◇ Brayan Marco Guamán Sarango , brayan.guaman@unl.edu.ec
- ◇ Daniel Andrés Novillo Tacuri, daniel.novillo@unl.edu.ec

Tutor:

- Hernán Torres, Mg. Sc.



Carrera de Ingeniería en
Sistemas / Computación

LOJA - ECUADOR
2022

Certificación de Tutoría

En calidad de Tutor del Proyecto de Trabajo de Integración Curricular TIC, certificamos la tutela a Brayan Marco Guamán Sarango y a Daniel Andrés Novillo Tacuri, con el tema **Desarrollo de un módulo para la gestión de prácticas pre profesionales de la Universidad Nacional de Loja** - , quienes han cumplido con todas las observaciones requeridas. Es todo cuanto puedo decir en honor a la verdad, facultando a los interesados hacer uso de la presente, así como el trámite de pertinencia del presente proyecto.

Loja, 16 de enero de 2023

Atentamente,
Hernán Torres, Mg. Sc.
TUTOR

1 Problemática

1.1 Situación Problemática

Las prácticas preprofesionales son un requisito de carácter obligatorio que un estudiante deberá cumplir a lo largo de toda la formación de la carrera, ya sea de forma continua o no continua, por lo que es indispensable que exista una agilización de procesos en la gestión de prácticas preprofesionales que no sólo beneficie al estudiante, sino a todos los involucrados en esta actividad.

De acuerdo con el reglamento de régimen académico de la Universidad Nacional de Loja [1], las prácticas preprofesionales están orientadas a la aplicación de conocimientos y al desarrollo de las capacidades profesionales de los estudiantes. Estas prácticas están vinculadas de forma directa con el beneficio de la sociedad.

Actualmente, en la Universidad Nacional de Loja, las prácticas preprofesionales no se encuentran centralizadas, y por lo tanto, se las realiza de forma independiente en cada carrera. Esto conlleva a que cada carrera maneje sus propios repositorios y procesos desligados de los demás y se termine con varios formatos diferentes.

En la Universidad no se tiene una base de datos centralizado con información de cartas de compromiso de prácticas preprofesionales que cada carrera legaliza en cada una de las facultades. Debido a esto, surge otro problema cuando la Dirección de Vinculación con la Sociedad (DVS) solicita informes finales que cada gestor de carrera debe otorgar y dirección tiene que constantemente solicitar a través de oficios, correos electrónicos o mensajes de texto, esto se convierte en información incompleta puesto que algunas carreras no otorgan esos informes.

Los docentes encargados de prácticas preprofesionales tienen que realizar una planificación para los estudiantes que vayan a realizar las mismas, a través de convenios o algún problema que surja en la Universidad que deba ser solucionado, al final de esta actividad, solicitan a cada estudiante a preparar su informe final con las actividades realizadas, registro de asistencia, firmas de los tutores de prácticas avalando el cumplimiento de las mismas y una evaluación de desempeño del estudiante que la realiza el tutor y el mismo estudiante. Otra problemática surge al momento de evaluar el desempeño y el registro de asistencias, en algunas localidades se las realiza de forma física, y el tutor debe realizar la evaluación del estudiante y luego volver a entregarle para que

el estudiante coloque en su informe final.

1.2 Problema de Investigación

La tecnología ofrece un abanico de posibilidades para mejorar y facilitar la vida de las personas, entre ellas permitirnos automatizar y agilizar procesos. la Universidad Nacional de Loja requiere de una gestión de prácticas preprofesionales automatizada, con el objetivo facilitar y agilizar procesos para todos los involucrados de las prácticas preprofesionales, para ello, se ha planteado la siguiente pregunta:

¿La automatización de la gestión de prácticas preprofesionales en la Universidad Nacional de Loja, mejorará la gestión de las mismas?

2 Justificación

El Reglamento de Régimen Académico de la Universidad Nacional de Loja en el artículo 338 menciona que: “la Coordinación de Vinculación con la Sociedad, elaborará el instructivo para prácticas preprofesionales, el cual contendrá entre otras cosas: el procedimiento a seguir para el desarrollo de las prácticas preprofesionales, los formatos para el desarrollo del programa de prácticas, el plan de actividades, los informes de prácticas preprofesionales, etc.; el mismo que aprobado por el Rector(a) será de aplicación en todas las carreras y programas vigentes en la Universidad Nacional de Loja.” [1]. Esto otorga la facultad a Dirección de Vinculación con la Sociedad (DVS) a buscar soluciones tecnológicas a problemas como los mencionados en la problemática del presente proyecto.

Dirección de Vinculación con la Sociedad posee la facultad de buscar una manera óptima de realizar sus funciones, se debe mencionar también que las instituciones de tercer nivel trabajan en tres funciones sustantivas que son: Docencia, Investigación y Vinculación con la sociedad que se ve reflejado en el Plan Estratégico de Desarrollo Institucional (PEDI) cuyo objetivo es direccionar el desarrollo institucional de la Universidad Nacional de Loja y mejorar el quehacer de las funciones sustantivas mencionadas anteriormente y con ello, cumplir las metas proyectadas en el PEDI para el 2019-2023 [2]. Con base en lo mencionado, este proyecto estaría reforzando el resultado esperado 3(R.1.3) del proyecto 1 (P.1) referente a la función sustantiva Vinculación con la Sociedad que se denomina: Fortalecimiento e Innovación de la función de Vinculación con la Sociedad. El R.1.3 se denomina: “Plataforma tecnológica de gestión, seguimiento y control de los proyectos y convenios de Vinculación con la Sociedad, desarrollada y en su primera fase de funcionamiento.” [2].

El módulo de gestión de prácticas preprofesionales (MGPP) deberá cumplir con la misma línea de acción del Reglamento de Régimen Académico de la Universidad Nacional de Loja[1] que tiene como objetivo especificar las bases fundamentales respecto al proceso de prácticas preprofesionales en toda la Universidad, y, en conjunto con Dirección de Vinculación con la Sociedad, se deberán gestionar los procesos y actividades que difieren entre carreras como son: número de horas laborales por cumplir y créditos.

Desarrollar este módulo sería un gran beneficio para todos los elementos que conforman la actividad de prácticas preprofesionales como: Estudiantes, docentes encargados y Dirección de Vinculación con la Sociedad (DVS). El beneficio se representaría en

virtud a la facilidad y agilidad de los procesos involucrados como: Cartas de compromiso, Desarrollo de PP, registro de actividades realizadas, informes finales de prácticas preprofesionales y Registro centralizado de prácticas preprofesionales. El desarrollo del módulo buscará agilizar los procesos anteriormente mencionados, permitiendo tener un seguimiento y control seguros y confiables.

Con la implementación del módulo, se podrá realizar una centralización de las actividades de prácticas preprofesionales, permitiendo a todas las carreras de la universidad seguir un orden, formato y pasos específicos para una planificación adecuada y un registro de la actividad que pueda ser accedida por todas las partes involucradas. Con esto, ya no habrán procesos engorrosos ni formatos independientes a cada carrera y facilitará el proceso de prácticas preprofesionales.

Tomando en cuenta los conocimientos adquiridos durante estos años de estudio: ingeniería de software, desarrollo web, base de datos y computabilidad avanzada, entre otras, se considera que académicamente en el presente proyecto se logrará plasmar la unificación de todos estos a la vez que incrementarán los mismos, lo cuál servirá para demostrar una mayor experiencia en el área y hacerse acreedor del título de Ingeniero en Computación.

3 Objetivos

3.1 General

Desarrollar un módulo para la gestión de prácticas preprofesionales que se integre al Sistema de Información Académico Administrativo Financiero (SIAAF) de la Universidad Nacional de Loja.

3.2 Específicos

- Realizar el análisis de requisitos para el desarrollo del módulo de prácticas preprofesionales.
- Desarrollar el módulo de gestión de prácticas preprofesionales basado en el análisis de requisitos.
- Evaluar la funcionalidad del módulo con los departamentos involucrados en el proceso de prácticas preprofesionales en un ambiente controlado.

4 Alcance

De forma preliminar a elaborar el módulo de gestión de prácticas pre profesionales, se deberá tener en consideración las siguientes actividades referentes al reglamento del Consejo de Educación Superior y el de la Universidad Nacional de Loja con el objetivo de obtener conocimiento legal de esta actividad:

1. Realizar el análisis y lectura del capítulo V del reglamento de Régimen Académico de Consejo de Educación Superior (CES)[3].
2. Realizar el análisis y lectura del capítulo III del reglamento de Régimen Académico de la Universidad Nacional de Loja (UNL)[1].

En cuanto al desarrollo del módulo, se han considerado las siguientes actividades:

1. Realizar el análisis de requisitos para el desarrollo del módulo de prácticas pre-profesionales.
 - Analizar el Reglamento de Régimen Académico de la Universidad Nacional de Loja.
 - Realizar entrevistas.
 - Realizar mesas de trabajo.
 - Especificar los requerimientos funcionales y no funcionales.
2. Desarrollar el módulo de gestión de prácticas preprofesionales basado en el análisis de requisitos.
 - Elaborar los casos de uso general.
 - Elaborar los diagramas de clases.
 - Elaborar el diseño arquitectónico del módulo de software.
 - Desarrollar el módulo de gestión de prácticas preprofesionales siguiendo la metodología XP.
3. Evaluar la funcionalidad del módulo con los departamentos involucrados en el proceso de prácticas preprofesionales en un entorno simulado.
 - Realizar pruebas unitarias.
 - Realizar pruebas de integración.
 - Realizar pruebas de aceptación.

- Realizar pruebas de funcionalidad en un entorno simulado.

Finalmente, al ser un módulo que será integrado al sistema de la Universidad Nacional de Loja, el mismo se deberá regir a las normas que implementa la Dirección de Tecnologías de la Información (DTI) de la Universidad, por lo que deberá realizarse con la metodología de desarrollo de software, métricas de calidad, lenguajes de programación y marcos de trabajo (frameworks) que esta área indique. Las pruebas de seguridad serán realizadas en base al nivel de acceso de usuario dentro de la aplicación.

5 Marco Teórico

Para el completo entendimiento de este anteproyecto, es necesario realizar un repaso sobre los conceptos de los términos que se usarán. Es necesario comprender de forma teórica conceptos como el ciclo de vida de software, la evaluación de calidad, entre otros. En esta sección se detalla a profundidad los términos importantes que serán implementados a lo largo del documento.

5.1 Reglamento de prácticas pre profesionales

Este anteproyecto tiene su base fundamental en las prácticas pre profesionales en la Universidad Nacional de Loja, por lo tanto, es necesario conocer los reglamentos que se rigen dentro de la Universidad a través del Reglamento de Régimen Académico de la Universidad Nacional de Loja (RRA) [1]. Tomando como base el documento mencionado anteriormente, se procede a recuperar artículos que se consideran fundamentales para tener un contexto claro sobre lo que representan las prácticas preprofesionales en la Universidad Nacional de Loja.

5.1.1 Responsabilidades

5.1.1.1 Coordinación con la vinculación de la sociedad

Artículo 331 (RRA) [1] . Conceptualización. “Las prácticas preprofesionales en las carreras de tercer nivel son actividades de aprendizaje, orientadas a la aplicación de conocimientos y/o al desarrollo de capacidades profesionales. Se realizarán en entornos organizacionales, institucionales, empresariales comunitarios u otros relacionados a los ámbitos profesionales de la carrera, públicos o privados, nacionales o internacionales.

Estarán orientadas por la Coordinación de Vinculación con la Sociedad quién entre otras, tendrá las siguientes responsabilidades:

- Orientar a los directores o docentes responsables de las prácticas preprofesionales de las carreras en la elaboración del plan, ejecución, seguimiento e informe de las prácticas preprofesionales.
- Coordinar la difusión y socialización de las actividades de prácticas preprofesionales a lo interno y externo de la institución.

- Promover el uso y seguimiento de los convenios y/o cartas compromiso interinstitucionales relacionados con las prácticas preprofesionales.”

5.1.1.2 Docente responsable de las prácticas preprofesionales

Art. 335 (RRA) [1]. Docente responsable de las prácticas preprofesionales.

“Cada carrera designará a un docente como responsable de las prácticas preprofesionales, que será el encargado de las siguientes actividades:

- Gestionar en conjunto con la Coordinación de Vinculación con la Sociedad la elaboración y suscripción de convenios y/o cartas compromiso para la realización de las prácticas preprofesionales y revisar las existentes, verificando que se encuentren vigentes.
- Socializar la normativa, convenios y/o cartas de compromiso, formatos, y demás informaciones referentes al procedimiento de prácticas preprofesionales con los tutores académicos y estudiantes.
- Gestionar la incorporación de los estudiantes en las instituciones y/o dependencias para la realización de las prácticas preprofesionales de acuerdo a los resultados de aprendizaje de la asignatura vinculada con la práctica. La incorporación de los estudiantes puede darse a pedido de la Universidad Nacional de Loja o de la institución o dependencia receptora.
- Elaborar el programa de prácticas preprofesionales en sus dos componentes. El programa será presentado al Director(a) de la carrera para su aprobación, al inicio de cada periodo académico que corresponda la práctica preprofesional.
- Realizar el informe de cumplimiento de la práctica preprofesional, en el que conste el número de identificación del estudiante, nombre de los estudiantes, nombre de la institución o dependencia donde se realizó la práctica preprofesional o del proyecto de vinculación para el caso de las prácticas de servicio comunitario, fecha de inicio y finalización, número de horas cumplidas, nombre y número de cédula o documento de identificación del tutor académico y la evaluación correspondiente; además se anexará los informes de las actividades cumplidas por los estudiantes, todo esto será entregado al Director(a) de la carrera para el registro y certificación correspondiente.”

5.1.1.3 Tutor académico de las prácticas preprofesionales

Art. 336 (RRA) [1] . Tutor académico de las prácticas preprofesionales.

“El tutor académico de las prácticas preprofesionales en sus dos componentes será el docente de la asignatura a la que está articulada la práctica o el personal de apoyo académico. Cuando la práctica preprofesional no está articulada a ninguna asignatura,

el tutor académico será el profesor o el personal de apoyo académico que designe el Director de carrera con formación afín al ámbito de la práctica y será el encargado de las siguientes actividades:

- Orientar la elaboración del plan de actividades que cumplirá el estudiante en la institución o dependencia receptora o en el proyecto de vinculación para el caso de las prácticas de servicio comunitario; el mismo que será legalizado por el tutor académico
- Guiar la ejecución, realizar el seguimiento, retroalimentación y evaluación del estudiante durante el desarrollo de la práctica preprofesional.
- Revisar y orientar la elaboración del informe de las actividades cumplidas por el estudiante en cada práctica preprofesional.”

5.1.1.4 Tutor institucional de las prácticas preprofesionales

Art. 337 (RRA) [1] . Tutor institucional de las prácticas preprofesionales. “Cuando las prácticas sean laborales, la institución receptora o dependencia en caso de ser dentro de la Universidad Nacional de Loja, nombrará un tutor quien será el responsable de:

- Monitorear el desarrollo de las actividades del estudiante de acuerdo al plan establecido, durante su permanencia en la institución o dependencia.
- Comunicar al tutor académico de prácticas preprofesionales, cualquier incumplimiento por parte del estudiante.
- Emitir un informe final sobre la ejecución de las prácticas preprofesionales, en donde conste: Información general, nombre del estudiante, actividades desarrolladas, fechas y cantidad de horas cumplidas.

En el caso de las prácticas de servicio comunitario no es necesaria la designación del tutor institucional.”

5.1.1.5 Ejecución de las prácticas preprofesionales

Art. 338 (RRA) [1] . De la ejecución de las prácticas preprofesionales. “Las prácticas preprofesionales pueden realizarse dentro o fuera de la Universidad Nacional de Loja en períodos académicos ordinarios y extraordinarios, siempre que sean de carácter formativo y supongan la aplicación o integración de conocimientos o capacidades profesionales desarrolladas a lo largo del proceso de enseñanza-aprendizaje.

La ejecución de las prácticas preprofesionales se realizará conforme lo establecido en el programa de prácticas de la carrera y el plan de actividades de los estudiantes. Si

por situaciones de fuerza mayor debidamente justificadas o incumplimiento de compromisos por parte de la entidad receptora o del estudiante, no se puede cumplir con la planificación establecida en los tiempos previstos, el estudiante podrá ser reubicado en otra entidad o completar las horas faltantes en el próximo período académico.

Durante el desarrollo de la práctica preprofesional se realizará el seguimiento respectivo por parte del tutor académico a las actividades estudiantiles y por parte del responsable de prácticas preprofesionales a las actividades establecidas en cada unidad de organización curricular.

Al finalizar cada práctica pre profesional de acuerdo a lo establecido en el programa curricular, el estudiante entregará al tutor académico el informe del cumplimiento de la práctica preprofesional.

El tutor académico entregará al responsable de prácticas preprofesionales el informe de los estudiantes que desarrollaron sus prácticas preprofesionales bajo su tutoría.

El responsable de las prácticas preprofesionales entregará al Director(a) de la carrera el informe consolidado de seguimiento al cumplimiento de la práctica preprofesional.

La secretaría de la carrera creará un portafolio por cada estudiante el mismo que contendrá: el programa de prácticas preprofesionales y los informes de seguimiento al cumplimiento de cada práctica preprofesional.

Las horas de prácticas de servicio comunitario serán realizadas en el marco de un proyecto de vinculación con la sociedad formalmente aprobado.

La Coordinación de Vinculación con la Sociedad, elaborará el instructivo para prácticas preprofesionales, el mismo que contendrá: el procedimiento a seguir para el desarrollo de las prácticas preprofesionales, los formatos para el desarrollo del programa de prácticas, el plan de actividades, los informes de prácticas preprofesionales, etc; el mismo que aprobado por el Rector(a) será de aplicación en todas las carreras y programas vigentes en la Universidad Nacional de Loja.”

5.1.2 Duración

Según el RRA [1], para las carreras de licenciatura, títulos profesionales y veterinaria en modalidad presencial ejecutarán 240 horas de prácticas preprofesionales laborales y 120 horas de prácticas preprofesionales de servicio comunitario. Para la modalidad a distancia se planificará 256 horas de prácticas laborales y 144 de prácticas de servicio comunitario[1].

También se menciona que la carrera de derecho deberá planificar 480 horas de prácticas laborales y 120 horas de prácticas de servicio comunitario. Las carreras de Odontología y Laboratorio Clínico ejecutarán 480 horas de prácticas preprofesionales laborales y 120 horas de servicio comunitario.

Finalmente se menciona también que, para las carreras de medicina y enfermería que cumplen sus prácticas preprofesionales a través del internado rotativo deberán planificar 832 horas y 780 horas por cada rotación respectivamente.

5.2 Ciclo de vida del desarrollo de software

El desarrollo de software resulta ser muy complejo y costoso si no se tiene en cuenta su ciclo de vida. El ciclo de vida de desarrollo de software es un proceso por el cual todo software debería pasar, este ciclo ayuda a establecer fases que se permita, de una forma ordenada, el desarrollo del software. Existen varios modelos y metodologías que tienen un distinto enfoque específico, sin embargo, tienen fases generales que siguen de una forma y otra[4], entre las fases generales se encuentran[5]:

1. **Análisis de requerimientos:** Se estudian las necesidades del cliente para llegar a una definición de requisitos del software, hardware o del sistema. Dependiendo de la metodología, existen algunas que definen al inicio del proyecto todos los requisitos y son inalterables, también existen otras que permiten evolucionar los requerimientos conforme el cliente lo sugiera.
 2. **Diseño:** Se define la arquitectura de software, los componentes, módulos, interfaces y la información de un sistema de software de satisfacer requerimientos específicos. Dependiendo del proyecto, se pueden implementar diferentes niveles y tipos de diseño como: diseño preliminar, funcional, arquitectónico, entre otros.
 3. **Implementación:** Se plasma el diseño a un producto software o hardware. Estos productos deben ser evaluados para verificar su correcto funcionamiento. Existen metodologías donde sólo se puede realizar la implementación una vez, sin embargo, en metodologías ágiles se pueden realizar iteraciones que permiten volver a rediseñar e implementar.
 4. **Evaluación:** Se evalúa la implementación acorde a entornos y condiciones específicas, los resultados de la evaluación deberán ser registradas y evaluadas de forma independiente y colectiva. El objetivo de esta fase es implementar todos los escenarios posibles a los que el software será enfrentado cuando se encuentre en fase de despliegue.
-

5. **Despliegue:** El software es puesto en operación y cualquier problema de transición de desarrollo a producción son solventados. En esta fase el software ya es funcional y ofrece los servicios por los cuales fue creado.
6. **Mantenimiento:** Cualquier problema que surja durante la producción del software deberá ser resuelto en la brevedad posible. Esta fase se encarga de otorgar servicios de mantenimiento al software como: revisión de base de datos, resolución de problemas, reparación de conexiones, entre otros.

5.2.1 Framework SCRUM

El término Scrum, no proviene de una sigla, se refiere a un concepto del deporte Rugby cuando ocurre una infracción y se deba recuperar una formación de manera rápida[6].

Scrum es un marco de trabajo que consiste en la colaboración eficaz del equipo de trabajo, se basa en un conjunto de reglas y se definen roles para generar una estructura con un funcionamiento correcto[7].

Esta metodología tiene tres roles:

- Scrum master: Lidera el equipo, controlando el proyecto, basándose en las reglas y procesos de la metodología.
- Dueño del producto (Product Owner): Representa a los accionistas y a los clientes que usan el software.
- Equipo de desarrollo: Son los encargados de convertir los requisitos(Backlog) en funcionalidades del software.

Una de las características más representativas de esta metodología es Sprint, una etapa de trabajo en la que se crea una versión utilizable del producto, se considera a cada sprint como un proyecto individual. El sprint está compuesto por:

- Reunión de planeación del sprint.
- Reunión diaria.
- Trabajo de desarrollo.
- Revisión y retrospectiva del Sprint.

5.3 Tecnologías de desarrollo web

El desarrollo web ha ido evolucionando constantemente desde su concepción, Existe una gran cantidad de tecnologías para desarrollar sistemas web con el objetivo de ofrecer mejores servicios a los usuarios, las cuales un desarrollador deberá elegir conforme a

las necesidades y requerimientos que se deben cumplir para satisfacer los objetivos del cliente. Existen tecnologías web tanto por el lado del cliente (Front-End) y por lado del servidor (Back-End). En esta sección se enmarcan las tecnologías tentativas a utilizar en el presente proyecto.

5.3.1 Lenguaje de maquetado HTML

HTML (HyperText Markup Language, Lenguaje de Marcado de Hipertexto) es un lenguaje de maquetado que se utiliza para el desarrollo web, se conforma de etiquetas que sirven para dar una estructura o formato a una página web[8], El hipertexto es un texto que tiene referencias como enlaces, vínculos, a otro texto. HTML surgió con la necesidad de procesar información de una forma mecánica entre ordenadores conectados en la red. HTML posee dos ventajas que son imprescindibles en el diseño web que son la compatibilidad y la facilidad de aprendizaje[9].

5.3.2 Lenguaje de maquetado CSS

Permite dar formato de estilo a lenguajes de maquetado como HTML expandiendo sus funciones directamente en la vista, si bien este último permite agregar ciertos atributos, esto son reducidos y a partir de HTML5 quedaron obsoletos. Las hojas de estilo en cascada además pueden ser almacenadas en la caché del navegador lo cual reduce el tráfico de la red[10].

5.3.3 Lenguaje de programación Javascript

Javascript es el lenguaje más utilizado principalmente en la construcción de páginas web, tiene una sintaxis muy similar a Java y a C, javascript es un lenguaje interpretado, algo que aporta gran valor al no necesitar compilación para ejecutar los programas creados. Al ser un lenguaje interpretado, estos programas pueden ser ejecutados directamente en un navegador web sin necesidad de procesos intermedios[11].

5.3.4 Lenguaje de programación Python

“Python es un lenguaje de programación interpretado e interactivo, capaz de ejecutarse en una gran cantidad de plataformas. Se desarrolla como un proyecto de código abierto, administrado por PYTHON Software Foundation” [12]

Este lenguaje de programación está especializado para ser implementado en el lado del servidor, ya que se encarga del comportamiento y la implementación de la página web en el servidor.

Python es un lenguaje de programación de alto nivel orientado a objetos, cuyas ventajas principales de python son: la facilidad de aprendizaje, estructuras de datos eficientes y sintaxis dinámica sencilla gracias a su filosofía de legibilidad de código [11].

5.3.5 Framework Django

“Es un framework web de código abierto escrito en Python que permite construir aplicaciones web más rápido y con menos código, fue inicialmente desarrollado para gestionar aplicaciones web de páginas orientadas a noticias de World Online, más tarde se liberó bajo licencia BSD.”[12].

Django es un marco de trabajo (Framework) gratuito para desarrollar aplicaciones web con el lenguaje de programación Python, este framework respeta el patrón de diseño Modelo-Vista-Controlador (MVC), el MVC es una arquitectura que se encarga de separar la lógica del código en una aplicación, donde la vista es la interfaz gráfica del usuario y ésta cambia conforme a un controlador que gestiona todo el contenido que se muestra en ella [12].

Django facilita el desarrollo de aplicaciones web gracias a algunas funcionalidades cuyo objetivo es evitar la escritura de todo el código y evitar pérdida de tiempo en búsqueda de errores de código en el framework [13].

5.3.6 Framework jQuery

Jquery es un framework de Javascript que aporta con librerías que facilitan la interacción con documentos html, manejo de eventos, animaciones, entre otras. Al tener funciones y herramientas ya implementadas, se asegura que la funcionalidad de las mismas sean exactamente igual en los distintos navegadores actuales, a la vez que ahorra tiempo de desarrollo al no tener que programar funciones básicas que ya existen en este framework[14].

5.3.7 Framework Vue.js

Pertenece a la familia de los Frameworks Javascript, llamado marco progresivo por sus creadores, esto debido a que su uso puede comenzar con un esfuerzo mínimo ya que su lógica central se basa en la capa de la vista, tiene el sentido de adoptabilidad incremental, es decir a medida que la aplicación crece se pueden agregar funcionalidades[15]. Además su uso puede darse como un decorador, agregando funcionalidades o como un marco completo de arquitectura[16].

5.3.8 Framework Angular

Angular es un framework elaborado y mantenido por Google para desarrollar aplicaciones web que es ejecutado con el lenguaje de programación Typescript, el cual es una versión avanzada de Javascript [17].

Uno de los objetivos principales de Angular como framework es fortalecer la estructura MVC y las aplicaciones de una sola página (Single Page application *SPA*). Las SPA son páginas que cargan únicamente un archivo HTML (HyperText Markup Language) en el cliente y que el contenido cambia dinámicamente conforme el usuario interactúa con la página [18].

5.4 Patrones de Diseño

5.4.1 Arquitectura Modelo-Vista-Controlador (MVC)

Es uno de los patrones de diseño más importantes y utilizados en el desarrollo de software, este tipo de patrón puede ser utilizado para pequeños módulos o sistemas completos, su definición no está limitada lo cual permite implementaciones alternativas. La clave para una implementación satisfactoria haciendo uso de este diseño es prestar atención a los objetos, sus roles y la comunicación entre ellos[19]. La intención de esta arquitectura es separar la parte visual de la parte lógica, permitiendo: reutilización de la aplicación o componentes visuales, capacidad de desarrollo separado (visual, lógico) y la herencia desde diferentes partes[20].

5.4.1.1 Modelo

.- Almacenan, encapsulan y abstraen datos. No debe contener funciones propias del sistema que lo hagan funcionar.

5.4.1.2 Vista

.- Abarca desde lo genérico hasta la parte específica de mostrar información en base a los modelos. La parte visual interactúa con el usuario iniciando acciones en base a eventos recibidos.

5.4.1.3 Controlador

.- Implementan las acciones de la aplicación, generalmente en base a los eventos de la vista generados por parte del usuario[19].

5.5 Especificación de Requisitos de Software (ERS)

En el ciclo de vida de desarrollo de software, el análisis de requisitos es, quizás, la sección más importante, a través de ella, el cliente o las partes interesadas dan a conocer lo que ellos requieran, las funcionalidades que deberá tener el software, o lo que consideran pertinente, en pocas palabras, se determinan los “planos” para el nuevo software[21].

Al ser una etapa crítica del proceso de creación del software, lo recomendable es basarse en normas que aseguren la correcta recolección e identificación de requisitos, por ejemplo, la norma estándar de especificación de requisitos IEEE 830, el resultado de esta etapa es la producción de un documento el cual se describe lo que el sistema podrá hacer y sus limitaciones.

El documento deberá ser socializado con las partes interesadas para comprobar que el contexto es el mismo y que se han comprendido de forma correcta las necesidades del software. Basándose en el documento generado y aprobado por el cliente, se procede a la representación de las funcionalidades que debe realizar el sistema y los datos que fluyen entre ellas.

Los objetivos de la especificación de requisitos de software son[21]:

1. Ayudar a los clientes a describir las funcionalidades del software, el cliente debe participar de forma activa en la especificación de requisitos.
2. Ayudar a los desarrolladores a comprender las necesidades del cliente, hay ocasiones donde el cliente no sabe con exactitud esas necesidades y a través de la ERS se permite al cliente definir todos los requisitos que se requiera, evitando costes elevados por cambios durante la creación de la aplicación.
3. Servir de base para el desarrollo de estándares de ERS particulares. Cada entidad puede desarrollar su propia especificación de requisitos.

5.5.1 IEEE 830

El estándar IEEE 830 consiste en la especificación de requerimientos, ofreciendo recomendaciones prácticas para la elaboración de un documento consistente gracias a la estructura y organización de toda la información que se debe incluir en un documento de especificación de requerimientos de software. Este estándar tiene como objetivos: Ayudar a los clientes a describir con precisión lo que se requiera del software, y ayudar a los encargados de recibir la información a establecer una estructura estándar para la ERS en sus organizaciones[22].

5.5.1.1 Estructura del documento IEEE 830

La estructura del documento resultante con base en la norma estándar IEEE 830-1998 es la siguiente[23]:

1. Introducción: Se propone de una introducción a todo el documento de ERS, contiene las subsecciones: propósito, ámbito del sistema, definiciones, referencias y visión general del documento.
2. Descripción General: Se describen los factores que afectan al producto y a los requisitos. No detallan los requisitos en específico pero sí el contexto en el que se encuentran, a través de ello, se logrará definir con detalle los requisitos en la siguiente sección, logrando un mejor entendimiento. Consta de las subsecciones: Perspectiva del producto, funciones del producto, características de los usuarios, restricciones, factores que se asumen y futuros requisitos.
3. Requisitos específicos: Contiene los requisitos detallados para permitir a los desarrolladores diseñar un sistema acorde con los mismos, y también permite que el equipo de pruebas pueda planificar y realizar las pruebas que demuestren si el sistema cumple con los requisitos.
4. Apéndices: Contiene todo tipo de información relevante para la ERS pero que no forma parte del documento, como: Formatos de entrada/salida de datos, resultados de análisis de costos, restricciones respecto al lenguaje de programación.

5.6 Evaluación de calidad

5.6.1 Calidad

“La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad.” García(1995, p.1)[24].

La calidad de software es medible y es diferente en cada proyecto, una característica de la calidad de software es que se puede evaluar luego de la culminación del producto, sin embargo, esto puede resultar muy costoso al no detectar problemas a tiempo que causen imperfecciones en el diseño, por lo que es recomendable tener en cuenta la calidad de software como su control durante el ciclo de vida de desarrollo del software[24].

5.6.2 ISO/IEC 25000

La ISO/IEC 25000 - Requisitos y Evaluación de calidad de Productos de Software (SQuaRE) es un conjunto o familia de normas que agrupa todas las normas de cali-

dad ISO/IEC relacionadas al producto de software, de esta forma se sustituyeron las existentes[25]. Estas normas proporcionan guías para el uso de nuevas series y estándares internacionales para orientar la evaluación de calidad de productos de software estableciendo criterios para la especificación de requisitos de calidad de software, considerando las métricas y evaluación.[26]

Las normas ISO/IEC 25000 se subdividen en las siguientes ramas[26]:

- ISO/IEC 2500n, Gestión de calidad: Estos estándares definen los modelos, términos y definiciones comunes. Comprende: ISO/IEC 25000: Guía de SQuaRE; ISO/IEC 25001: Planificación y gestión.
- ISO/IEC 2501n, Modelo de calidad: Detalla las características para la calidad interna, externa y en uso. Comprende: ISO/IEC 25010.
- ISO/IEC 2502n, Medición de calidad: Esta familia de normas incluyen un modelo de referencia de calidad del producto de software, definiciones matemáticas para las métricas de calidad y una guía práctica para la aplicación. Comprende: ISO/IEC 25020: Modelo de referencia para la medida con guía, ISO/IEC 25021: Primitivas, ISO/IEC 25022: Medidas de calidad en uso, ISO/IEC 25023: Medidas de calidad del producto software.
- ISO/IEC 2503n, Requisitos de calidad: Comprende ISO/IEC 25030, que se orienta en ayudar en la especificación de requisitos para un producto software que va a ser desarrollado, o para un proceso de evaluación.
- ISO/IEC 2504n, Evaluación de calidad: Este conjunto proporciona requisitos, recomendaciones y guías para la evaluación de un producto software, ya sea por evaluadores independientes, compradores o desarrolladores. Comprende: ISO/IEC 25040: Proceso de evaluación, ISO/IEC 25041: Guía para la evaluación para desarrolladores, compradores y evaluadores independientes.

5.7 Trabajos Relacionados

5.7.1 Diseño de un sistema de gestión para la coordinación de practicas pre profesionales de la Universidad Técnica Estatal de Quevedo

Este trabajo tiene como objetivo el automatizar del desarrollo de las prácticas pre-profesionales de la Universidad Técnica Estatal de Quevedo ya que como antecedentes tienen el desarrollo de las mismas de manera manual. Para el desarrollo de la investigación hicieron uso del método analítico y para su programación se basaron en el lenguaje de programación C .NET [?].

5.7.2 Sistema de Control de Actividades de Prácticas Pre-Profesionales para la Universidad Estatal Península de Santa Elena

Este trabajo tiene como objetivo el desarrollo de una plataforma para facilitar y gestionar las actividades de los estudiantes dentro de las empresas donde vayan a realizar sus prácticas. Para el desarrollo del proyecto se utilizaron herramientas web como: java, zk framework, bootstrap, maven, entre otras; y para el almacenamiento de información el motor de base de datos SQL Server que está instalado en el servidor de la Facultad de Sistemas y Telecomunicaciones de este establecimiento[?].

5.7.3 Aplicación web para la gestión de las prácticas pre profesionales de la Carrera de Sistemas de la UNIANDES

Esta aplicación aparece en base a la necesidad de automatizar el proceso de desarrollo de las prácticas preprofesionales, este trabajo inicia con un estudio de la fundamentación científica sobre la gestión de los procesos que se llevan a cabo dentro de las prácticas preprofesionales, utilizando como metodología de desarrollo XP, su arquitectura es una de 3 capas con equilibrio de carga[?].

5.7.4 Desarrollo de un sistema web para la gestión de prácticas preprofesionales y pasantías – ESFOT.

Parte de un problema inicial que es el de llevar el control de las horas que realiza cada estudiante de la ESFOT al realizar sus prácticas preprofesionales, al iniciar su estudio este proceso lo realizan de manera manual que mencionan como un proceso lento y que puede presentar errores o resultados incorrectos, para ello plantean un sistema web bajo el framework Laravel y como base de datos Postgress que pueda solventar estos problemas utilizando para su desarrollo la metodología Scrum[?].

5.7.5 Desarrollo e implementación de un sistema web de seguimiento y evaluación de las prácticas pre-profesionales para la Facultad de Ingeniería Escuela Civil de la PUCE

El problema nace por la necesidad de llevar un registro adecuado de las actividades realizadas por los estudiantes de una facultad, al inicio de este proyecto este proceso lo realizan de forma manual en archivos, o medios de almacenamiento lo cual de acuerdo a este estudio no garantiza la seguridad de la información, para ello se plantea un sistema web basado en el lenguaje de programación PHP y haciendo uso de la metodología en cascada [?].

6 Metodología

Para el presente proyecto se optará por emplear metodologías para cada sección del mismo, al ser un proyecto de desarrollo de software, se enfocará fuertemente en el método científico.

6.1 Método científico

Esta metodología se empleará a lo largo del proyecto, desde la observación de la problemática hasta los resultados finales, gracias a la obtención y análisis de procesos que se involucran en la gestión de prácticas preprofesionales.

6.2 Método deductivo

El ciclo de vida de desarrollo de software contempla la recolección de datos como una forma de adquirir los requisitos funcionales y no funcionales del sistema que será creado. Para la recolección de datos, se hará uso de la metodología deductiva, a través de entrevistas como un procedimiento científico, esto se hará con la ayuda de preguntas abiertas para permitir la creatividad de los entrevistados, que en este caso serán las partes interesadas del proyecto. Una vez se hayan identificado los requisitos acorde a las necesidades de las partes interesadas, se generarán historias de usuarios, para luego ser socializadas y comprobar que los requisitos reflejen las funcionalidades necesarias para satisfacer a los clientes.

6.3 Metodología de desarrollo

Este proyecto debe enfocarse en metodologías ágiles como Scrum, sin embargo, esta metodología es mucho más eficaz con un equipo de desarrollo, que en este proyecto no se cuenta, por esos motivos, se ha optado por emplear la metodología Extreme Programming (XP). Con esta metodología se puede realizar un seguimiento eficaz acorde con los objetivos que se planteen en cada iteración, a través de una planificación, historias de usuario, prototipos, entre otras. Esta metodología de desarrollo, al ser una metodología ágil, se podrá reaccionar de forma eficaz ante cambios de enfoque de la aplicación, lo que otorga más flexibilidad y minimiza el riesgo que podría afectar al tiempo de desarrollo.

6.3.1 Tipos de metodologías de desarrollo de software

Antes de hablar de metodologías ágiles, es necesario mencionar a las metodologías de desarrollo tradicionales, para tener el contexto de la creación de las primeras; las metodologías tradicionales fueron las primeras en surgir y por ello es que las características son aplicables en la mayoría de proyectos, sin embargo, no tendría la eficacia que aplicar una metodología específica para el tipo de proyecto.

6.3.1.1 Metodologías tradicionales

La característica principal de las metodologías tradicionales radica en su rigurosidad respecto a los pasos a seguir, ya que éstas son orientadas por planeación. Por lo general, se inicia el desarrollo de un proyecto con un proceso de captura de requisitos, siguiendo las etapas de análisis y diseño. [6]

Al ser metodologías estrictas, se sigue un proceso secuencial de una dirección, por lo que resulta complejo poder volver pasos atrás, esto supone una desventaja ante las metodologías ágiles, por el simple hecho que no se puede reaccionar de forma rápida ante algún incidente. Otra desventaja resulta en los requerimientos, éstos sólo se acuerdan una vez y para todo el proyecto, algo que afecta en la comunicación con el cliente[6].

6.3.1.2 Metodologías ágiles

De forma opuesta a las tradicionales, las metodologías ágiles se destacan por su flexibilidad, ya que pueden ser modificadas acorde a cada proyecto. Al poder otorgar una flexibilidad, se tiene una mayor comunicación con el cliente, lo que permite perfeccionar el software con requisitos que solicite el software a mitad del desarrollo, permitiendo reaccionar ante ellos de una forma eficiente[6]. Este tipo de metodología suele ser más efectiva con equipos de trabajo pequeños que resuelven problemas concretos, a diferencia del desarrollo de grandes sistemas cuyo factor de éxito es la modularización del mismo[27].

6.3.2 Xtreme Programming (XP)

Otra metodología ágil, es la XP, se centra en trabajo en conjunto, de esta forma se potencian la relación del equipo de trabajo, el aprendizaje de los desarrolladores y adquiriendo un buen entorno de trabajo. Esta metodología es adecuada para proyectos con clientes indecisos cuyos requisitos no son claros y pueden cambiar en la mitad del desarrollo, lo que causa un gran riesgo técnico[28].

La metodología XP según la propuesta del creador contiene los siguientes roles[28]:

- Programador: Produce el código del sistema y las pruebas unitarias.
-

- Cliente: Otorga las especificaciones y escribe pruebas funcionales del software para validar su implementación, decide qué especificaciones e historias de usuario tendrán prioridad y serán implementadas en cada iteración.
- Encargado de pruebas (Tester): Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas de software y comunica los resultados al equipo de desarrollo.
- Encargado de seguimiento (Tracker): Verifica los avances y toma en cuenta las estimaciones realizadas contra el tiempo real dedicado.
- Entrenador (Coach): Es responsable de todo el proyecto de forma global, se encarga de hacer cumplir las prácticas de XP.
- Consultor: Posee conocimientos específicos de algún tema que sea necesario para el proyecto.
- Gestor: Se encarga de la coordinación del equipo.

El ciclo de desarrollo de XP consiste en[28]:

1. El cliente define el valor del negocio.
2. El programador estima el esfuerzo necesario para la implementación.
3. El cliente elige qué construir, dependiendo las prioridades, el tiempo, y el costo.
4. El programador construye el valor de negocio.
5. Se repite el paso 1.

7 Cronograma

El cronograma se muestra a través del siguiente enlace:

<https://drive.google.com/file/d/1WUhTZsY-5wSrtGO8q5daY2NPJLr3jGJ/view?usp=sharing>

8 Presupuesto

A continuación se muestran los valores que serán necesarios cubrir para la parte de recursos humanos conjuntamente materiales, servicios y valores imprevistos, los recursos humanos se manejan con el número de horas necesarias para la realización del proyecto.

Tabla 8.1: Talento humano.

Persona	Horas	Costo por Hora (\$)	Costo Total (\$)
Estudiante 2 (Autores)	400	10	8000
Director	64	10.45	668.80
Total			1468.80

Tabla 8.2: Bienes.

Equipo	Cantidad	Costo (\$)	Costo Total (\$)
Computadora	2	1200	2400
Total			2400.00

Tabla 8.3: Servicios.

Descripción	Tiempo	Cantidad	Costo (\$)	Costo Total (\$)
Servicio de internet	4 (meses)	2	30 x 1 mes x 1 servicio	240
Servidor de testing	2 (meses)	1	15 x 1 mes	30
Total				270.00

Tabla 8.4: Imprevistos.

Valor Imprevistos (\$)
827.76

Tabla 8.5: Presupuesto Total.

Descripción	Total (\$)
Talento humano	1468.80
Bienes	2400.00
Servicios	270.00
Imprevistos	827.76
Total	4966.56

Bibliografía

- [1] Universidad Nacional de Loja, “Reglamento de régimen académico de la universidad nacional de Loja,” *Secretaría General de la Universidad Nacional de Loja*, p. 187, 2021.
- [2] Universidad Nacional de Loja, “Plan estratégico de desarrollo institucional 2019-2023,” tech. rep., Universidad Nacional de Loja, Loja, 2019.
- [3] C. de Educación Superior, “Reglamento de regimen academico,” *Resolución del Consejo de Educación*, pp. 1–51, 2019.
- [4] S. Radack, “Security considerations in the system development life cycle,” *National Institute of Standards and Technology*, pp. 1–7, 2009.
- [5] F. Berzal, “El ciclo de vida de un sistema de información,” 2004.
- [6] M. J. D. F. . V. J. M. Cadavid, A. N., “Revisión de metodologías ágiles para el desarrollo de software.,” *Prospectiva*, vol. 11, pp. 30–39, 2013.
- [7] B. M. Montero, H. V. Cevallos, and J. D. Cuesta, “Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software,” *Revista multidisciplinaria de investigación Espirales*, vol. 2, pp. 113–121, 2018.
- [8] Y. R. Cruz, *Desarrollo del módulo de maquetado para formularios del generador de aplicaciones enriquecidas de internet modeladas bajo el patrón arquitectónico MVC usando UML e IFML*. PhD thesis, 2020.
- [9] E. Vértice, *Diseño básico de páginas web en HTML*, vol. 1. Editorial Vértice, 1 ed., 2009.
- [10] J. Krause, “Css: Cascading style sheets,” *Introducing Web Development*, pp. 65–86, 2016.
- [11] M. R. Valarezo Pardo, J. A. Honores Tapia, A. S. Gómez Moreno, and L. F. Vínces Sánchez, “Comparación de tendencias tecnológicas en aplicaciones web,” *3C Tecnología, Glosas de Innovación aplicadas a la pyme*, vol. 7, no. 1, pp. 28–49, 2018.

- [12] J. R. Molina Ríos, N. M. Loja Mora, M. P. Zea Ordóñez, and E. L. Loaiza Sojos, “Evaluación de los Frameworks en el Desarrollo de Aplicaciones Web con Python,” *Revista Latinoamericana de Ingenieria de Software*, vol. 4, no. 4, p. 201, 2016.
 - [13] C. L. Vidal-Silva, A. Sánchez-Ortiz, J. Serrano, and J. M. Rubio, “Experiencia académica en desarrollo rápido de sistemas de información web con Python y Django,” *Formacion Universitaria*, vol. 14, no. 5, pp. 85–94, 2021.
 - [14] M. A. Alvarez, “Manual de jquery,” vol. 17, pp. 1–37, 2010.
 - [15] B. Nelson, “Getting to know vue.js,” *Getting to Know Vue.js*, 2018.
 - [16] P. So, “Vue.js,” *Decoupled Drupal in Practice*, pp. 381–397, 2018.
 - [17] T. J. Barboto Álvarez, *Análisis comparativo de los frameworks móviles entre Vue y Angular*. PhD thesis, Universidad Técnica de Babahoyo, 2022.
 - [18] V. Zuñiga and Ramon, “Desarrollo de aplicaciones web utilizando Angular como framework,” p. 56, 2020.
 - [19] J. Bucanek, “Model-view-controller pattern,” *Learn Objective-C for Java Developers*, pp. 353–402, 2009.
 - [20] J. Hunt, “The model-view-controller architecture,” *Smalltalk and Object Orientation*, pp. 266–278, 1997.
 - [21] I. of Electrical and E. Engineers, “Especificación de requisitos según el estándar de iee 830,” *Especificacion de Requisitos segun el estandar de IEEE 830*, p. 27, 2008.
 - [22] E. Bonilla Huerta, F. Ramírez Cruz, and E. Sanchez Lucero, “Advances in intelligent information technologies,” *Research in Computing Science, LXXIX*, 2014.
 - [23] G. Méndez, “Especificación de requisitos según el estándar de iee 830,” *Facultad de Informática, Universidad Complutense de Madrid*, 2008.
 - [24] D. García León and A. Beltrán Benavides, “Un enfoque actual sobre la calidad del software,” *ACIMED*, vol. 3, pp. 40 – 42, 12 1995.
 - [25] P. G. Tello, *Evaluación de calidad de un producto de software*. PhD thesis, Universidad Nacional de la Plata, 2016.
 - [26] E. A. Balseca Chisaguano, “Evaluación de calidad de productos de software en empresas de desarrollo de software aplicando la norma iso/iec 25000,” B.S. thesis, Quito, 2015., 2014.
-

-
- [27] S. D. Amaro Calderón and J. C. Valverde Rebaza, *Metodologías Ágiles*. PhD thesis, Universidad Nacional de Trujillo, 2007.
- [28] P. Letelier, M. Canós, E. Sánchez, and M. Penadés, “Metodologías Ágiles en el desarrollo de software,” *Valencia, Valencia, España*, pp. 1–8, 2003.
-