**Departamento de Ciencias de la Computación (DCCO)**

**Carrera de Ingeniería en Software**

**Prácticas Pre Profesionales (PPP)**

Perfil del Proyecto

Presentado por: Pilataxi Miranda, Diego

Tutor académico: Ruiz Robalino, Jenny

Ciudad: Quito

Fecha: 26/06/2025

**Índice Pág.**

[**Introducción** 5](#_Toc200114325)

[**Planteamiento del trabajo** 5](#_Toc200114326)

[**Formulación del problema** 5](#_Toc200114327)

[**Justificación** 5](#_Toc200114328)

[**Sistema de Objetivos** 5](#_Toc200114329)

[**Objetivo General** 5](#_Toc200114330)

[**Objetivos Específicos** 6](#_Toc200114331)

[**Alcance** 6](#_Toc200114332)

[**Marco Teórico** 7](#_Toc200114333)

[**Metodología (Marco de trabajo 5W+2H)** 7](#_Toc200114334)

[**Ideas a Defender** 8](#_Toc200114335)

[**Resultados Esperados** 9](#_Toc200114336)

[**Responsables** 9](#_Toc200114337)

[**8.1 Humana** 9](#_Toc200114338)

[**Conclusiones y recomendaciones** 11](#_Toc200114339)

[**9.1 Conclusiones** 11](#_Toc200114340)

[**9.2 Recomendaciones** 11](#_Toc200114341)

# **Introducción**

En respuesta a la necesidad de modernizar sistemas monolíticos y mejorar la escalabilidad y mantenibilidad del software, el presente proyecto aborda la migración de módulos de la aplicación Node Analytics, desarrollado originalmente en PHP, hacia una arquitectura basada en microservicios utilizando NestJS. Esta iniciativa surge como solución a las limitaciones del sistema anterior y como una oportunidad para implementar prácticas actuales como la integración continua, el uso de control de versiones con GitLab y el despliegue automatizado, aportando así una mejora significativa al proceso de desarrollo dentro de la empresa INNOBIX.

# **Planteamiento del trabajo**

## **Formulación del problema**

El sistema Wifi-Analytics desarrollado en PHP presentaba dificultades de mantenimiento, escalabilidad y estructuración del código, lo cual afectaba la eficiencia del desarrollo y la integración de nuevas funcionalidades. Estas limitaciones técnicas motivaron la necesidad de migrar el sistema hacia una arquitectura moderna basada en microservicios con NestJS, permitiendo una mejor organización, pruebas más claras y despliegue más eficiente.

## **Justificación**

La migración del sistema representa una mejora significativa en términos de rendimiento, mantenimiento y buenas prácticas de desarrollo. Además, permitió al pasante aplicar conocimientos adquiridos en entornos reales, incluyendo herramientas modernas como GitLab, SSH, CI/CD y pruebas funcionales, generando así un impacto práctico y profesional relevante tanto para la empresa como para su formación.

# **Sistema de Objetivos**

Los objetivos que se han propuesto para la realización de Prácticas Pre-profesionales han sido los siguientes

## **Objetivo General**

Desarrollar la migración del sistema Wifi-Analytics, originalmente implementado en PHP, hacia una arquitectura modular basada en microservicios utilizando el framework NestJS, con el fin de mejorar la escalabilidad, mantenibilidad y eficiencia en el desarrollo y despliegue del sistema dentro del entorno profesional de la empresa INNOBIX.

## **Objetivos Específicos**

* Analizar y comprender la estructura y funcionamiento del sistema Wifi-Analytics actual para identificar los módulos clave que deben ser migrados.
* Integrar prácticas de automatización mediante la configuración de pipelines CI/CD que permitan realizar despliegues controlados y eficientes del sistema en su nueva arquitectura.
* Implementar la migración progresiva de los módulos del sistema, aplicando principios de desarrollo limpio, buenas prácticas de codificación y empleando herramientas como GitLab, SSH y control de versiones.

|  |  |  |
| --- | --- | --- |
| **N.º** | **Actividad / Paquete de trabajo** | **Descripción** |
| 1 | Inducción y familiarización | Recepción de credenciales, configuración del entorno y diagnóstico técnico. |
| 2 | Análisis del sistema actual | Revisión del sistema Wifi-Analytics en PHP y mapeo de módulos clave. |
| 3 | Migración modular a NestJS | Codificación de cada módulo en arquitectura de microservicios con NestJS. |
| 4 | Configuración de repositorios y automatización | Uso de GitLab, creación de llaves SSH y variables de entorno. |
| 5 | Integración de CI/CD | Configuración de pipelines para automatizar pruebas y despliegue. |
| 6 | Pruebas finales y documentación | Validación funcional, generación de documentación técnica y cierre de prácticas. |

# **Alcance**

El proyecto contempla la migración funcional de los principales módulos del sistema Wifi-Analytics, originalmente desarrollado en PHP, hacia una arquitectura moderna utilizando NestJS. Entre las funcionalidades consideradas se incluye la codificación de módulos como index, campaigns, devices, people, options y bienvenida, así como la implementación de flujos de trabajo en GitLab, control de versiones, despliegue mediante CI/CD, y la validación funcional de los servicios migrados. Este proyecto se limita al backend de la aplicación, sin intervenir en la interfaz gráfica o lógica del frontend.

# **Marco Teórico**

Durante el desarrollo del proyecto se utilizaron diversas herramientas y entornos de desarrollo que facilitaron la migración del sistema Wifi-Analytics. El IDE principal fue Visual Studio Code, elegido por su compatibilidad con el framework NestJS, su ligereza y extensiones para desarrollo backend en TypeScript. Para el control de versiones se utilizó Git, en conjunto con GitLab como plataforma de almacenamiento y colaboración de código. Adicionalmente, se implementaron pipelines de CI/CD configurados en GitLab, los cuales permitieron automatizar el proceso de integración, pruebas y despliegue. Otras herramientas relevantes incluyeron el uso de Postman para pruebas de endpoints y Excel para el control del cronograma y bitácora de actividades.

## **Metodología (Marco de trabajo 5W+2H)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **¿QUÉ?** | **¿CÓMO?** | **¿QUIÉN?** | **¿CUÁNDO?** | **¿POR QUÉ?** | **¿CUÁNTO?** | **% DE CUMPLIMIENTO** |
| Migración del sistema Wifi-Analytics de PHP a NestJS | Refactorizando módulos del sistema en estructura modular NestJS | Pasante (tú) | Del 5 al 31 de mayo de 2025 | Para modernizar el sistema y mejorar su mantenibilidad y escalabilidad | 00 | 100% |
| Configuración de repositorio y SSH en GitLab | Creando llaves SSH y repositorio privado | Pasante | Semana 2 | Para permitir control de versiones y colaboración segura | 1 día | 100% |
| Implementación de CI/CD con GitLab | Configurando pipelines y variables en .gitlab-ci.yml | Pasante con guía del tutor | Semana 3 | Para automatizar pruebas y despliegue del sistema | 2 días | 100% |
| Pruebas funcionales y validación de módulos | Mediante tests manuales y revisión de endpoints | Pasante | Semana 4 | Para garantizar que los servicios migrados funcionen correctamente | 2 días | 100% |

Tabla 1 Marco de trabajo 5W+2H

# **Ideas a Defender**

Este proyecto defiende la idea de que migrar sistemas monolíticos desarrollados en tecnologías tradicionales como PHP hacia arquitecturas modernas basadas en microservicios, como las que permite NestJS, mejora significativamente la mantenibilidad, escalabilidad y calidad del software. Se sustenta en principios fundamentales de la Ingeniería de Software como el desarrollo modular, la reutilización de código, el uso de control de versiones (Git), automatización de procesos (CI/CD) y pruebas funcionales, aplicados en un entorno real de trabajo. Asimismo, se reafirma la importancia de aplicar buenas prácticas de programación para generar soluciones limpias, eficientes y sostenibles.

# **Resultados Esperados**

Como resultado del desarrollo de este proyecto, se espera lograr una migración funcional y eficiente del sistema Wifi-Analytics hacia un backend modular en NestJS, con una estructura de microservicios clara y operativa. Adicionalmente, se espera la implementación correcta de flujos de trabajo con Git y GitLab, la configuración de pipelines de integración continua, y la validación completa del sistema mediante pruebas funcionales. Estos resultados no solo evidencian la correcta aplicación de conocimientos adquiridos en Ingeniería de Software y Fundamentos de Programación, sino que también aportan valor real al entorno empresarial donde se desarrolló la pasantía.

# **Responsables**

## **8.1 Humana**

**8.1.1 Tutor Empresarial**

Ing. Esteban Andaluz

**• Responsabilidades**

* Supervisar el desarrollo de las actividades del pasante en la empresa.
* Asignar tareas técnicas relacionadas con el proyecto de migración.
* Brindar retroalimentación continua sobre avances y correcciones.
* Validar el cumplimiento de objetivos prácticos y técnicos.

**8.1.2 Tutor Académico**

Ing. Jenny Ruiz

* Guiar al estudiante en el cumplimiento del perfil académico del proyecto.
* Realizar seguimiento institucional y validar entregables según la malla curricular.
* Servir como enlace entre la empresa y la universidad para asegurar coherencia entre objetivos educativos y empresariales.

**8.1.3 Estudiante**

Diego Pilataxi

**• Responsabilidades**

* Ejecutar la migración del sistema Wifi-Analytics de PHP a NestJS.
* Aplicar conocimientos de programación, herramientas DevOps y control de versiones.
* Documentar el proceso y presentar resultados a tutores.
* Cumplir con los cronogramas y actividades asignadas durante el período de pasantía.

**8.2 Tecnológica**

* Laptop personal con especificaciones mínimas: Intel CORE i7, 21GB RAM, 1TB SSD.
* Visual Studio Code como IDE de desarrollo.
* Git y GitLab para control de versiones y colaboración.
* Docker para pruebas y contenedores.
* Postman para pruebas de servicios.
* Excel para cronogramas y registro de actividades.
* NestJS, Node.js, y otras librerías backend como base tecnológica.

**8.2.1 Hardware**

|  |  |  |
| --- | --- | --- |
|  | Requisitos mínimos | Disponibilidad |
| Memoria RAM | 24 GB de RAM | Alta |
| Almacenamiento | 1 TB de espacio de almacenamiento | Alta |

Tabla 3 Requisitos de Hardware

**8.2.2 Software**

|  |  |  |
| --- | --- | --- |
|  | Requisitos mínimos | Disponibilidad |
| Sistema Operativo | Windows 11 | Alta |
| IDE | Es recomendable Visual Studio Code debidoa su conexión con FTP, sin embargo, cualquier IDE con esta funcionalidad funciona. | Alta |

Tabla 4 Requisitos de Software

# **Conclusiones y recomendaciones**

## **9.1 Conclusiones**

## AL FINAL ESPECIFICAR

## **9.2 Recomendaciones**

## AL FINAL ESPECIFICAR