



*Instituto Tecnológico Superior de
Acatlán de Osorio*

SEP SEIT TNM

DIVISIÓN DE LA CARRERA DE INGENIERÍA INFORMÁTICA

“Proyecto de Residencia Profesional”

Proyecto

“Desarrollo y despliegue para un Sistema de Vinculación

**Universitaria a través de DevOps en el Instituto Nacional de Astrofísica, Óptica y
Electrónica”**

Que para obtener el título de:

Ingeniero Informático

Presenta

Ximena Castillo Martínez

Número de Control

191009007

Acatlán de Osorio, Pue., Enero de 2024

Agradecimientos

Me gustaría expresar mi gratitud a todas las personas que han formado parte de mi vida, principalmente quiero agradecer a mis padres, quienes han sido mi fuente constante de apoyo, orientación y aliento para poder cumplir todos mis objetivos personales y académicos. Su amor incondicional y sacrificios han sido la fuerza impulsora detrás de cada logro, y siempre han sido mis mejores guías de vida.

A mi familia, incluyendo a mis hermanos, tíos, tíos y abuelitas, por su respaldo tanto económico como emocional.

Quisiera dedicar un agradecimiento especial a mis dos abuelitos, quienes ahora son mis angelitos y ya no pudieron presenciar el momento en que su nieta se convertiría en ingeniera, pero me inspiran a seguir esforzándome y honrando su memoria en cada paso que doy.

A Dios por brindarme la fortaleza, la sabiduría y las oportunidades que me han guiado y llegado hasta este momento.

Gracias a mis compañeros de la carrera por enseñarme quien si y quien no, a mis amigos externos y sobre todo a mis roomies que si han sido parte fundamental en este último trayecto de residencia.

Finalmente, a mi asesor interno y externo, por compartir sus conocimientos de manera profesional e invaluable, por su dedicación y perseverancia. Y a todos los docentes que fueron fundamentales para mi formación académica, que me dieron las bases del conocimiento en cada etapa que pase.

Resumen

El presente proyecto tiene como objetivo principal proporcionar un sistema que apoye los procesos de captación de estudiantes externos de nivel superior (licenciatura e ingenierías) de universidades (públicas y privadas), para que realicen prácticas, servicios o estancias profesionales con la comunidad académica y científica del Instituto Nacional de Astrofísica, Óptica y Electrónica. Por lo tanto, el proyecto contempla los procedimientos de convocatoria para proyectos de trabajo internos o externos, así como su realización.

Este sistema permitirá atraer de manera accesible, electrónica y rápida, mediante las nuevas tecnologías, a estudiantes de universidades, tecnológicos e instituciones de educación superior, al mismo tiempo que promoverá nuestros posgrados y líneas de investigación.

El resultado final abarcará el desarrollo de la infraestructura, la plataforma web y su implementación. Para llevar a cabo este proyecto, se basará en el uso de DevOps, sistemas web y diversas herramientas universales para que pueda interpretarse en cualquier dispositivo a través del navegador. Esto dará como resultado un proceso gestionable, medible y retroalimentativo.

INDICE DE CONTENIDO

1. Introducción	3
2. Problemas a resolver	4
3. Objetivos	5
3.1 Objetivo general.	5
3.2 Objetivos específicos.....	5
4. Justificación	6
5. Marco teórico.....	7
5.1. Antecedentes del Proyecto.....	7
5.2. ¿Qué es un sistema de vinculación universitaria?	8
5.3. Metodología DevOps	9
5.4. Infraestructura del proyecto	10
5.5. Ambiente de Desarrollo	11
5.6. DevOps.....	12
5.6.1Importancia	12
5.7. Automatización de procesos.....	13
5.7.1. Integración Continua.....	13
5.7.2. Entrega Continua	14
5.7.3. Despliegue Continuo.....	14
5.8. Source Code Management- Gestión de Código Fuente SCM	15
5.8.1. GitLab	15
5.9. Virtualización	15
5.9.1. Vagrant.....	15
5.9.2. Virtualbox	16
5.10. Sistema operativo	16
5.10.1. Linux.....	16
5.10.2. Debían	16
5.11. Las herramientas de Infraestructura.....	17
5.11.1. Ansible	17
5.12. Herramientas de desarrollo	18
5.13. Entornos de Desarrollo Integrado (IDE).....	18
5.13.1. Visual Studio Code	18

5.14. Base de Datos	19
5.14.1. PostgreSQL	19
5.14.2. PgAdmin	19
5.15. Lenguaje de programación.....	20
5.15.1. PHP	20
5.15.2. JavaScript	20
5.15.3. HTML	20
5.15.4. CSS	21
5.16. Frameworks de Desarrollo Web.....	21
5.16.1. Laravel	21
5.16.1. Composer.....	22
5.17. Servidores Web.....	22
5.17.1. Apache	23
5.18. La computación de alto rendimiento (HPC)	23
5.18.1. Grafana	23
5.19. Direcciones IP	24
5.19.1. Direcciones IP públicas.....	24
5.19.2. Direcciones IP privadas (reservadas).....	24
5.19.3. Direcciones IP estáticas (fijas).....	24
5.19.4. Direcciones IP dinámicas.....	24
6. Cronograma de actividades	25
7. Procedimiento y descripción de las actividades realizadas.....	26
7.1. Planeación y gestión de infraestructura tecnológica	26
7.2. Identificar las partes interesadas.....	28
7.3. Reunión con las partes interesadas.....	29
.....	30
7.4. Identificar las características no funcionales de proyecto	30
7.5 Definición e implementación de metodología y procesos de desarrollo seguro	31
7.6 Análisis y diseño de la arquitectura, infraestructura y configuración del sistema.....	34
7.7 Diseño de interfaces de los módulos registro y postulación de proyecto ...	37

7.8 Desarrollo de componentes (Modelos, Controlador y Vistas)	48
7.9 Configuración de servidores, cuentas de acceso a la red y servicios del software.	49
7.10 Creación y pruebas para el código en entorno de desarrollo.....	51
7.11 Pruebas no funcionales.....	54
7.12 Integración de codificación	55
7.13 Automatización de implementación.....	57
8. Resultados	61
	62
9. Conclusiones, recomendaciones y experiencia profesional adquirida	68
9.1. Conclusiones.....	68
9.2. Recomendaciones.....	68
9.3. Experiencia profesional.....	69
10. Competencias desarrolladas y/o aplicadas	70
11. Fuentes de información.....	71

TABLA DE FIGURAS

Figura 1 Cronograma de actividades	25
Figura 2 Herramientas DevOps.....	27
Figura 3 Partes interesadas del Sistema de Vinculación Universitaria.....	29
Figura 4 Reunión con las partes interesadas.....	30
Figura 5 DevOps. La cultura de buenas prácticas	31
Figura 6 Muro principal de Microsoft Teams	32
Figura 7 Clasificación de carpetas del proyecto en Microsoft Teams.....	33
Figura 8 Calendario de actividades en Microsoft Teams	33
Figura 9 Codificación del Vagranfile	34
Figura 10 Inicio de la máquina virtual.....	36
Figura 11 Interfaz de Laravel instalado mediante el vagrantfile.....	37
Figura 12 Diagrama General de Actividades de Registro de Proyecto	38
Figura 13 Diagrama General de Actividades de Postulación del Proyecto.....	39
Figura 14 Diagrama General de Estado de Registro del Proyecto	40
Figura 15 Diagrama General de Estado de Postulación del Proyecto.....	41
Figura 16 Centro de mensajes E-mail y Alertas	41
Figura 17 Pre-interfaz visual del primer pre-registro del aspirante.....	42
Figura 18 Pre-interfaz de la confirmación del primer pre-registro del aspirante.....	43
Figura 19 Pre-Interfaz del formulario para el registro del proyecto	43
Figura 20 Especificaciones de las interfaces	44
Figura 21 Herramienta draw.io	45
Figura 22 Diagrama de la Base de Datos.....	46
Figura 23 Concentrado del módulo del registro del proyecto	47
Figura 24 Concentrado del módulo de la postulación del proyecto.....	47
Figura 25 Diagrama de MCV	49
Figura 26 Diagrama de clonado del código y base de datos	50
Figura 27 Codificación de Vagrantfile integrado con GitLab	52
Figura 28 Inicialización de las máquinas virtuales para GitLab	53
Figura 29 Diagrama de CI/CD con GitLab.....	56
Figura 30 Código GitLab CI/CD	56
Figura 31 Diagrama de despliegue.....	57
Figura 32 Codificación con Ansible	58
Figura 33 Playbook de Ansible para la automatización.	59
Figura 34 Ejecución de las maquinas con Ansible.....	60
Figura 35 Diagrama representando los submodulos	61
Figura 36 Interfaz lista de proyectos	62
Figura 37 Pre-registro del aspirante	63
Figura 38 Interfaz del responsable del proyecto	64
Figura 39 Interfaz del responsable del proyecto para la evaluación de entrevista	65
Figura 40 Interfaz del aspirante registro oficial	66

Figura 41 Interfaz de Dirección de Formación Académica 67

1.Introducción

El presente trabajo tiene como propósito explorar el desarrollo y despliegue de un Sistema de Vinculación Universitaria a través de la metodología DevOps, entendida como la amalgama de prácticas y herramientas destinadas a incrementar la capacidad de una organización para entregar aplicaciones y servicios a alta velocidad. Este enfoque es especialmente relevante en el ámbito de la educación superior, donde la eficiencia y la innovación son cruciales para la formación de profesionales y la gestión de la colaboración entre la universidad y su entorno.

En este sentido, el sistema de Vinculación Universitaria se presenta como un componente esencial en la articulación de relaciones estratégicas entre el instituto y diversos actores, instituciones privadas o públicas. La implementación de este sistema mediante DevOps promete no solo una integración continua y un despliegue rápido, sino también una mayor colaboración entre los equipos de desarrollo y operaciones, lo que resulta en un servicio más resiliente y adaptable a las necesidades cambiantes del instituto.

A lo largo de este trabajo, se abordarán las bases teóricas de DevOps, así como su aplicación práctica en el desarrollo de un sistema de Vinculación Universitaria. Se analizarán casos de estudio, se identificarán las mejores prácticas y se evaluarán los retos y beneficios de esta metodología en el contexto educativo. Con este enfoque, se busca ofrecer una perspectiva integral que permita comprender la importancia de la adopción de DevOps para la mejora continua y la innovación en los procesos de vinculación y colaboración universitaria.

2. Problemas a resolver

Anteriormente el proceso de selección de Alumnos externos para la realización de tesis, prácticas profesionales, servicio social, era de manera directa con el Investigador o tecnólogo, de una manera informal y que no abarcaba todas las áreas y líneas de investigación institucionales. Sino con algunas personas de manera informal. Una vez definida la colaboración el alumno se registraba en la Dirección de Formación Académica y entregando su carta de presentación.

Estos procesos no tienen actualmente un control ni un histórico de cuantos alumnos realizan prácticas y cuántos de ellos tienen un impacto Institucional, de la misma manera se desconoce quién o qué coordinación atiende más alumnos bajo este concepto. Por tal motivo se ha visto en la necesidad de crear un sistema que gestione todo aquello relacionado a la Vinculación Universitaria.

Es por eso que se desarrollará este sistema, que permitirá atraer a estudiantes de universidades, tecnológicos e instituciones de educación superior, de una manera más apropiada, electrónica y rápida, así mismo tener un control de cuantos estudiantes realizan el desarrollo de proyectos de Tesis, Prácticas, Estadías y Servicio Social

3. Objetivos

3.1 Objetivo general.

Desarrollar un sistema que apoye a los procesos de captación de estudiantes externos mediante un sistema electrónico para el desarrollo de proyectos de Tesis, Prácticas, Estadias y Servicio Social en el Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE).

3.2 Objetivos específicos.

- Definir las características funcionales y no funcionales de Postulación y Registro de Proyecto.
- Planificar y gestionar la infraestructura tecnológica.
- Definir e implementar de metodologías y procesos de desarrollo seguro.
- Realizar análisis y diseño de arquitectura de software.
- Conducir pruebas unitarias, integrales y funcionales con el usuario interno.
- Diseñar interfaces (Formulario de postulación y registro de proyecto).

4. Justificación

Anteriormente el proceso de selección de Alumnos externos para la realización de tesis, prácticas profesionales, servicio social, era de manera directa con el Investigador o tecnólogo, de una manera informal y que no abarcaba todas las áreas y líneas de investigación institucionales. Sino con algunas personas de manera informal. Una vez definida la colaboración el alumno se registraba en la Dirección de Formación Académica y entregando su carta de presentación.

Estos procesos no tienen actualmente un control ni un histórico de cuantos alumnos realizan prácticas y cuántos de ellos tienen un impacto Institucional, de la misma manera se desconoce quién o qué coordinación atiende más alumnos bajo este concepto. Por tal motivo se ha visto en la necesidad de crear un sistema que gestione todo aquello relacionado a la Vinculación Universitaria.

Es por eso que se desarrollará este sistema, que permitirá atraer a estudiantes de universidades, tecnológicos e instituciones de educación superior, de una manera más apropiada, electrónica y rápida, así mismo tener un control de cuantos estudiantes realizan el desarrollo de proyectos de Tesis, Prácticas, Estadías y Servicio Social.

5. Marco teórico

Este fundamento teórico proporciona una base sólida para comprender la intersección entre el desarrollo de software, DevOps y la vinculación universitaria, sentando las bases para el diseño e implementación exitosos del sistema propuesto.

5.1. Antecedentes del Proyecto

Como antecedentes podemos citar que anteriormente el proceso de selección de Alumnos externos para la realización de tesis, prácticas profesionales, servicio social, era de manera directa con el Investigador o tecnólogo, de una manera informal y que no abarcaba todas las áreas y líneas de investigación institucionales. Sino con algunas personas de manera informal. Una vez definida la colaboración el alumno se registraba en la Dirección de Formación Académica y entregando su carta de presentación.

Estos procesos no tienen actualmente un control ni un histórico de cuantos alumnos realizan prácticas y cuántos de ellos tienen un impacto Institucional, de la misma manera se desconoce quién o qué coordinación atiende más alumnos bajo este concepto. Por tal motivo se ha visto en la necesidad de crear un sistema que gestione todo aquello relacionado a la Vinculación Universitaria.

Como se desea una interacción con el alumno el Sistema de Vinculación Universitaria desde ahora denominado (SVU), tendrá que tener una modalidad web debido a que esta puede ser accesible por todos, la cual propone de lenguajes que pueden compilar desde cualquier equipo por medio de su navegador. Teniendo en cuenta esto el SVU contempla aspectos importantes como los tiempos de recepción de cada trabajo, el registro y vinculación entre las Instituciones (refuerzo académico), evaluaciones de desempeño y la eficiencia terminal. El desarrollo y despliegue de un sistema de Vinculación Universitaria mediante la implementación de DevOps implica integrar una serie de conceptos y teorías fundamentales.

5.2. ¿Qué es un sistema de vinculación universitaria?

Un sistema de vinculación universitaria se refiere a un conjunto de políticas, programas y actividades diseñadas para establecer y fortalecer las relaciones entre una universidad y diversas entidades externas, como la industria, el gobierno, organizaciones sin fines de lucro y la comunidad en general. Este sistema busca promover la colaboración y la interacción entre la universidad y su entorno, con el objetivo de enriquecer la experiencia educativa de los estudiantes, fomentar la investigación aplicada, y contribuir al desarrollo social y económico.

Las universidades suelen implementar programas de vinculación con varios propósitos, entre ellos:

Colaboración en Investigación: Facilitar la colaboración entre la universidad y la industria para realizar investigaciones conjuntas, proyectos de desarrollo y transferencia de tecnología.

Desarrollo de Habilidades: Establecer programas de prácticas, pasantías o proyectos que permitan a los estudiantes aplicar sus conocimientos en entornos del mundo real y desarrollar habilidades prácticas.

Transferencia de Conocimientos: Facilitar la transferencia de conocimientos y experiencias entre la universidad y otros sectores de la sociedad, contribuyendo así al desarrollo de la comunidad y la resolución de problemas locales.

Innovación y Emprendimiento: Estimular la innovación y el espíritu emprendedor al brindar apoyo a iniciativas empresariales y proyectos de estudiantes, profesores o emprendedores locales.

La vinculación universitaria es esencial para crear puentes entre la teoría académica y la aplicación práctica, así como para fortalecer la relevancia de la educación superior en el contexto social y económico.

5.3. Metodología DevOps

La metodología DevOps es un enfoque que busca la colaboración y la integración entre los equipos de desarrollo de software (**Dev**) y las operaciones de tecnología de la información (**Ops**), con el objetivo de acelerar el ciclo de vida del desarrollo de software y mejorar la calidad de las entregas.” (NetApp, 2019).

Para seguir la metodología DevOps, generalmente se puede dividir en las siguientes fases:

Planificación (Plan): En esta fase, los equipos de desarrollo y operaciones colaboran para definir los objetivos del proyecto, identificar las necesidades y establecer un plan que incluye la arquitectura del sistema, los requisitos y los plazos.

Diseño: Diseñar la arquitectura del sistema y las infraestructuras necesarias para la implementación.

Desarrollo (Develop): Los desarrolladores crean el código fuente del software siguiendo las especificaciones y requisitos establecidos en la fase de planificación. Es importante escribir código de alta calidad y realizar pruebas unitarias en esta etapa.

Pruebas (Test): Se realizan pruebas de calidad, seguridad y rendimiento para garantizar que el software funcione correctamente. Esto puede incluir pruebas automatizadas y manuales.

Integración (Integration): Los cambios de código se integran de manera continua en un entorno compartido, lo que permite la detección temprana de problemas de integración y conflictos entre códigos.

Despliegue (Deploy): El software se implementa en entornos de producción o de prueba, generalmente de manera automatizada, para que esté listo para su uso.

Monitoreo y Retroalimentación (Monitor & Feedback): Se implementan herramientas de monitoreo para rastrear el rendimiento del software en tiempo real y recopilar comentarios de los usuarios. Esto permite identificar y abordar problemas rápidamente.

5.4. Infraestructura del proyecto

En este documento, se describe la infraestructura sobre el desarrollo y despliegue de un sistema de vinculación universitaria utilizando la metodología DevOps. Esta infraestructura busca optimizar la colaboración, agilidad y confiabilidad en todo el ciclo de vida del desarrollo y asegurar un proceso de despliegue fluido y seguro.

Se tomaron en cuenta la utilización de las herramientas de tecnologías para el despliegue con DevOps, la infraestructura es la siguiente:

Tabla 1

Equipo requerido para el despliegue

Infraestructura de Despliegue	
Sistema Operativo	
Debian 12	Debian es una distribución de sistema operativo basada en el kernel de Linux.
Software	Función
Vagrant	Vagrant permite definir la configuración de una máquina virtual a través de un archivo de configuración simple y legible.
Virtual Box	Permite crear máquinas virtuales (VM) en las cuales se pueden instalar sistemas operativos adicionales. Esto es útil para propósitos de desarrollo, pruebas y experimentación sin afectar el sistema operativo principal.
Ansible	Ansible automatiza la ejecución de tareas y operaciones en sistemas, eliminando la necesidad de realizar estas acciones manualmente. Esto ahorra tiempo y reduce errores asociados con la intervención humana.
Base de Datos	Es almacenar datos de manera estructurada. Esto permite organizar la

PostgreSQL	información de manera que sea fácilmente accesible y manipulable.
Gitlab	Proporciona repositorios de código Git privados o públicos para almacenar y organizar el código fuente de los proyectos.
Laravel	Es un popular framework de desarrollo web en PHP que proporciona una serie de funciones y características para simplificar y acelerar el desarrollo de aplicaciones web

Nota: Información específica sobre los componentes de la infraestructura de despliegue.
(Ximena,2024)

5.5. Ambiente de Desarrollo

El ambiente de desarrollo se configura para permitir a los equipos de desarrollo iterar rápidamente en el código y colaborar de manera efectiva. Se utiliza un repositorio centralizado basado en control de versiones, como Git, para gestionar el código fuente.”
(Explicación de los entornos de Desarrollo Integrado - AWS, s. f.)

Las siguientes herramientas y componentes son esenciales en el ambiente de desarrollo:

- Reppositorio de Código Fuente: Se utiliza Git para almacenar y gestionar el código fuente de manera colaborativa.
- Integración Continua (CI): Se emplea una plataforma de CI, como Jenkins o GitLab CI/CD, para automatizar la integración de código y ejecutar pruebas automáticas en cada cambio.
- Automatización de Pruebas: Se configuran pruebas automatizadas, como pruebas unitarias, pruebas de integración y pruebas de regresión, para asegurar la calidad del código.

Las herramientas clave incluyen:

- ❖ Configuración Automatizada: Se automatiza la configuración de servidores y servicios mediante scripts y plantillas.

- ❖ Entorno de Despliegue: El entorno de despliegue está diseñado para facilitar la entrega continua y confiable del sistema a producción. Se establece una tubería de entrega (pipeline) que automatiza el proceso de despliegue y prueba.

Los componentes clave son:

- Tubería de Entrega (Pipeline): Se configura una tubería de entrega automatizada que comprende etapas para construcción, pruebas, implementación y monitorización.
- Despliegue Continuo: Las fases de implementación y despliegue se automatizan, lo que permite liberar nuevas versiones del sistema de manera rápida y sin problemas.

5.6. DevOps

DevOps es el acrónimo de las palabras en inglés Development (Desarrollo) y Operations (Operaciones) que, en TI, se refiere a la integración de los equipos de desarrollo, de administración y de operación de sistemas, empleando principios Lean y Agile con objeto de acelerar las entregas de software.” (¿Qué es DevOps? | IBM, s. f.)

DevOps es una de las disciplinas más nuevas en el ámbito del desarrollo de aplicaciones web y, por ello, en los últimos años han surgido toda una serie de herramientas que vienen a facilitar los procesos de despliegue, integración continua y testing de los proyectos. DevOps ofrece un ecosistema muy rico de utilidades que continúa madurando mes a mes, por ello es a veces difícil saber qué herramientas acabarán estableciéndose como dominadores de cada área.”(¿Qué es DevOps? | IBM, s. f.)

5.6.1 Importancia

Investigaciones realizadas durante más de 4 años y documentadas por (Forsgren, Humble, & Kim, 2018) dan cuenta que organizaciones de todos los tamaños, industrias y tipo de tecnologías de software que han implementado prácticas DevOps obtienen mejores resultados, comparativamente contra aquellas compañías donde las prácticas son incipientes o nulas, así:

- Despliegue de código 46 veces más frecuente.

- Tiempo Commit to Deploy (tiempo que transcurre entre el momento de la adición de un cambio en las líneas de código al código fuente y el instante en el cual dichas líneas de código han sido implementadas en producción) 440 veces más corto.
- Tiempo promedio de recuperación (MTTR) de un error desplegado en producción 170 veces más corto.
- Tasa de Falla de los cambios desplegados en producción 5 veces menor.

Adicionalmente, otro estudio de (Puppet Labs, 2014) asegura tener evidencia cuantitativa que el desempeño de TI y las prácticas DevOps contribuyen al desempeño organizacional. Este estudio encontró que las compañías con áreas de TI de alto desempeño tienen una probabilidad dos veces mayor de superar sus metas de rentabilidad, participación de mercado y productividad.

En resumen, la implementación de DevOps no solo se trata de tecnologías y herramientas, sino también de una transformación cultural que mejora la colaboración, la eficiencia y la calidad en el desarrollo y despliegue de software. Esta cultura tiene un impacto significativo en la capacidad de una organización para mantenerse competitiva y responder de manera ágil a los cambios en el mercado y las necesidades de los usuarios.

5.7. Automatización de procesos

La automatización de procesos es una práctica del desarrollo de software que busca la reducción de errores operativos basándose en la capacidad de automatizar dichos procesos, como por ejemplo lo podrían ser los procesos de revisión de componentes de software, la integración entre ellos y hasta la misma la puesta en producción de los mismos. "(Sydle, 2024)

Entre los procesos automatizados requeridos para el pilar tecnológico de DevOps se encuentran los siguientes:

5.7.1. Integración Continua

Una de las prácticas de desarrollo de software contempladas dentro del pilar tecnológico de DevOps, que consiste en la automatización del proceso común de los desarrolladores

en el que se compilan los componentes que han sido enviados al SCM central (Source Code Manager, en español sistemas de control de código fuente) y posteriormente se ejecutan las pruebas unitarias que los mismos requieran, en este caso de manera automatizada, esto con el fin de minimizar los posibles errores de procedimiento por la repetición humana de los procesos, encontrar también defectos en componentes de software de manera temprana para tomar medidas correctivas y optimizar los tiempos de validación de los mismos para garantizar mejores entregas. (Pruebas automatizadas | AWS, s. f.)

5.7.2. Entrega Continua

Es una de las prácticas de desarrollo de software contempladas dentro del pilar tecnológico de DevOps, la entrega continua consiste en el proceso de la preparación de los componentes para convertirse en artefactos listos para producción, lo cual implica básicamente que dentro de su proceso se ha contemplado la integración continua.” (Amazon Web Services, s. f.)

Durante la entrega continua, se llevan a cabo procesos automatizados de pruebas de las compilaciones que van más allá de las pruebas de unidad, incluyen pruebas como lo son las de integración, carga, regresión, aceptación, entre otras, con el fin de realizar una validación integral del release, que será ubicado usualmente en un gestor de artefactos de software para la posterior aprobación manual y entrega a producción por el encargado del proceso.” (Amazon Web Services, s. f.)

5.7.3. Despliegue Continuo

Una de las prácticas de desarrollo de software contempladas dentro del pilar tecnológico de DevOps, el despliegue continua es muy similar a la entrega continua pero con una ligera variación, en este caso el despliegue en producción del release se realiza también de manera automatizada, lo que concluiría en la automatización total de la línea de producción de software, lo cual sería ideal para un proceso muy maduro que requiere de la mayor celeridad para la liberación del producto o de constantes actualizaciones. (¿Qué Es El Despliegue Continuo?, n.d.)

5.8. Source Code Management- Gestión de Código Fuente SCM

Herramientas que coordinan la gestión eficaz del código fuente entre los desarrolladores de un proyecto. Sus componentes principales son el gestor de ficheros y el control de versiones para una correcta gestión. Se permite el trabajo concurrente sobre los ficheros, habida cuenta de que se lleva un control estricto y documentado del trabajo de cada persona en cada bloque de código. (Atlassian, n.d.)

5.8.1. GitLab

Es un servicio de alojamiento de código y gestión de versiones, unido a una completa plataforma DevOps. Ofrece un servicio web de control de versiones y desarrollo de software colaborativo basado en Git. Además de ser un gestor de repositorios, ofrece generación de wikis y un sistema de seguimiento de errores. GitLab es una suite completa que permite gestionar, administrar, crear y conectar los repositorios con diferentes aplicaciones y hacer todo tipo de integraciones, ofreciendo un ambiente y una plataforma para realizar varias etapas. (Amazon Web Services, n.d.)

5.9. Virtualización

Son herramientas que posibilitan la prueba de software complejo en entornos controlados y aislados. La virtualización permite crear esos entornos controlados con las mismas características y requisitos que el sistema donde el software será instalado. (Arsys, 2023)

5.9.1. Vagrant

Es una herramienta para construir y gestionar entornos de máquinas virtuales. Funciona sobre Linux, FreeBSD, macOS, y Microsoft Windows. Su función básica es reemplazar el interfaz (tanto gráfico como de texto) de estas plataformas, proporcionando un interfaz de texto, programable y homogéneo que permite preparar las máquinas, levantarlas, configurarlas, etc. (Arsys, 2023)

El Vagrantfile es un archivo de configuración utilizado por Vagrant para definir y configurar máquinas virtuales y entornos de desarrollo. Este archivo se encuentra en el directorio raíz

de tu proyecto Vagrant y contiene instrucciones sobre cómo debería ser configurada la máquina virtual cuando la inicias con Vagrant

5.9.2. Virtualbox

Es una aplicación que sirve para hacer máquinas virtuales con instalaciones de sistemas operativos. Esto quiere decir que, si tienes un ordenador con Windows, GNU/Linux o incluso macOS, puedes crear una máquina virtual con cualquier otro sistema operativo para utilizarlo dentro del que estés usando. (Arsys, 2023)

Vagrant utiliza VirtualBox como backend para crear y ejecutar máquinas virtuales según las especificaciones proporcionadas en el Vagrantfile.

5.10. Sistema operativo

Es el software que coordina y dirige todos los servicios y aplicaciones que utiliza el usuario en una computadora por eso es el más importante y fundamental. Se trata de programas que permiten y regulan los aspectos más básicos del sistema. Los sistemas operativos a utilizar son Linux y Windows. (Equipo editorial, Etecé, 2023)

5.10.1. Linux

Es un sistema operativo de libre distribución. Este sistema operativo posee un gran volumen de desarrolladores y es muy reconocido a nivel mundial. El uso más importante que se le suele dar a este sistema operativo es el de actuar como servidor tanto en pequeñas redes como en corporaciones, incluyendo servidores web, de correo electrónico, etc. (De Zúñiga, 2023)

5.10.2. Debían

Es un sistema operativo basado en Linux adecuado para un amplio rango de dispositivos incluyendo portátiles, ordenadores de escritorio y servidores. Proporcionamos una configuración predeterminada razonable para cada paquete, así como actualizaciones de seguridad con regularidad durante su ciclo de vida. (De Zúñiga, 2023)

5.11. Las herramientas de Infraestructura

Como código (IaC) permiten automatizar las tareas realizadas por los departamentos de IT de forma rápida y dinámica mediante el uso de lenguajes de programación de scripts, que permiten administrar, crear, manipular y distribuir múltiples recursos informáticos a gran escala dentro de una infraestructura de Cloud Computing."(Dominguez-Quintero, n.d.)

Las herramientas de Infraestructura como código Chef, Puppet y Ansible generan una representación virtual de toda la infraestructura física y escalable de una plataforma Cloud Computing y de Centro de Datos, facilitando que esta sea programable y dinámica.

5.11.1. Ansible

Es una herramienta de orquestación, implementación y gestión de configuración de TI de código abierto. Esto ayuda a automatizar el despliegue de aplicaciones, la gestión de la configuración, el aprovisionamiento en la nube, la actualización de estaciones de trabajo y servidores, entre otras muchas tareas. También puede abastecer máquinas virtuales, contenedores y la red, así como infraestructuras de informática en la nube completas." (Manjaly, 2023)

Flujo de trabajo con Ansible:

- 1) Cree un playbook y un inventory en la máquina local.
- 2) Cree SSH para los nodos de destino.
- 3) Ansible Server recopila los datos de los nodos de destino para obtener la indicación de los nodos de destino.
- 4) Los playbooks se envían a los nodos.
- 5) Los playbooks se ejecutan en los nodos.

Términos importantes utilizados en Ansible:

- Servidor Ansible: La máquina donde está instalado Ansible y desde la cual se ejecutarán todas las tareas y playbooks.
- Task: Una tarea es una sección que consta de un único procedimiento a completar.
- Rol: Una forma de organizar tareas y archivos relacionados para luego llamarlos en un playbook
- Inventory: Archivo que contiene datos sobre los servidores del cliente ansible.

- Playbooks: Archivo YAML con una extensión yml o yaml, el cual contiene por lo menos un play y sirve para definir el estado deseado de un sistema.

5.12. Herramientas de desarrollo

Una herramienta de desarrollo se refiere a cualquier software o conjunto de programas que los desarrolladores utilizan para facilitar y agilizar el proceso de construcción, diseño, prueba y mantenimiento de sitios web o aplicaciones web. Estas herramientas son esenciales para simplificar tareas repetitivas, mejorar la productividad y garantizar la calidad del código y la funcionalidad de las aplicaciones web.” (IBM, n.d.)

Existen dos áreas en las que se pueden dividir las tecnologías:

Lado del servidor: Aquí, querrás ver las tecnologías de backend que utilizas. Por ejemplo, deberás tener en cuenta la elección del alojamiento y el servidor web, el almacenamiento de valores clave y SQL, los marcos de aplicación y automatización que vayas a utilizar y, por supuesto, el lenguaje de programación.

Del lado del cliente: La elección de las herramientas de frontend será posiblemente más sencilla, especialmente en lo que respecta a los lenguajes que se utilizan. Mientras que HTML, CSS y JavaScript serán los protagonistas, la elección del marco de trabajo de JavaScript (y el marco de trabajo de automatización) necesitará alguna consideración.

5.13. Entornos de Desarrollo Integrado (IDE)

Son aplicaciones que proporcionan un entorno completo para el desarrollo, que incluye un editor de código, herramientas de depuración, administración de proyectos y a menudo integración con sistemas de control de versiones.” (AWS, n.d.)

Ejemplo:

5.13.1. Visual Studio Code

Es un editor de código optimizado que proporciona muchas facilidades tiene una buena integración con Git, cuenta con soporte para depuración de código, y dispone de un sinnúmero de extensiones, que básicamente te da la posibilidad de escribir y ejecutar

código en cualquier lenguaje de programación.” (Documentation for Visual Studio Code, 2021)

5.14. Base de Datos

Es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático. Normalmente, una base de datos está controlada por un sistema de gestión de bases de datos (DBMS).”(Modelado De Datos Con Amazon DocumentDB, n.d.)

5.14.1. PostgreSQL

Es una base de datos de código abierto que tiene una sólida reputación por su fiabilidad, flexibilidad y soporte de estándares técnicos abiertos. A diferencia de otros RDMBS (sistemas de gestión de bases de datos relacionales), PostgreSQL (soporta tipos de datos relacionales y no relacionales. Esto la convierte en una de las bases de datos relacionales más compatibles, estables y maduras disponibles actualmente. (What Is PostgreSQL?, 2023)

5.14.2. PgAdmin

Básicamente es una aplicación gráfica desarrollada con python que implementa las siguientes tecnologías: Flask para el Backend, Javascript/Jquery/Backbone para el FrontEnd y Bootstrap.” (Page, n.d.)

Ésta aplicación sirve para gestionar el gestor de BD de PostgreSQL, y es considerada una de las más completas y populares con licencia Open Source y está disponible en diferentes idiomas. Su interfaz gráfico soporta todas las características de PostgreSQL y facilita de gran manera la administración ya que nos permite desde hacer búsquedas SQL hasta desarrollar toda nuestra base de datos de forma muy fácil e intuitiva: directamente desde la interfaz gráfica.” (Page, n.d.)

5.15. Lenguaje de programación

5.15.1. PHP

Es el acrónimo recursivo de Hypertext Preprocessor, un lenguaje de programación de uso general que se utiliza, sobre todo, en el entorno del desarrollo web. Este lenguaje se utiliza generalmente para desarrollar el backend de una web, el lado del servidor. Aun así, tiene numerosas utilidades en frontend. Es por esto que es uno de los principales lenguajes de programación en el mundo de la programación web.” (López, 2023)

Beneficios de usar PHP:

- ✓ Tiene una comunidad muy grande de desarrolladores que están constantemente actualizando y mejorando el lenguaje.
- ✓ Es uno de los lenguajes más sencillos que puedan encontrar en el mundo de la programación.
- ✓ Puedes usarlo en cualquier tipo de servidor y distintos sistemas operativos virtuales.
- ✓ Soporta gran cantidad de bases de datos diferentes, por esta razón es ideal para la creación de web y aplicaciones basadas en bases de datos.
- ✓ PHP puede combinarse otros lenguajes de programación como HTML, CSS o JavaScript.

5.15.2. JavaScript

Es un lenguaje de alto nivel, dinámico e interpretado. Se define como orientado a objetos, basado en prototipos, imperativo y débilmente tipado. Junto al HTML y a CSS, es una de las tres tecnologías esenciales en la producción mundial webs: la mayoría de los sitios web la emplean y es compatible con todos los navegadores modernos sin necesidad de plugins. JavaScript posee una API para trabajar con texto, arrays, fechas y expresiones regulares. (Aprende Desarrollo Web | MDN, n.d.)

5.15.3. HTML

Lenguaje de Marcas de Hipertexto, del inglés HyperText Markup Language es el componente más básico de la Web. Define el significado y la estructura del contenido web. Además de HTML, generalmente se utilizan otras tecnologías para describir la apariencia/presentación de una página web (CSS) o la funcionalidad/comportamiento (JavaScript).”(HTML: Lenguaje De Etiquetas De Hipertexto | MDN, n.d.)

5.15.4. CSS

Abreviado en sus siglas en inglés, Cascading Style Sheets, que significa hojas de estilo en cascada, el CSS es una función que se agrega a HTML que proporciona tanto a los desarrolladores de sitios Web, así como a los usuarios, más control sobre cómo se muestran las páginas.”(CSS | MDN, 2023)

Con CSS, los diseñadores y los usuarios pueden crear hojas de estilo que definen cómo aparecen los diferentes elementos, como los encabezados y los enlaces. Estas hojas de estilo se pueden aplicar a cualquier página Web, y nos permite optimizarla para mejorar su posicionamiento.” (CSS | MDN, 2023)

5.16. Frameworks de Desarrollo Web

Son conjuntos de herramientas, bibliotecas y estándares que proporcionan una estructura para el desarrollo de aplicaciones web. Estos facilitan la creación rápida y eficiente de aplicaciones al ofrecer una base predefinida y patrones de diseño. En este caso se utilizó:

5.16.1. Laravel

Es un framework PHP y utiliza un lenguaje de scripting en lugar de ser un lenguaje de programación PHP estricto. Aunque los lenguajes de scripting y los de programación están relacionados, tienen varias diferencias notables, principalmente en la facilidad de uso y la velocidad de ejecución.” (Laravel - the PHP Framework for Web Artisans, n.d.)

Los lenguajes de scripting son lenguajes interpretados específicos de la plataforma. Por el contrario, los lenguajes de programación son lenguajes compilados e independientes de la plataforma. Al ser específicos de la plataforma, los lenguajes de scripting dependen del compilador de la aplicación en la que se trabaja.

Ventajas:

- a) Productividad mejorada: Laravel ofrece una amplia gama de herramientas y características que permiten a los desarrolladores ser más eficientes en su trabajo. Su sintaxis limpia y clara facilita la escritura de código rápido y de calidad.
- b) Arquitectura MVC: Laravel sigue el patrón de diseño Modelo-Vista-Controlador (MVC), lo que proporciona una estructura organizada para el desarrollo de

- aplicaciones web. Esto permite una mayor separación de preocupaciones y facilita el mantenimiento del código en el tiempo.
- c) Abundantes bibliotecas y paquetes: Laravel cuenta con una gran cantidad de bibliotecas y paquetes listos para usar, lo que acelera el proceso de desarrollo al no tener que reinventar la rueda. Estas bibliotecas abarcan desde la autenticación y el manejo de sesiones hasta el procesamiento de imágenes y el envío de correos electrónicos.
 - d) Comunidad activa: Laravel cuenta con una comunidad de desarrolladores muy activa que ofrece soporte, tutoriales, ejemplos de código y actualizaciones frecuentes de la tecnología. Esto facilita el aprendizaje y permite estar al tanto de las últimas tendencias y mejores prácticas.
 - e) Base de datos ORM: Laravel incluye Eloquent, un ORM (Mapeo objeto-relacional) poderoso y fácil de usar.

5.16.1. Composer

Es una herramienta de gestión de dependencias en PHP. Permite gestionar, para cada proyecto, la lista de módulos y bibliotecas necesarios para su funcionamiento, así como sus versiones. Sirve para descargar y gestionar las librerías y dependencias de un proyecto PHP de manera automatizada.”(Qué Es Composer Y Cómo Usarlo, n.d.)

Cuando se desarrolla una aplicación o sitio web en PHP, a menudo se utilizan diferentes librerías o frameworks de terceros para añadir funcionalidades adicionales. Composer permite gestionar estas dependencias de manera sencilla, descargándolas y manteniéndolas actualizadas de manera automática.

5.17. Servidores Web

Son un componente de los servidores que tienen como principal función almacenar, en web hosting, todos los archivos propios de una página web (imágenes, textos, videos, etc.) y transmitirlos a los usuarios a través de los navegadores mediante el protocolo HTTP (Hypertext Transfer Protocol).

Se realizan configuraciones específicas en el servidor web para asegurar que la aplicación se ejecute correctamente. Esto puede incluir la configuración de variables de entorno, ajustes de seguridad, etc.

5.17.1. Apache

Es un servidor web que se encarga de almacenar, procesar y servir las páginas web a los usuarios de las mismas. Se distribuye bajo una licencia de código abierto, lo que quiere decir que es gratuito y fácilmente adaptable, personalizable y reutilizable. No es un servidor físico, sino un software que se ejecuta en un servidor HTTP.

La función de Apache es la de servir de enlace entre el servidor y los navegadores de los usuarios del sitio web, como Firefox, Google Chrome o Safari, mientras intercambia información entre ellos. (Expósito, 2023)

5.18. La computación de alto rendimiento (HPC)

La HPC permite a los usuarios procesar grandes cantidades de datos más rápido que una computadora estándar, lo que genera conocimientos más rápidos y ofrece a las organizaciones la capacidad de mantenerse por delante de la competencia. (IBM, n.d.)

5.18.1. Grafana

Es una solución de código abierto que se utiliza principalmente para monitorizar infraestructuras IT y aplicaciones. Permite visualizar métricas, analizar y comprender la información de forma clara y sencilla, gracias al uso de herramientas de visualización gráficas que elaboran informes en tiempo real a partir de los datos recolectados. (Martínez, 2023)

5.19. Direcciones IP

Una dirección IP (Protocolo de Internet) es una etiqueta numérica única asignada a cada dispositivo que participa en una red de computadoras que utiliza el protocolo de Internet para la comunicación. Estos identificadores numéricos se utilizan para ubicar y identificar de manera única a cada dispositivo en una red. (Qué Es Una Dirección IP: Definición Y Explicación, 2023)

Las direcciones IP se clasifican en:

5.19.1. Direcciones IP públicas.

Son visibles en todo Internet. Un host con una IP pública es accesible (visible) desde cualquier otro host conectado a Internet. Para conectarse a Internet es necesario tener una dirección IP pública. (Qué Es Una Dirección IP: Definición Y Explicación, 2023)

5.19.2. Direcciones IP privadas (reservadas).

Son visibles únicamente por otros hosts de su propia red o de otras redes privadas interconectadas por routers. Se utilizan en las empresas para los puestos de trabajo. Los hosts con direcciones IP privadas pueden salir a Internet por medio de un Router (o proxy) que tenga una IP pública. Sin embargo, desde Internet no se puede acceder a ordenadores con direcciones IP privadas. (Qué Es Una Dirección IP: Definición Y Explicación, 2023)

A su vez, las direcciones IP pueden ser:

5.19.3. Direcciones IP estáticas (fijas).

Un host que se conecte a la red con dirección IP estática siempre lo hará con una misma IP. Las direcciones IP públicas estáticas son las que utilizan los servidores de Internet con objeto de que estén siempre localizables por los usuarios de Internet. Estas direcciones hay que contratarlas. (Qué Es Una Dirección IP: Definición Y Explicación, 2023)

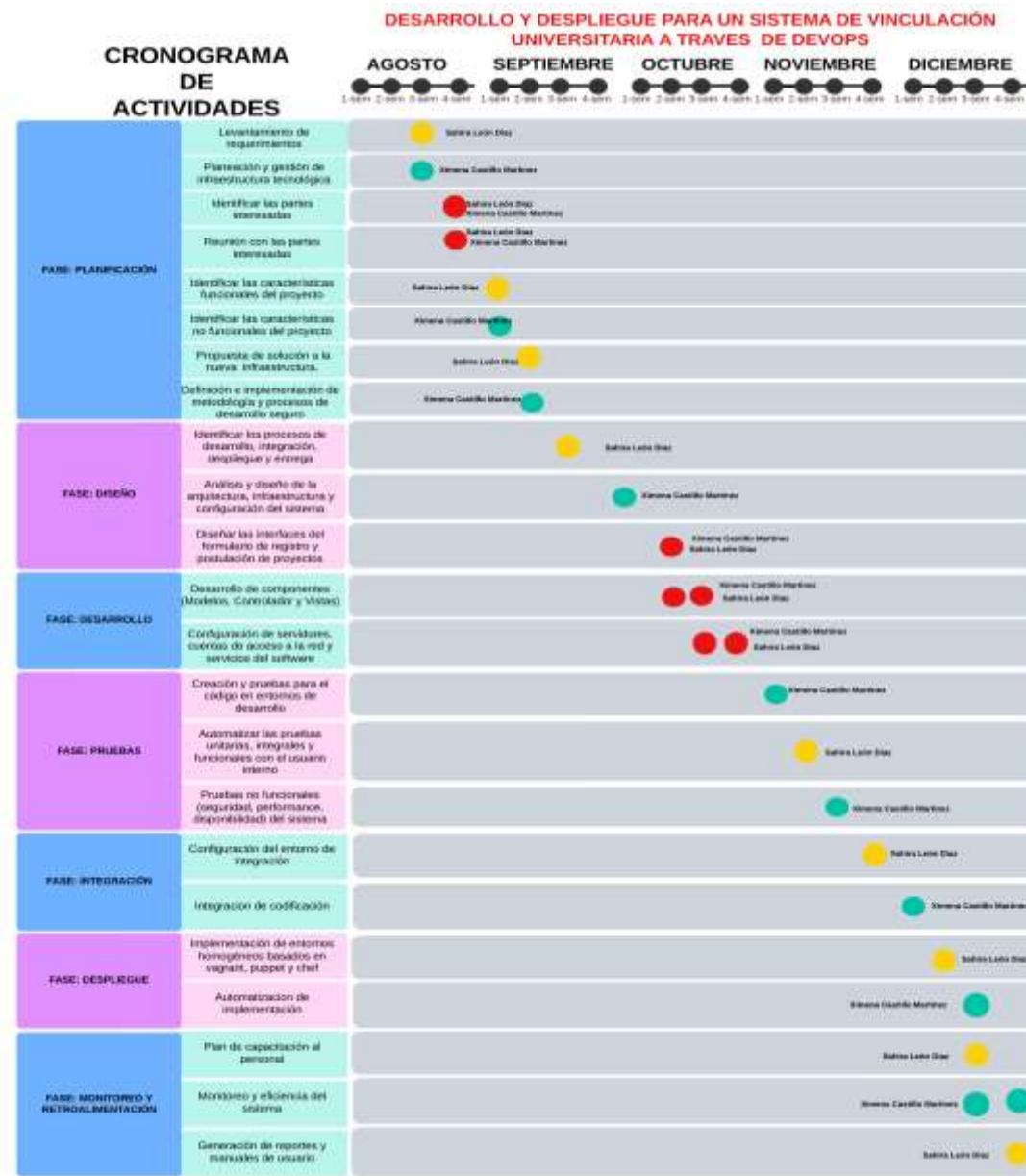
5.19.4. Direcciones IP dinámicas.

Un host que se conecte a la red mediante dirección IP dinámica, cada vez lo hará con una dirección IP distinta. Las direcciones IP públicas dinámicas son las que se utilizan en las conexiones a Internet mediante un módem. Los proveedores de Internet utilizan direcciones IP dinámicas debido a que tienen más clientes que direcciones IP (es muy improbable que todos se conecten a la vez). (Qué Es Una Dirección IP: Definición Y Explicación, 2023)

6. Cronograma de actividades

Este cronograma de actividades es una herramienta que permite planificar y visualizar las tareas durante el desarrollo del proyecto. En este caso, el proyecto se divide en semanas y se plasman las tareas a desarrollar cada una.

Figura 1 Cronograma de actividades



Nota: Actividades realizadas durante este lapso de tiempo. (Ximena, 2023)

7. Procedimiento y descripción de las actividades realizadas

7.1. Planeación y gestión de infraestructura tecnológica

El Sistema de Vinculación Universitaria (SVU) debe estar disponible en un sistema web, ya que esto garantiza su accesibilidad para todos los usuarios. Esta modalidad permite que los usuarios puedan utilizar el sistema desde cualquier dispositivo a través de un navegador web. Para lograr esto, se propone la utilización de lenguajes de programación que sean compatibles y ejecutables en cualquier equipo.

En cuanto a DevOps (Desarrollo y Operaciones), nos permitirá integrar las tareas de desarrollo y operaciones mediante el uso de diversas herramientas. Se utilizarán herramientas como Vagrant, Virtualbox, PHP, Apache, PostgreSQL, Laravel, GitLab entre otras, para facilitar y agilizar las operaciones de desarrollo y gestión de infraestructura del sistema web.

Se tomaron en cuenta los siguientes puntos:

Evaluación de Requisitos:

- Comprender los requisitos específicos del Sistema de Vinculación Universitaria en términos de capacidad, rendimiento, escalabilidad y seguridad.
- Colaborar con los equipos de desarrollo y operaciones para recopilar requisitos técnicos y no funcionales.

Selección de Tecnologías:

- Evaluar y seleccionar las tecnologías de infraestructura que mejor se adapten a las necesidades del sistema.
- Considerar aspectos como servidores físicos, máquinas virtuales, contenedores y servicios en la nube según la escala y los requerimientos.

Arquitectura de Infraestructura:

- Diseñar la arquitectura de la infraestructura teniendo en cuenta la separación de entornos (desarrollo, prueba, producción) y la escalabilidad horizontal y vertical.
- Definir prácticas de seguridad, redundancia y respaldo para garantizar la disponibilidad y la integridad del sistema.

Implementación de la Infraestructura:

- Utilizar herramientas de automatización de infraestructura en este caso Ansible, para implementar y gestionar la infraestructura de manera eficiente y reproducible.
- Automatizar la creación de entornos y la configuración de servicios.

Seguridad de la Infraestructura:

- Implementar prácticas sólidas de seguridad, como firewalls, configuración segura de servidores y monitoreo constante de amenazas.
- Garantizar el cifrado de datos en tránsito y en reposo.

Figura 2 Herramientas DevOps



Nota: Estas herramientas se utilizaron para el desarrollo del sistema. (Ximena, 2023)

7.2. Identificar las partes interesadas

Durante este proceso el Sistema de Vinculación Universitaria involucrará una amplia gama de partes interesadas incluyendo:

Coordinaciones de Área: Representantes de las diferentes áreas académicas del instituto que desempeñan un papel clave en la definición de los requisitos y la validación del sistema en función de las necesidades específicas de cada área.

Representantes Docentes: Los docentes o investigadores designados serán quienes podrán publicar alguna vacante disponible a las cuales los estudiantes externos pueden postularse para realizar sus prácticas profesionales, tesis, estadías o servicio social.

Dirección de Formación Académica: Tiene como función principal supervisar y controlar el progreso de cada estudiante externo a través del sistema, mantener un registro completo de su historial académico y gestionar la emisión de cartas de liberación.

Área de Vinculación: Los profesionales encargados de gestionar las relaciones con la comunidad y las instituciones externas, quienes contribuirán a asegurar que el sistema facilite eficazmente la vinculación entre la universidad y su entorno.

Área de Administración General de Cómputo: El equipo de TI responsable de proporcionar la infraestructura tecnológica y el soporte técnico necesarios para el sistema.

Instituciones Universitarias: Los estudiantes de las universidades podrán realizar su servicio social, prácticas profesionales, tesis y estadías, lo cual permitirá de una manera accesible, electrónica y rápida este proceso.

Figura 3 Partes interesadas del Sistema de Vinculación Universitaria



Nota: En esta imagen se muestra la jerarquía de cómo están las partes interesadas del sistema. (Agustín, 2023)

7.3. Reunión con las partes interesadas

Las reuniones con las partes interesadas se han llevado a cabo con el propósito de crear un espacio adecuado para la coordinación, colaboración, comunicación y toma de decisiones en relación con el proyecto. En cada una de estas reuniones, se resalta la importancia de aclarar cualquier duda existente, obtener comentarios y aportes, así como también se dedica tiempo para revisar los avances y desarrollos recientes. Esto se hace con el objetivo de garantizar que todos los involucrados tengan un entendimiento claro y compartido de los objetivos del proyecto.

Esto ha servido como referencia posterior y ha ayudado a garantizar que los compromisos se cumplan. Después de la reunión, he colaborado estrechamente con el equipo del proyecto para implementar las acciones acordadas y comunicar los resultados a las partes interesadas. La retroalimentación continua ha sido

fundamental para mantener a las partes interesadas informadas y comprometidas en todo momento.

Figura 4 Reunión con las partes interesadas



Nota: En esta foto se visualiza que se tenían reuniones con las partes interesadas sobre el sistema. (Ana María, 2023)

7.4. Identificar las características no funcionales de proyecto

Esta actividad se desglosan las características no funcionales identificadas para contribuir a garantizar que el sistema satisfaga las necesidades del usuario.

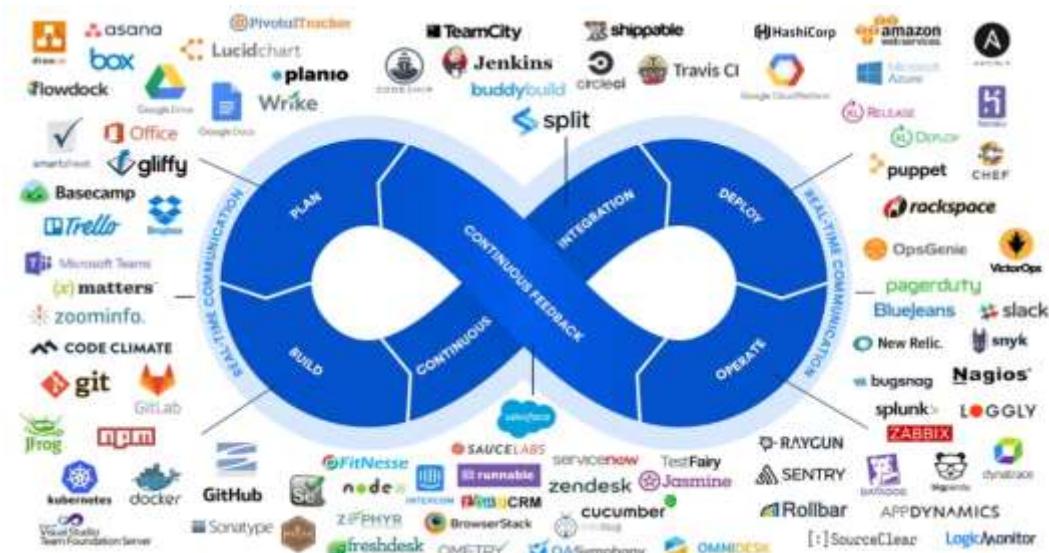
- ✓ La capacidad de permitir a los usuarios tanto internos como externos que obtengan acceso a la aplicación sin que sea necesario descargar software alguno.
- ✓ La capacidad para interactuar con la aplicación o Sistema en cada procedimiento que se genere información relativa a este.
- ✓ La capacidad para incorporar enrutamiento y notificaciones semi automatizadas con base a reglas de publicación o registro de proyectos.
- ✓ La capacidad de poder visualizar los proyectos una vez registrados en un formato claro y conciso.
- ✓ La capacidad de permitir a los usuarios tanto internos como externos la visualización de los proyectos publicados de manera que los alumnos/estudiantes interesados puedan postularse.

- ✓ La capacidad de mostrar de manera clara a los Alumnos/estudiantes la información de cada proyecto.
- ✓ La capacidad de poder contar con más de un rol en el sistema para la asignación de privilegios dentro del sistema.
- ✓ La capacidad de poder generar reportes donde se pueda mostrar el número de proyectos creados, la cantidad de postulantes por periodo, el proyecto con mayor número de postulantes, el proyecto con más rechazos, entre otros.

7.5 Definición e implementación de metodología y procesos de desarrollo seguro

Esta actividad se realizó y se seguirá implementando la metodología DevOps para la cual tenga el despliegue exitoso del Sistema de Vinculación Universitaria (SVU). Esto implica la identificación y adopción de estrategias que garanticen tanto la calidad del software como la seguridad de los datos y la infraestructura en todo el ciclo de vida del proyecto. A través de la integración de herramientas y prácticas de DevOps, se busca automatizar tareas, acelerar la entrega de software, y asegurar que los cambios se implementen de manera segura y confiable. Es por ello que tiene como objetivo final optimizar la eficiencia operativa y garantizar la satisfacción de los usuarios finales para SVU seguro y de alta calidad.

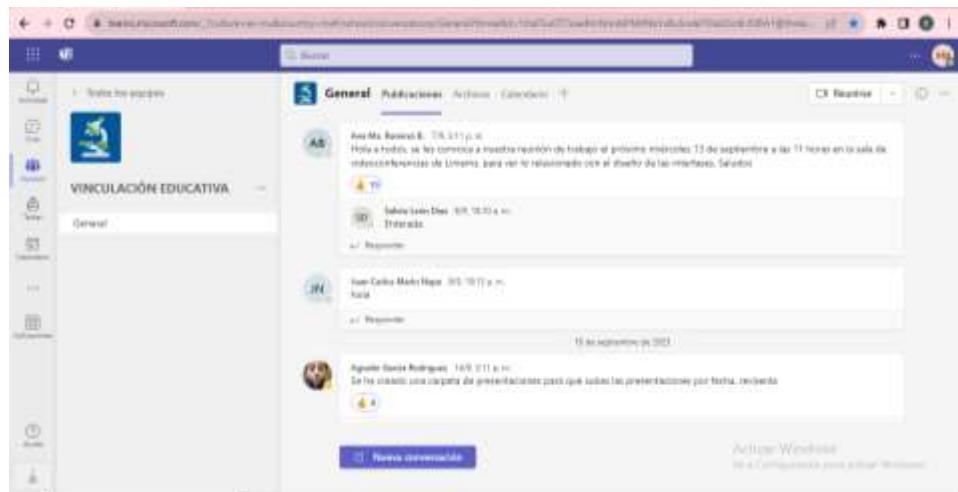
Figura 5 DevOps. La cultura de buenas prácticas



Nota: Conjunto de prácticas, herramientas y filosofía cultural. (Macchia, 2023).

En esta metodología utilizamos la plataforma Microsoft Teams, en la cual es nuestro medio de colaboración, así como también facilitando la comunicación efectiva. Cuenta con la capacidad de realizar publicaciones, aquí podemos compartir las actualizaciones, anuncios u otra información relevante al proyecto que nos sirve para mantenernos informados.

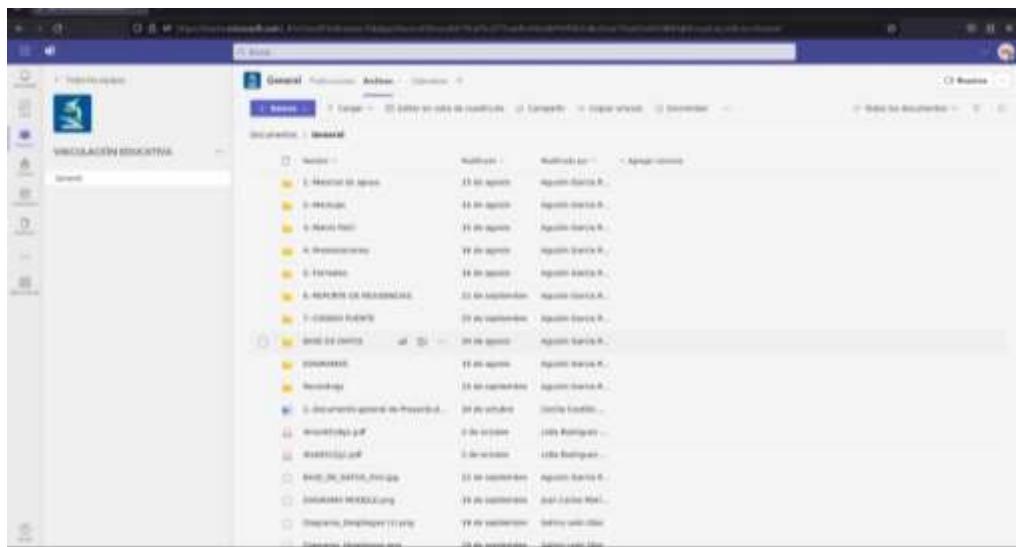
Figura 6 Muro principal de Microsoft Teams



Nota: Visualización de la interfaz de Teams en donde se hacen las publicaciones. (Ximena, 2023)

Además, tiene la función de subir, editar y compartir documentos, presentaciones y otros recursos en tiempo real, fomentando la colaboración más eficiente y eliminando la necesidad de enviar múltiples versiones. Todo lo trabajamos dentro de carpetas para tener un orden.

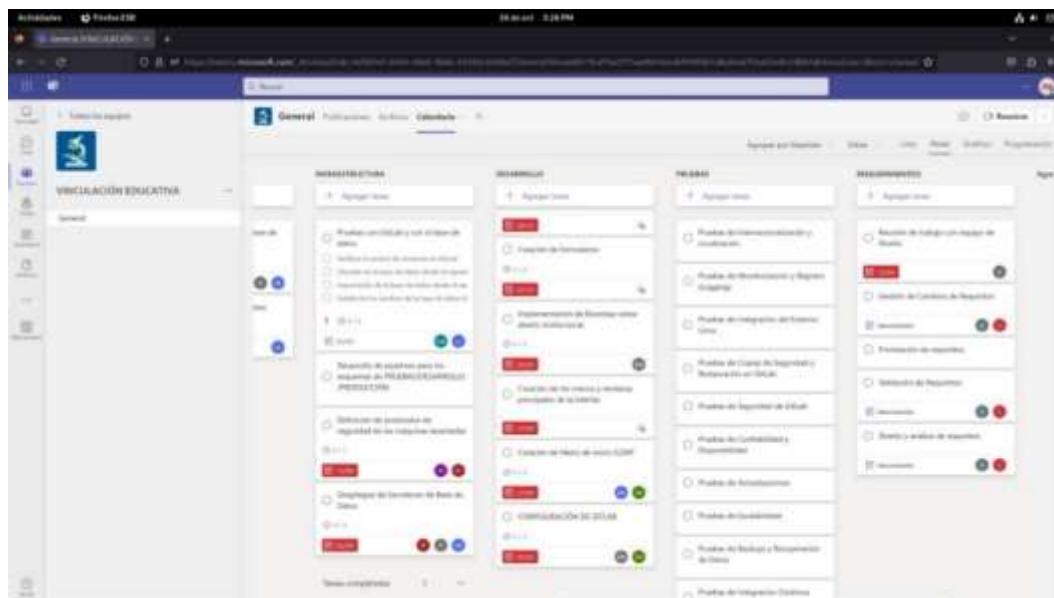
Figura 7 Clasificación de carpetas del proyecto en Microsoft Teams



Nota: Se desglosan las carpetas en las cuales se clasifica toda la información con la que se trabaja el sistema (Ximena, 2023)

También está integrado un calendario el cual proporciona una visión más clara de los eventos y plazos importantes. Aquí programamos las reuniones, asignamos tareas y coordinamos todo lo del proyecto. Esto mejora la planificación y la organización, manteniendo a todos en sintonía con los plazos y las actividades.

Figura 8 Calendario de actividades en Microsoft Teams



Nota: En esta interfaz se muestra las actividades asignadas a realizar. (Ximena, 2023)

7.6 Análisis y diseño de la arquitectura, infraestructura y configuración del sistema.

Como primera instancia para el desarrollo del Sistema de Vinculación Educativa(SVU), se está trabajando con el sistema operativo Debian12. Para la infraestructura se está utilizando Vagrant que es una herramienta que nos permite manejar y configurar máquinas virtuales en este caso Oracle VM VirtualBox manteniendo un mismo entorno de trabajo que pueden ser reproducidos y compartidos de una forma muy fácil. Esto también facilita el proceso de desarrollo de software cuando este es montado por más de una persona, debido a que evita los problemas de compatibilidad entre sistemas y brinda la posibilidad de compartir sus archivos, donde se centralizan las configuraciones creadas.

En las siguientes capturas se muestra como se está haciendo la automatización con Vagrant para descargar los softwares a instalar y configurar (PHP, Apache, PostgreSQL, Composer, Laravel y GitLab). Cada línea de código está detalladamente descrita, proporcionando un entendimiento más completo de las acciones llevadas a cabo.

Figura 9 Codificación del Vagranfile

```

$ pg_ctl -D /var/lib/pgsql/10/data -l /var/log/pgsql/10/postmaster.log start
[1] 11888 pts/0 0:00 pg_ctl -D /var/lib/pgsql/10/data -l /var/log/pgsql/10/postmaster.log start
pg_ctl: server starting
[1]+ 11888 pts/0 0:00 pg_ctl -D /var/lib/pgsql/10/data -l /var/log/pgsql/10/postmaster.log start
pg_ctl: server started
[1]+ 11888 pts/0 0:00 pg_ctl -D /var/lib/pgsql/10/data -l /var/log/pgsql/10/postmaster.log start

* PostgreSQL 10.0-rc1 on x86_64-pc-linux-gnu, compiled by gcc (Ubuntu 5.4.0-6ubuntu1~16.04.4) 5.4.0, 2016-06-09 12:50:52 UTC, on 2016-06-10 12:50:52 UTC
* PostgreSQL is running as process 11888.
* PostgreSQL's config file is /var/lib/pgsql/10/data/postgresql.conf
* PostgreSQL's shared memory segment identifier is 11888.
* PostgreSQL's port number is 5432.
* PostgreSQL's PID file is /var/run/postgresql/10.pid.
* PostgreSQL's log file is /var/log/pgsql/10/postmaster.log.

$ psql -U postgres
psql: could not connect to server: Connection refused
Is the server running locally and accepting
connections on port 5432?

```

*Nota: En esta captura se codifica el código para que se hagan las instalaciones y configuraciones.
(Ximena, 2023)*

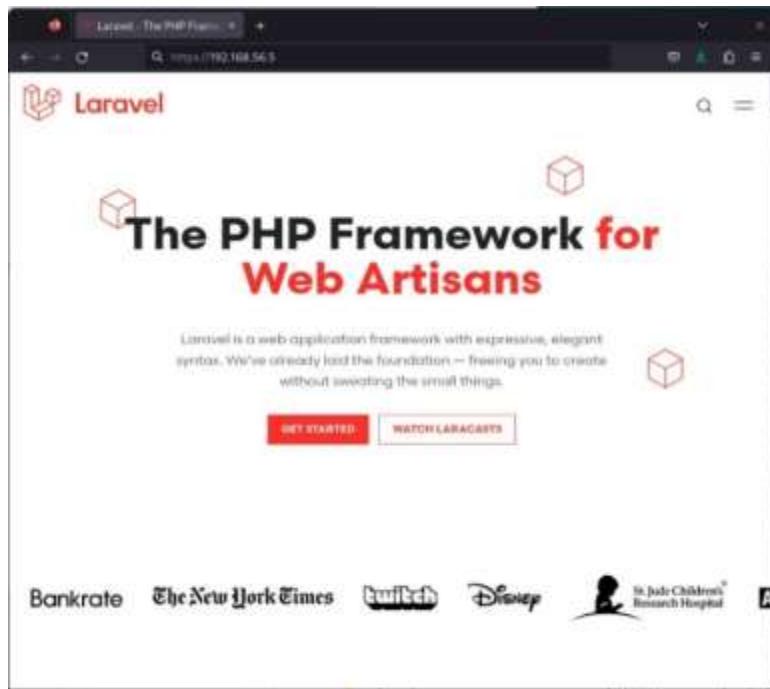
Después de generar el código que incluye las instalaciones y configuraciones correspondientes, procedemos a abrir la terminal y nos dirigimos a la ubicación de nuestro archivo Vagrantfile. Posteriormente, ejecutamos el comando “vagrant up” para inicializar la máquina virtual, esperando a que concluya el proceso para obtener los resultados deseados, que consisten en las instalaciones ya configuradas. Es importante destacar que, esta automatización es rápida y efectiva.

Figura 10 Inicio de la máquina virtual

```
alexander@debian: ~[Laravel]
alexander@debian: ~[Laravel]
alexander@debian: ~[Laravel] vagrant up
Bringing machine 'Config' up with 'virtualbox' provider...
*** Config: Importing base box 'genesis/Ubuntu2204'...
*** Config: Matching MAC address for NAT networking...
*** Config: Checking if box 'genesis/Ubuntu2204' version '4.2.18' is up to date...
*** Config: A newer version of the box 'genesis/Ubuntu2204' for provider 'virtualbox' is
available! You currently have version '4.2.18'. The latest is version
*** Config: 4.2.19. Run 'vagrant box update' to update.
*** Config: Setting the name of the VM: Laravel_Config_1784914481757_74838
*** Config: Clearing any previously set network interfaces...
*** Config: Preparing network interfaces based on configuration...
Config: Adapter 1: nat
Config: Adapter 2: hostonly
*** Config: Forwarding ports...
Config: 22 (guest) -> 2222 (host) (adapter 1)
*** Config: Running 'pre-boot' VM customizations...
*** Config: Booting VM...
*** Config: Waiting for machine to boot. This may take a few minutes...
Config: SSH address: 127.0.0.1:2222
Config: SSH username: vagrant
Config: SSH auth method: private key
Config: Warning: Connection reset. Retrying...
Config: Warning: Remote connection disconnect. Retrying...
Config: Warning: Connection reset. Retrying...
Config: Warning: Remote connection disconnect. Retrying...
Config:
Config: Vagrant insecure key detected. Vagrant will automatically replace
Config: this with a newly generated keypair for better security.
Config:
Config: Inserting generated public key within guest...
Config: Removing insecure key from the guest if it's present...
Config: Key inserted! Disconnecting and reconnecting using new SSH key...
*** Config: Machine booted and ready!
but different exports about parallelized hostadditions version.
```

Nota: En esta captura se inició la máquina virtual con el vagranfile para las instalaciones y configuraciones solicitadas. (Ximena, 2023)

Figura 11 Interfaz de Laravel instalado mediante el vagrantfile



Nota: Captura de la interfaz después de que se ejecutara el Vagranfile. (Ximena, 2023)

7.7 Diseño de interfaces de los módulos registro y postulación de proyecto

El diseño de interfaces es esencial para garantizar que los usuarios puedan navegar de manera intuitiva y eficiente a través de este módulo.

Para realizar estas interfaces primero fue necesario primero trabajar en las siguientes actividades:

- ✓ **Diagramas de actividades:**

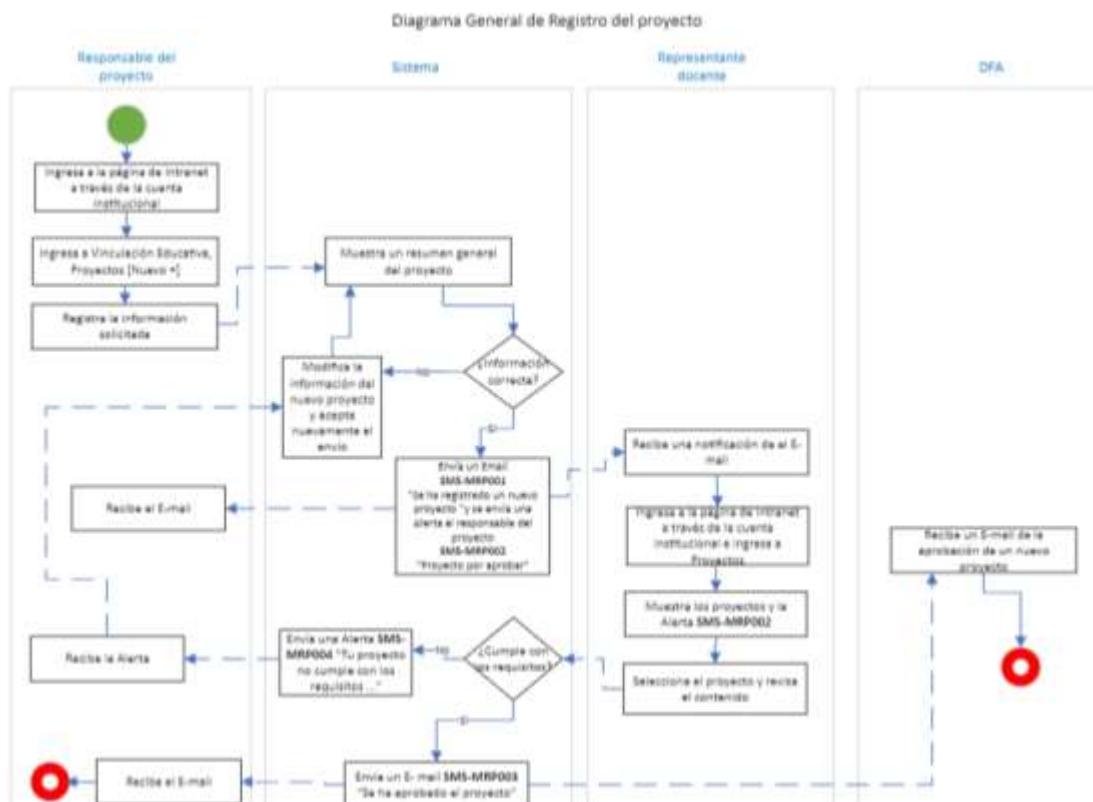
Los diagramas de actividades son herramientas visuales que posibilitan la modelación y visualización de procesos y flujos de trabajo dentro de este sistema. Su función principal es representar las distintas actividades presentes en un proceso y establecer cómo estas se relacionan entre sí. Como parte integral de nuestro enfoque, se han creado diagramas de

actividades específicos para cada módulo del sistema contando con cinco módulos y submódulos, proporcionando una representación detallada y clara de las operaciones y su interconexión en cada área funcional.

Acciones realizadas:

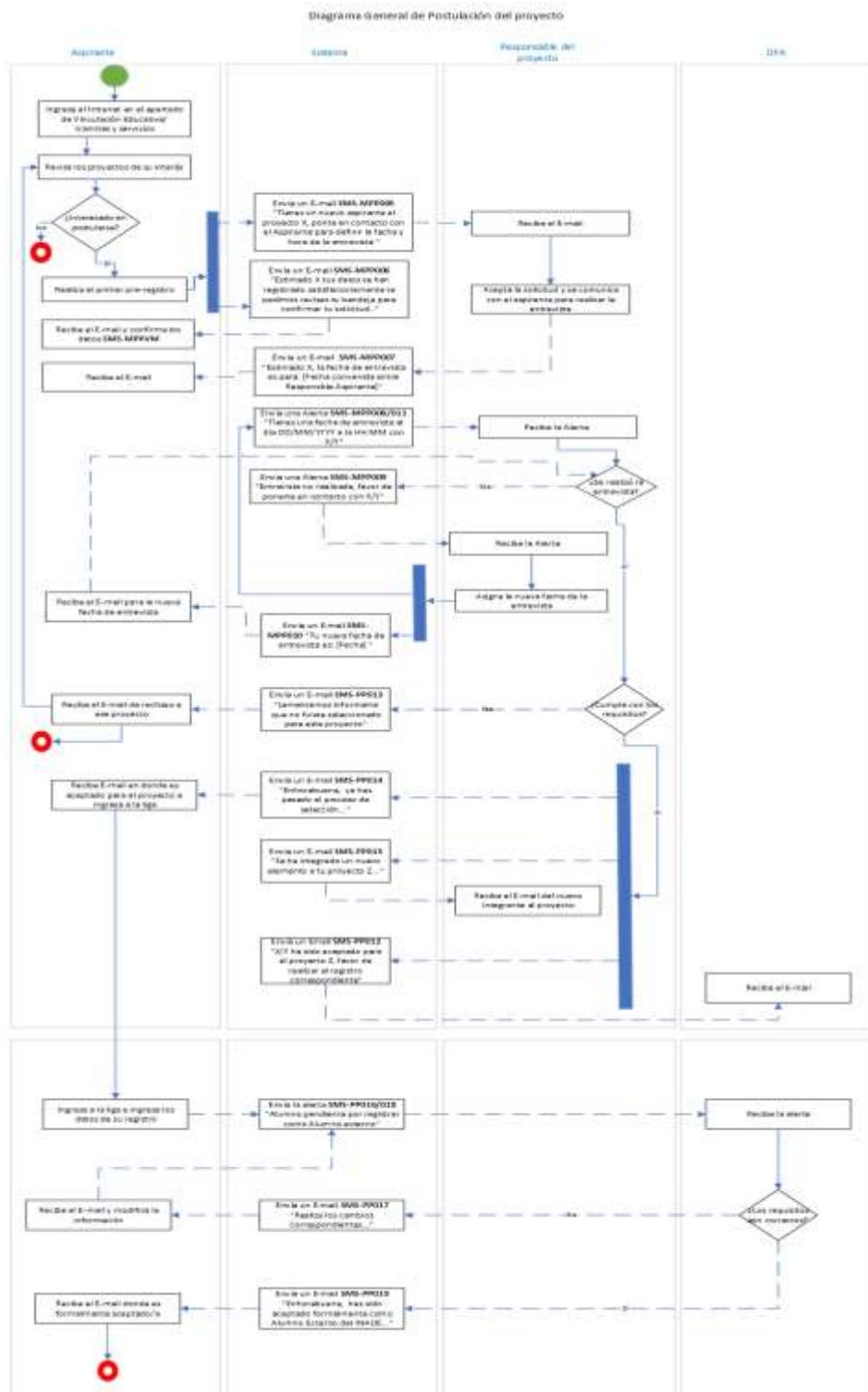
- Creación de Identificación y modelado de las actividades clave y sus interrelaciones, brindando una representación visual detallada del comportamiento operativo del Sistema de Vinculación Universitaria.
- Definición de flujos de trabajo específicos, destacando la secuencia lógica de tareas y decisiones que caracterizan cada actividad.
- Utilización de notaciones gráficas estándar para mejorar la comprensión y facilitar la comunicación efectiva entre los miembros del equipo de desarrollo y los stakeholders.

Figura 12 Diagrama General de Actividades de Registro de Proyecto



Nota: Diagrama realizando las actividades que va hacer tanto el responsable, sistema, responsable docente y la DFA. (Ximena, 2023)

Figura 13 Diagrama General de Actividades de Postulación del Proyecto



Nota: Diagrama realizando las actividades que va hacer tanto el aspirante, el sistema, responsable docente y la DFA. (Ximena, 2023)

✓ **Diagramas de estado:**

Los diagramas de estado se emplean para ilustrar el comportamiento de un sistema mediante la representación de sus distintos estados y las transiciones que ocurren entre ellos. En consecuencia, se han desarrollado diagramas de estado específicos para abordar esta dinámica, brindando una representación visual detallada de cómo el sistema responde y evoluciona en función de las condiciones cambiantes.

Acciones realizadas:

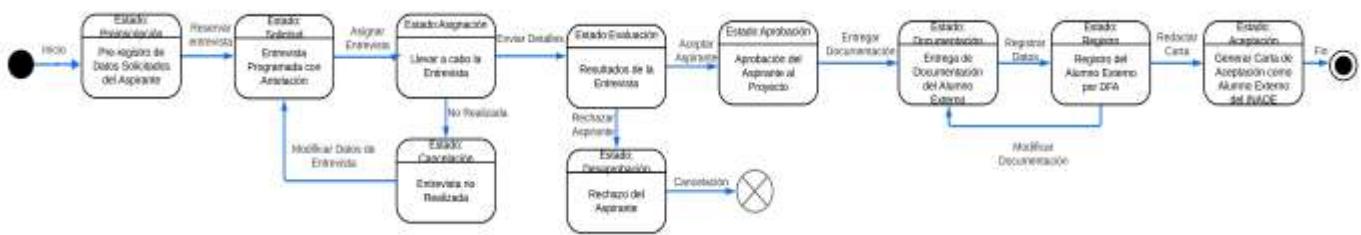
- Identificación y modelado de los estados fundamentales que caracterizan las diversas fases y condiciones dentro del sistema.
- Establecimiento claro de las transiciones entre los estados, permitiendo una representación visual coherente del comportamiento dinámico del Sistema de Vinculación Universitaria.
- Integración de notaciones gráficas para mejorar la interpretación de los diagramas, facilitando su utilidad en la toma de decisiones y comunicación efectiva.

Figura 14 Diagrama General de Estado de Registro del Proyecto



Nota: Diagrama que muestra los estados en los que se va a encontrar el proceso. (Ximena,2023)

Figura 15 Diagrama General de Estado de Postulación del Proyecto



Nota: Diagrama que muestra los estados en los que se va a encontrar el proceso. (Ximena,2023)

✓ Centro de mensajes:

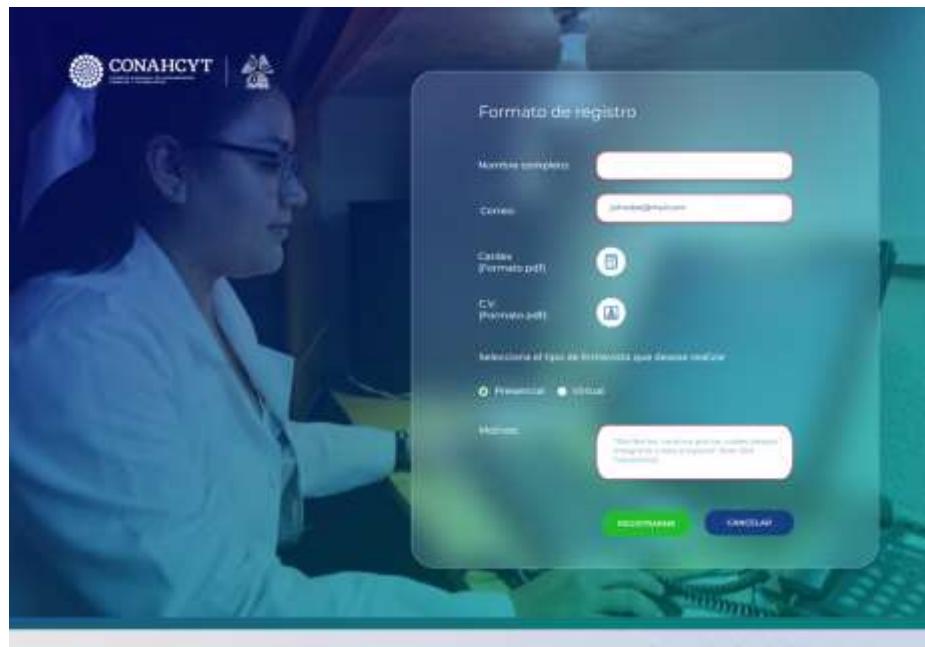
El Centro de Mensajes se erige como un componente fundamental dentro de las interfaces, ya que simplifica la comunicación entre usuarios y suministra notificaciones críticas. En esta función vital, hemos categorizado los mensajes en dos tipos distintos: correo electrónico (Email) y alertas. Esta clasificación permite una gestión eficiente de la información, garantizando que los usuarios tengan acceso a las comunicaciones relevantes de manera organizada y oportuna.

Figura 16 Centro de mensajes E-mail y Alertas

Nota: Excel en donde se realizaron los mensajes y alertas que aparecerán en el sistema.

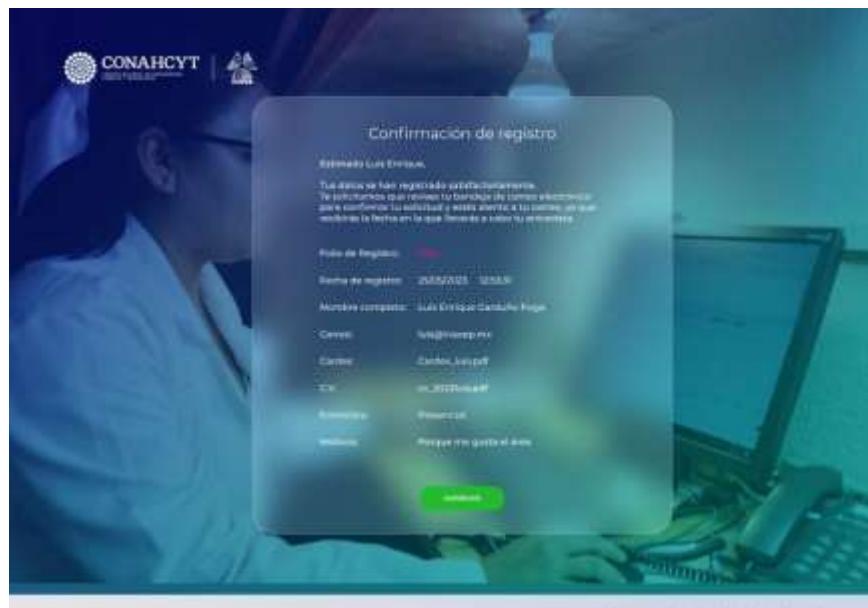
Después de elaborar todas estas actividades se realizó unas pre-interfaces de todo esto proceso como solo un ejemplo ya que es bastante extenso, pero se está haciendo cada una detalladamente con su respectiva especificación.

Figura 17 Pre-interfaz visual del primer pre-registro del aspirante.



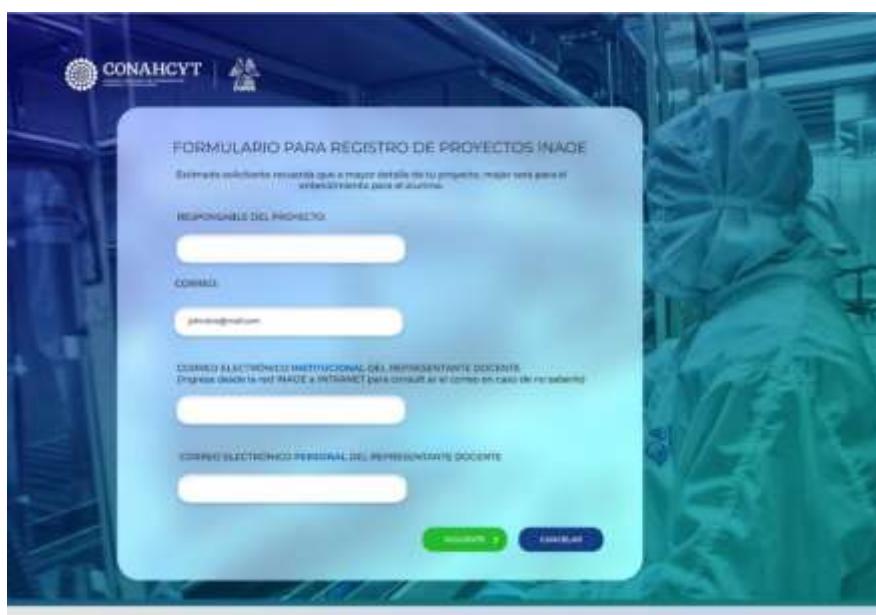
Nota: Se realizó esta pre-interfaz para la visualización de cómo se podría ver el sistema. (Ximena, 2023)

Figura 18 Pre-interfaz de la confirmación del primer pre-registro del aspirante



Nota: Se realizó esta pre-interfaz para la visualización de cómo se podría ver el sistema con la confirmación de datos. (Ximena, 2023)

Figura 19 Pre-Interfaz del formulario para el registro del proyecto



Nota: Se realizó esta pre-interfaz para el registro del proyecto. (Ximena, 2023)

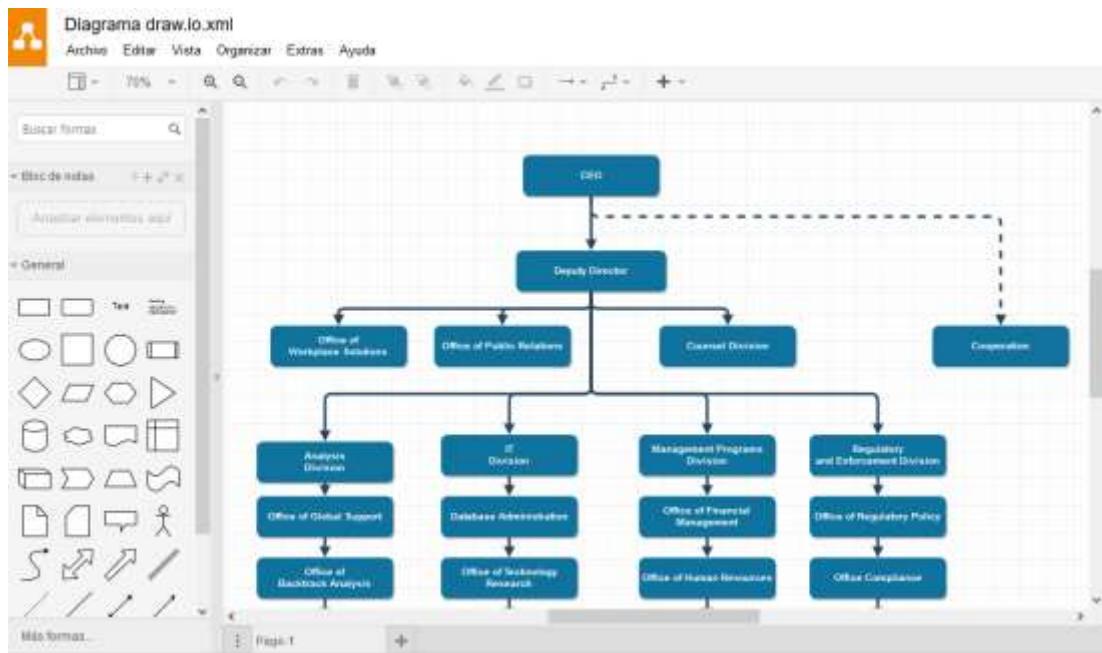
Figura 20 Especificaciones de las interfaces

Formulario de Pre-registro del aspirante									
Todas las interfaces de llenado de datos deberán tener el guardado automático									
Tabla matriz	Nombre del campo en la DB	Descripción	Ejemplo de uso	Campo obligatorio	Enriquecido	Type	placeholder	Restricciones	
Persona	nombre	Este campo permite al usuario ingresar su nombre o sus nombres, según sea el caso. Cadena de texto (String).	Melanie Lida	Si	(Input)	text	Ingrese su nombre	Largo: máxima: 50 caracteres.	
Persona	apaterno	En este campo, el usuario debe ingresar su apellido paterno, es decir, el primer apellido de su familia.	Rodríguez	Si	(Input)	text	Ingrese su apellido paterno	Largo: máxima: 50 caracteres.	
Persona	apaterno	Similar al campo anterior, aquí el usuario debe ingresar su apellido materno, que es el segundo apellido de su familia.	López	Si	(Input)	text	Ingrese su apellido materno	Largo: máxima: 50 caracteres.	
Persona	correo	En este campo, el usuario proporcionará su dirección de correo electrónico, que debe ser válida y utilizada para fines de contacto.	ldiazdiaz94@gmail.com	Si	(Input)	email	Ingrese su correo electrónico	Formato válido de correo electrónico	
Persona	telefono	En este campo, el usuario debe ingresar un número de teléfono que se utilizará para su posterior contacto.	7971308552	Si	(Input)	tel	Ingrese su número telefónico	Formato válido de número telefónico	
Persona	cv	Este campo permite al usuario cargar un archivo que contiene su Curriculum Vitae.	(El usuario selecciona y carga un archivo con su Curriculum Vitae)	Si	(Input)	file	Adjunte su currículum	Tipo de archivo permitido: PDF	
Persona	karden	Similar al campo anterior, aquí el usuario puede cargar un archivo que contiene su historial académico o	(El usuario selecciona y carga un archivo con su historial académico)	Si	(Input)	file	Adjunte su Karden	Tipo de archivo permitido:	

Nota: Tabla en excel para especificar los tipos de datos que debe llevar las interfaces. (Ximena, 2023)

También se utilizó una herramienta de diseño para crear prototipos y wireframes, para el diseño de las interfaces de los módulos, Draw.io que es una herramienta de diagramación en línea que permite a los usuarios crear una variedad de diagramas, como organigramas, diagramas de flujo, diagramas de red, entre otros.

Figura 21 Herramienta draw.io

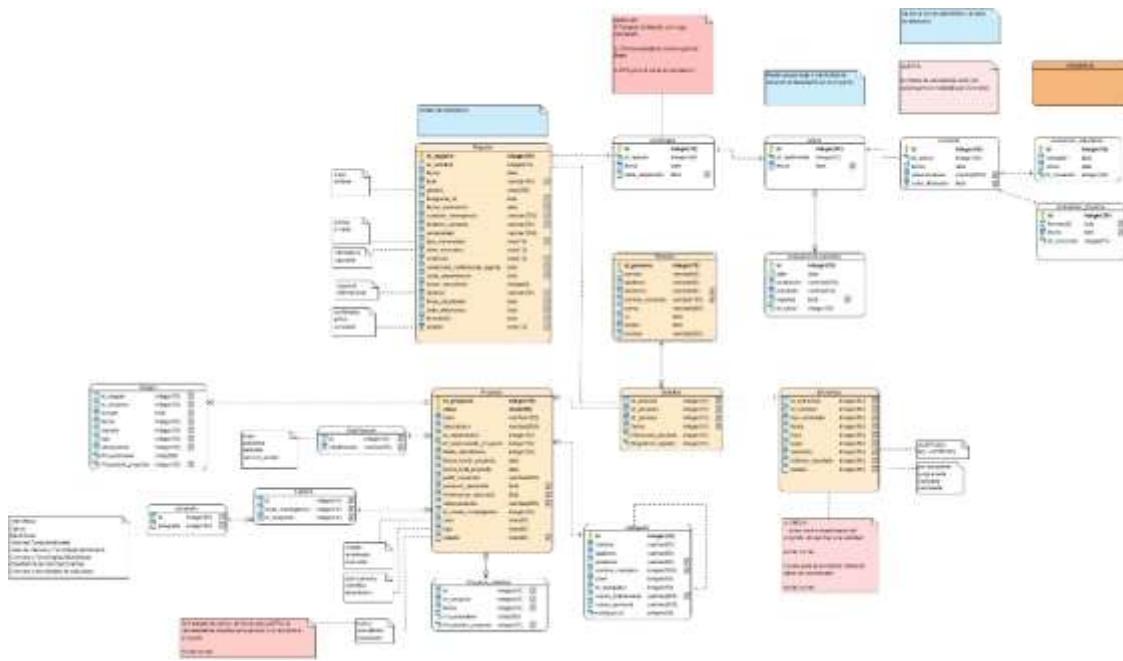


Nota: Captura de Herramienta en línea utilizada para la creación de interfaces y diagramas. (Ximena, 2023)

✓ **Desarrollo y diseño de la base de datos.**

Para el diseño y desarrollo de la base de datos se está utilizando PostgreSQL y pgAdmin ambos forman una combinación robusta y altamente funcional para el diseño, desarrollo y gestión de bases de datos relacionales. Su versatilidad y potencia los convierten en elecciones populares en diversos entornos de desarrollo de software.

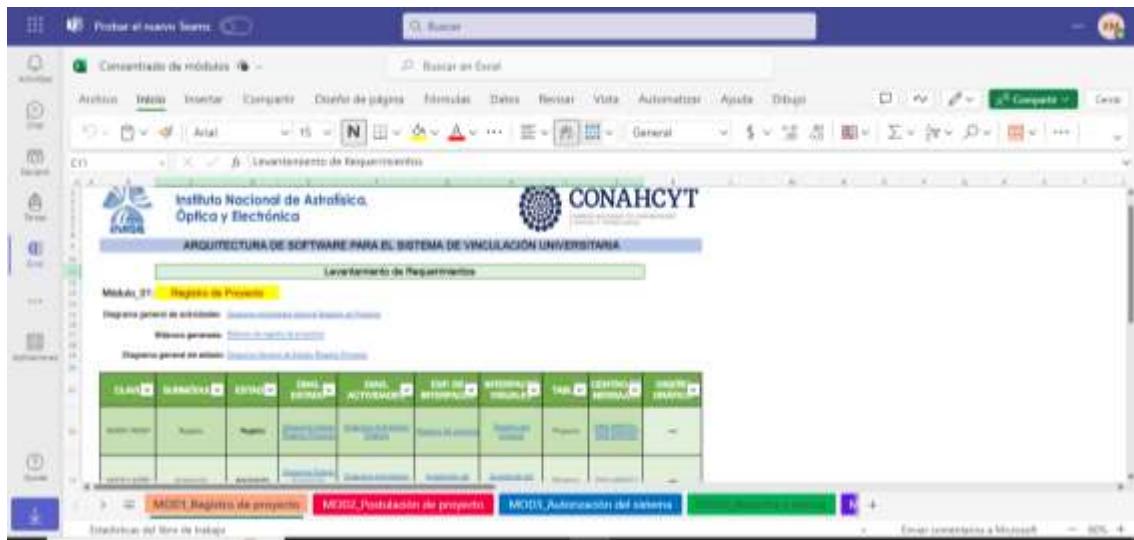
Figura 22 Diagrama de la Base de Datos



Nota: Se elaboró el diagrama de la base de datos a utilizar en el sistema.

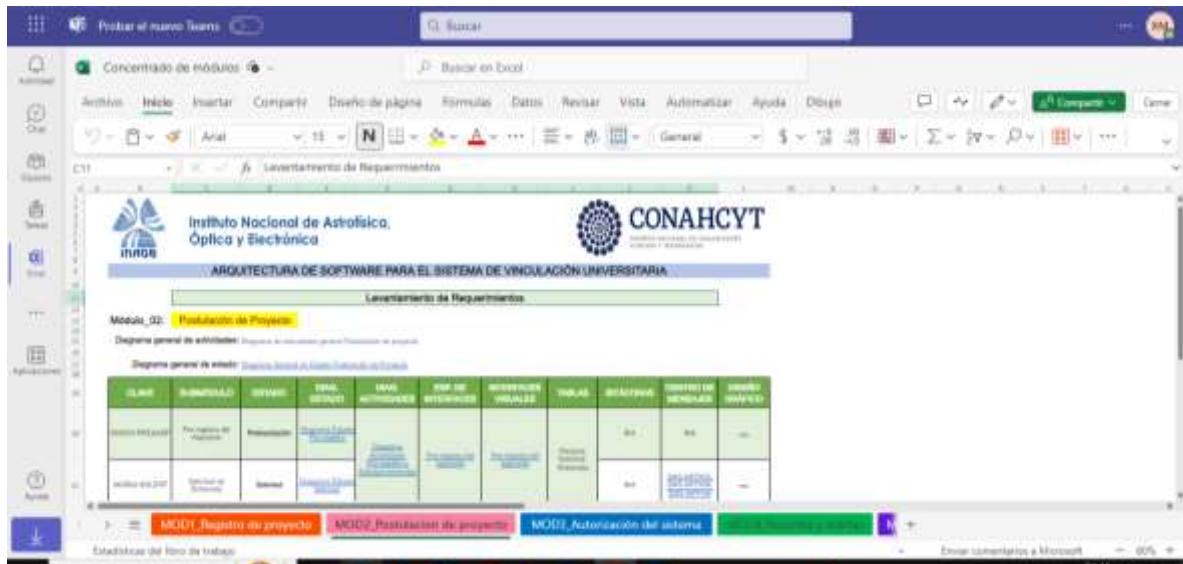
Finalmente, para llevar una organización sobre los recursos para el desarrollo de las interfaces, se organizó de forma formal la información en un Concentrado de Módulos, con las partes fundamentales sobre los estados del sistema, los diagramas de actividades, la visualización de las interfaces, la relación de las tablas de la base de datos y los mensajes o alertas.

Figura 23 Concentrado del módulo del registro del proyecto



Nota: Se creó el modulo dentro de Microsoft Teams para mayor facilidad para acceder a los diagramas, mensajes, alertas, especificaciones de las interfaces, tablas de la base de datos y las interfaces visuales. (Ximena, Castillo)

Figura 24 Concentrado del módulo de la postulación del proyecto



Nota: Se creó el modulo dentro de Microsoft Teams para mayor facilidad para acceder a los diagramas, mensajes, alertas, especificaciones de las interfaces, tablas de la base de datos y las interfaces visuales. (Ximena, 2023)

7.8 Desarrollo de componentes (Modelos, Controlador y Vistas)

Para el desarrollo de componentes, que incluyen Modelos, Controladores y Vistas, se tomó como parte esencial en la construcción de aplicaciones de software, particularmente en el contexto de la arquitectura de software Modelo-Vista-Controlador (MVC).

Modelo:

- Lógica de Negocio: Implementar la lógica de negocio del Sistema de Vinculación Universitaria. Esto puede incluir reglas de validación, procesamiento de datos y la interacción con bases de datos.
- Manejo de Datos: Crear clases o componentes responsables de acceder y manipular datos. Utilizar ORM (Mapeo Objeto-Relacional) si es necesario para interactuar con bases de datos de manera más orientada a objetos.
- Pruebas Unitarias: Desarrollar pruebas unitarias para asegurar que el modelo funcione correctamente y cumpla con los requisitos de negocio.

Controlador:

- Coordinación de la Lógica de Negocio: Implementar el controlador para coordinar las interacciones entre el modelo y la vista. Este componente maneja las solicitudes del usuario y actualiza el modelo según sea necesario.
- Manejo de Rutas y URL: Configurar las rutas y URL de la aplicación. Utilizar un enrutador para gestionar las distintas acciones del controlador basadas en las solicitudes del usuario.
- Pruebas de Controladores: Desarrollar pruebas de controladores para asegurar que la lógica de coordinación funcione adecuadamente y responda correctamente a las solicitudes del usuario.

Vista:

- Diseño de Interfaz de Usuario: Crear las vistas que representarán la interfaz de usuario. Utilizar tecnologías como HTML, CSS y JavaScript para construir páginas web interactivas y visualmente atractivas.

- Interacción con el Usuario: Implementar componentes de interfaz de usuario interactivos utilizando bibliotecas o frameworks
- Integración con el Controlador: Conectar la vista con el controlador para manejar eventos del usuario y enviar actualizaciones al modelo cuando sea necesario.

Figura 25 Diagrama de MCV



Nota: Este diagrama muestra cómo funciona el modelo, vista y controlador (Alvarez, 2023)

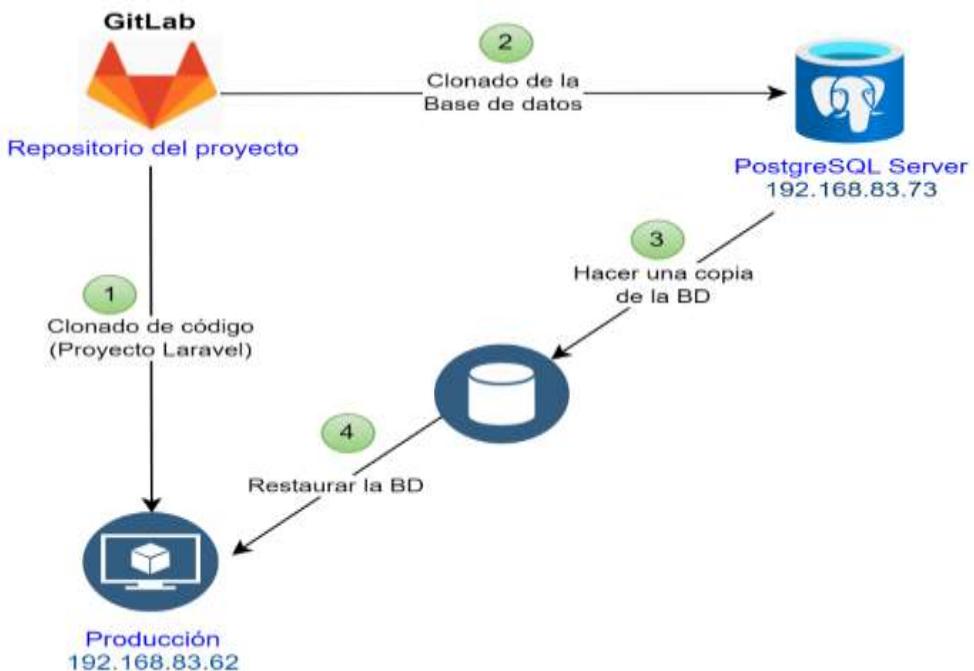
7.9 Configuración de servidores, cuentas de acceso a la red y servicios del software.

La primera etapa implicó la configuración de servidores para albergar y ejecutar el software de vinculación educativa. Se configuro la parte de Gitlab, jugo un papel fundamental en la implementación de prácticas de DevOps al proporcionar una plataforma integral que aborda múltiples aspectos del ciclo de vida del desarrollo de software.

Evalúe detalladamente los requisitos del sistema y seleccioné la infraestructura adecuada, considerando factores como la capacidad de procesamiento, almacenamiento y conectividad de red. Instalé y configuré los sistemas operativos y servicios esenciales para garantizar un entorno de servidor estable y seguro.

Una vez configurados los servidores, procedí a la creación de cuentas de acceso a la red. Definí roles y privilegios según las necesidades del personal involucrado en el proyecto. Implementé medidas de seguridad, como contraseñas robustas y políticas de acceso, para proteger la integridad y confidencialidad de los datos. La creación de cuentas personalizadas aseguró un acceso controlado y eficiente a la red.

Figura 26 Diagrama de clonado del código y base de datos



Nota: Este diagrama muestra cómo funciona Gitlab con la base de datos. (Ximena,2023)

7.10 Creación y pruebas para el código en entorno de desarrollo

Durante la fase de creación y pruebas en el entorno de desarrollo, los desarrolladores se dedican a construir y perfeccionar el código que constituirá la base del sistema, es crucial para garantizar la calidad y la eficiencia. Este proceso implica varias etapas clave:

Desarrollo de Código

- ✓ Los desarrolladores trabajan en colaboración en un entorno de desarrollo integrado (IDE) para escribir y revisar el código fuente del Sistema de Vinculación Universitaria.
- ✓ Se siguen las prácticas de codificación definidas y se aplican patrones de diseño para asegurar la consistencia y la escalabilidad del código.

Pruebas Unitarias y de Integración:

- ✓ Se crean pruebas unitarias para cada componente del sistema, evaluando la funcionalidad a nivel modular.
- ✓ Las pruebas de integración se enfocan en garantizar que los distintos módulos del sistema interactúen de manera efectiva.

Control de Versiones y Ramificación:

- ✓ Se utiliza un sistema de control de versiones, como Git, para gestionar el código fuente y facilitar la colaboración.
- ✓ El desarrollo de nuevas características y correcciones de errores se realiza en ramas separadas, y las fusiones se gestionan de manera controlada.

Figura 27 Codificación de Vagrantfile integrado con GitLab

Vaquero - Visual Studio Code

Archivo Editar Sección Ver Ir Explorador Terminal Ayuda

■ Vaquero.vsl

```
1 // Vaquero
2
3 // Crea un servicio de host para Vaquero
4 config ve {
5   config ve_configs("27.00.100.100") {
6     # Especificación de los que se utilizará para crear la IP
7     config ve_box = "genericUbuntu2004"
8
9     # Configuración de la dirección IP escritora
10    config ve_networks("escritor_realm") {
11      ip = "192.168.88.127"
12    }
13
14    # Configuración de recursos de la VM
15    config ve_provider("virtualBox") {
16      vs_memory = 2448 #C 32 GB RAM
17      vs_processor = 2 #Número de núcleos
18      vs_cpus = 2 # C 2 CPU
19    }
20
21    config ve_hostname = "luisvm1" #Nombre de la máquina
22  }
23
24  config ve_synced_folders("www") {
25    path = "/var/www/html"
26  }
27
28  # Verificar si existe un archivo de nombre de usuario y correo en gitfile
29  gitfile_data_file = "gitfile_data.txt"
30  gitfile_data = File.exist?(gitfile_data_file) ? File.read(gitfile_data_file).strip.split(',') : []
31
32  # Si no existen los datos en el archivo, solicitarlos
33  if gitfile_data.empty? || gitfile_data.length != 2
34    require 'open-uri'
35    puts "Por favor, ingresa tu nombre de usuario en luis (name)"
36    gitfile_username = STDIN.gets.chomp
37    puts "Por favor, ingresa tu correo de correo (mail)"
38    gitfile_email = STDIN.gets.chomp
39
40    # Guardar los datos en el archivo
41    File.write(gitfile_data_file, "#{gitfile_username},#{gitfile_email}")
42  end
43
44  # Configuración de las máquinas virtuales
45  config ve_box = "genericUbuntu2004"
46  config ve_network("public_network", ip: "192.168.88.127")
47  config ve_provider("virtualBox"),
48    name: "luisvm1",
49    initcmd: "ip route add default via 192.168.88.1"
50
51  # Configuración de recursos de la VM
52  config ve_provider("virtualBox") {
53    vs_memory = 2448 # C 32 GB RAM
54    vs_processor = 2 # C 2 CPU
55  }
56
57
58 // LOTES DE CANTOS
59 echo "Instalando Lotes"
60 composer global require laravel/installer
61 echo "Asignar PATH=$HOME/composer/vendor/bin:$PATH" >> .bashrc
62 source ~/.bashrc
63
64 cd ~
65 git config --global http.sslVerify false
66
67 # Crear e instalar la carpeta gitlab en el directorio del usuario
68 echo "GITLAB" >> gitconfig
69 echo "  name = luis@luisvm1" >> gitconfig
70 echo "  email = luis@luisvm1" >> gitconfig
71 echo "  https" >> gitconfig
72 echo "  university = false" >> gitconfig
73
74 cd ~/www/html
75 git pull
76
77 git config --global http.sslVerify false
78 git clone https://gitlab.com/luis/luisvm1.git
79 git_www_dir=$PWD/project/luisvm1
80 rm -rf $git_www_dir/.gitignore
81 rm -rf $git_www_dir/.gitmodules
82 rm -rf $git_www_dir/.gitattributes
83 rm -rf $git_www_dir/.git
84 rm -rf $git_www_dir/.gitconfig
85
86 # Clonar el repositorio del proyecto Laravel
87 cd $git_www_dir
88 composer install
89 sudo chmod +w www-data
90 sudo chmod +w storage
91
92
93 # Cambiar el valor de APP_URL en el archivo .env en Laravel
94 sed -i 's/APP_URL=127.0.0.1:8000/APP_URL=192.168.88.127:8000/g' .env
95
96 # Configurar Laravel en el archivo apache.conf
97 echo "ServerName 192.168.88.127:8000" >> /etc/apache2/sites-available/laravel.conf
98
99 # Cambiar directorio de Laravel
100 sed -i 's/DocumentRoot /var/www/html/laravel/public/' /etc/apache2/sites-available/laravel.conf
101
102 # Cambiar permisos de los directorios
103 chown -R www-data:www-data /var/www/html/laravel/public
104
105 # Reiniciar Apache
106 sudo systemctl restart apache2
```

Nota: Captura de la integración para la configuración de GitLab en el vagrantfile. (Ximena, 2023)

Figura 28 Inicialización de las máquinas virtuales para GitLab

The figure consists of two screenshots of a terminal window titled "ximena@debian:~/Gitlab".

The top window shows the command "vagrant up" being run. The output indicates that the user is prompted for their GitLab username ("Por favor, ingresa tu nombre de usuario de GitLab (inaoe): ximena") and email ("Por favor, ingresa tu correo de GitLab (inaoe): ximena.castillo@inaoe.edu.mx"). The vagrant provider is set to "virtualbox". The process continues with steps like "Bringing machine 'default' up with 'virtualbox' provider...", "default: Importing base box 'generic/ubuntu2204'", "default: Matching MAC address for NAT networking...", "default: Checking if box 'generic/ubuntu2204' version '4.2.16' is up to date", and "default: Setting the name of the VM: Testing". It also handles network interfaces, forwarding ports (port 22 mapped to 2222), and booting the VM. It ends with a warning about connection resets.

```
ximena@debian:~$ cd Gitlab
ximena@debian:~/Gitlab$ vagrant up
Por favor, ingresa tu nombre de usuario de GitLab (inaoe):
ximena
Por favor, ingresa tu correo de GitLab (inaoe):
ximena.castillo@inaoe.edu.mx
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'generic/ubuntu2204'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'generic/ubuntu2204' version '4.2.16' is up to date
...
==> default: Setting the name of the VM: Testing
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
    default: Adapter 2: hostonly
    default: Adapter 3: bridged
==> default: Forwarding ports...
    default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Running 'pre-boot' VM customizations...
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2222
    default: SSH username: vagrant
    default: SSH auth method: private key
    default: Warning: Connection reset. Retrying...
    default: Warning: Remote connection disconnect. Retrying...
```

The bottom window shows the command "sudo apt update" being run. The output lists packages to be installed (PostgreSQL 14), suggested packages (postgresql-doc-14), and packages to be upgraded (libcommon-sense-perl, libjson-perl, libjson-xs-perl, liblvm2m2, libpq5, libtypes-serialiser-perl). It also shows the download of packages from mirrors.edge.kernel.org/ubuntu/jammy/main amd64. The user is prompted for confirmation to proceed with the upgrade.

```
default: Instalando PostgreSQL 14...
default: Reading package lists...
default: Building dependency tree...
default: Reading state information...
default: The following additional packages will be installed:
  default:   libcommon-sense-perl libjson-perl libjson-xs-perl liblvm2m2 libpq5
5
  default:   libtypes-serialiser-perl postgresql-client-14 postgresql-client-common
  default:   postgresql-common ssl-cert
  default: Suggested packages:
  default:   postgresql-doc-14
  default: The following NEW packages will be installed:
  default:   libcommon-sense-perl libjson-perl libjson-xs-perl liblvm2m2 libpq5
5
  default:   libtypes-serialiser-perl postgresql-14 postgresql-client-14
  default:   postgresql-client-common postgresql-common postgresql-contrib ssl-cert
  default: 8 upgraded, 12 newly installed, 0 to remove and 178 not upgraded.
  default: Need to get 41.9 MB of archives.
  default: After this operation, 160 MB of additional disk space will be used.
  default: Ign:1 https://mirrors.edge.kernel.org/ubuntu/jammy/main amd64 libcommon-sense-perl amd64 3.75-2build1
  default: Ign:2 https://mirrors.edge.kernel.org/ubuntu/jammy/main amd64 libjson-perl all 4.84000-1
  default: Ign:3 https://mirrors.edge.kernel.org/ubuntu/jammy/main amd64 libtypes-serialiser-perl all 1.81-1
  default: Ign:4 https://mirrors.edge.kernel.org/ubuntu/jammy/main amd64 libjson-xs-perl amd64 4.030-1build3
```

2 figuras

7 .11 Pruebas no funcionales

Las pruebas no funcionales son críticas para evaluar aspectos que van más allá de la funcionalidad del sistema y se centran en su rendimiento, seguridad y usabilidad.

Pruebas de Rendimiento:

- La utilización de herramientas como Apache para simular cargas de usuarios y evaluar el rendimiento del sistema bajo diferentes condiciones de carga.
- Identificación y optimización de cuellos de botella para garantizar que el sistema maneje eficientemente la carga esperada.

Pruebas de Seguridad:

- Realización de pruebas de seguridad, como análisis de vulnerabilidades y pruebas de penetración, para identificar posibles brechas de seguridad.
- Garantizar el cumplimiento de estándares de seguridad y la protección contra ataques comunes.

Pruebas de Estrés y Resistencia:

- Sometimiento del sistema a cargas extremas para evaluar su capacidad de resistencia y para identificar posibles puntos de fallo bajo condiciones extremas.
- Monitoreo de la estabilidad del sistema durante pruebas prolongadas.

Pruebas de Escalabilidad:

- Evaluación de la capacidad del sistema para escalar vertical u horizontalmente para manejar aumentos súbitos en la carga.
- Implementación de estrategias de escalabilidad en entornos de nube para garantizar la flexibilidad del sistema.

7. 12 Integración de codificación

La integración continua es esencial para mantener un flujo de desarrollo eficiente, detectar problemas tempranamente y garantizar la estabilidad del sistema en evolución. Este enfoque proporciona una base sólida para la implementación exitosa.

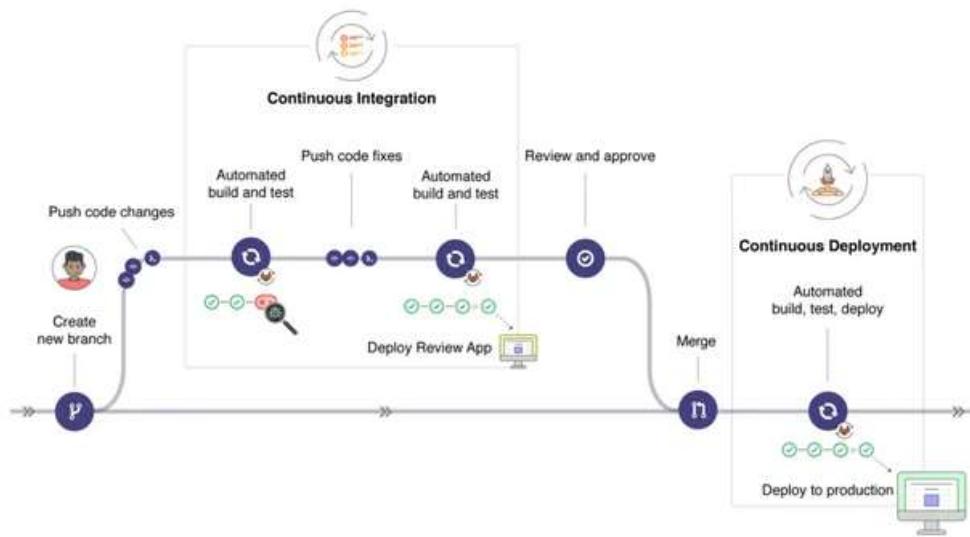
Se ha continuado utilizando GitLab como herramienta esencial para el control de versiones, implementando integración continua (CI) y despliegue continuo (CD). Esta práctica garantiza la automatización de procesos, asegurando la coherencia y la estabilidad del sistema en todas las etapas del desarrollo.

Acciones realizadas:

- Configuración de GitLab para integración continua (CI) y despliegue continuo (CD).
- Desarrollo e implementación de pipelines automatizados para pruebas y despliegue.
- Automatizar el proceso de compilación y prueba para garantizar la consistencia y la calidad del código.
- Desarrollar y ejecutar pruebas automáticas unitarias y de integración como parte del proceso de integración continua.
- Desplegar automáticamente el código integrado en entornos de pruebas para realizar pruebas adicionales antes de la implementación en producción.

Si el proceso de integración es exitoso, considerar la posibilidad de extender la práctica hacia el despliegue continuo, donde los cambios integrados automáticamente se implementan en entornos de producción.

Figura 29 Diagrama de CI/CD con GitLab



Nota: GitLab para el proceso de construcción, integración e implementación (Magnolia CMS Documentation, s. f.)

Figura 30 Código GitLab CI/CD

Este es un fragmento de código de configuración para GitLab CI/CD. El archivo se llama 'gitlab-ci.yml' y contiene las siguientes líneas:

```

# This is a template - please edit with your project's requirements.
# Help: https://gitlab.com/help/ci/yaml/index.html

stages:
  - build
  - test
  - deploy

variables:
  DB_HOST: "db_ximena" # Nombre de la base de datos PostgreSQL
  DB_USER: "ximena" # Usuario de la base de datos PostgreSQL
  DB_PASSWORD: "ximena22" # Contraseña de la base de datos PostgreSQL

before_script:
  # Llamadas a ejecutar antes de cada etapa del proyecto.
  - apt-get update -y
  - apt-get install -y curl git

script:
  # Etapa para construir el proyecto.
  - echo "Preparing environment" # Mensaje para utilizar la preparación del sistema.
  - apt-get update -y
  - apt-get install -y curl git
  # Etapa para construir el proyecto.
  - docker build -t ximena
  # Etapa para ejecutar pruebas.
  - docker test
  # Etapa de ejecución del proyecto.
  - curl

```

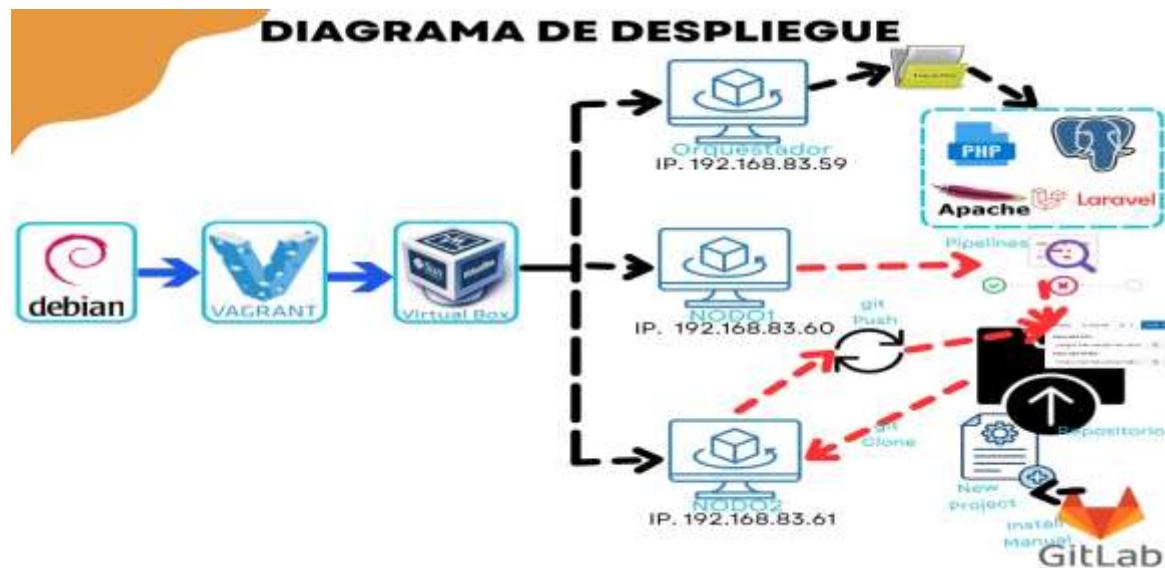
Nota: En esta captura muestra la codificación del código para GitLab. (Ximena, 2023)

7. 13 Automatización de implementación

La automatización de la implementación es fundamental para agilizar el desarrollo de software y garantizar la consistencia en los entornos de producción. Es por ello que se utilizó la herramienta de ansible que permite la automatización de la configuración y la gestión de sistemas.

Se diseñó el diagrama de despliegue de procesos de ejecución con ansible para las máquinas virtuales, para visualizar los programas y las instalaciones que se utilizaron para la orquestación.

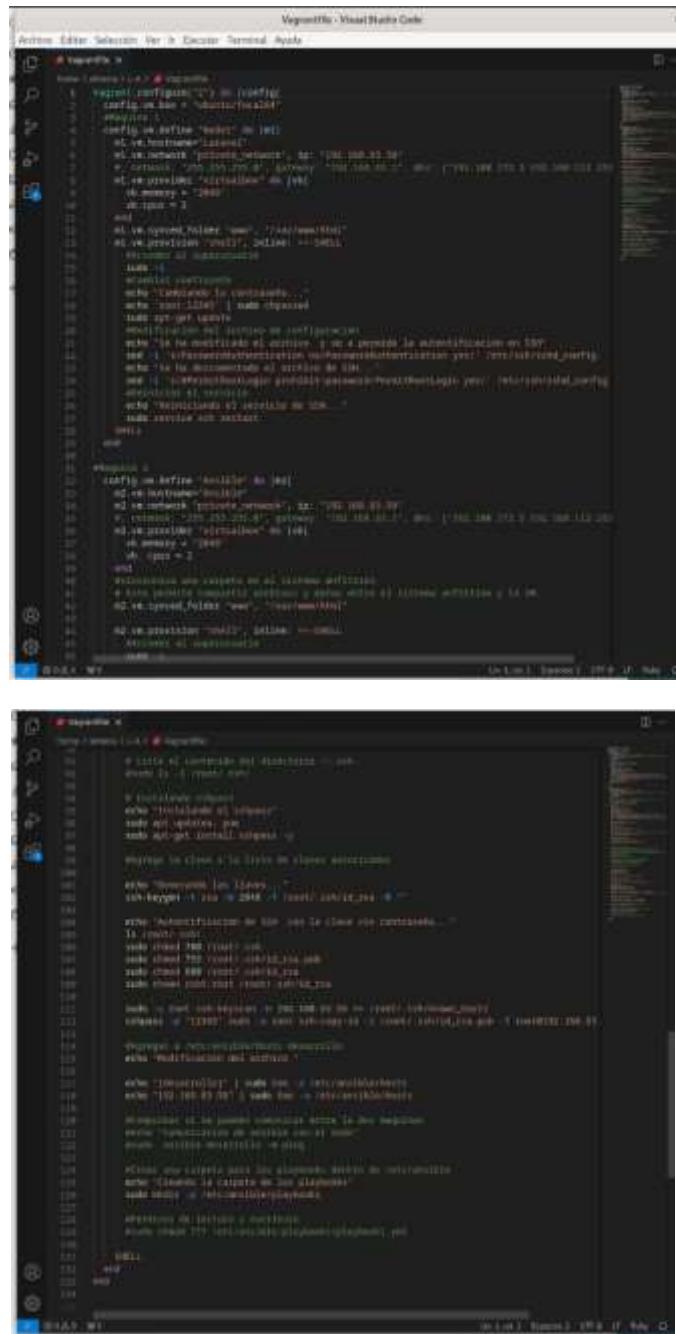
Figura 31 Diagrama de despliegue



Nota: El diagrama muestra cómo se hace el despliegue con las máquinas virtuales por medio de ansible. (Ximena, 2023)

Mediante vagrantfile se crearon dos máquinas virtuales y la configuración de cada una para que tuvieran conexión entre ambas.

Figura 32 Codificación con Ansible



The image shows two tabs of code in Visual Studio Code:

Top Tab (Vagrantfile):

```
Ansible Editor Selección Ver Iniciar Terminal Ayuda
Vagrantfile - Visual Studio Code
Vagrantfile
1 #!/usr/bin/vagrant
2
3 Vagrant.configure("2") do |config|
4   config.vm.box = "UbuntuServer"
5   config.vm.network "public_network",
6     :ipaddress => "192.168.93.30"
7   config.vm.provider "virtualbox",
8     :memory => 2048
9   config.vm.synced_folder "src", "/vagrant"
10  config.vm.provider "virtualbox" do |vb|
11    vb.customize ["modifyvm", "--nictype1", "Intel PRO/100 MT Desktop"]
12    vb.customize ["modifyvm", "--nictype2", "Bridged Adapter"]
13    vb.customize ["modifyvm", "--bridgeon", "1"]
14    vb.customize ["modifyvm", "--macaddress1", "080020C0A9E0"]
15    vb.customize ["modifyvm", "--macaddress2", "080020C0A9E1"]
16    vb.customize ["modifyvm", "--eth0", "Bridged Adapter"]
17    vb.customize ["modifyvm", "--eth1", "Intel PRO/100 MT Desktop"]
18    vb.customize ["modifyvm", "--eth2", "Bridged Adapter"]
19    vb.customize ["modifyvm", "--eth3", "Bridged Adapter"]
20    vb.customize ["modifyvm", "--eth4", "Bridged Adapter"]
21    vb.customize ["modifyvm", "--eth5", "Bridged Adapter"]
22    vb.customize ["modifyvm", "--eth6", "Bridged Adapter"]
23    vb.customize ["modifyvm", "--eth7", "Bridged Adapter"]
24    vb.customize ["modifyvm", "--eth8", "Bridged Adapter"]
25    vb.customize ["modifyvm", "--eth9", "Bridged Adapter"]
26    vb.customize ["modifyvm", "--eth10", "Bridged Adapter"]
27    vb.customize ["modifyvm", "--eth11", "Bridged Adapter"]
28    vb.customize ["modifyvm", "--eth12", "Bridged Adapter"]
29    vb.customize ["modifyvm", "--eth13", "Bridged Adapter"]
30    vb.customize ["modifyvm", "--eth14", "Bridged Adapter"]
31    vb.customize ["modifyvm", "--eth15", "Bridged Adapter"]
32    vb.customize ["modifyvm", "--eth16", "Bridged Adapter"]
33    vb.customize ["modifyvm", "--eth17", "Bridged Adapter"]
34    vb.customize ["modifyvm", "--eth18", "Bridged Adapter"]
35    vb.customize ["modifyvm", "--eth19", "Bridged Adapter"]
36    vb.customize ["modifyvm", "--eth20", "Bridged Adapter"]
37    vb.customize ["modifyvm", "--eth21", "Bridged Adapter"]
38    vb.customize ["modifyvm", "--eth22", "Bridged Adapter"]
39    vb.customize ["modifyvm", "--eth23", "Bridged Adapter"]
40    vb.customize ["modifyvm", "--eth24", "Bridged Adapter"]
41    vb.customize ["modifyvm", "--eth25", "Bridged Adapter"]
42    vb.customize ["modifyvm", "--eth26", "Bridged Adapter"]
43    vb.customize ["modifyvm", "--eth27", "Bridged Adapter"]
44    vb.customize ["modifyvm", "--eth28", "Bridged Adapter"]
45    vb.customize ["modifyvm", "--eth29", "Bridged Adapter"]
46    vb.customize ["modifyvm", "--eth30", "Bridged Adapter"]
47    vb.customize ["modifyvm", "--eth31", "Bridged Adapter"]
48    vb.customize ["modifyvm", "--eth32", "Bridged Adapter"]
49    vb.customize ["modifyvm", "--eth33", "Bridged Adapter"]
50    vb.customize ["modifyvm", "--eth34", "Bridged Adapter"]
51    vb.customize ["modifyvm", "--eth35", "Bridged Adapter"]
52    vb.customize ["modifyvm", "--eth36", "Bridged Adapter"]
53    vb.customize ["modifyvm", "--eth37", "Bridged Adapter"]
54    vb.customize ["modifyvm", "--eth38", "Bridged Adapter"]
55    vb.customize ["modifyvm", "--eth39", "Bridged Adapter"]
56    vb.customize ["modifyvm", "--eth40", "Bridged Adapter"]
57    vb.customize ["modifyvm", "--eth41", "Bridged Adapter"]
58    vb.customize ["modifyvm", "--eth42", "Bridged Adapter"]
59    vb.customize ["modifyvm", "--eth43", "Bridged Adapter"]
59  end
60
61
62
63
64
65
66
67
68
69
69
```

Bottom Tab (roles/01-interfaces/defaults/main.yml):

```
Ansible Editor Selección Ver Iniciar Terminal Ayuda
roles/01-interfaces/defaults/main.yml - Visual Studio Code
roles/01-interfaces/defaults/main.yml
1 #! /usr/bin/ansible
2
3 #! /bin/bash
4
5 #! /bin/bash
6
7 #! /bin/bash
8
9 #! /bin/bash
10
11 #! /bin/bash
12
13 #! /bin/bash
14
15 #! /bin/bash
16
17 #! /bin/bash
18
19 #! /bin/bash
20
21 #! /bin/bash
22
23 #! /bin/bash
24
25 #! /bin/bash
26
27 #! /bin/bash
28
29 #! /bin/bash
30
31 #! /bin/bash
32
33 #! /bin/bash
34
35 #! /bin/bash
36
37 #! /bin/bash
38
39 #! /bin/bash
40
41 #! /bin/bash
42
43 #! /bin/bash
44
45 #! /bin/bash
46
47 #! /bin/bash
48
49 #! /bin/bash
50
51 #! /bin/bash
52
53 #! /bin/bash
54
55 #! /bin/bash
56
57 #! /bin/bash
58
59 #! /bin/bash
60
61 #! /bin/bash
62
63 #! /bin/bash
64
65 #! /bin/bash
66
67 #! /bin/bash
68
69 #! /bin/bash
70
71 #! /bin/bash
72
73 #! /bin/bash
74
75 #! /bin/bash
76
77 #! /bin/bash
78
79 #! /bin/bash
80
81 #! /bin/bash
82
83 #! /bin/bash
84
85 #! /bin/bash
86
87 #! /bin/bash
88
89 #! /bin/bash
90
91 #! /bin/bash
92
93 #! /bin/bash
94
95 #! /bin/bash
96
97 #! /bin/bash
98
99 #! /bin/bash
100
101 #! /bin/bash
102
103 #! /bin/bash
104
105 #! /bin/bash
106
107 #! /bin/bash
108
109 #! /bin/bash
110
111 #! /bin/bash
112
113 #! /bin/bash
114
115 #! /bin/bash
116
117 #! /bin/bash
118
119 #! /bin/bash
120
121 #! /bin/bash
122
123 #! /bin/bash
124
125 #! /bin/bash
126
127 #! /bin/bash
128
129 #! /bin/bash
130
131 #! /bin/bash
132
133 #! /bin/bash
134
135 #! /bin/bash
136
137 #! /bin/bash
138
139 #! /bin/bash
140
141 #! /bin/bash
142
143 #! /bin/bash
144
145 #! /bin/bash
146
147 #! /bin/bash
148
149 #! /bin/bash
150
151 #! /bin/bash
152
153 #! /bin/bash
154
155 #! /bin/bash
156
157 #! /bin/bash
158
159 #! /bin/bash
160
161 #! /bin/bash
162
163 #! /bin/bash
164
165 #! /bin/bash
166
167 #! /bin/bash
168
169 #! /bin/bash
170
171 #! /bin/bash
172
173 #! /bin/bash
174
175 #! /bin/bash
176
177 #! /bin/bash
178
179 #! /bin/bash
180
181 #! /bin/bash
182
183 #! /bin/bash
184
185 #! /bin/bash
186
187 #! /bin/bash
188
189 #! /bin/bash
190
191 #! /bin/bash
192
193 #! /bin/bash
194
195 #! /bin/bash
196
197 #! /bin/bash
198
199 #! /bin/bash
200
201 #! /bin/bash
202
203 #! /bin/bash
204
205 #! /bin/bash
206
207 #! /bin/bash
208
209 #! /bin/bash
210
211 #! /bin/bash
212
213 #! /bin/bash
214
215 #! /bin/bash
216
217 #! /bin/bash
218
218
```

Nota: Codificación para la automatización con Ansible. (Ximena, 2023)

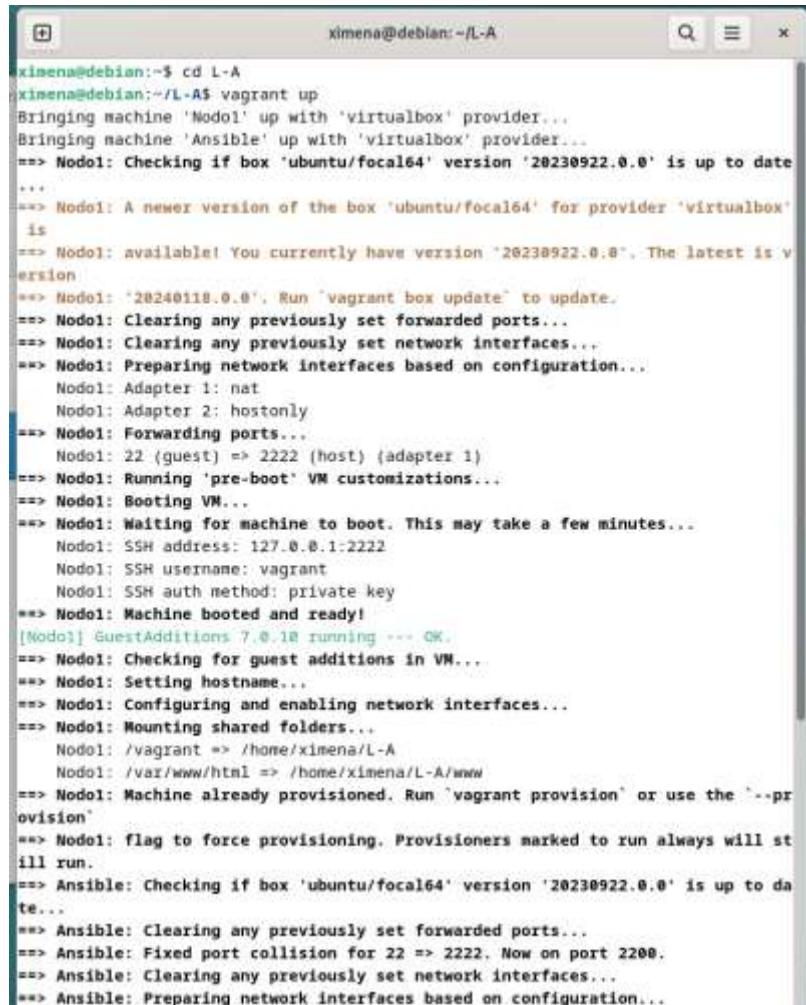
Se codificó un playbook especificando las tareas que iba realizar, las descargas de las herramientas y sus configuraciones.

Figura 33 Playbook de Ansible para la automatización.

Nota: Creación de los playbooks para automatizar el proceso.

Y finalmente se ejecutó desde la terminal, teniendo un despliegue con Ansible para una automatización más rápida y efectiva.

Figura 34 Ejecución de las maquinas con Ansible



A screenshot of a terminal window titled "ximena@debian: ~/L-A". The window displays the output of a command-line session. The user runs "cd L-A" followed by "vagrant up". The vagrant command performs several steps: it checks the box version, finds a newer version available, runs pre-boot customizations, boots the VM, waits for machine to boot, and configures guest additions. It also handles forwarded ports, network interfaces, and shared folders. Finally, it runs Ansible to provision the machine, checking its version and preparing network interfaces. The terminal uses color-coded output for different log levels.

```
ximena@debian:~/L-A$ cd L-A
ximena@debian:~/L-A$ vagrant up
Bringing machine 'Nodo1' up with 'virtualbox' provider...
Bringing machine 'Ansible' up with 'virtualbox' provider...
==> Nodo1: Checking if box 'ubuntu/focal64' version '20230922.0.0' is up to date
...
==> Nodo1: A newer version of the box 'ubuntu/focal64' for provider 'virtualbox'
is
==> Nodo1: available! You currently have version '20230922.0.0'. The latest is v
ersion
==> Nodo1: '20240118.0.0', Run 'vagrant box update' to update.
==> Nodo1: Clearing any previously set forwarded ports...
==> Nodo1: Clearing any previously set network interfaces...
==> Nodo1: Preparing network interfaces based on configuration...
Nodo1: Adapter 1: nat
Nodo1: Adapter 2: hostonly
==> Nodo1: Forwarding ports...
Nodo1: 22 (guest) => 2222 (host) (adapter 1)
==> Nodo1: Running 'pre-boot' VM customizations...
==> Nodo1: Booting VM...
==> Nodo1: Waiting for machine to boot. This may take a few minutes...
Nodo1: SSH address: 127.0.0.1:2222
Nodo1: SSH username: vagrant
Nodo1: SSH auth method: private key
==> Nodo1: Machine booted and ready!
[Nodo1] GuestAdditions 7.0.10 running --- ok.
==> Nodo1: Checking for guest additions in VM...
==> Nodo1: Setting hostname...
==> Nodo1: Configuring and enabling network interfaces...
==> Nodo1: Mounting shared folders...
Nodo1: /vagrant => /home/ximena/L-A
Nodo1: /var/www/html => /home/ximena/L-A/www
==> Nodo1: Machine already provisioned. Run 'vagrant provision' or use the '--pr
ovision'
==> Nodo1: flag to force provisioning. Provisioners marked to run always will st
ill run.
==> Ansible: Checking if box 'ubuntu/focal64' version '20230922.0.0' is up to da
te...
==> Ansible: Clearing any previously set forwarded ports...
==> Ansible: Fixed port collision for 22 => 2222. Now on port 2200.
==> Ansible: Clearing any previously set network interfaces...
==> Ansible: Preparing network interfaces based on configuration...
```

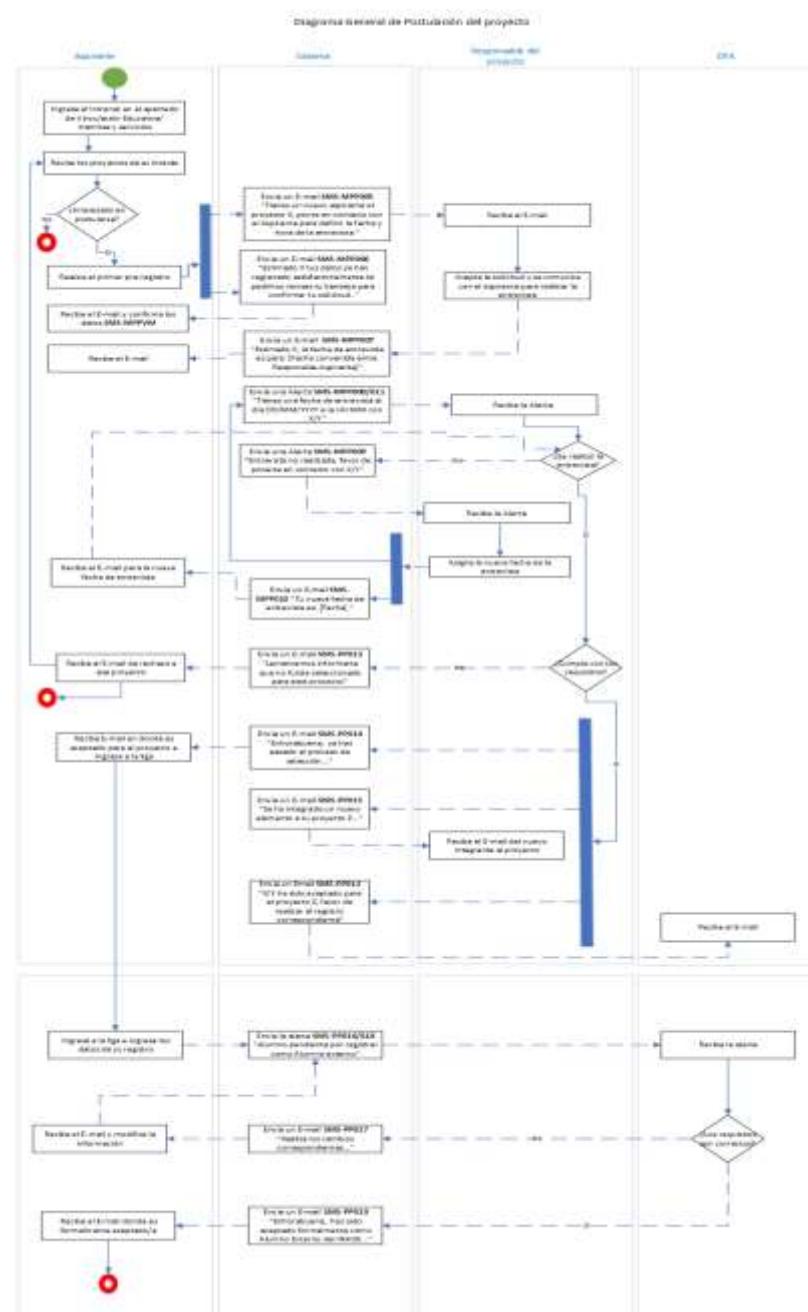
Nota: Ejecución de las maquinas desde la terminal para la automatización.

8. Resultados

Como resultado final se tiene el módulo de postulación del proyecto el cual se dividen en varios submodulos.

La figura 35 indica todas las actividades que realiza el aspirante, el sistema, el responsable del proyecto y la DFA para una mayor comprensión del proceso que hace el módulo.

Figura 35 Diagrama representando los submodulos



La figura 36 el aspirante ve reflejado todos proyectos y puede filtrar de acuerdo a su área de interés y postularse para ese proyecto.

Figura 36 Interfaz lista de proyectos

The figure displays two screenshots of a web-based application interface for managing projects.

Top Screenshot (Lista de Proyectos):

- Header:** PROYECTOS
- Search Bar:** Filtro: Buscar ▾
- Filter Buttons:** Clasificación: ▾ | Fecha de inicio: ▾ | Fecha fin: ▾ | Área: ▾ | Detalles del proyecto: ▾
- Table:**| Nº | Título | Clasificación | Fecha de inicio | Fecha fin | Área | Detalles del proyecto |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Investigación sobre fenómenos estelares para mejorar la Eficiencia de los satélites | Seguro | 19/10/23 | 06/12/23 | Astro | |
| 2 | Título del proyecto | Desarrollado | Fecha inicio | Fecha inicio | Astro | |
| 3 | Título del proyecto | Desarrollado | Fecha inicio | Fecha inicio | Astro | |
| 4 | Título del proyecto | Desarrollado | Fecha inicio | Fecha inicio | Astro | |
| 5 | Título del proyecto | Desarrollado | Fecha inicio | Fecha inicio | Astro | |
- Pagination:** 1 2 3 4 5 6 7 8 9 10

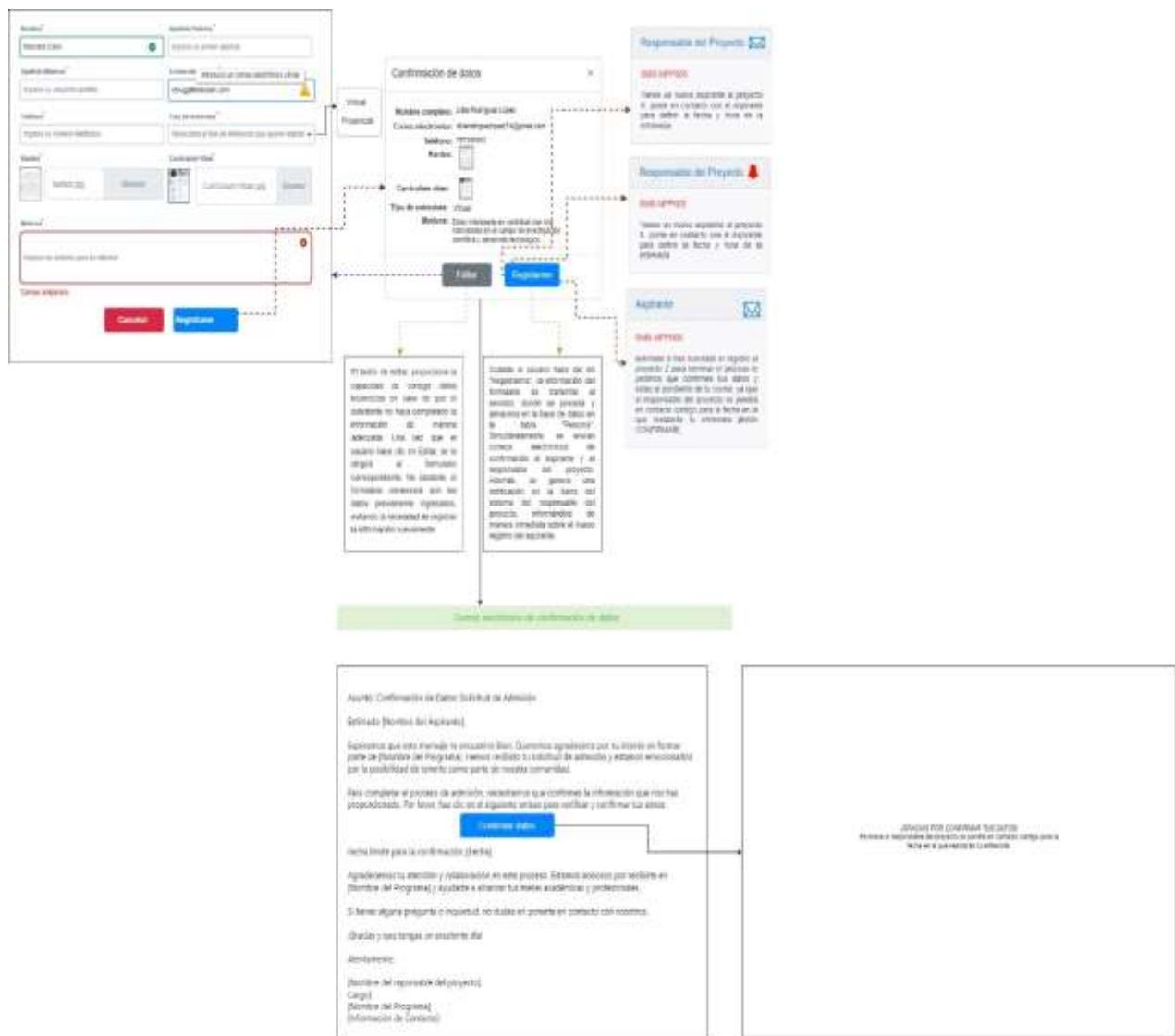
Bottom Screenshot (Detalles del proyecto):

- Section Header:** Detalles del proyecto
- Image:** Imagen del proyecto:
- Form Fields:**
 - Nombre del proyecto: Desarrollo del Sistema de Monitoreo de Satélites
 - Descripción: El proyecto tiene como objetivo crear un sistema de seguimiento y monitoreo avanzado para satélites...
 - Nombre del responsable del proyecto: Orlando Rodríguez Martínez
 - Clasificación: Seguro
 - Límite de resultados: 10
 - Fecha inicial del proyecto: 18/09/2023
 - Fecha final del proyecto: 08/10/2024
 - Producto general: Tesis
 - Período requerido:
 - Información adicional:
 - Observaciones:
 - Línea de investigación: Astronáutica
 - Tipo: Ciencias
- Buttons:** Cancelar | Postear
- Text at the bottom:** Una vez que se da click en postear se deberá aparecer una interfaz de reporte.

Nota: Se visualiza la interfaz que la aspirante vera cuando los proyectos ya estén publicados.
(Ximena,2023)

En la figura 37 se visualiza que después de postularse al proyecto el aspirante hace un pre-registro de sus datos y seleccionar si su entrevista será presencial o virtual. Al registrarse le llegara al responsable del proyecto un E-mail y una alerta informando que hay un aspirante a su proyecto, también le llegara un E-mail al aspirante para confirmación de datos.

Figura 37 Pre-registro del aspirante



Nota: Interfaz de como el aspirante primero hará su pre-registro. (Ximena, 2023)

En la figura 38 se visualiza como en la interfaz del responsable del proyecto se muestran los aspirantes para la realización de la entrevista, los alumnos que ya son aceptados en sus proyectos y los estados en los que se encuentran, así como también los filtrados de si están haciendo Tesis, Prácticas profesionales, Estadías y Servicio social. Finalmente, también se podrá filtrar el área Ciencias computacionales, Astrofísica, Electrónica, Óptica, etc.

Cuando el aspirante tenga el estado de “Para entrevistar” el responsable del proyecto asignara la hora y fecha de la realización de la entrevista, ya definida se le enviara un correo al aspirante notificándole su entrevista.

Figura 38 Interfaz del responsable del proyecto

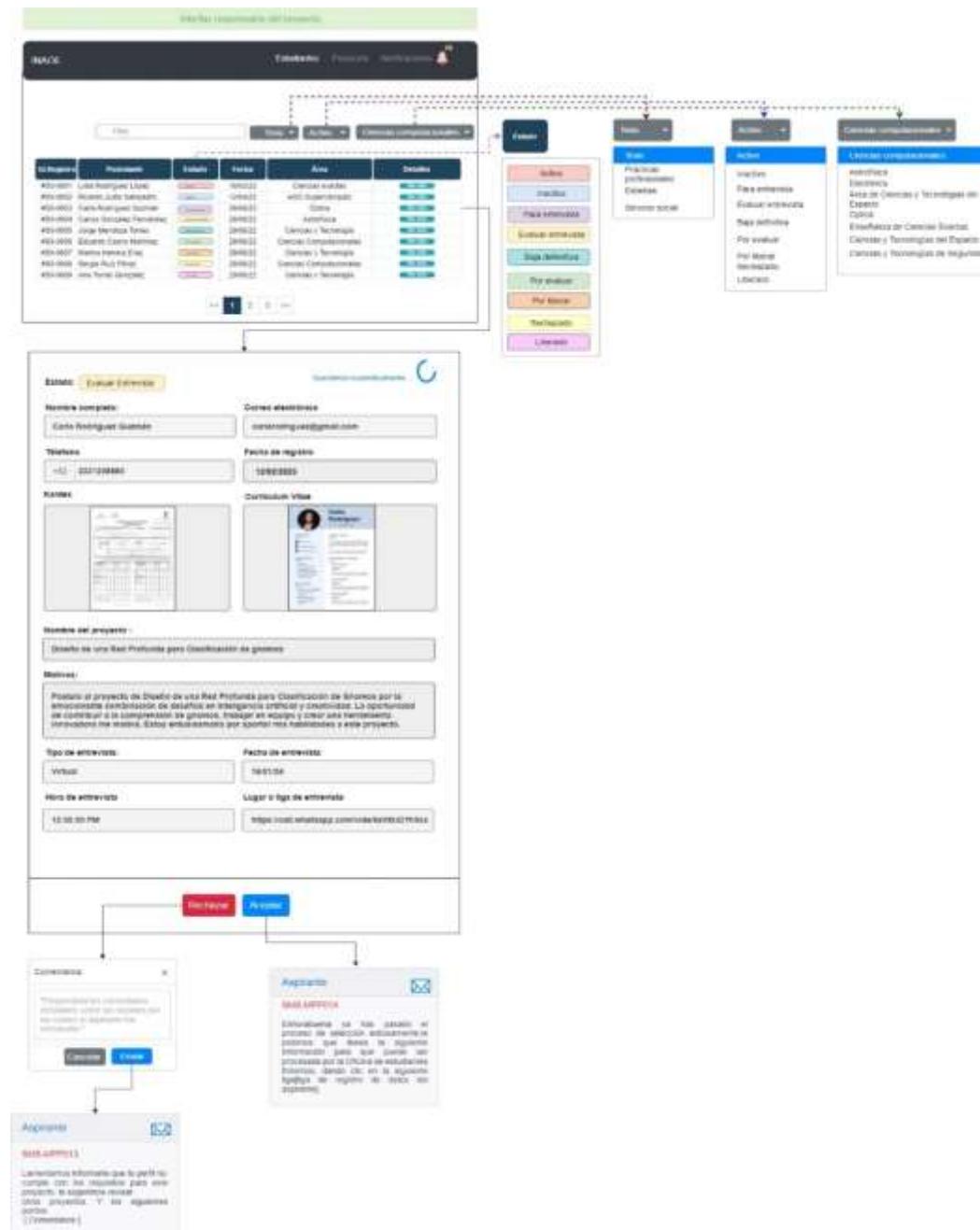
The figure displays a complex web-based application interface for managing project applications. At the top, a navigation bar includes 'Iniciar sesión/registro del alumno', 'Iniciar', 'Proyectos', 'Nuevo proyecto', and a user icon. A sidebar on the left lists 'Proyecto' and 'Alumno'. The main content area is divided into several sections:

- Aplicaciones Pendientes:** Shows a table with columns 'ID', 'Nombre completo', 'Email', 'Fecha', 'Área', and 'Estado'. It lists various student applications with their names, emails, dates, areas (e.g., Ciencias Computacionales, Astrofísica, Electrónica, Óptica), and current status (e.g., 'Para entrevistar').
- Filtros:** A sidebar with dropdown menus for 'Área' (with options like 'Tesis', 'Práctica profesional', 'Estadía', 'Servicio social'), 'Estado' (with options like 'Para entrevistar', 'Entrevistado', 'Por evaluar', 'Por aceptar', 'Por rechazar', 'Listo para', 'Rechazado', and 'Libre'), and 'Área de formación' (with options like 'Arquitectura', 'Zootecnología', 'Área de Ciencias y Tecnologías de la Vida', 'Óptica', 'Estructuras de Computación', 'Electrónica y Tecnologías del Desarrollo', and 'Ciencias y Tecnologías de la Ingeniería').
- Solicitar Entrevista:** A form for scheduling an interview. It includes fields for 'Nombre completo' (David Rodríguez Guadiana), 'Email' (davidrodriguezguadiana@gmail.com), 'Número' (+52 2212000001), 'Fecha de nacimiento' (12/03/1995), 'Número de proyecto' (1234567890), 'Carrera' (Electrónica), 'Nombre del proyecto' (Tesis de una Red Profunda para Clasificación de Sonidos por la Explotación combinación de descriptores en entornos artificiales y naturales), 'Motivo' (Prestar al proyecto de Tesis de una Red Profunda para Clasificación de Sonidos por la Explotación combinación de descriptores en entornos artificiales y naturales. La oportunidad de contribuir a la comprensión de sistemas de generación, análisis en tiempo real de señales acústicas que podrían ser utilizadas para detectar más enfermedades a sede temprana.), 'Tipo de entrevista' (Virtual), 'Fecha de entrevista' (November 2018 calendar showing the 30th), 'Hora de entrevista' (13:30), 'Lugar e tipo de entrevista' (Videoconferencia), and 'Comentarios' (None).
- Área:** A section at the bottom with the text 'Solicitar Entrevista' and the note 'Completa el formulario de solicitud de entrevista en para iniciar conversación con tu supervisor o profesor'.

Nota: Interfaz del responsable del proyecto para visualizar a los aspirantes que eligieron su proyecto.
(Ximena, 2023)

En la figura 39 se mostrará la misma interfaz el responsable del proyecto, pero en este caso ya realizada la entrevista evaluará al aspirante para ver si cumple con los sus requisitos para el proyecto, sea rechazado o aceptado se le notificará al aspirante con un correo.

Figura 39 Interfaz del responsable del proyecto para la evaluación de entrevista



Nota: Interfaz del responsable del proyecto aceptar o rechazar al aspirante. (Ximena.2023)

En la figura 40 solo le aparecerá al aspirante si es aceptado para el proyecto, la interfaz muestra el registro para seguir con el procedimiento de ser aceptado como un alumno externo. Se llenarán los campos que con la primera información del primer pre-registro y ya solo se pedirán otros datos importantes. Tendrá el botón de modificar los datos si hubo un error y el botón de registrar el cual al momento se le enviará una alerta y correo a Dirección de Formación Académica de que ya hay un Alumno externo por registrar.

Figura 40 Interfaz del aspirante registro oficial

Este diagrama ilustra el flujo de trabajo para el registro oficial de un aspirante. Se divide en tres secciones principales:

- Formulario Principal:** Muestra los campos para los datos personales (Nombre: Mariana Martínez Flores, Fecha de nacimiento: 01/01/1990, Género: Seleccionar su género), datos académicos (Universidad: Universidad Interamericana del Estado de Puebla Alcaldía, Nivel educativo: Preparatoria, Carrera: Psicología, Carrera institucional vigente: Psicología, Carta presentación: Subir), y datos de contacto (Teléfono: +52 222 123 4567, Contacto de emergencia: 100, Correo electrónico: marianamartinezTA@gmail.com). Los botones "Siguiente" y "Finalizar" están disponibles.
- Confirmación de datos:** Una pantalla que muestra los datos ingresados y una lista de errores o advertencias. Los errores incluyen:
 - Nombre: Mariana Martínez Flores
 - Fecha de nacimiento: 01/01/1990
 - Género: Seleccionar su género
 - Carrera: Psicología
 - Carrera institucional vigente: Psicología
 - Carta presentación: Subir
 - Teléfono: +52 222 123 4567
 - Contacto de emergencia: 100
 - Correo electrónico: marianamartinezTA@gmail.com
- Sistema de seguimiento:** Muestra dos paneles:
 - DFA:** Notificación: Alumno pendiente por registrar como Alumno externo.
 - SPD:** Notificación: Alumno pendiente por registrar como Alumno externo.

Un cuadro de texto detallado explica el proceso de envío de datos:

El sistema no solo se procesan los datos que se corregirán los errores; en caso de que el aspirante no haya cumplido la información de matrícula establecida. Una vez que el usuario hace clic en "Finalizar", se le dirá que el formulario correspondiente ha sido enviado al formulario correspondiente con los datos previamente ingresados, así como se le recordará la necesidad de registrar la información nuevamente.

Nota: El aspirante que fue aceptado podrá hacer su primer registro oficial para ser aceptado.

La figura 41 presenta la interfaz de la Dirección de Formación Académica (DFA), donde se lleva a cabo la verificación de los datos del aspirante. Esta validación es crucial para la generación de la carta de aceptación que permitirá al individuo ser admitido en el instituto. En caso de que se detecten errores en la información proporcionada, se enviará un correo electrónico al aspirante para notificarle la necesidad de realizar los cambios correspondientes. Este proceso de retroalimentación se repetirá hasta que los datos del solicitante sean precisos y completos.

Figura 41 Interfaz de Dirección de Formación Académica



Nota: La DFA verificará si los datos del registro son correctos para generar su carta de aceptación y sea formalmente aceptado.

9. Conclusiones, recomendaciones y experiencia profesional adquirida

9.1. Conclusiones

En la culminación de este proyecto de Sistema de Vinculación Universitaria, es evidente que hemos alcanzado metas significativas que no solo mejoran la eficiencia operativa de nuestra institución educativa, sino que también fortalecen los lazos entre la universidad, sus estudiantes y las entidades colaboradoras. Durante el transcurso de este esfuerzo, hemos experimentado de primera mano cómo la implementación de un sistema moderno y centrado en el usuario puede transformar la manera en que interactuamos y colaboramos dentro de nuestra comunidad educativa.

Los beneficios de este sistema son palpables, desde la simplificación de procesos administrativos hasta la mejora de la experiencia del estudiante en su búsqueda de oportunidades de vinculación con el entorno profesional. La integración de herramientas de DevOps ha permitido un desarrollo ágil y despliegue eficiente, garantizando la entrega puntual de nuevas funcionalidades y la rápida adaptación a las necesidades cambiantes.

En conclusión, el Sistema de Vinculación Universitaria no es simplemente un logro puntual, sino más bien un hito en constante evolución que ha transformado la forma en que nuestra institución se relaciona con el mundo que la rodea. Este proyecto es un testimonio de nuestro compromiso con la mejora continua y la excelencia en la experiencia educativa.

9.2. Recomendaciones

Las recomendaciones, o sugerencias de mejora, son esenciales para guiar a la organización en la evolución y optimización continua del proyecto de Sistema de Vinculación Universitaria.

- Hacer del conocimiento a toda comunidad de la propuesta del sistema de Vinculación Educativa con el objetivo de incentivar el uso.
- Proporcionar los recursos Humanos para el desarrollo total de la aplicación.
- Proporcionar el recurso tecnológico (Almacenamiento, pólizas de software, etc.) para poder brindar la capacidad tecnológica en los siguientes años.

Estas recomendaciones forman una base para la mejora continua y el perfeccionamiento del Sistema de Vinculación Universitaria, asegurando que evolucione de manera efectiva para satisfacer las necesidades cambiantes de la comunidad universitaria.

9.3. Experiencia profesional

Durante mi experiencia profesional en este proyecto, tuve la oportunidad de adquirir valiosos conocimientos en diversos ámbitos. La implementación de una nueva metodología ágil marcó el inicio de este aprendizaje, permitiéndome comprender y aplicar prácticas más eficientes en el desarrollo de software.

La utilización de herramientas de software de última generación fue esencial para el proyecto, brindándome la oportunidad de familiarizarme con tecnologías actuales y mejorar mis habilidades técnicas. Además, la ingeniería de requerimientos detallada se convirtió en una parte fundamental del proceso, enseñándome la importancia de una planificación precisa en la ejecución de proyectos complejos.

A lo largo del desarrollo y despliegue, exploré una amplia gama de conceptos que enriquecieron mi comprensión del campo. Esta experiencia no solo se limitó al ámbito técnico, sino que también impulsó mi capacidad de investigación y análisis. La aplicación práctica de los conceptos y conocimientos adquiridos resultó fundamental para el éxito del proyecto.

10. Competencias desarrolladas y/o aplicadas

- Adquirí habilidades en la implementación de metodologías ágiles, facilitando la adaptabilidad y la entrega incremental de resultados.
- Mejoré mis habilidades en el desarrollo de software mediante la utilización de herramientas y tecnologías de última generación, especialmente aquellas asociadas con DevOps.
- Desarrollé la capacidad de realizar una ingeniería de requerimientos detallada, asegurando una comprensión precisa de las necesidades del proyecto.
- Implementé prácticas de Despliegue Continuo (CI/CD), automatizando procesos y acelerando el ciclo de vida del desarrollo de software.
- Aprendí a manejar grandes conjuntos de datos de manera eficiente, garantizando la seguridad y la integridad de la información.
- Se incentivó y fortaleció mi capacidad de investigación y análisis, crucial para abordar desafíos y tomar decisiones fundamentadas.
- Desarrollé habilidades para colaborar de manera efectiva con profesionales de diferentes disciplinas, optimizando la comunicación y el trabajo en equipo.
- Fomenté una mentalidad orientada a resultados, asegurando que cada tarea y actividad contribuyera al logro de los objetivos del proyecto.
- Desarrollé la capacidad de gestionar cambios de manera eficaz, adaptándome a nuevas circunstancias y optimizando procesos según las necesidades del proyecto.

11. Fuentes de información

Introducción al Ciclo de Vida de DevOps. (s/f). Kranio.io. <https://www.kranio.io/blog/introduccion-al-ciclo-de-vida-de-devops>

Sánchez, J. E. C. (s/f). *Dev Sec Ops: Implementación de seguridad en Dev Ops a través de herramientas open source.* Uoc.edu. <https://openaccess.uoc.edu/bitstream/10609/126766/11/jcastrosancheTFM122Omemoria.pdf>

Valenzuela, R., & Hernán, J. (2019). *Plataforma de desarrollo de aplicaciones en el DCC basada en técnicas de DevOps.* <https://repositorio.uchile.cl/handle/2250/170717>

Introducción al Ciclo de Vida de DevOps. (s/f). Kranio.io. <https://www.kranio.io/blog/introduccion-al-ciclo-de-vida-de-devops>

Metodología DevOps: qué es y en qué consiste. (2020, noviembre 13). Inesdi. <https://www.inesdi.com/blog/metodologia-devops-que-es-y-en-que-consiste/>

Melys, I. D., & Delgado, B. (s/f). EL PARADIGMA DEVOPS Y SUIMPLEMENTACIÓN EN EL DESARROLLO DE SOFTWARE THE DEVOPS PARADIGM AND ITS IMLEMENTATION IN THE SOFTWARE DEVELOPMENT. Unica.cu.

<https://revistas.unica.cu/index.php/uciencia/article/download/1604/pdf/8871>

De, S. (s/f). Miguel Ortuño Escuela Técnica Superior de Ingeniería deTelecomunicación. Urjc.es. <https://gsyc.urjc.es/~mortuno/lagrs/vagrant.pdf>

¿Qué es Vagrant y para qué sirve? (2022, febrero 25). KeepCoding Bootcamps. <https://keepcoding.io/blog/que-es-vagrant-y-para-que-sirve/>

Juana, R. (2019, diciembre 3). Cinco herramientas para automatizar tuinfraestructura IT. MuyComputerPRO. <https://www.muycomputerpro.com/2019/12/03/cinco-herramientas-para-automatizar-tu-infraestructura-it>

Dominguez-Quintero, L., & Vargas-Lombardo, M. (s/f). Herramientas de infraestructura como código: ansible, terrafom, chef, puppet. Revista de I+D Tecnológico, 17(2). <http://portal.amelica.org/ameli/journal/339/3392429002/html/>

Carrero, L. (2023, junio 29). 31 distribuciones de Linux populares. Stackscale.

<https://www.stackscale.com/es/blog/distribuciones-linux-populares/>

Mbaabu, M. H. (2021, agosto 16). DevOps 101 : Introduction to Ansible. DEV Community.

<https://dev.to/grayhat/devops-101-introduction-to-ansible-1n64>

il, M. B. (2023, agosto 18). Análisis de las ventajas y desventajas de Laravel: Todo lo que debes saber. Ventajas y desventajas top. <https://ventajasydesventajastop.com/laravel-ventajas-y-desventajas/>

NetApp. (2019). ¿Qué es DevOps y para qué sirve? | NetApp. <https://www.netapp.com/es/devops-solutions/what-is-devops/>

¿Qué es un IDE? - Explicación de los entornos de Desarrollo Integrado - AWS. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/ide/> Sydle. (2024, 11 enero). Automatización de procesos: ¿cómo funciona? ¿Cuáles son los beneficios? Blog SYDLE. <https://www.sydle.com/es/blog/automatizacion-de-procesos-6070ae4c9b901904c4349dcb>

Integración continua del software | Pruebas automatizadas | AWS. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/devops/continuous-integration/#:~:text=La%20integraci%C3%B3n%20continua%20se%20refiere%20a%20la%20fase%20de%20creaci%C3%B3n,la%20creaci%C3%B3n%20y%20las%20pruebas.>

¿Qué es la entrega continua? – Amazon Web Services. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/devops/continuous-delivery/>

Arsys. (2023, May 23). Herramientas imprescindibles para DevOps. Blog De arsys.es. <https://www.arsys.es/blog/herramientas-devops>

Equipo editorial, Etecé. (2023, November 19). Sistema Operativo - Concepto, usos, tipos, funciones y ejemplos. Concepto. <https://concepto.de/sistema-operativo/>

Manjaly, S. (2023, 5 abril). Qué es Ansible: la herramienta DevOps para automatizar tareas de IT. Qué es Ansible: la herramienta DevOps para automatizar tareas de IT. <https://blog.invgate.com/es/ansible>

Documentation for Visual Studio code. (2021, November 3). <https://code.visualstudio.com/docs>

López, M. (2023, October 18). *Qué es PHP | Definición, Significado y Ejemplos*. Arimetrics.

<https://www.arimetrics.com/glosario-digital/php>

¿Qué es JavaScript? - Aprende desarrollo web | MDN. (n.d.). MDN Web Docs.

https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript

LaRavel - the PHP framework for web artisans. (n.d.). <https://laravel.com/docs/10.x>

Martínez, J. (2023, April 27). Qué es Grafana y primeros pasos. OpenWebinars.net.

<https://openwebinars.net/blog/que-es-grafana-y-primeros-pasos/>

Macchia, G. O. (2023, 12 enero). DevOps - Germán Osses Macchia - Medium. Medium.

<https://medium.com/@germanmacchia/devops-79a580fde534>