UNIVERSIDAD CENTRAL DEL ECUADOR

FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

TEMA:

DESARROLLO DE UN SISTEMA CLOUD NATIVE PARA LA GESTIÓN DEL PROCESO DE PLAN DE TITULACIÓN EN LA FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS DE LA UNIVERSIDAD CENTRAL DEL ECUADOR.

PLAN DEL TRABAJO DE TITULACIÓN, MODALIDAD PROYECTO INTEGRADOR

PRESENTADO COMO REQUISITO PARA APROBAR EL TRABAJO DE TITULACIÓN, PARA OPTAR EL TÍTULO DE INGENIERO EN CIENCIAS DE LA COMPUTACIÓN.

AUTORES:
LUIS FERNANDO MOSQUERA ROMERO
PABLO ALEXIS SUNTAXI HEREDIA

PROFESOR GUÍA
ING. JAIME SALVADOR

QUITO – ECUADOR

2024

ACEPTACIÓN DEL TUTOR

Por la presente dejo constancia que he leído el plan del trabajo de titulación, modalidad proyecto integrador, cuyo título es **DESARROLLO DE UN SISTEMA CLOUD NATIVE PARA LA GESTIÓN DEL PROCESO DE PLAN DE TITULACIÓN EN LA FACULTAD DE INGENIER**ÍA Y CIENCIAS APLICADAS DE LA UNIVERSIDAD CENTRAL DEL ECUADOR, presentado por los Sr. LUIS FERNANDO MOSQUERA ROMERO y PABLO ALEXIS SUNTAXI HEREDIA.

Y en mi calidad de Tutor/a del Trabajo de Titulación, acepto asesorar a la estudiante en calidad de Tutor, durante el desarrollo del trabajo hasta la elaboración del informe final, presentación y evaluación.

Ing. JAIME SALVADOR M.
DOCENTE-TUTOR/A
C.C. 1002437521

Señor Ingeniero

GEOVANNY MONCAYO, MSc

DIRECTOR DE LA CARRERA DE INGENIERÍA EN COMPUTACIÓN

Presente

De mis consideraciones

Por medio de la presente, yo con número de identificación 1750958843, estudiante de la Facultad de Ingeniería y Ciencias Aplicadas en la carrera de Ingeniería en Computación, me dirijo a usted para entregar el plan de trabajo de titulación perteneciente al período 2024-2024.

Por la favorable respuesta que se digne dar a la presente, agradezco su gentil atención.

Atentamente,

Nombre Apellido
CI: 1750958843

PLAN DEL TRABAJO DE TITULACIÓN

CONTENIDO (INDICE)

1. Resumen

2. Análisis del problema

2.1. Antecedentes

Basado en el análisis de los desafíos actuales del proceso de titulación en la Universidad Central del Ecuador, se identifican varias deficiencias significativas que afectan la eficiencia y la transparencia del sistema. Entre estas deficiencias se incluyen la falta de automatización, la documentación inadecuada, la lentitud en los procesos, la falta de control a futuro, la limitada administrabilidad y las notificaciones ineficientes hacia los usuarios involucrados. Estas limitaciones han generado una necesidad urgente de mejorar y modernizar el manejo del proceso de titulación mediante la implementación de soluciones tecnológicas avanzadas.

La Facultad de Ingeniería y Ciencias Aplicadas de la Universidad Central del Ecuador no cuenta con un sistema sobre control del PLAN DEL TRABAJO DE TITULACIÓN, lo que con lleva a demoras e incertidumbre del proceso. Actualmente se cuenta con una hoja electrónica (convocatoria, carrera, estudiantes, tema, tutores y revisores) de una forma manual sin saber el estado real en el que se encuentra, por lo que se debe consultar de forma individual, ya que la información no se encuentra centralizada.

Al realizar todo este proceso de forma manual es susceptible a que se comentan errores humanos en el manejo de estas hojas electrónicas. Este tipo de errores producen en tiempo adicional en la aprobación del tema de titulación por tal motivo repercute al estudiante en el ámbito retraso profesional, económico y educativo; la facultad de igual forma se ve afectada ya que no se aprueba en el periodo inscrito lo que con lleva un control extra en la parte administrativa.

Al no tener una gestión eficiente con la aprobación del plan de titulación no se puede realizar el TRABAJO DE TITULACIÓN en el tiempo idóneo (6 meses), por lo que se debe pagar por prorroga, como consta en el instructivo de titulación. Este tiempo de espera el estudiante busca otro tipo de actividad, económica o educativa (cursos-talleres) por consiguiente no va a dedicar el 100% de su tiempo a la culminación del TRABAJO DE TITULACIÓN.

Las aplicaciones Cloud Native resultan ser una opción atractiva para abordar estas necesidades, debido a que ofrece plataformas robustas y escalables, diseñadas para soportar cambios en la carga de trabajo, ya sea si se presentan picos altos de tráfico o periodos de baja actividad (Vitale & Long, 2022).

2.2. Planteamiento del Problema

La Facultad de Ingeniería y Ciencias Aplicadas de la Universidad Central del Ecuador enfrenta un desafío significativo en la gestión del proceso de titulación de sus estudiantes. Actualmente, este proceso se lleva a cabo de manera manual y con el apoyo de hojas de cálculo de Excel, lo cual genera varios inconvenientes y retrasos. La gestión manual y el uso de hojas de cálculo resultan en procedimientos lentos y propensos a errores, lo que

incrementa el tiempo necesario para completar las tareas administrativas relacionadas con la titulación.

Además, los estudiantes y el personal administrativo tienen dificultades para realizar un seguimiento efectivo del estado de los procesos de titulación. Esto se debe a la dispersión de la información y la falta de un sistema centralizado y accesible en tiempo real. La manipulación manual de los datos y el uso de hojas de cálculo aumentan el riesgo de errores y duplicación de información, lo que puede llevar a inconsistencias y confusiones en el registro de los avances y requerimientos de titulación.

Hay periodos de tiempo muertos significativos durante el proceso de titulación debido a la espera de aprobaciones, la validación de documentos y la actualización de la información en las hojas de cálculo. Estos tiempos muertos afectan la capacidad de los estudiantes para graduarse a tiempo. Además, el sistema actual no se adapta fácilmente a un incremento en la cantidad de estudiantes o a cambios en los requisitos de titulación, lo que limita la capacidad de la universidad para manejar eficientemente el proceso a medida que la demanda crece.

Estas dificultades no solo afectan la experiencia de los estudiantes, retrasando su graduación, sino que también incrementan la carga de trabajo del personal administrativo, afectando la eficiencia operativa de la universidad. Es evidente la necesidad de un sistema más eficiente y confiable que pueda ser gestionado de manera integral en la nube, mejorando así la eficiencia, la transparencia y la experiencia tanto de los estudiantes como del personal administrativo.

3. Investigación Bibliográfica (PorRealizarse)

Diseño de un Sistema de Planificación de Recursos Empresariales (ERP) para una Microempresa (Acosta Vega et al., 2017)

Este artículo aborda la implementación de un sistema informático para la planificación de recursos empresariales (ERP) de una microempresa del sector textil, siguiendo una metodología estructurada para su desarrollo. Resultando en el diseño completo del sistema y su implementación parcial en la organización, que por falta de recursos económicos se implementaron los módulos más relevantes para el mejoramiento competitivo de las empresas. En este proyecto las fases de modelización y parametrización fueron claves para la construcción adecuada de sistema.

Diseño de un sistema para la gestión de inventarios de las PYMES en el sector alimentario (Carreño Dueñas et al., 2019)

Este proyecto se centra en la implementación de un sistema informático para la gestión de ventas e inventario para una microempresa dedicada a la venta de productos básicos llamada “La Económica”, que trabajaba en su totalidad con procesos manuales. Para esto se realizó una investigación para determinar y evaluar los procesos actuales de la empresa, con lo que se logró proponer una solución que mejorase la eficiencia en la gestión empresarial y contribuyese al crecimiento del negocio. El sistema cuenta con una base de

datos relacional y lenguaje de programación Java para backend y para el frontend se utilizó HTML junto a Java Script.

Aplicación web para gestionar las ventas de la microempresa Vida Sana en el cantón El Carmen utilizando la metodología Design Thinking (Benavides Rivera & Buñay Guisñan, 2024)

Este proyecto consistió en el desarrollo de una aplicación web para la gestión de ventas de la microempresa "Vida Sana" en el cantón El Carmen en Manabí. Se utilizó la metodología Design Thinking compuesta de cinco fases para comprender los requerimientos del usuario a lo largo del desarrollo.

Taming the Complexity of Elasticity, Scalability and Transferability in Cloud Computing Cloud-Native Applications for SMEs (Quint & Kratzke, 2017)

El artículo resalta el potencial de la computación en la nube para empresas y señala los inconvenientes de las MIPYMES en la implementación de aplicaciones Cloud Native como la falta de recursos. Se presenta el sistema C4S de código abierto para que estas empresas desplieguen y operen su aplicación en contenedores con características como autoescalado, balanceador de carga y elasticidad, también soporta la migración de contenedores entre las diferentes plataformas de Infraestructura como Servicio (IaaS).

4. Justificación

La implementación de un Sistema Integral de Gestión nativo en la nube para el proceso de titulación en la Universidad Central del Ecuador representa una oportunidad significativa para aprovechar las ventajas que ofrece esta tecnología. Un sistema integral basado en tecnologías cloud-native permitirá centralizar y organizar la información de manera efectiva, proporcionando acceso rápido y seguro a los datos necesarios para todos los actores involucrados.

Además, los extensos periodos en la gestión de las solicitudes puede ser un desafío para los estudiantes y el personal administrativo. La implementación de un sistema automatizado reducirá significativamente el tiempo de procesamiento, optimizando el flujo de trabajo y permitiendo una respuesta más ágil y eficiente.

Otro aspecto crucial es la necesidad de un control y seguimiento a futuro. Un sistema de gestión basado en microservicios permitirá monitorear y planificar de manera continua y precisa el progreso de los estudiantes, facilitando una administración proactiva y efectiva del proceso de titulación.

La administración y las notificaciones son también áreas que se beneficiarán enormemente de un sistema moderno. La integración de mecanismos de notificación automáticos garantizará que los estudiantes, tutores y el personal administrativo reciban información oportuna y precisa sobre el estado de sus trámites y tareas pendientes, mejorando la comunicación y reduciendo la incertidumbre.

5. Objetivos

5.1. Objetivo General

Desarrollar un Sistema Integral de Gestión cloud native para la optimización del proceso de titulación de la Facultad de Ingeniería y Ciencias Aplicadas en la Universidad Central del Ecuador, mejorando la eficiencia, transparencia y escalabilidad de los procedimientos administrativos y académicos relacionados.

5.2.Objetivos Específicos

- Analizar los requerimientos y necesidades de la Facultad de Ingeniería y Ciencias Aplicadas de la Universidad Central del Ecuador en cuanto a la gestión del proceso de titulación (Plan de trabajo de Titulación), basado en el marco del Reglamento de Régimen Académico y la normativa interna de la Universidad Central del Ecuador.
- Diseñar la arquitectura del Sistema Integral de Gestión cloud native para la optimización del proceso de titulación en la Universidad Central del Ecuador, definiendo los componentes, módulos, bases de datos, interfaces.
- Implementar un Sistema Integral de Gestión cloud napara optimizar el proceso de titulación en la Universidad Central del Ecuador, desarrollando una aplicación web que utilice tecnologías y frameworks modernos para asegurar un funcionamiento eficiente, robusto y seguro.

- Desarrollar una interfaz intuitiva y fácil de usar que requiera un mínimo de capacitación para los usuarios.
- Implementar funcionalidades de seguridad informática para proteger la integridad y confidencialidad de los datos del sistema.

6. Alcance

En este proyecto se propone desarrollar una aplicación Cloud Native para la gestión del proceso del plan de titulación para la Facultad de Ingeniería y Ciencias Aplicadas de la Universidad Central del Ecuador, que contempla el proceso de la aprobación del plan de titulación. Además, se implementará un sistema de notificaciones automáticas para informar a los revisores y estudiantes sobre asignaciones, fechas límite y recordatorios. También se mantendrá un historial de todas las asignaciones y se permitirá el seguimiento del progreso de cada revisión, asegurando transparencia y eficiencia en el proceso.

Se desarrollará un portal en línea donde se podrá presentar solicitudes de titulación, cargar documentos y realizar seguimientos del estado de su solicitud. Este portal incluirá validaciones automáticas de los requisitos para garantizar que las solicitudes cumplan con todos los criterios antes de ser enviadas para revisión.

La optimización del proceso de aprobación es un aspecto crucial. Se crearán flujos de trabajo automatizados que guiarán a los documentos a través de los distintos niveles de aprobación, desde la presentación hasta la aprobación final. Esto reducirá los tiempos de espera y mejorará la eficiencia del proceso.

Para el almacenamiento de información, se establecerá una base de datos relacional centralizada y segura que albergará toda la información relacionada con el proceso de titulación. El acceso a esta base de datos será controlado estrictamente para garantizar la integridad de la información.

Se mejorará comunicación y colaboración con la creación de una plataforma integrada que permita la comunicación fluida entre estudiantes, revisores y administradores. Esta plataforma facilitará la colaboración en tiempo real entre revisores y estudiantes, mejorando la calidad de las revisiones y acelerando el proceso de titulación.

Se implementarán herramientas para la generación de reportes detallados sobre el progreso de las solicitudes, el desempeño de los revisores y los tiempos de aprobación. Además, se utilizará análisis de datos para identificar cuellos de botella en el proceso y áreas de mejora, permitiendo ajustes y optimizaciones continuas.

El diseño de la interfaz de usuario será intuitivo y fácil de usar tanto para estudiantes como para el personal administrativo. La plataforma será accesible desde múltiples dispositivos y navegadores, asegurando que todos los usuarios puedan acceder a ella sin inconvenientes.

Por consiguiente, se desarrollará el backend de la aplicación utilizando un framework específico para el desarrollo de aplicaciones Cloud Native. El frontend será desarrollado con Angular para proporcionar una interfaz de usuario intuitiva y estará conectado con un API REST. El sistema se diseñará siguiendo los principios de Cloud Native.

Finalmente, se ofrecerán programas de capacitación y un manual de usuario para que los usuarios comprendan cómo utilizar la nueva plataforma de manera efectiva. Esto garantizará una integración rápida y eficiente al nuevo sistema, mejorando así el proceso de titulación.

7. Limitaciones

- El sistema estará enfocado en la gestión interna del proceso de aprobación del plan de titulación para la Facultad de Ingeniería y Ciencias Aplicadas, por lo que no se contemplará la integración con otros sistemas externos.
- Se desarrollarán reportes por periodo académico, y por estado de aprobación.
- La aplicación será responsiva, lo que permitirá su uso en todo tipo de dispositivos, incluyendo PC y dispositivos móviles.
- Se limita al despliegue en un único entorno de nube pública o privada.

8. Marco teórico

8.1. Tecnología en la gestión de procesos burocráticos educativos.

Según (Huamantumba & Delgado Bardales, 2020) en el contexto de las universidades públicas, la gestión de procesos burocráticos ha sido identificada como un área crítica que

requiere simplificación y modernización. La investigación sugiere que la incorporación de tecnología en estos procesos puede significativamente mejorar la eficiencia, reducir costos y aumentar la competitividad institucional.

La simplificación administrativa en las universidades públicas se refiere a la reducción de tiempos de servicio, la optimización de procesos administrativos y la disminución de costos. Según (Huamantumba & Delgado Bardales, 2020) el 70% de las investigaciones revisadas destacan que la eficiencia en los servicios y la reducción de cargas documentales son esenciales para cumplir con estos objetivos. El otro 30% sugiere que la integración de tecnología virtual es crucial. Esta integración debe estar fundamentada en políticas internas que faciliten la implementación de trámites más accesibles y menos costosos, promoviendo el desarrollo institucional.

Los beneficios de modernizar la gestión administrativa mediante tecnología son numerosos: la automatización de procesos permite realizar trámites de manera más rápida y con menos recursos, lo que se traduce en ahorros significativos tanto para la institución como para los usuarios. La tecnología permite ofrecer servicios más oportunos y personalizados, mejorando la satisfacción del usuario. Las instituciones con procesos administrativos eficientes y modernos pueden responder mejor a las demandas de estudiantes y profesores, posicionándose mejor frente a otras instituciones educativas; la tecnología no solo mejora los procesos administrativos, sino que también puede ser un motor para el desarrollo institucional, facilitando la implementación de nuevas políticas y programas académicos (Huamantumba & Delgado Bardales, 2020).

La simplificación administrativa, acompañada de tecnología, es fundamental para el desarrollo de las universidades públicas. La planificación basada en políticas internas, junto con la adopción de herramientas tecnológicas, puede transformar la gestión de procesos burocráticos, reduciendo tiempos y costos, y mejorando la competitividad y la calidad del servicio. Es imperativo que las instituciones educativas adopten estas estrategias para enfrentar los desafíos actuales y futuros en la gestión educativa.

8.2.Computación en la nube FER

El Programa de Computación en la Nube (NCCP, por sus siglas en inglés) del NIST se formó en mayo de 2010 con el propósito de fomentar y garantizar la adopción segura y efectiva de la computación en la nube en el Gobierno de los Estados Unidos mediante el examen de los requisitos estratégicos de alta prioridad en seguridad, interoperabilidad y portabilidad. Para ello, examinó el panorama tecnológico actual para determinar las normas, las orientaciones y la tecnología pertinentes que se necesitan para satisfacer los requisitos (get_pdf.cfm (nist.gov)).

La computación en la nube reduce significativamente los gastos de capital relacionados con la compra de bienes raíces para los centros de datos y la inversión en infraestructura de TI y, en su lugar, cambia los modelos de costos a una estrategia de gastos operativos (OpEx) (https://csrc.nist.gov/Projects/cloud-computing/events).

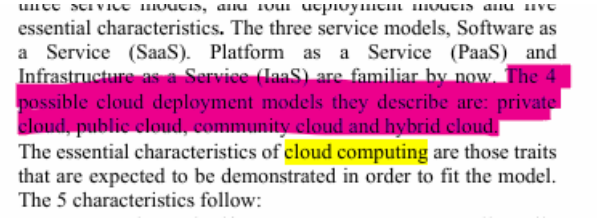
Las características esenciales de la computación en la nube son autoservicio bajo demanda, acceso amplio a la red, agrupación de recursos, la elasticidad y, por último, tenemos el pago por uso, donde el usuario solo debe pagar por los servicios consumidos

durante un determinado periodo de tiempo. (The Path to Cloud Federation through Standardization | NIST).

La computación en la nube ha revolucionado los métodos mediante los cuales se almacenan, procesan y transmiten los datos digitales (NIST.IR.8006.pdf). Es una tecnología que permite acceder a recursos informáticos (como almacenamiento, servidores y aplicaciones) a través de internet, en lugar de usar hardware y software locales. Proporciona una solución ágil, escalable y rentable para empresas y usuarios, facilitando el acceso rápido a tecnologías innovadoras, mejorando la productividad y reduciendo costos operativos.

8.3. Modelos de despliegue PABLO

Según la NIST (Lee et al., 2020), existen cuatro modelos de implementación para los servicios en la nube: Nube Privada, Nube Pública, Nube Híbrida y Nube Comunitaria.


three service models, and four deployment models and five essential characteristics. The three service models, Software as a Service (SaaS). Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) are familiar by now. The 4 possible cloud deployment models they describe are: private cloud, public cloud, community cloud and hybrid cloud. The essential characteristics of cloud computing are those traits that are expected to be demonstrated in order to fit the model. The 5 characteristics follow:

The Path to Cloud Federation through Standardization | NIST

- **Nube privada:** La infraestructura de la nube es proporcionada para ser utilizada por una sola organización, puede ser gestionada por la propia organización o por un tercero y puede estar alojada en las instalaciones o en un lugar externo (Vitale & Long, 2022).

- **Nube pública:** La infraestructura de la nube se proporciona para uso abierto por el público en general. Suele ser propiedad del proveedor de la nube, quien la administra y opera. Se aloja en las instalaciones del proveedor (Mell & Grance, 2011).

- **Nube Comunitaria:** La infraestructura de la nube es proporcionada para uso exclusivo de una comunidad de consumidores que tienen necesidades en común, como requisitos de seguridad, políticas, entre otros (Mell & Grance, 2011).

- **Nube Híbrida:** Es una combinación de dos o tres de los modelos descritos anteriormente, brindando servicios como si se tratara de un entorno único (Vitale & Long, 2022).

8.4. Modelos de Servicios en Cloud Computing FER

Cloud Computing está cambiando la infraestructura empresarial y de negocios. Las organizaciones ya no necesitan alojar sistemas de información en servidores internos ni alquilar espacios en rack en centros de datos (https://csrc.nist.gov/Projects/cloud-computing/events).

Los principales modelos de servicios en la nube según la NIST (Mell & Grance, 2011) son tres: Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS). Adicionalmente, podemos destacar los modelos: Container as a Service (CaaS) y Function as a Service (FaaS).

- **Software as a Service (SaaS):** Proporciona infraestructura de hardware y software para construir y mantener aplicaciones, generalmente a través de API. Los proveedores de la nube hospedan herramientas de desarrollo de hardware y software en sus centros de datos(https://aws.amazon.com/es/what-is/saas/).
- **Platform as a Service (PaaS):** Es un entorno completo de desarrollo e implementación en la nube, incluye infraestructura (servidores, almacenamiento y redes), pero también middleware, herramientas de desarrollo, servicios de inteligencia empresarial (BI), sistemas de gestión de bases de datos (What is PaaS? Platform as a Service | Microsoft Azure).
- **Infrastructure as a Service (IaaS):** Es la disponibilidad bajo demanda de recursos de computación muy escalables como servicios a través de Internet. De este modo, se elimina la necesidad de aprovisionar, configurar o gestionar los recursos y solo se paga por el uso que se hace de estos recursos (¿Qué es IaaS (infraestructura como servicio)? | Google Cloud).
- **Container as a Service (CaaS):** son servicios de nube que permiten gestionar e implementar las aplicaciones usando el aislamiento en contenedores, ya sea en las instalaciones o en la nube (¿Qué son los contenedores como servicio (CaaS)? (redhat.com)). Los proveedores de CaaS ofrecen innumerables funciones, incluidos (entre otros) tiempos de ejecución de contenedores, capas de orquestación y administración de almacenamiento persistente (¿Qué son los contenedores como servicio (CaaS)? | IBM). Kubernetes ha llegado a ser el estándar en la implementación de CaaS, ofrece las funciones de organización y gestión de los contenedores para implementarlos según sea necesario en varios hosts de servidores con diferentes capas de seguridad, mientras gestiona su estado a lo largo del tiempo (¿Qué son los contenedores como servicio (CaaS)? (redhat.com)).
- **Function as a Service (FaaS):** Es un subconjunto de las funciones sin servidor, es un servicio de cloud computing que permite a los clientes ejecutar código en respuesta a sucesos, sin gestionar la compleja infraestructura asociada normalmente a la creación y lanzamiento de aplicaciones de microservicios (¿Qué es la FaaS (función como servicio)? | IBM).

8.5. Aplicaciones Cloud Native PABLO

Según la Cloud Native Computing Foundation (CNCF) (CLOUD NATIVE COMPUTING FOUNDATION, n.d.-c), Cloud Native se refiere a las tecnologías nativas de la nube que capacitan a las organizaciones para construir y ejecutar aplicaciones escalables en entornos dinámicos como nubes públicas, privadas o híbridas. Estas tecnologías incluyen: contenedores, microservicios, APIs declarativas, entre otras, que permiten sistemas con bajo acoplamiento, resilientes, manejables y observables. Las

aplicaciones cloud native deben diseñarse específicamente para la nube y tener propiedades que puedan ser aprovechadas en el entorno de la nube. A pesar de que se pueda trasladar una aplicación tradicional a la nube, esto no hace que la aplicación sea nativa de la nube (Vitale & Long, 2022).

8.6. Propiedades de las Aplicaciones Cloud Native FER

Según (Who We Are | CNCF), Cloud Native permiten sistemas de bajo que son resilientes, manejables y observables. Combinados con una automatización robusta, permiten a los ingenieros realizar cambios de alto impacto con frecuencia y de manera predecible con un esfuerzo mínimo.

- **Escalabilidad:** Las aplicaciones Cloud Native deben escalar horizontalmente, agregando o eliminando nodos o contenedores en lugar de aumentar la potencia de las máquinas existentes.
- **Bajo acoplamiento:** Busca que las partes de un sistema tengan el menor conocimiento posible entre sí, para que permitir cambios sin afectar a otras partes.
- **Resiliencia:** Capacidad de sistema para mantener sus servicios siempre disponibles, incluso ante fallas o cambios en el entorno.
- **Manejabilidad:** Facilidad con la que se puede cambiar el comportamiento de una aplicación sin tener la necesidad de modificar su código.
- **Observabilidad:** Capacidad para entender el estado interno de un sistema a partir de sus salidas externas como métricas, registros y trazas.

8.7. Topologías Cloud Native PABLO

- **Contenedores:** Un contenedor empaqueta todo lo necesario para que una aplicación pueda ser ejecutada en cualquier lugar, incluyendo bibliotecas y dependencias. Los contenedores comparten el mismo kernel del sistema operativo del host y aprovechan las características del sistema operativo para la partición de recursos y controlar la cantidad de CPU, memoria y disco a la que pueden acceder (Vitale & Long, 2022) (IBM, n.d.). Consumen menos recursos y son portables, en comparación con las máquinas virtuales, que necesitan su propio sistema operativo. Pueden ejecutarse varios contenedores en una misma máquina distribuyendo la carga de recursos y creando un uso eficiente de la máquina física (CLOUD NATIVE COMPUTING FOUNDATION, n.d.-a).
- **Orquestación:** La orquestación de contenedores se encarga del manejo y automatización del ciclo de vida de una aplicación contenerizada. Un orquestador debe realizar despliegues, monitoreo, autoreparación y escalar automáticamente (CLOUD NATIVE COMPUTING FOUNDATION, n.d.-b). Funcionan con instrucciones declarativas, es decir, que indican qué hacer y no cómo hacerlo, por ejemplo, mediante un archivo YAML (Vitale & Long, 2022). El orquestador más común es Kubernetes.
- **Serverless:** Permite a los desarrolladores centrarse en la lógica de negocio de sus aplicaciones, sin preocuparse por la gestión de la infraestructura, ya que la

plataforma es la que se encarga de ello, incluyendo máquinas virtuales, contenedores y escalado dinámico (Vitale & Long, 2022).

8.8. Arquitecturas para aplicaciones Cloud Native FER

La arquitectura nativa en la nube combina componentes de software que los equipos de desarrollo utilizan para crear y ejecutar aplicaciones escalables nativas en la nube. La CNCF enumera la infraestructura inmutable, los microservicios, las API declarativas, los contenedores y las mallas de servicios como los bloques tecnológicos de la arquitectura nativa en la nube. (¿Qué es la tecnología nativa en la nube? - Explicación de la tecnología nativa en la nube - AWS (amazon.com)):

- **Microservicios:** Dividen la aplicación en componentes independientes con funcionalidades propias. Cada microservicio se ejecuta en su propio proceso y se comunica a través de protocolos ligeros, lo que permite una mayor escalabilidad y flexibilidad.
- **Infraestructura inmutable:** es un enfoque en el que los componentes de la infraestructura (como servidores, contenedores y máquinas virtuales) no se modifican una vez desplegados. En lugar de actualizar o cambiar estos componentes, se reemplazan completamente con versiones nuevas cuando se requiere una actualización o corrección**.**
- **API:** Permiten definir la configuración deseada del sistema en un archivo de manifiesto, y la plataforma de orquestación (como Kubernetes) se encarga de gestionar el estado actual para que coincida con la configuración deseada. Esto simplifica la gestión y el despliegue de aplicaciones.
- **Contenedores:** Una tecnología que permite empaquetar una aplicación junto con todas sus dependencias en un solo paquete ejecutable. Los contenedores garantizan que la aplicación se ejecute de manera consistente en diferentes entornos y facilitan la portabilidad, escalabilidad y aislamiento.
- **Malla de servicios:** Una capa de infraestructura dedicada para gestionar la comunicación entre microservicios. Proporcionan características como balanceo de carga, descubrimiento de servicios, encriptación de tráfico, autenticación, autorización y monitoreo.

8.9. Prácticas para el desarrollo de aplicaciones Cloud Native PABLO

El desarrollo Cloud Native se basa en tres conceptos clave: automatización, entrega continua y DevOps (Vitale & Long, 2022):

- **Automatización:** Elimina tareas manuales repetitivas, como la construcción y despliegue de aplicaciones, provisión y configuración de la infraestructura. Acelerando la entrega y despliegue de las aplicaciones.
- **Entrega continua:** Implementar características en ciclos cortos. Se basa en la integración continua (CI), donde los cambios se integran al menos una vez al

día, y en la entrega continua (CD), donde el software puede ser desplegado en cualquier momento.

- **DevOps:** Cultura de colaboración entre los diferentes roles como desarrolladores, testers, operadores y expertos en seguridad.

8.10. Principios para el desarrollo de aplicaciones Cloud Native: 12 Factores

La metodología de los 12 factores propone un conjunto de principios para diseñar y construir aplicaciones nativas de la nube, denominadas como web apps o *software as a service* (SaaS). Abordan características como: ser adecuadas para el despliegue en la nube, escalables y portables, facilitar la implementación continua y la agilidad. Los doce factores son los siguientes:

- **Código base (Codebase):** Representa el repositorio central de código gestionado por un sistema de control de versiones como Git. Es único para toda la aplicación, asegurando consistencia y un punto de referencia común para todos los despliegues. Cada despliegue constituye una instancia ejecutable de la aplicación en diferentes entornos, aunque pueden existir variaciones en las versiones desplegadas en cada entorno debido a comité específicos o desarrollos en curso, todos comparten la misma base de código. Esta estructura permite mantener la coherencia y facilita la gestión de versiones (Wiggins, 2017).
- **Dependencias:** La gestión de dependencias se enfoca en declarar y aislar explícitamente todas las librerías y herramientas necesarias mediante un manifiesto de dependencias. Esto asegura que la aplicación no dependa de paquetes instalados globalmente en el sistema operativo, sino que cada dependencia se especifique claramente y se gestione dentro del entorno de la aplicación (Wiggins, 2017).
- **Configuración:** Se centra en separar estrictamente la configuración del código base, almacenando todas las variables de configuración, como recursos de base de datos, credenciales de servicios externos y ajustes específicos de despliegue, en variables de entorno. Este enfoque asegura que la configuración pueda variar entre despliegues sin necesidad de modificar el código fuente (Wiggins, 2017).
- **Backing services:** Son tratados como recursos conectables esenciales para el funcionamiento de la aplicación. Estos servicios incluyen bases de datos, sistemas de mensajería y de colas, los servicios SMTP de email y otros servicios gestionados localmente o por terceros, accesibles a través de la red mediante URL o credenciales almacenadas en la configuración (Wiggins, 2017).
- **Construcción, distribución y ejecución**. La fase de construcción transforma el código base en una construcción ejecutable, incluyendo todas las dependencias y compilando los binarios necesarios. Luego, en la fase de distribución, esta construcción se combina con la configuración específica del entorno de despliegue para crear una distribución lista para ser ejecutada.

Finalmente, la fase de ejecución se encarga de lanzar la aplicación en el entorno de ejecución (Wiggins, 2017).

- **Proceso:** Las aplicaciones se ejecutan como uno o más procesos sin estado, lo que significa que cada proceso no retiene información persistente localmente. Cualquier dato que necesite persistencia debe ser almacenado en un "backing service" con estado, como una base de datos. Los procesos son diseñados para ser "share-nothing", compartiendo información solo a través de servicios conectables como bases de datos o sistemas de cache. Esto asegura que la aplicación sea escalable y robusta (Wiggins, 2017).
- **Asignación de puertos:** Cada aplicación web se configura para escuchar peticiones HTTP en un puerto específico, garantizando así su autocontención y portabilidad. Durante el desarrollo se accede a la aplicación mediante URLs locales que especifican el puerto asignado. En producción, una capa de enrutamiento maneja las solicitudes entrantes y las dirige al proceso correspondiente basado en la asignación de puertos (Wiggins, 2017).
- **Concurrencia:** Se maneja escalando mediante un modelo de procesos donde cada proceso es independiente y maneja un tipo específico de carga de trabajo. Los procesos en "twelve-factor" son livianos y se gestionan utilizando gestores de procesos del sistema operativo. Esto permite escalar horizontalmente los procesos según sea necesario, facilitando una gestión eficiente de la concurrencia y asegurando que la aplicación pueda dividirse en múltiples procesos ejecutándose en diferentes máquinas físicas si es necesario (Wiggins, 2017).
- **Desechabilidad:** Los procesos son diseñados para ser desechables, lo que permite iniciar y finalizar de manera rápida y segura según sea necesario. Esto facilita un escalado ágil y flexible, así como despliegues rápidos de código y configuraciones (Wiggins, 2017).
- **Paridad entre desarrollo y producción:** Se busca reducir al mínimo las diferencias entre los entornos de desarrollo, preproducción y producción. Esto se logra mediante prácticas como despliegues continuos que permiten a los desarrolladores tener su código en producción en cuestión de horas o minutos (Wiggins, 2017).
- **Historiales:** Los historiales de la aplicación se tratan como transmisiones continuas de eventos, capturadas desde la salida estándar de todos los procesos y servicios en ejecución. Estos eventos son registros ordenados y en formato de texto, sin un principio o final definido, que fluyen constantemente mientras la aplicación está activa. Pueden ser consultados, analizados y utilizados para generar alertas y gráficos de tendencia (Wiggins, 2017).
- **Administración de procesos:** Las tareas de gestión y administración, como ejecutar migraciones de base de datos o scripts de mantenimiento, se deben manejar como procesos únicos que se ejecutan una sola vez. Estos procesos deben operar en un entorno idéntico al de la aplicación en producción, utilizando el mismo código base y configuración. Es crucial que estos procesos estén integrados en el repositorio de la aplicación para mantener la coherencia y evitar problemas de sincronización (Wiggins, 2017).

9. Metodología Experimental

Se definió para este proyecto implementar la metodología SCRUM, ya que es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos (https://proyectosagiles.org/que-es-scrum/).

De igual forma se emplearán los principios SCRUM: Control de Proceso Empírico, Auto-Organización, Colaboración, Priorización en valor, Time Boxing, Iteración.

- **Roles**
  - Product Owner: Luis Fernando Mosquera Romero y Pablo Alexis Suntaxi Heredia.
  - Scrum Master: Luis Fernando Mosquera Romero.
  - Equipo de desarrollo: Luis Fernando Mosquera Romero y Pablo Alexis Suntaxi Heredia.
- **Eventos**
  - **Planificación del Sprint (semanal):**
    - Propósito: Definir qué trabajo se realizará durante el Sprint y cómo se logrará
    - Actividad: El Product Owner presentará las historias de usuario y los requisitos prioritarios. El equipo seleccionará las tareas que pueden completar durante la semana y definirá el objetivo del Sprint.
  - **Reunión Diaria (Daily Scrum):**
    - Propósito: Sincronizar actividades y planificar el trabajo del día.
    - Actividad: Cada miembro del equipo responderá a tres preguntas: ¿Qué hice ayer? ¿Qué voy a hacer hoy? ¿Hay algún impedimento? Se identificarán y resolverán obstáculos rápidamente.
  - **Revisión del Sprint (semanal):**
    - Propósito: Presentar el trabajo completado y obtener retroalimentación.
    - Actividad: El equipo mostrará las funcionalidades desarrolladas. Se discutirá cualquier cambio requerido basado en la retroalimentación recibida y se actualizará el backlog del producto.
  - **Retrospectiva del Sprint (semanal):**
    - Propósito: Reflexionar sobre el Sprint finalizado y encontrar formas de mejorar.
    - Actividad: Se revisará qué aspectos funcionaron bien, qué se podría mejorar y se definirá un plan de acción para implementar mejoras en el próximo Sprint.
- **Herramientas:**

- Jira: Es una herramienta de gestión de proyectos que soporta metodologías ágiles.
- GitHub: Proporciona un sistema de control de versiones basado en Git, que facilita la colaboración en el desarrollo de software. Permite a los miembros del equipo trabajar en paralelo, gestionar el código fuente y realizar revisiones de código.
- Microsoft Teams: Se usará para la comunicación y colaboración diaria entre los miembros del equipo, permitiendo realizar videoconferencias, chat y compartir archivos de manera eficiente.

- **Control de Calidad y Pruebas:** Las actividades de control de calidad y pruebas se integrarán en cada Sprint para asegurar la calidad del producto final. La estrategia será la siguiente.
  - **Responsables:** Todos los miembros del equipo serán responsables de la calidad del trabajo que realizan. Además, se designará un miembro específico del equipo para liderar las actividades de pruebas y control de calidad.
  - **Actividades**
    - Revisión de Código: Cada pieza de código desarrollada será revisada por otro miembro del equipo para asegurar su calidad y detectar posibles errores.
    - Pruebas Unitarias: Se realizarán pruebas unitarias de cada componente desarrollado para asegurar que cada unidad de código funcione correctamente.
    - Pruebas de Integración: Al final de cada Sprint, se realizarán pruebas de integración para asegurar que todos los componentes del sistema funcionen juntos de manera correcta.
    - Pruebas de Aceptación del Usuario: Se llevarán a cabo pruebas de aceptación con los usuarios para validar que el producto cumpla con sus expectativas y requerimientos.
- **Beneficios Esperados:** Se recopilará la opinión de los usuarios y se revisará el producto final. Esta información guiará las posibles mejoras futuras.
  - **Mejoras en la Productividad del Equipo:** La metodología propuesta fomenta la colaboración y la auto-organización, lo que puede llevar a un aumento en la eficiencia y productividad del equipo.
  - **Entrega Más Rápida de Valor al:** La estructura de Sprints cortos permite entregar incrementos del producto de manera regular, lo que asegura que el cliente reciba valor continuamente a lo largo del proyecto.
  - **Adaptabilidad a Cambios en los Requisitos:** Permite adaptarse fácilmente a los cambios en los requisitos, asegurando que el producto final cumpla con las expectativas del cliente.
  - **Mejora Continua:** La revisión y retroalimentación constantes en cada Sprint facilitan la mejora continua del proceso y del producto.
  - **Transparencia y Visibilidad:** El uso de herramientas como Jira y Microsoft Teams facilita la transparencia y la visibilidad del progreso del

proyecto, asegurando que todos los miembros del equipo y las partes interesadas estén alineados.

10. Cronograma

	CAPÍTULO I	**DÍAS**
1	**Análisis del problema**	**7 días**
1.1	Antecedentes	1 día
1.2	Planteamiento del problema	2 días
1.3	Justificación	2 días
1.4	Objetivos	2 días
	CAPITULO II	
2	**Marco Teórico**	**14 días**
2.1	RIMPE – Negocios Populares y Tecnología en la gestión empresarial de pequeños negocios y microempresas	1 día
2.2	Computación en la nube	1 día
2.3	Modelos de despliegue de la nube	1 día
2.4	Modelos de Servicios en Cloud Computing	2 días
2.5	Aplicaciones Cloud Native	1 día
2.6	Propiedades de las aplicaciones Cloud Native	2 días
2.7	Topologías Cloud Native	1 día
2.8	Arquitecturas para las aplicaciones Cloud Native	1 días
2.9	Prácticas para el desarrollo de aplicaciones Cloud Native	1 día
2.10	Principios para el desarrollo de aplicaciones Cloud Native: 12 Factores	3 días
	CAPÍTULO III	
3	**DESARROLLO DE LA APLICACIÓN**	**120 días**
3.1	Investigación	14 días
3.2	Requerimientos	7 días
3.3	Diseño, Desarrollo y Análisis	82 días
3.4	Pruebas	7 días
	CAPÍTULO IV	
4	Conclusiones y Recomendaciones	**3 días**

11. Planificación del desarrollo de la aplicación

1 mes	Capítulo 1 • Investigación sobre las tecnologías y herramientas para el desarrollo de la aplicación. • Diseño de la arquitectura Cloud Native. • Elaboración de la documentación del capítulo 1.
2 mes	Capítulo 2 • Recolección de requerimientos mediante entrevistas a usuarios. • Definir los requisitos funcionales y no funcionales del sistema
3 mes	Capítulo 3 • Diseño de la arquitectura del sistema, incluyendo la definición de la base de datos relacional. • Implementación del esquema de base de datos. • Desarrollo del backend: API REST.
4 mes	Capítulo 3 • Desarrollo de la interfaz de usuario. • Conexión del frontend con backend. • Realización de pruebas de unidad e integración. • Pruebas de usuario final y corrección de errores.
5 mes	Capítulo 3 • Implementación del sistema en el entorno de producción en la nube. • Configuración de la seguridad y autenticación. • Pruebas, correcciones y ajustes finales. • Documentación del código
6 mes	Capítulo 4 • Fin de la documentación • Entrega de la versión final de la aplicación.

12. Recurso humano

El recurso humano involucra únicamente al estudiante y al tutor que se le asignará para asistirlo en el trabajo.

Durante la recolección de necesidades y las pruebas de usuario final, se emplearán técnicas específicas como entrevistas y encuestas dirigidas a grupos focales seleccionados. En particular, se llevarán a cabo entrevistas con la dueña del Minimarket Leche y Miel, así como con sus empleados. Estas entrevistas se centrarán en recopilar datos detallados sobre los requisitos funcionales y operativos del sistema. Además, se utilizarán cuestionarios para obtener información estructurada y cuantitativa sobre las

necesidades y expectativas de los usuarios. Estas técnicas asegurarán una comprensión profunda y precisa de los requisitos, lo que contribuirá a la eficacia del sistema.

13. Presupuesto

PRESUPUESTO				
GASTO DE PERSONAL				
Investigador	**Dedicación (horas semanales)**	**Número de meses**	**Costo por hora**	**VALOR TOTAL**
Joselyn Moncayo	20	6	6,25	3.000,00
TOTAL				**3.000,00**
SOFTWARE, EQUIPO TECNOLÓGICO, MAQUINARIA Y EQUIPO				
Rubro	**Justificación**		**Valor**	**VALOR TOTAL**
Equipo Tecnológico (Laptop)	Desarrollo del Proyecto		850,00	850,00
TOTAL				**850,00**
OTROS GASTOS DIVERSOS				
Rubro	Descripción		Valor	VALOR TOTAL
Papelería y fotocopias			50,00	50,00
Impresiones			70,00	70,00
Otros			10,00	10,00
TOTAL				**150,00**
TOTAL DE PRESUPUESTO				**4000,00**

14. Referencias

Acosta Vega, R. K., Ospino Ayala, Ó. J., & Valencia Espejo, V. E. (2017). Diseño de un sistema de planificación de recursos empresariales (ERP) para una microempresa. *INGE CUC*, *13*(1), 84–100. https://doi.org/10.17981/ingecuc.13.1.2017.08

Benavides Rivera, L. A., & Buñay Guisñan, P. A. (2024). *Aplicación web para gestionar las ventas de la microempresa Vida Sana en el cantón Carmen utilizando la metodología design thinking*. http://dspace.unach.edu.ec/handle/51000/12430

Carreño Dueñas, D. A., Amaya González, L. F., Ruiz Orjuela, E. T., & Javier Tiboche, F. (2019). Diseño de un sistema para la gestión de inventarios de las pymes en el sector alimentario. *Industrial Data*, *22*(1), 113–132. https://doi.org/10.15381/idata.v22i1.16530

CLOUD NATIVE COMPUTING FOUNDATION. (n.d.-a). *Contenedores | Cloud Native Glosario*. Retrieved April 24, 2024, from https://glossary.cncf.io/es/container/

CLOUD NATIVE COMPUTING FOUNDATION. (n.d.-b). *Orquestación de Contenedores | Cloud Native Glosario*. Retrieved April 24, 2024, from https://glossary.cncf.io/es/container-orchestration/

CLOUD NATIVE COMPUTING FOUNDATION. (n.d.-c). *Who We Are | CNCF*. Retrieved April 22, 2024, from https://www.cncf.io/about/who-we-are/

Huamantumba, E. J. S., & Delgado Bardales, J. M. (2020). Gestión de simplificación administrativa en el desarrollo de las universidades públicas. *Ciencia Latina*, *4*(2), 1839–1856. https://doi.org/10.37811/cl_rcm.v4i2.197

IBM. (n.d.). *What Are Containers? | IBM*. Retrieved April 24, 2024, from https://www.ibm.com/topics/containers

Lee, C. A., Bohn, R. B., & Michel, M. (2020). *NIST Cloud Computing Program - NCCP | NIST*. https://doi.org/10.6028/NIST.SP.500-332

Mell, P. M., & Grance, T. (2011). *The NIST definition of cloud computing*. https://doi.org/10.6028/NIST.SP.800-145

Quint, P.-C., & Kratzke, N. (2017). Taming the Complexity of Elasticity, Scalability and Transferability in Cloud Computing-Cloud-Native Applications for SMEs Taming the Complexity of Elasticity, Scalability and Transferability in Cloud Computing Cloud-Native Applications for SMEs. *Article in International Journal on Advances in Networks and Services*, *9*. http://www.iariajournals.org/intelligent_systems/2016,

Vitale, T., & Long, J. (2022). *Cloud Native Spring in action: with Spring Boot and Kubernetes*.

Wiggins, A. (2017). *The Twelve-Factor App*. https://12factor.net/