

UNIVERSIDAD PERUANA UNIÓN
FACULTAD DE INGENIERÍA Y ARQUITECTURA
Escuela Profesional de Ingeniería de Sistemas



Arquitectura Multi-Tenant para la implementación de un software como servicio (SaaS) y su influencia en la usabilidad de las MYPES del sector comercio del Nor Oriente del Peruano

Tesis para obtener el Título Profesional de Ingeniero de Sistemas

Autores:

Edwin Alexander Bautista Villegas
Johann James Valles Paz

Asesor:

Mg. Hitler Collantes Chules
Mg. Nancy Esther Casildo Bedón

Tarapoto, septiembre de 2023

DECLARACIÓN JURADA DE ORIGINALIDAD DE TESIS

Yo Mg. Hitler Collantes Chules, docente de la Facultad de Ingeniería y Arquitectura, Escuela Profesional de Ingeniería de Sistemas, de la Universidad Peruana Unión.

DECLARO:

Que la presente investigación titulada: **“Arquitectura Multi-Tenant para la implementación de un software como servicio (SaaS) y su influencia en la usabilidad de las MYPES del sector comercio del Nor Oriente del Peruano”** de los autores Edwin Alexander Bautista Villegas y Johann James Valles Paz tiene un índice de similitud de 14 % verificable en el informe del programa Turnitin, y fue realizada en la Universidad Peruana Unión bajo mi dirección.

En tal sentido asumo la responsabilidad que corresponde ante cualquier falsedad u omisión de los documentos como de la información aportada, firmo la presente declaración en la ciudad de Tarapoto, a los 15 días del mes de septiembre del año 2023.



Mg. Hitler Collantes Chules

ACTA DE SUSTENTACIÓN DE TESIS

En San Martín, Tarapoto, Morales, a 01 día(s) del mes de setiembre del año 2023, siendo las 11:00 horas, se reunieron los miembros del jurado en la Universidad Peruana Unión Campus Tarapoto, bajo la dirección del (de la) presidente(a): Mg. Danny Lévano Rodríguez, el (la) secretario(a): Mg. Joyse Baldwin Huamán Labán y los demás miembros: Dr. Godofredo Apaza Romero y el (la) asesor(a) Mg. Hitler Collantes Chules Coasesor: Mg. Nancy Esther Casildo Bedón con el propósito de administrar el acto académico de sustentación de la tesis titulado: **Arquitectura Multi-Tenant para la implementación de un software como servicio (SaaS) y su influencia en la usabilidad de las MYPES del sector comercio del Nor Oriente del Peruano.**

del(los) bachiller(es): a) Edwin Alexander Bautista Villegas
 b) Johann James Valles Paz
 c) conducente a la obtención del título profesional de:

Ingeniero de Sistemas
 (Denominación del Título Profesional)

El Presidente inició el acto académico de sustentación invitando al (a la) / a (los) (las) candidato(a)s hacer uso del tiempo determinado para su exposición. Concluida la exposición, el Presidente invitó a los demás miembros del jurado a efectuar las preguntas, y aclaraciones pertinentes, las cuales fueron absueltas por al (a la) / a (los) (las) candidato(a)s. Luego, se produjo un receso para las deliberaciones y la emisión del dictamen del jurado.

Posteriormente, el jurado procedió a dejar constancia escrita sobre la evaluación en la presente acta, con el dictamen siguiente:

Bachiller-(a): Edwin Alexander Bautista Villegas

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	
Aprobado	20	A+	Excelente	Excelencia

Bachiller -(b): Johann James Valles Paz

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	
Aprobado	20	A+	Excelente	Excelencia

Bachiller -(c):

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	

(*) Ver parte posterior

Finalmente, el Presidente del jurado invitó al (a la) / a (los) (las) candidato(a)s a ponerse de pie, para recibir la evaluación final y concluir el acto académico de sustentación procediéndose a registrar las firmas respectivas.

 Presidente/a



 Secretario/a

 Asesor/a

 Miembro

 Miembro

 Bachiller (a)

 Bachiller (b)

 Bachiller (c)

RESUMEN:

En los últimos años, ha habido un crecimiento exponencial en la adopción de servicios en la nube, lo cual ha llevado a las empresas de tecnología de la información a identificar una oportunidad para brindar soluciones basadas en la nube, como el software como servicio, con el fin de atender las demandas de los clientes de manera efectiva. El objetivo de esta investigación fue determinar en qué medida la implementación de la arquitectura multi-Tenant en un software como servicio (SaaS) incrementa su usabilidad en las MYPES del sector comercio del Nor Oriente del Peruano. Para ello, se realizó una investigación pre-experimental con una muestra de 32 MYPES a quienes se les aplicó un cuestionario para medir el grado de usabilidad del software puesto en producción. En promedio, el 38% de encuestados están totalmente de acuerdo con el nuevo sistema propuesto, y 62% reconoció mejoras en la solución propuesta, lo que marca una gran diferencia entre el pre y pos-test. En conclusión, el software como servicio (SaaS) desarrollado bajo la arquitectura Multi-Tenant ha aumentado la usabilidad del sistema implementado, generando beneficios significativos en términos de rentabilidad para la empresa CEATECSOFT E.I.R.L. En primer lugar, se ha logrado una reducción notable en los costos relacionados con el personal técnico, al eliminar la necesidad de gestionar múltiples versiones y al minimizar los gastos asociados con las visitas técnicas a los clientes, así como los costos relacionados con el registro de nuevos clientes. Estas mejoras han llevado a un incremento en la eficiencia operativa y una optimización de los recursos empresariales.

Palabras clave: Empresa privada; sistema de información; usabilidad; arquitectura multi-Tenant; software integrado.

ABSTRACT:

In recent years, there has been exponential growth in the adoption of cloud services, leading information technology companies to identify an opportunity to provide cloud-based solutions such as Software as a Service (SaaS) to effectively meet customer demands. The objective of this research was to determine to what extent the implementation of multi-tenant architecture in a Software as a Service (SaaS) increases its usability in small and medium-sized enterprises (SMEs) in the commerce sector of northeastern Peru. To achieve this, a pre-experimental investigation was conducted with a sample of 32 SMEs, to whom a questionnaire was administered to measure the degree of usability of the software implemented in production. On average, 92% of the respondents acknowledged improvements in the proposed solution, indicating a significant difference between the pre-test and post-test results. In conclusion, the Software as a Service (SaaS) developed under the Multi-Tenant architecture has increased the usability of the implemented system, generating significant benefits in terms of profitability for the company CEATECSOFT E.I.R.L. Firstly, a notable reduction in costs related to technical personnel has been achieved by eliminating the need to manage multiple versions and minimizing expenses associated with technical visits to clients, as well as costs related to registering new clients. These improvements have led to an increase in operational efficiency and optimization of business resources.

Keywords: Private company; information system; usability; multi-tenant architecture; embedded software.

INTRODUCCIÓN

El Software como Servicio (SaaS) es un servicio en la nube que se ha convertido en una de las soluciones más productivas y utilizadas actualmente, gracias a su impacto en la economía y su capacidad para resolver problemas de negocios [1].

Muchas empresas de TI están ofreciendo cada vez más servicios SaaS como parte de sus ofertas de productos y servicios [2]. [3] afirma que la tendencia hacia el uso de servicios en la nube ha aumentado significativamente en los últimos años, y las empresas de TI han visto una oportunidad para ofrecer soluciones basadas en la nube, como el software como servicio, para satisfacer las necesidades de los clientes [4].

Ofrecer servicios SaaS puede ser beneficioso para las empresas de TI, ya que les permite generar ingresos recurrentes a través de la suscripción de los clientes, en lugar de solo ingresos únicos por la venta de licencias de software [5]. Además, el modelo de SaaS permite una mayor flexibilidad y escalabilidad para las empresas, lo que puede ayudar a satisfacer las demandas cambiantes de los clientes [6].

La arquitectura Multi-tenant es importante para desarrollar servicios SaaS por varias razones: eficiencia en la utilización de recursos; al utilizar una arquitectura Multi-tenant, múltiples clientes pueden compartir la misma infraestructura y recursos de hardware [7]–[9], lo que permite una mejor utilización de los recursos disponibles y una reducción en los costos de infraestructura [10], [11]. Escalabilidad, los servicios pueden escalar de manera más eficiente para acomodar un mayor número de clientes y un mayor tráfico, lo que permite que los servicios SaaS crezcan con el tiempo [12]. Mantenimiento simplificado, los desarrolladores pueden realizar actualizaciones y mantenimiento en una única instancia del software, en lugar de tener que hacerlo para cada cliente individualmente [13]. Esto reduce la complejidad y el tiempo necesario para mantener el software y garantiza una mayor consistencia en el rendimiento [14]. Personalización de la experiencia del usuario: a pesar de que los clientes comparten la misma instancia de software, una arquitectura Multi-tenant permite la personalización de la experiencia del usuario, lo que permite a los clientes personalizar sus propias configuraciones y datos sin afectar a otros clientes [15]. Seguridad, la arquitectura Multi-tenant permite implementar una seguridad más sólida y centralizada, lo que facilita la gestión de políticas de seguridad y el monitoreo de las actividades de los usuarios [16].

La arquitectura multi-tenant es un enfoque de diseño de software que permite que una sola instancia de una aplicación de software sirva a múltiples clientes o "inquilinos" (tenants) a la vez [1]. En este enfoque, los inquilinos comparten recursos comunes, como la infraestructura, la base de datos y las aplicaciones, pero cada uno tiene su propio espacio de trabajo y sus datos están completamente separados y seguros [1]. La arquitectura multi-tenant ofrece una serie de ventajas, como la reducción de costos, la escalabilidad, la eficiencia, la personalización y la seguridad, que hacen que sea una opción popular para los proveedores de SaaS y otras aplicaciones empresariales [10].

La compañía CEATECSOFT E.I.R.L, que se dedica al desarrollo y venta de software personalizado en la región de San Martín - Perú, tiene como principal objetivo expandirse y mejorar su estabilidad financiera. Para lograrlo, han decidido adoptar la arquitectura "Multi-Tenant" para crear aplicaciones basadas en la nube. El objetivo es reducir los costos de mantenimiento de su aplicación SaaS (Software como Servicio) y acelerar la adaptación a nuevos clientes (suscriptores). En la actualidad, CEATECSOFT E.I.R.L está experimentando un problema que requiere una

evaluación cuidadosa. A través del análisis de implicados en la organización, se ha identificado una disminución en la rentabilidad económica, la cual está relacionada con varios factores, tales como el costo de la mano de obra y las horas extras del personal técnico, la falta de eficacia en la recaudación y facturación de licencias, la falta de un sistema Multi-Tenant y los gastos en actualizaciones del sistema de escritorio.

La consecuencia de esto es que se produce una disminución en la calidad del servicio, se reactivan servicios que se encontraban fuera del contrato, los clientes reactivados deben hacer pagos adicionales por los servicios, se pierde la capacidad de hacer un seguimiento adecuado al soporte al cliente y se pueden recibir sanciones económicas por parte de la SUNAT (Superintendencia Nacional de Aduanas y de Administración Tributaria - Perú).

El propósito de este estudio es resolver un problema específico, y para lograrlo, se busca

determinar en qué medida la implementación de la arquitectura Multi-Tenant en un software como servicio (SaaS) incrementa su usabilidad en las MYPES del sector comercio del Nor Oriente del Peruano.

Métodos o Metodología Computacional

Análisis estadístico

Este estudio se realizó a través de un enfoque aplicativo y experimental, con un diseño pre- experimental (Pre y post test). Se usó una encuesta como técnica para recopilar datos, utilizando el cuestionario "System Usability Scale" (Sistema de Escalas de Usabilidad) desarrollado por John

Brooke en 1986; que incluye una escala de valoración tipo Likert de 1 a 5. La población y muestra estuvo conformada por de 32 clientes, obtenida mediante un muestreo probabilístico aleatorio simple.

Para calcular e interpretar los resultados del cuestionario SUS, se seguirá el siguiente procedimiento: se sumarán las respuestas de los enunciados impares y se les restará 5; luego se sumarán las respuestas de los enunciados pares y se restará 25; finalmente, se sumarán ambos resultados y se multiplicarán por 2,5. El puntaje obtenido será interpretado de acuerdo con la escala siguiente (Figura 1).

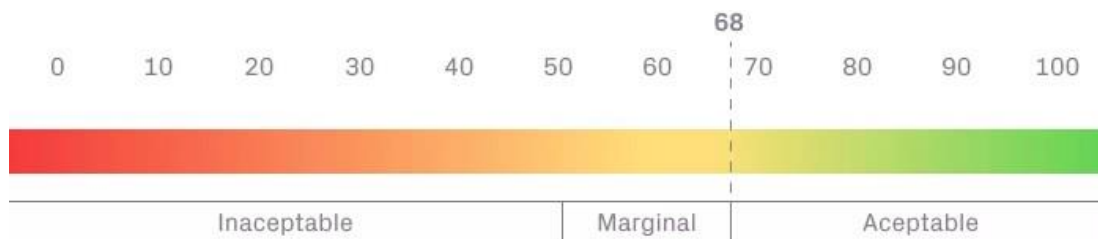


Fig. 1. Representación de los resultados de un SUS

La estadística proporciona un fundamento objetivo y sólido para examinar y respaldar las hipótesis en la investigación. Esto nos asiste tomar decisiones informadas, la comprensión de la incertidumbre relacionada con los resultados y su efectiva comunicación. Bajo este contexto, se realizó un análisis inferencial, el nivel de

confianza para el cálculo fue del 95% y margen de error 5%. Se utilizó técnica paramétrica dado a que la “data” cumplió los 3 supuestos, es decir, es variable numérica, tiene distribución normal, y sus datos son homogéneos.

Para responder a la hipótesis planteada se aplicó la prueba t para muestras relacionados, dado que la muestra fue la misma para el pre y pos-test.

Por último, se realizó una categorización de variables para una mejor comprensión de las variables de estudio, resultando los niveles y rangos siguientes.

Niveles y rangos	Baja	Regular	Alta
Variable dependiente (VD)	[10 - 23]	[24 - 37]	[38 - 50]

Fases de desarrollo

Se utilizó un método evolutivo incremental, similar a OpenUp [17], el cual se describe en la Tabla

1. Este método está planteado por cuatro (04) fases: inicio, elaboración, construcción y transición; permitiendo asumir que, en la fase de inicio, la especificación de requerimientos funcionales y no funcionales puede cambiar en cualquier momento del ciclo de vida del proyecto [18].

Tabla 1. Fases del desarrollo de la solución

FASE	ACTIVIDAD
Inicio	<ul style="list-style-type: none"> Establecer el alcance del proyecto Especificación de los requerimientos funcionales y no funcionales Planificación de la solución a nivel: presupuesto, cronograma de actividades, riesgos y calidad del proyecto
Elaboración	<ul style="list-style-type: none"> Diseñar, implementar y validar la arquitectónica Multi-Tenant Definir las herramientas tecnológicas a utilizar Desarrollar prototipos y modelos Establecer un ambiente de desarrollo
Construcción	<ul style="list-style-type: none"> Desarrollar el código Realizar pruebas Verificar la calidad del software Gestionar los cambios Preparar la transición del proyecto
Transición	<ul style="list-style-type: none"> Puesta en producción del software Capacitar al cliente Establecer el soporte y mantenimiento

Con el fin de lograr nuestro primer objetivo, hemos llevado a cabo la fase de inicio, en la cual nos enfocamos en identificar los objetivos del proyecto, evaluar su viabilidad y definir sus requisitos. Durante esta fase, realizamos un análisis preliminar del proyecto con el propósito de establecer su visión, alcance y objetivos. Para el cumplimiento del segundo objetivo, hemos realizado la definición detallada de los requisitos, la arquitectura del sistema y la planificación del proyecto. Durante esta fase, se definieron las características y funcionalidades del software.

En la fase de construcción nos centramos en desarrollar el software utilizando los requisitos y diseños definidos en la fase de elaboración. En la fase transición, el software se entrega al cliente para su validación y se prepara para su implementación en el entorno de producción.

Resultados y discusión

Caracterización del sistema actual

CEATECSOFT E.I.R.L dispone de un sistema de escritorio que ha sido instalado en 32 MYPES a nivel nacional. Sin embargo, tras el análisis de implicados, se han identificado diversas deficiencias y limitaciones en este programa. Entre las más destacadas se encuentran la limitada accesibilidad, costos asociados a las actualizaciones y mantenimiento, problemas de seguridad y gastos en hardware.

Según [19]–[21], los sistemas de escritorio están quedando obsoletos, debido a una característica en particular: los usuarios tienen que descargar y actualizar manualmente el software en sus computadoras cada vez que hay una nueva versión. Esta limitación se debe a la arquitectura propia de los sistemas de escritorio, tal como se muestra en la figura 2.

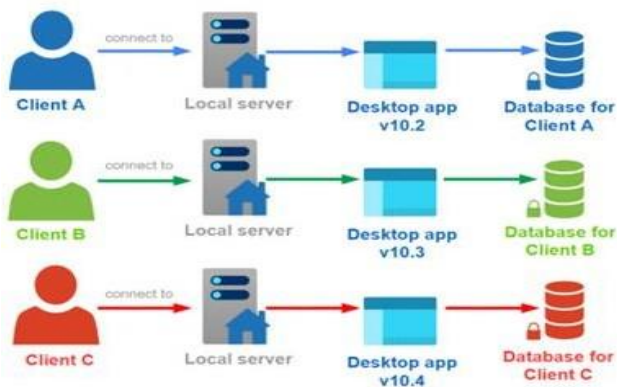


Fig. 2. Arquitectura del sistema de escritorio

Con el propósito de evaluar lo mencionado anteriormente, se aplicó un cuestionario de 10 preguntas con escala de Likert. Este instrumento tuvo como objetivo recopilar la opinión de los usuarios sobre la usabilidad del sistema de escritorio en sus respectivas MYPES. A continuación, se presenta el resultado respecto a la caracterización de la situación actual.

Tabla 2. Resultado de la aplicación del cuestionario System Usability Scale (SUS)

MYPES encuestadas	Puntaje obtenido	Estado	Puntaje ideal
32	55	Marginal	100

Es importante recalcar que, hemos aprovechado la encuesta SUS para analizar la situación actual del proceso, dado a que es una encuesta validada y aprobada para medir la satisfacción de un sistema de información. Bajo ese contexto se obtuvo los resultados siguientes:

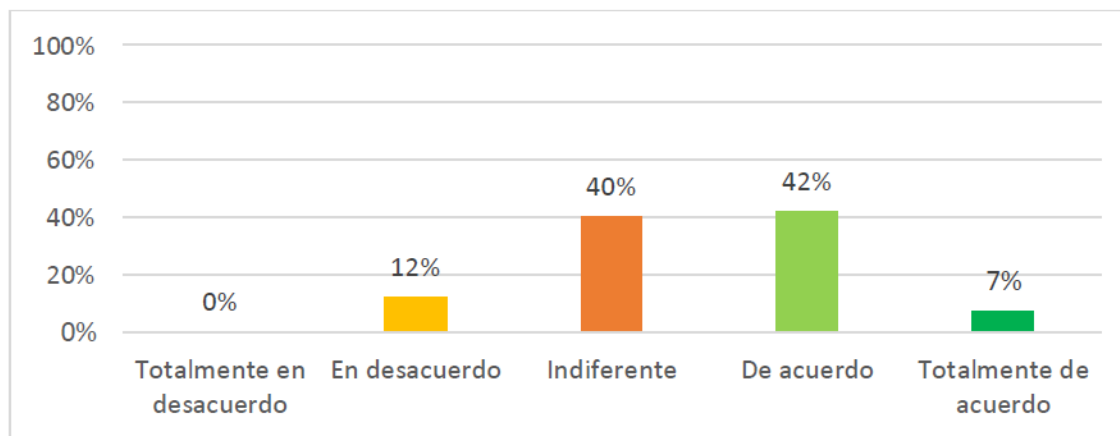


Fig 3. Percepción de los clientes respecto a la usabilidad del sistema de escritorio

De total de encuestados el 7% de encuestados expresó estar totalmente de acuerdo, mientras que el 42% manifestó su acuerdo con respecto al sistema implementado (Sistema de escritorio); Esta situación plantea inquietudes, especialmente considerando que la empresa se dedica a ofrecer servicios de software. En esta línea, la evaluación positiva de sus productos y servicios cobra una importancia significativa.

Después de realizar un análisis de frecuencias, se llegó a la conclusión de que el 100% de los encuestados (un total de 32 participantes) opinan que el sistema actual se ubica en el nivel de usabilidad "Regular".

Tabla 3. Análisis de frecuencias

Nivel de usabilidad	Frecuencia	Porcentaje
Regular	32	100,0

La tabla 2 y 3 revela que el software de escritorio no cumple con las expectativas de los usuarios en varios aspectos, destacando la falta de usabilidad. Un puntaje de 55 y nivel de usabilidad "Regular" indican que el software no es ampliamente utilizado, lo que podría llevar a CEATECSOFT E.I.R.L a perder toda su base de clientes en el futuro [20].

En base a lo anterior, nuestro propósito al implementar la arquitectura multi-tenant es potenciar la rentabilidad de CEATECSOFT E.I.R.L, enfocándonos en abordar los

desafíos actuales que enfrenta. Estos incluyen los elevados gastos en personal especializado para llevar a cabo actualizaciones, la necesidad de gestionar las distintas versiones y mantener el código fuente separado para cada cliente, los costos relacionados con las visitas técnicas a cada cliente, y las deficiencias y los gastos al registrar a un nuevo cliente.

Arquitectura Multi-Tenant implementada en un software como servicio (SaaS)

La arquitectura multi-tenant es una forma de diseño de software en la que una sola instancia del software es utilizada por múltiples clientes, lo que permite a cada cliente tener su propia configuración y datos, pero compartiendo los recursos de la misma instancia [22]. La arquitectura multi-tenant ofrece una forma eficiente, escalable y segura de proporcionar software personalizado a múltiples clientes al mismo tiempo [23].

Para cumplir con nuestro segundo objetivo de estudio y ofrecer una solución a la problemática existente, hemos desarrollado un sistema de información utilizando la metodología multi-tenant. En la Figura 4 presentamos el flujograma que explica la integración de esta arquitectura en nuestro proyecto.

Cuando se utiliza una arquitectura multi-tenant, donde diferentes clientes comparten una misma instancia de una aplicación [11], es fundamental contar con un método eficiente y efectivo para identificar e inicializar los tenants. Esto implica configurar correctamente los recursos, como la conexión a la base de datos, para cada tenant. Aunque Laravel, un popular framework de desarrollo de aplicaciones web en PHP, no ofrece de forma predeterminada una solución específica para gestionar aplicaciones multi-tenant, se ha desarrollado una solución para abordar la identificación e inicialización de tenants en aplicaciones basadas en Laravel. Para satisfacer esta necesidad, hemos elegido utilizar la librería "Tenancy for Laravel". Esta librería proporciona una solución integral y autónoma para gestionar tanto la identificación como la inicialización de los tenants, y también se enfoca en optimizar el rendimiento y fortalecer la seguridad en entornos multi-tenant [24].

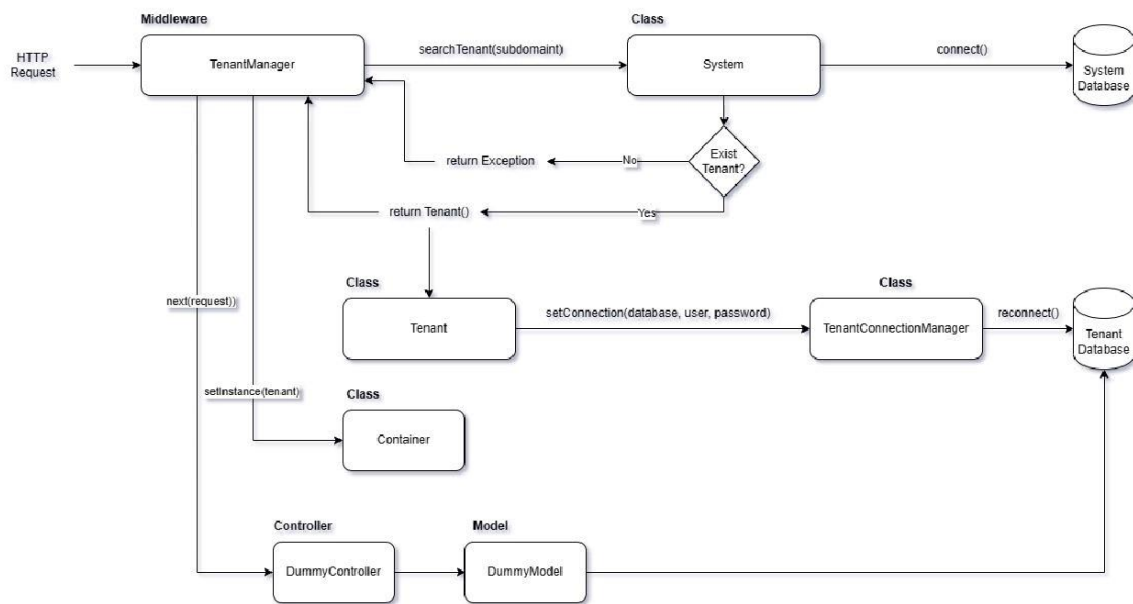
En la Figura 3, se puede observar cómo el flujo de trabajo comienza con la intervención del middleware "InitializeTenancyByDomainOrSubdomain". Este componente de software es responsable de dirigir inicialmente el proceso de identificación del inquilino. Dependiendo del dominio de la solicitud, este middleware bifurca el flujo hacia "InitializeTenancyByDomain" o "InitializeTenancyBySubdomain". Esta bifurcación tiene en cuenta la estructura de la solicitud y separa las solicitudes que provienen de dominios y subdominios.

A continuación, se invoca el método "initializeTenancy", el cual se encuentra en la clase abstracta "IdentificationMiddleware" de la cual extienden "InitializeTenancyByDomain" y "InitializeTenancyBySubdomain". El método "initializeTenancy" recibe dos parámetros importantes: primero, la solicitud (request) que hace referencia a una instancia de la clase Request de Laravel, y segundo, un array de argumentos que contienen datos del subdominio. Dentro de este método, también se instancia la clase "Tenancy" a través del método "initialize", utilizando los argumentos del subdominio o dominio como parámetro.

Una vez dentro del alcance de la clase "Tenancy", se busca activamente el inquilino correspondiente dentro del modelo "Tenant". Esta acción representa la fase real de identificación, donde se busca el inquilino específico que realiza la solicitud. La identificación exitosa del inquilino desencadena dos eventos sucesivos: "InitializingTenancy" y "BootstrappingTenancy". El primero simboliza el inicio del proceso de establecimiento de la tenencia, mientras que el segundo representa la conclusión de la inicialización.

"BootstrappingTenancy" es particularmente relevante porque incorpora la clase "DatabaseManager". Este componente es responsable de establecer la conexión con la base de datos correspondiente al inquilino identificado. Este proceso de conmutación de la base de datos es lo que finalmente permite a la aplicación interactuar con los datos específicos del inquilino.

Fig. 4. Flujograma de la arquitectura multi-tenant implementada en la solución



Note:
 DummyController simulates any controller that is in the application.
 DummyModel simulates any model that is in the application.

En la Figura 4 se muestra el esquema del sistema de información desarrollado. El sistema consta de "N" clientes que realizan diversas peticiones al servidor, que se comunica a través de la vista (SPA - VUE) y consume la API (Api - Laravel). En esta etapa se realizan las validaciones

correspondientes y se redirecciona a cada cliente a su respectiva base de datos.

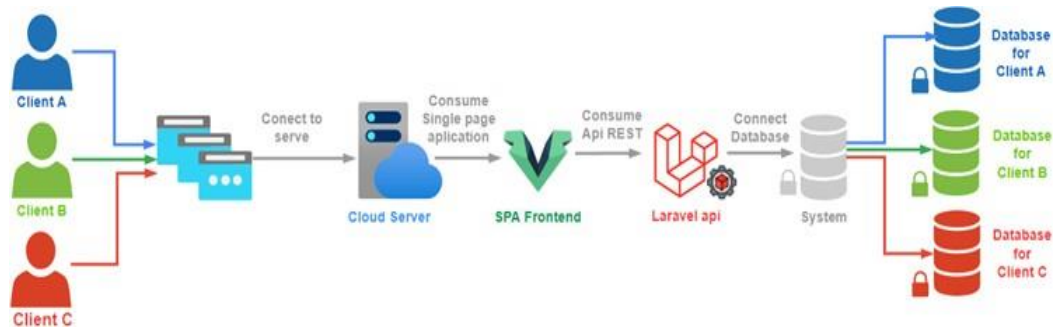


Fig. 5. Esquema de Sistema de información puesto en producción

El software desarrollado se compone de cinco módulos principales, cada uno de los cuales cuenta con sus respectivos submódulos y microservicios. Estos paquetes se ofrecen a las micro y pequeñas empresas (MYPES) bajo un modelo de licencia mensual.

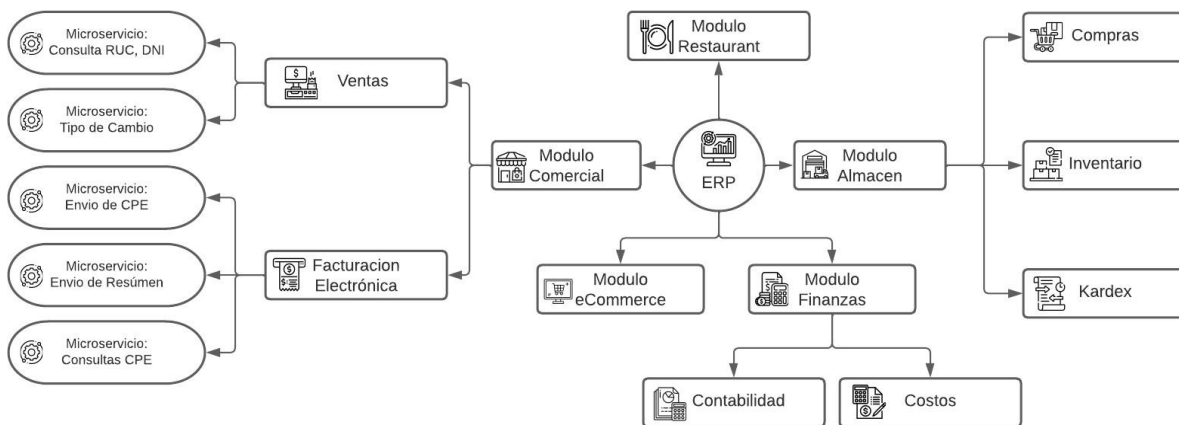


Fig. 6. Sistema de información (ERP) desarrollado

Evaluar la influencia del sistema desarrollado en la usabilidad de las MYPES

En el marco de nuestro tercer objetivo, buscaremos analizar el impacto del sistema de información basado en la arquitectura Multi-Tenant en términos de usabilidad para las micro y pequeñas empresas (MYPES) del sector comercial en la región Nor Oriente de Perú.

Tabla 4. Resultado de la aplicación del cuestionario System Usability Scale (SUS)

MYPES encuestadas	Puntaje obtenido	Estado	Puntaje ideal
32	92	Aceptable	100

La Tabla 3 refleja los resultados obtenidos después de la implementación del sistema

de información basado en la arquitectura Multi-Tenant. Con una puntuación de 92, casi perfecta, se demuestra que el nuevo sistema en producción cumple con las expectativas de los clientes, presentando altos niveles de aceptabilidad y usabilidad.

Para [5], [15], [20], [21] la estadística se utiliza en una investigación con el objetivo de analizar y comprender los datos recopilados, obtener conclusiones significativas y tomar decisiones informadas. Bajo ese concepto, aplicaremos la estadística inferencial para procesar y evaluar los datos de la investigación.

Inicialmente, se llevó a cabo un análisis para determinar qué tipo de estadística se emplearía en este estudio, es decir, si se utilizaría estadística paramétrica o no paramétrica. Para ello, se llevó a cabo el análisis de los tres supuestos de la estadística paramétrica: variable cuantitativa numérica, distribución normal y homogeneidad de varianzas. Nuestra investigación cumple los tres supuestos: las variables son cuantitativas, los datos presentan una distribución normal (Tabla 5) y presentan homogeneidad de varianzas (Tabla 6).

Tabla 5. Prueba de normalidad

		Shapiro-Wilk		
		Estadístico	gl.	Sig.
	Pre-test	,883	32	,052
	Post-test	,812	32	,060

Nota. P valor $\geq 0.05 \rightarrow$ Se acepta la H_0 y rechaza la H_1 .

Tabla 6. Prueba de homogeneidad de varianza

		Estadístico de Levene	gl1	gl2	Sig.
Pre y post Test	Se basa en la media	2,311	1	62	,134
	Se basa en la mediana	2,067	1	62	,156
	Se basa en la mediana y con gl ajustado	2,067	1	49,348	,157
	Se basa en la media recortada	2,455	1	62	,122

Nota. P valor $\geq 0.05 \rightarrow$ Se acepta la H_0 y rechaza la H_1 .

Dado que los supuestos de una técnica paramétrica se cumplen, se aplicó la técnica estadística paramétrica. Se utilizó la prueba “t” para muestras relacionadas dado a que se trabajó con la misma muestra para el pre y post test; cuyos resultados se presentan a continuación.

Tabla 7. Prueba t para muestras relacionadas

	t	gl	Sig. (bilateral)
Pre y post test	-26,092	31	,000

Los datos de la Tabla 7 permiten corroborar que se cumplió nuestro objetivo principal, porque la Arquitectura Multi-Tenant implementado en un software como servicio (SaaS) influyó en la usabilidad de las MYPES del sector comercio del Nor Oriente del Peruano. Esto se debe eventualmente a que el valor que obtiene “Sig.” (0,000) es menor a 0.05.

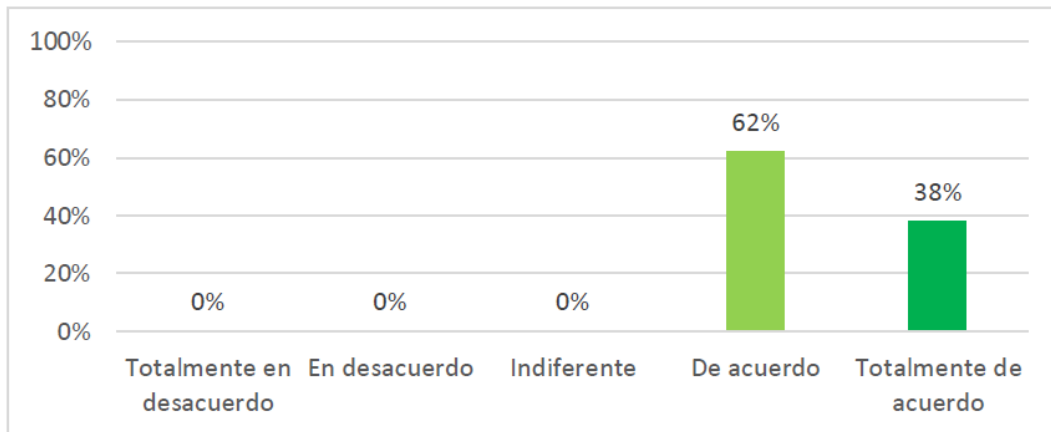


Fig 7. Percepción de los clientes respecto a la usabilidad del sistema SaaS con arquitectura Multi-Tenant

Del total de encuestados el 38% están totalmente de acuerdo con el nuevo sistema puesto en producción y 62% están de acuerdo. Estos resultados indican claramente que nuestra propuesta contribuye a elevar tanto la usabilidad como la rentabilidad de CEATECSOFT E.I.R.L. En resumen, a raíz del análisis de frecuencias efectuado, se ha llegado a la conclusión de que el nivel de usabilidad del sistema puede considerarse "Alto".

Tabla 8. Análisis de frecuencias

Nivel de usabilidad	Frecuencia	Porcentaje
ALTA	32	100,0

CONCLUSIONES

Basándonos en los resultados de la prueba estadística t para muestras relacionadas, presentados en la Tabla 7, se concluye que el estudio ha logrado mejorar la usabilidad del software como servicio (SaaS) desarrollado bajo la arquitectura Multi-Tenant. Esto ha llevado a un aumento en la rentabilidad de CEATECSOFT E.I.R.L. en varios aspectos. Primero, hemos reducido significativamente los costos asociados con el personal técnico, al disminuir la necesidad de desarrolladores. Además, hemos eliminado la gestión de distintas versiones y el mantenimiento de código fuente separado para cada cliente, lo que ha generado ahorros adicionales. También se han reducido los gastos relacionados con las visitas técnicas a cada cliente y los costos asociados al registro de nuevos clientes.

Hemos identificado los indicadores que tienen un impacto directo en la usabilidad de un sistema de información. Basándonos en estos hallazgos, nos hemos asegurado de satisfacer las necesidades de los clientes al cumplir rigurosamente con los requerimientos funcionales y no funcionales.

En cuanto a la implementación del software, esta se llevó a cabo de manera eficiente, ya que los clientes lo adoptaron sin dificultades, como se evidencia en la tabla 4.

Como resultado, podemos afirmar que el software como servicio (SaaS) desarrollado bajo la arquitectura Multi-Tenant ha tenido un impacto positivo en la usabilidad de las micro y pequeñas empresas (MYPES) del sector comercial en el Nor Oriente del Perú.

La solución propuesta ha sido diseñada de manera que pueda ser implementada en cualquier empresa proveedora de software. Basándonos en los resultados favorables encontrados, recomendamos su aplicación, teniendo en cuenta las características y realidades específicas de las organizaciones donde se pretende implantar.

REFERENCIAS

- [1] R. Santana, A. Malucelli, and S. Reinehr, "Práticas de Customização de Software Aplicadas em SaaS Multi-Tenant," in *WORKSHOP DE TESES E DISSERTAÇÕES EM QUALIDADE DE SOFTWARE - SIMPÓSIO BRASILEIRO DE QUALIDADE DE SOFTWARE (SBQS)*, 2020, pp. 55–60[Online]. Available: https://sol.sbc.org.br/index.php/sbqs_estendido/article/view/14193/14040.
- [2] J. Herrera-cubides, N. Gelvez-García, and D. López-sarmiento, "LMS SaaS: Una alternativa para la formación virtual SaaS LMS: An alternative to the virtual training," *Rev. Chil. Ing.*, vol. 27, no. 1, pp. 164–179, 2019, [Online]. Available: <https://www.scielo.cl/pdf/ingeniare/v27n1/0718-3305-ingeniare-27-01-00164.pdf>.
- [3] T. Satpathy, *Una Guía para el Conocimiento de SCRUM*, 3rd ed., vol. 44, no. 8. 2017.
- [4] F. Matloob *et al.*, "Software defect prediction using ensemble learning: A systematic literature review," *IEEE Access*, vol. 9, no. 1, pp. 98754–98771, 2021, doi: 10.1109/ACCESS.2021.3095559.
- [5] O. Agudelo-Varela, J. Martínez-Baquero, and S. Valbuena-Rodríguez, "Administración de TI en la facultad de ingeniería de la Universidad de los Llanos," *Rev. Politécnica*, vol. 16, no. 31, pp. 68–76, 2020, doi: 10.33571/rpolitec.v16n31a5.
- [6] S. F. Ruiz-Paz, R. Santaolaya-Salgado, O. G. Fragoso-Díaz, F. J. Álvarez-Rodríguez, and J. C. Rojas-Pérez, "Modelo de orquestación dinámica para flujos de trabajo del software como servicio," *Ing. Investig. y Tecnol.*, vol. 20, no. 3, pp. 1–12, 2019, doi: 10.22201/fi.25940732e.2019.20n3.034.
- [7] D. Strode, T. Dingsøyr, and Y. Lindsjorn, "A teamwork effectiveness model for agile software development," *Empir. Softw. Eng.*, vol. 27, no. 2, pp. 1–50, 2022, doi: 10.1007/s10664-021-10115-0.
- [8] E. M. Arvanitou, A. Ampatzoglou, A. Chatzigeorgiou, and J. Carver, "Software engineering practices for scientific software development: A systematic mapping study," *J. Syst. Softw.*, vol. 172, no. 1, pp. 1–18, 2021, doi: 10.1016/j.jss.2020.110848.
- [9] A. A. K. A *et al.*, "Software architecture for quantum computing systems—A systematic review," *J. Syst. Softw.*, vol. 201, no. 1, p. 111682, 2023, doi: 10.1016/j.jss.2023.111682.
- [10] B. Fang, X. Zeng, and M. Zhang, "NestDNN: Resource-aware multi-tenant on-device deep learning for continuous mobile vision," *Proc. Annu. Int. Conf. Mob. Comput. Networking, MOBICOM*, pp. 115–127, 2018, doi: 10.1145/3241539.3241559.
- [11] M. H. Hilman, M. A. Rodriguez, and R. Buyya, "Multiple workflows scheduling in multi-tenant distributed systems: A taxonomy and future directions," *ACM Comput. Surv.*, vol. 53, no. 1, 2020, doi: 10.1145/3368036.
- [12] Y. Pérez *et al.*, "Design Thinking en la Planificación de Pruebas de Software," *Rev. Innovación y Softw.*, vol. 1, no. 2, pp. 40–51, 2020, [Online]. Available: <https://www.redalyc.org/journal/6738/673870835004/673870835004.pdf>.
- [13] A. R. Ortega, M. Noguera, J. L. Garrido, K. Benghazi, and L. Chung, "Component-based design for multi-tenant multi-target support in the cloud," in

Lecture Notes in Business Information Processing, 2013, vol. 153, pp. 146–160, doi: 10.1007/978-3-642-41638-5_10.

[14] A. Sunardi and Suharjito, “MVC architecture: A comparative study between laravel framework and slim framework in freelancer project monitoring system web based,” *Procedia Comput. Sci.*, vol. 157, pp. 134–141, 2019, doi: 10.1016/j.procs.2019.08.150.

[15] V. Castro-Rivera, R. Herrera-Acuña, and M. Villalobos-Abarca, “Desarrollo de un software web para la generación de planes de gestión de riesgos de software,” *Inf. Technol.*, vol. 31, no. 3, pp. 135–148, 2020, doi: 10.4067/S0718-07642020000300135.

[16] L. Mendoza-Munar, “El software como servicio y el habeas data: una aproximación desde el derecho privado y constitucional en Colombia,” *Dixi*, vol. 20, no. 27, p. 13, 2018, doi: 10.16925/di.v20i27.2396.

[17] P. S. F. Yip *et al.*, “The opportunities and challenges of the first three years of open up, an online text-based counselling service for youth and young adults,” *Int. J. Environ. Res. Public Health*, vol. 18, no. 24, pp. 1–10, 2021, doi: 10.3390/ijerph182413194.

[18] J. Medina, E. Pineda, and F. Téllez, “Requerimientos de software: prototipado, software heredado y análisis de documentos,” *Ing. y Desarro.*, vol. 37, no. 02, pp. 327–345, 2022, doi: 10.14482/inde.37.2.1053.

[19] A. Uribe Arévalo and E. Norman Acevedo, “Internacionalización de la pequeña y mediana industria del software y de las tecnologías informáticas (SW & TI) a través del efecto trampolín del gremio,” *Cuad. Latinoam. Adm.*, vol. 16, no. 31, pp. 1–14, 2020, doi: 10.18270/cuaderlam.v16i31.3068.

[20] G. A. Vizalote-Rodriguez, “Contribución de las TI en la mejora de la productividad de las PYME,” *Rev. Amaz. Digit.*, vol. 1, no. 1, p. e164, 2022, doi: 10.55873/rad.v1i1.164.

[21] R. Prada, M. Rueda, and P. Ocampo, “Clima de trabajo y su relación con la productividad laboral en empresas de tecnología,” *Espacios*, vol. 41, no. 45, pp. 57–75, 2020, doi: 10.48082/espacios-a20v41n45p06.

[22] A. Quincho, “Arquitectura Multi-tenant en aplicaciones basadas en la nube para mejorar la rentabilidad en la empresa Group Solution Technology sociedad anónima cerrada,” Universidad Nacional del Centro del Perú, 2020.

[23] L. Oll, J. Roa, and R. Robles, “Reestructuración de la Arquitectura de Software para la Implementación de Software,” Universidad Cooperativa de Colombia, 2019.

[24] M. Laaziri, K. Benmoussa, S. Khouliji, and M. Kerkeb, “A Comparative study of PHP frameworks performance,” *Procedia Manuf.*, vol. 32, pp. 864–871, 2019, doi:10.1016/j.promfg.2019.02.295.

Evidencia de Sumisión



[ingeniare] Envío recibido

1 mensaje

Kristopher Chandía Valenzuela <noreply.ojs@scielo.org>
Para: Jeison Elí Sánchez Calle <jeisonelisanchez@gmail.com>

jue, 29 jun. 2023 a la hora 10:47 p. m.

Jeison Elí Sánchez Calle:

Gracias por enviar el artículo titulado "Arquitectura Multi-Tenant para la implementación de un software como servicio (SaaS) y su influencia en la usabilidad de las MYPES del sector comercio del Nor Oriente del Peruano" a Ingeniare. Revista chilena de ingeniería, pronto informaremos sobre la decisión editorial correspondiente (si será o no sometido al proceso de evaluación).

Gracias al sistema de gestión de revistas online que usamos, podrá seguir su progreso a través del proceso editorial identificándose en el sitio web de la revista:

URL del manuscrito: <https://cl.submission.scielo.org/index.php/ingeniare/authorDashboard/submission/9479>
Nombre de usuario/o: jeisonsanchez

Si tiene cualquier pregunta, no dude en contactarse con nosotros/as. Gracias por tener en cuenta esta revista para difundir su trabajo.

Kristopher Chandía Valenzuela

Dr. Kristopher Chandía Valenzuela Editor

Ingeniare. Revista chilena de ingeniería <http://cl.submission.scielo.org/index.php/ingeniare>

Esta mensagem pode conter informação confidencial, sendo seu sigilo protegido por lei. Se você não for o destinatário ou a pessoa autorizada a receber esta mensagem, não pode usar, copiar ou divulgar as informações nela contidas ou tomar qualquer ação baseada nessas informações. Se você recebeu esta mensagem por engano, por favor, avise imediatamente ao remetente, respondendo o e-mail e em seguida apague-a. Agradecemos sua cooperação.

This message may contain confidential information and its confidentiality is protected by law. If you are not the addressee or authorized person to receive this message, you must not use, copy, disclose or take any action based on it or any information herein. If you have received this message by mistake, please advise the sender immediately by replying the e-mail and then deleting it. Thank you for your cooperation.

Participantes

Carina García (cgarcia_c)

Jeison Elí Sánchez Calle (jeisonsanchez)

Mensajes

Nota

Estimados autores

Cordial saludo,

Hemos tomado una decisión sobre su presentación a
Ingeniare. Revista
chilena de ingeniería,

Trabajo recepcionado

Su trabajo está en línea editorial de la revista, por lo cual
ha sido
recepcionado

Se enviará a evaluar a los distintos árbitros y de acuerdo
a su

evaluación su artículo puede ser aceptado o rechazado.

Quedan pendientes

correcciones de formato, que se verán más adelante
una vez que avance proceso editorial.

Consultar por su artículo con el número i-9479 o título
de su trabajo de
preferencia por este medio o puede enviar correo.

Saluda atentamente a usted,

Carina García C.

Profesional Ingeniare. Revista chilena de ingeniería

<http://cl.submission.scielo.org/index.php/ingeniare>
ingeniare.uta.cl

Flujo de trabajo **Publicación****Envío** Revisión Editorial Producción

Ronda 1

Estado de ronda 1

Todos los revisores/as han contestado y hay que tomar una decisión.

Discusiones de revisión**Añadir discusión**

Nombre	De	Última respuesta	Respuestas	Cerrado
Revista Ingeniería Trabajo recepcionado 9479	cgarcía_c 2023-07-25 12:13 PM	-	0	<input type="checkbox"/>

Resolución de inscripción del perfil de proyecto de tesis



“AÑO DE LA UNIDAD, LA PAZ Y EL DESARROLLO”

RESOLUCIÓN N° 0067-2023/UPeU-FIA-CF-T

Lima, Ñaña 28 de febrero de 2023

VISTO:

El expediente de **Edwin Alexander Bautista Villegas**, identificado(a) con código universitario N° **201522202** y **Johan James Valles Paz** identificado(a) con código universitario N° **201322636** de la Escuela Profesional de Ingeniería de Sistemas de la Facultad de Ingeniería y Arquitectura de la Universidad Peruana Unión;

CONSIDERANDO:

Que la Universidad Peruana Unión tiene autonomía académica, administrativa y normativa, dentro del ámbito establecido por la Ley Universitaria N° 30220 y el Estatuto de la Universidad;

Que la Facultad de Ingeniería y Arquitectura de la Universidad Peruana Unión, mediante sus reglamentos académicos y administrativos, ha establecido las formas y procedimientos para la aprobación e inscripción del proyecto de tesis;

Que **Edwin Alexander Bautista Villegas** y **Johan James Valles Paz**, han solicitado la modificación de la denominación del proyecto de tesis titulado "Arquitectura Multi-Tenant en aplicaciones basadas en la nube y su influencia en la mejora de la rentabilidad de una empresa de TI";

Estando a lo acordado en la sesión del Consejo de la Facultad de Ingeniería y Arquitectura de la Universidad Peruana Unión, celebrada el 28 de febrero de 2023, y en aplicación del Estatuto y el Reglamento General de Investigación de la Universidad;

SE RESUELVE:

Aprobar la modificación de la denominación del proyecto de tesis titulado: "Arquitectura Multi-Tenant en aplicaciones basadas en la nube y su influencia en la mejora de la rentabilidad de una empresa de TI", por el de: "Arquitectura Multi-Tenant para la implementación de un software como servicio (SaaS) y su influencia en la usabilidad de las MYPES del sector comercio del Nor Oriente del Peruano", en el registro respectivo y disponer que con la orientación de su asesor el(la) **Asesor: Mg. Hitler Collantes Chules Coasesor: Mg. Nancy Esther Casildo Bedón**, sea desarrollado y ejecutado el proyecto de tesis por **Edwin Alexander Bautista Villegas** y **Johan James Valles Paz**, otorgándoles un plazo máximo de doce (12) meses para la ejecución, a partir de la inscripción inicial.

Regístrese, comuníquese y archívese.



Dra. Erika Inés Acuña Salinas
DECANA



Dr. Santiago Ramírez López
SECRETARÍA ACADÉMICA

CC:

- Interesado
- Asesor
- DGI
- Archivo

Instrumento de recolección de datos

N°	Pregunta	ESCALA DE LIKERT				
		Totalmente en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Totalmente de acuerdo
1	Creo que me gustaría utilizar frecuentemente este sitio web					
2	Encontré el sitio web innecesariamente complejo					
3	Pienso que el sitio web es fácil de usar					
4	Creo que necesitaré el apoyo de personal técnico para poder utilizar este sitio web					
5	Encontré que varias de las funciones en el sitio que estaban bien integradas					
6	Pensé que había demasiada inconsistencia en este sitio					
7	Me imagino que la mayoría de las personas podrían aprender a usar este sitio web muy rápido					
8	Encontré el sitio web muy difícil de usar					
9	Me sentí muy confiado (seguro) al utilizar el sitio web					
10	Necesité aprender muchas cosas antes de poder usar este sitio web					

Carta de autorización para ejecución de tesis

"Año del Fortalecimiento de la Soberanía Nacional"

CARTA N° 001-2022/CEATEC

A : **Mg. Danny Lévano Rodríguez**
Coordinador E.P. Ingeniería de Sistemas, Universidad Peruana Unión

ASUNTO : Carta de autorización para ejecución de proyecto de tesis.

REFERENCIA : CARTA N° 003-2022/FIA-EP.IS

FECHA : Tarapoto, 4 de julio de 2022

Es grato dirigirme a usted para saludarlo a nombre de la empresa **CEATECSOFT E.I.R.L.**

Por medio del presente se autoriza la ejecución del proyecto de tesis titulado **"Arquitectura Multi-Tenant en aplicaciones basadas en la nube y su influencia en la gestión logística y comercial de las pymes"** que será realizada por los bachilleres Edwin Alexander Bautista Villegas, con DNI N° 72462226, identificado con código universitario N°201522202 y Johann James Valles Paz, con DNI N° 48432529, identificado con código universitario N°201322636, de la carrera profesional de Ingeniería de Sistemas, Universidad Peruana Unión.

Sin otro particular me suscribo de usted, expresándole las muestras de mi especial consideración y estima.


CEATECSOFT E.I.R.L.
Ing. Héctor C. Pérez Ruiz
GERENTE GENERAL
Ing. Héctor C. Pérez Ruiz
Gerente General de CeatecSoft