

**COMUNIDAD VIRTUAL DE PRÁCTICA PARA PROPICIAR EL DESARROLLO
DE COMPETENCIAS EN LA CONSTRUCCIÓN DE SOFTWARE BASADA EN
EQUIPOS DESDE UN ENFOQUE ÁGIL**

**BRAYAN SEBASTIAN CÁRDENAS CÓRDOBA
JORGE OSWALDO RIASCOS ROMERO**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE SISTEMAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
SAN JUAN DE PASTO
2018**

**COMUNIDAD VIRTUAL DE PRÁCTICA PARA PROPICIAR EL DESARROLLO
DE COMPETENCIAS EN LA CONSTRUCCIÓN DE SOFTWARE BASADA EN
EQUIPOS DESDE UN ENFOQUE ÁGIL**

**BRAYAN SEBASTIAN CÁRDENAS CÓRDOBA
JORGE OSWALDO RIASCOS ROMERO**

Trabajo de grado presentado como requisito parcial para optar al título de
Ingeniero de Sistemas

Giovanni Albeiro Hernández Pantoja, Mg.
Director
Sandra Marleni Vallejo Chamorro, Mg.
Codirectora

UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE SISTEMAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
SAN JUAN DE PASTO
2018

NOTA DE RESPONSABILIDAD

“Las ideas y conclusiones aportadas en la tesis de grado son responsabilidad exclusiva de sus autores”.

Artículo 1o del acuerdo No 324 del 11 de octubre de 1966, emanado del honorable Consejo Directivo de la Universidad de Nariño.

“La Universidad de Nariño no se hace responsable de las opiniones o resultados obtenidos en el presente trabajo y para su publicación priman las normas sobre el derecho de autor”

Artículo 13, Acuerdo N. 005 de 2010 emanado del Honorable Consejo Académico.

NOTA DE ACEPTACIÓN

Presidente del jurado

FRANKLIN EDUARDO JIMENEZ GIRALDO
Jurado

JHON KELLY VILLOTA PISMAG
Jurado

20 de Abril de 2018.

AGRADECIMIENTOS

A Dios, gracias por su presencia a lo largo de mi vida y permitir que alcance una meta más.

A mi compañero, amigo y colega, Jorge Oswaldo Riascos Romero, por su compromiso, apoyo y trabajo en equipo.

Gracias a nuestros asesores, Giovanni Albeiro Hernández Pantoja y Sandra Marleni Vallejo, por su criterio, experiencia e intelecto, haciendo de lo difícil algo posible; por su tiempo y esfuerzo, creyendo y confiando en nosotros.

A los estudiantes y participantes de esa investigación, sin ellos no hubiese sido posible. Gracias por su tiempo y constancia.

Agradezco, infinitamente a mi madre por hacer más que un sacrificio durante toda mi vida, dando todo y más por verme cumplir mis metas. Ha hecho más de lo que debería, mi respeto y admiración absoluta para ella.

Finalmente gracias a mi familia, amigos y demás personas que de alguna manera hicieron parte de todo este proceso para hacerlo posible.

Sebastián.

Le doy gracias a mi familia por apoyarme incondicionalmente, alentarme y brindarme todo su afecto durante las etapas de mi vida.

A nuestros asesores, por compartir su tiempo, conocimiento y experiencia para guiarnos en este proceso de aprendizaje, también por su calidad humana y por brindarnos su amistad.

A mi compañero y amigo Sebastián por su disposición al trabajo, compromiso y responsabilidad.

Finalmente, agradezco a la Universidad de Nariño y al programa de Ingeniería de Sistemas por influir de manera positiva en nuestro proyecto de vida personal y profesional.

Jorge.

RESUMEN

Este trabajo, describe el proceso de conformación y ejecución de una comunidad virtual de práctica con el objetivo de incrementar las competencias en construcción de software en equipos con enfoque ágil en estudiantes de noveno semestre del periodo A de 2017 de Ingeniería de Sistemas en la Universidad de Nariño y determinar el nivel de aporte realizado por la comunidad.

Inicialmente, se identificaron niveles de desempeño en construcción de software basada en equipos que poseía la población en la dimensión del saber y saber hacer. Para ello se establecieron cinco variables (Etapa, Rol, Artefacto, Lineamiento y Herramienta), permitieron realizar la medición de niveles de desempeño durante todo el proceso. También, como técnicas de recolección de datos se utilizaron encuestas sociodemográficas, encuestas de conocimientos previos y talleres de aplicación.

Posteriormente, conformar la comunidad virtual de práctica, que se basó en los resultados iniciales para determinar temáticas y aspectos específicos a desarrollar. Para propiciar el desarrollo de competencias en construcción de software basada en equipos desde un enfoque ágil, se estableció una estrategia de aprendizaje basada en problemas y la práctica a través de unos recursos, herramientas y actividades que contribuyeron a un aprendizaje incremental. Finalmente, el proceso de aprendizaje de la comunidad virtual de práctica se replicaron encuesta y taller.

Al finalizar el proceso de la comunidad virtual de práctica, se realizó un análisis para determinar la evolución de competencias en construcción de software de la población objeto de estudio. Para ello se efectuó un contraste entre las competencias iniciales y finales en las dimensiones del saber y saber hacer acordes a los resultados de las respectivas encuestas y talleres, en el cual se encontró que en general la comunidad virtual de práctica generó un impacto positivo en las competencias en construcción de software en relación con las variables establecidas.

ABSTRACT

The present work describes the makeup and execution process of a virtual community of practice to increase the competences in team based software construction through an agile approach within System Engineering students of ninth semester during the first term of 2017 at the University of Nariño and to determine the level of contribution that was done by the community.

Firstly, the levels of performance in team building software construction were identified in the dimension of know to and know how to. To achieve this, five variables were set up (Stage, Role, Artefact, Guideline and Tool). This led to measure the levels of performance during the process. Techniques of collecting data like social surveys, knowledge surveys and an application activity.

Afterwards, the virtual community of practice was laid down, it was based in the initial results to determine the topics and specific aspects to develop. In order to create a favorable atmosphere in the development of competences in team based software construction through an agile approach, a problem based learning strategy was set up and through the practice with a set of resources, tools and activities which contributed to reach an incremental learning. In the final stage of the learning process of the virtual community of practice, the knowledge survey and the application activity were replied.

At the end of the virtual community of practice process, an analysis was done to determine the evolution of competences in team based software construction within the members of the community. To achieve this, a contrast between the initial and final competences was done, in which it was found that in general the virtual community of practice made a positive impact within the competences in software construction in regard to the variables.

TABLA DE CONTENIDO

GLOSARIO	13
INTRODUCCIÓN	14
1. MARCO TEÓRICO.....	20
2. METODOLOGÍA.....	25
3. RESULTADOS DE LA INVESTIGACIÓN.....	30
4. ANÁLISIS Y DISCUSIÓN DE LOS RESULTADOS.....	82
5. CONCLUSIONES.....	85
6. RECOMENDACIONES	87
BIBLIOGRAFÍA.....	88
ANEXOS	91

LISTA DE TABLAS

Tabla 1. Proceso de investigación _____	26
Tabla 2. Variables de la investigación _____	27
Tabla 3. Descripción sociodemográfica – Género _____	30
Tabla 4. Descripción sociodemográfica - Edad _____	31
Tabla 5. Descripción sociodemográfica - Lugar de origen _____	31
Tabla 6. Descripción sociodemográfica - Estrato _____	32
Tabla 7. Descripción sociodemográfica - Proyección del estudiante _____	33
Tabla 8. Competencias preliminares – Afinidad con el desarrollo de software ____	34
Tabla 9. Competencias preliminares - Conocimiento de metodologías _____	34
Tabla 10. Competencias preliminares – Etapa, etapa inicial _____	35
Tabla 11. Competencias preliminares – Etapa, etapa de ejecución _____	35
Tabla 12. Competencias preliminares – Etapa, etapa de cierre _____	36
Tabla 13. Competencias preliminares - Rol, cliente _____	36
Tabla 14. Competencias preliminares - Rol, Ingeniero de requisitos _____	37
Tabla 15. Competencias preliminares - Rol, Analista de software _____	37
Tabla 16. Competencias preliminares - Rol, Diseñador de software _____	38
Tabla 17. Competencias preliminares - Rol, Desarrollador _____	38
Tabla 18. Competencias preliminares - Rol, Jefe de calidad _____	39
Tabla 19. Competencias preliminares - Rol, jefe de pruebas _____	39
Tabla 20. Competencias preliminares - Rol, Líder de equipo _____	40
Tabla 21. Competencias preliminares - Rol, Líder de soporte _____	40
Tabla 22. Competencias preliminares - Artefacto, etapa inicial _____	41
Tabla 23. Competencias preliminares - Artefacto, etapa ejecución _____	42
Tabla 24. Competencias preliminares - Artefacto, etapa cierre _____	42
Tabla 25. Competencias preliminares - Lineamiento, etapa inicial _____	43
Tabla 26. Competencias preliminares - Lineamiento, etapa ejecución _____	44
Tabla 27. Competencias preliminares - Lineamiento, etapa cierre _____	45
Tabla 28. Competencias preliminares - Herramientas proceso construcción de software _____	45
Tabla 29. Actividades entregadas en el taller N° 1 _____	47
Tabla 30. Conocimientos posteriores - afinidad con el desarrollo de software ____	58
Tabla 31. Conocimientos posteriores - Conocimiento de metodologías _____	58
Tabla 32. Conocimientos posteriores - Variable Etapa, Etapa inicial _____	59
Tabla 33. Conocimientos posteriores - Variable Etapa, Etapa ejecución _____	59
Tabla 34. Conocimientos posteriores - Variable Etapa, Etapa cierre _____	60
Tabla 35. Conocimientos posteriores - Variable Rol _____	61
Tabla 36. Conocimientos posteriores - Variable Artefacto, Etapa inicial _____	61

Tabla 37. Conocimientos posteriores - Variable Artefacto, Etapa ejecución _____	62
Tabla 38. Conocimientos posteriores - Variable Artefacto, Etapa cierre _____	62
Tabla 39. Conocimientos posteriores - Variable Lineamiento, Etapa inicial _____	63
Tabla 40. Conocimientos posteriores - Variable Lineamientos, Etapa ejecución_	63
Tabla 41. Conocimientos posteriores - Variable Lineamientos, Etapa cierre _____	64
Tabla 42. Conocimientos posteriores - Variable Herramienta _____	64
Tabla 43. Conocimientos posteriores – Actividades entregadas _____	66

LISTA DE FIGURAS

Figura 1. Actividades entregadas Taller N° 1	48
Figura 2. Contraste resultados conocimientos previos variable Etapa	49
Figura 3. Resultados conocimientos previos variable Rol	50
Figura 4. Contraste resultados conocimientos previos variable Rol	50
Figura 5. Contraste resultados conocimientos previos variable Artefacto	51
Figura 6. Contraste resultados conocimientos previos variable Lineamiento	52
Figura 7. Resultado de conocimientos previos variable Herramienta	53
Figura 8. Arquitectura Comunidad virtual de práctica	56
Figura 9. Conocimientos posteriores - Actividades entregadas	66
Figura 10. Contraste resultados conocimientos posteriores variable etapa	67
Figura 11. Contraste resultados conocimientos posteriores variable Rol	68
Figura 12. Contraste resultados conocimientos posteriores variable Artefacto	69
Figura 13. Contraste resultados conocimientos posteriores variable Lineamiento	70
Figura 14. Contraste resultados conocimientos posteriores variable Herramienta	71
Figura 15. Comparación de competencias - Variable Etapa	72
Figura 16. Comparación de competencias - Variable Rol	73
Figura 17. Comparación de competencias - Variable Artefacto	73
Figura 18. Comparación de competencias - Variable Lineamiento	74
Figura 19. Comparación de competencias - Variable Herramienta	75
Figura 20. Comparación de competencias en Taller - Variable Etapa	76
Figura 21. Comparación de competencias en Taller - Variable Rol	77
Figura 22. Comparación de competencias en Taller - Variable Artefacto	77
Figura 23. Comparación de competencias en Taller - Variable Lineamiento	78
Figura 24. Comparación de competencias en Taller - Variable Herramienta	79
Figura 25. Resultados globales - Saber inicial	80
Figura 26. Resultados globales - Saber final	80
Figura 27. Resultados globales - Saber hacer inicial	81
Figura 28. Resultados globales - Saber hacer final	81
Figura 29. Material actividad Kanban, Lápices	101
Figura 30. Material actividad Kanban, Tableros Kanban	102
Figura 31. Material actividad Kanban, tarjetas visuales	102
Figura 32. Evidencias - Actividad Kanban	103
Figura 33. Actividades encontradas Actividad Kanban	104
Figura 34. Calificaciones del proceso Actividad Kanban	104

LISTA DE ANEXOS

Anexo A. Formulario información sociodemográfica _____	91
Anexo B. Formulario Competencias preliminares _____	95
Anexo C. Actividad del Taller N° 1. _____	98
Anexo D. Guía de observación _____	100
Anexo E. Actividad KANBAN _____	100
Anexo F. Material Audio Visual _____	105
Anexo G. Actividad del Taller N° 2 _____	105
Anexo H. Guía de instalación herramienta KUNAGI _____	107
Anexo I. Guía de uso Herramienta KUNAGI _____	119

GLOSARIO

Virtual: representación tecnológica del mundo real.

Medios Virtuales: herramienta en la que se apoyan procesos educativos y que facilitan la conformación de ambientes de aprendizaje.

Comunidad de práctica: espacio destinado para el aprendizaje en la cual los participantes interactúan de acuerdo a un interés común.

Identidad: tema central o interés común de una comunidad.

Comunidad: hace referencia a la población perteneciente o involucrada, que pueden estar divididos geográficamente.

Práctica: conjunto de actividades que propician el aprendizaje y desarrollo de un conjunto de competencias.

Competencias: conjunto de habilidades, actitudes y aptitudes dentro de un área específica.

Construcción de software: actividad intelectual y compleja para solucionar un problema dentro de un contexto mediante algoritmos y herramientas tecnológicas, generalmente se realiza en equipos de trabajo.

Enfoque ágil: consiste en una serie de lineamientos, que permite desligar al proceso de construcción de software de los métodos tradicionales y que busca agregar valor al proceso y producto software.

INTRODUCCIÓN

Las tecnologías de la información y las comunicaciones (TIC) han contribuido de gran manera al sector educativo, permitiendo que cada vez sea más fácil acceder a la información. Además, ha permitido utilizar herramientas para crear espacios digitales, los cuales son utilizados para apoyar procesos educativos y garantizar que el conocimiento llegue a los estudiantes teniendo acceso desde diferentes medios tecnológicos.

Adicionalmente, se hace necesario aprovechar el potencial que ofrecen las tecnologías existentes con el fin de fortalecer y transformar la educación, generando espacios alternativos de aprendizaje que fomenten una participación activa, de trabajo en equipo y de pensamiento colaborativo.

Por otra parte, las tecnologías de la información y las comunicaciones poseen elementos muy importantes que hacen posible un despliegue tecnológico y una constante evolución, uno de estos elementos es el Software. Debido a esto es indispensable que dicho Software sea de calidad y por esta razón, sea capaz de mantener los sistemas y las exigencias actuales. En virtud de ello los profesionales encargados de desarrollar Software deben poseer un conjunto de competencias para enfrentar los desafíos que conlleva esta actividad.

Por lo anterior, se pretende diseñar y conformar una comunidad virtual de práctica, con el fin de aportar al desarrollo de competencias en construcción de software basada en equipos en estudiantes de Ingeniería de Sistemas de la Universidad de Nariño. Para ello se realizará un estudio preliminar que permita determinar el nivel de desempeño actual que poseen los estudiantes en dichas competencias.

Posteriormente, se conformará una comunidad virtual de práctica, en la cual se aplicará una metodología de aprendizaje colaborativa, en donde los miembros interactúan entre sí, adquiriendo experticia en construcción de software basada en equipos y modelos de desarrollo ágiles durante el proceso de aplicación de la comunidad de práctica.

Finalmente, se realizará un análisis en el cual se contrasta el nivel de desempeño inicial y final de los miembros de la comunidad, estableciendo el grado de aporte que proporciona la comunidad virtual de práctica en el desarrollo de las competencias mencionadas.

Tema de investigación

Establecer el nivel de competencias en desarrollo de software basado en equipos desde un enfoque ágil, desarrollar dichas competencias mediante la conformación de una comunidad virtual de práctica.

Área de investigación

Esta investigación pertenece al área de Educación en ciencias de la computación e Ingeniería de software.

Línea de investigación

Línea Software y manejo de Información. Tiene como objetivo, planificar, analizar, diseñar, implantar, administrar sistemas complejos de información y de conocimiento.

Descripción del problema

Las tecnologías de la información y las comunicaciones (TIC) se han consolidado como una herramienta propicia para potencializar y transformar la educación, en especial la educación superior [1]. En este sentido, las Instituciones de Educación Superior (IES) como responsables del fomento del conocimiento en todos los ámbitos de la sociedad, ven en las TIC a un valioso aliado [2]. El uso de las TIC en estos procesos conlleva a modificar enfoques pasivos y tradicionales de la educación superior y enfrentar el desafío que impone una sociedad informatizada [3]. En este contexto, las comunidades virtuales de práctica (VCoP por sus siglas en inglés) aparecen como un aporte importante a la innovación académica. Introduciendo un cambio tanto en el ámbito de la modalidad (de presencial a virtual) como en la conceptualización de los procesos de formación de los estudiantes, basado en una cultura de aprendizaje autónomo, colaborativo y de conocimiento compartido, fundamentado en el desarrollo de competencias por medio de la práctica [4].

Una comunidad de práctica puede definirse como un grupo de personas que comparten un interés mutuo en un dominio específico, para lo cual participan en un proceso de aprendizaje colectivo [5]. En una comunidad virtual de práctica, los miembros (usuarios) son personas con ideas afines, pero geográficamente dispersas. Estos miembros no son pasivos en la construcción de saberes en línea, sino que también los crean y comparten, para desarrollar conocimientos, habilidades y actitudes en un tema específico de su interés [6]. Estas comunidades

de práctica tienen múltiples niveles y tipos de participación, los miembros pueden ser participantes centrales en una comunidad, y al mismo tiempo, ser participantes periféricos en otras. Incluso dentro de una misma comunidad, los miembros pueden ir y venir entre el núcleo y la periferia [7].

Por otra parte, diversos factores han revolucionado las formas como los estudiantes también interactúan con el mundo. Hoy en día es mucho más fácil acceder a la información [8] y existe una amplia difusión de información en Internet, redes sociales y dispositivos móviles. Estos elementos son usados de forma cotidiana por los estudiantes, por lo que llevar la educación a estos lugares u objetos es una prioridad, que debe aportar a cubrir las exigencias de un mercado globalizado que cada día exige mayor movilidad y agilidad en la información [9].

La movilidad y agilidad de la información, aportada por el sector de las TIC, brinda oportunidades únicas para el aumento y evolución de la económica, la política, el sector social y cultural a cualquier país [10]. Dentro del sector de las TI, la industria del Software al ser blanca, no contamina y genera fuentes de trabajo bien remuneradas [11], interviene de manera importante en la "nueva economía", debido a que algunas firmas (Empresas dedicadas a la construcción de software) en América Latina, ya se encuentran posicionadas y compitiendo por este mercado potencial.

Lo anterior, abre oportunidades como demanda de trabajadores profesionales en el desarrollo de software con el salario más competitivo de América Latina; para lo cual, se exige que los trabajadores sean competentes en la mayoría de los aspectos involucrados en el desarrollo de software [12]. Con base en las oportunidades laborales que presenta la industria del software, el Ministerio de Tecnologías de la Información y las Comunicaciones de Colombia, prevé que para el año 2018, 93.431 profesionales harían falta en la industria TI del país. Este reto, obliga a las Instituciones de Educación Superior a plantear alternativas que permitan el acceso a las personas a una formación que avance de consumidores a productores de software.

La construcción de software es una actividad compleja, que actualmente, se realiza por equipos, que deben dar respuesta a las necesidades crecientes de software en las organizaciones. No obstante, el trabajar en equipo tiene un conjunto de factores que inciden al momento de alcanzar los objetivos que esta labor traza. Putman, citado por McConell [13] logra demostrar que cuando el tamaño del equipo crece por encima de un grupo de personas, el esfuerzo se aumenta, pero el tiempo del proyecto, no se reduce. DeMarco [14], plantea que un equipo con exceso de trabajo, ocupado y sobresaturado no es garantía de mayor efectividad; y además, no permite

visualizar mayores beneficios para un proyecto. Así mismo, Brooks [15] a partir de la experiencia, éxito y fracaso en el desarrollo de software, logra concluir que, en un equipo de desarrollo, agregar más personas a un proyecto que tiene un retraso, provocará un retraso mayor. En trabajos intelectuales, como programar, escribir artículos, entre otros, las interrupciones son un mal que afecta enormemente a la productividad. Parnin y Rugaber [16] hicieron un estudio en el año 2010 sobre las interrupciones, siendo la conclusión más destacada que, lo normal es que a un programador le lleve de 10 a 15 minutos volver al estado de concentración previo al haber sido interrumpido. Otro aspecto fundamental, que incide en la productividad y desempeño de un equipo de trabajo es la comunicación. Los estudios realizados por Cockburn [17], permiten establecer que la forma más efectiva de comunicación corresponde a mantener al equipo interactuando en un sitio, frente a un tablero, formulando y respondiendo cuestionamientos e interrogantes. Este corresponde a un gran desafío en un periodo donde el teletrabajo y la contratación freelance, se ha convertido en una práctica habitual, por la escasez de mano de obra calificada para el desarrollo de software.

En este trabajo, una competencia se comprende como un conjunto de estructuras complejas en las dimensiones cognoscitiva (Conocimientos), actitudinal (Habilidades) y afectivo-motivacional (Actitudes) [18] que permiten solucionar problemas propios de la construcción de software basada en equipos. En este sentido, la comunidad de práctica pretende cubrir aspectos no incluidos dentro de la academia, dónde generalmente, se enseña a desarrollar software de manera individual, por imitación y utilizando ejercicios descontextualizados.

Por lo anteriormente descrito, cobra fuerza la necesidad de desarrollar competencias que les permita a los estudiantes del programa profesional de Ingeniería de Sistemas construir software a través de equipos de trabajo, como una oportunidad de innovar no únicamente en la educación, sino en otras áreas de conocimiento.

Formulación del problema

¿Cómo aportar al desarrollo de competencias para la construcción de software basada en equipos, en los estudiantes de Ingeniería de Sistemas de la Universidad de Nariño?

Sistematización del problema

- ¿Cuál es el nivel de desempeño que tienen los estudiantes de Ingeniería de Sistemas en la construcción de software basada en equipos?
- ¿Cómo conformar una comunidad virtual de práctica para el desarrollo de competencias en construcción de software basada en equipos?
- ¿Cuál es el nivel de aporte hecho por la comunidad virtual de práctica en el desarrollo de competencias en la construcción de software basada en equipos, a los miembros de la comunidad?

Objetivos de la investigación

General

Aportar al desarrollo de competencias para la construcción de software basada en equipos, en los estudiantes de Ingeniería de Sistemas de la Universidad de Nariño mediante el uso de una comunidad virtual de práctica.

Específicos

A continuación, se relacionan los fines específicos que deberán cumplirse para lograr en conjunto el objeto del presente estudio:

- Identificar el nivel de desempeño que tienen los estudiantes de Ingeniería de Sistemas en la construcción de software basada en equipos.
- Conformar una comunidad virtual de práctica para el desarrollo de competencias en construcción de software basada en equipos.
- Determinar el nivel de aporte hecho por la comunidad virtual de práctica en el desarrollo de competencias en construcción de software basada en equipos, a los miembros de la comunidad.

Justificación

Este trabajo es interesante, porque permite establecer el nivel en el que se encuentran los estudiantes con respecto a las competencias en construcción de software basada en equipos e identifica los niveles de desempeño. Además, se logra observar la evolución colectiva de dichas competencias durante la conformación y uso de la comunidad virtual de práctica, donde los integrantes

interactúan de manera activa recibiendo, generando y compartiendo conocimiento y experiencias en la construcción de software basada en equipos y adicionalmente se determina de manera directa, como la comunidad virtual influye en la enseñanza y desempeño de los miembros, también permite contrastar el nivel inicial y final de desempeño de los estudiantes para identificar aportes significativos y características a mejorar de la comunidad virtual de práctica.

Por otra parte, se puede identificar características comunes entre los estudiantes de Ingeniería de Sistemas y de esta forma planificar estrategias que faciliten el aprendizaje, desarrollo y fortalecimiento de las competencias en la construcción de software basada en equipos. Al finalizar todo el proceso, se puede determinar si la comunidad de práctica influye de manera positiva en el desarrollo de las competencias, pero también identificar aspectos a mejorar. Lo anterior, posibilitará tomar decisiones estratégicas en la creación y aplicación de futuras comunidades de práctica y generar nuevas estrategias de aprendizaje.

Por lo tanto, se puede decir que el objetivo de este trabajo investigativo se encuentra dentro de un marco novedoso ya que se crea un nuevo espacio de aprendizaje el cual se sale del esquema de aprendizaje tradicional, utilizando diversos medios y herramientas tecnológicas disponibles con el fin de que los estudiantes puedan ampliar y mejorar las competencias en la construcción de software basada en equipos.

Delimitación

Este trabajo tiene como fin contribuir al desarrollo de competencias para la construcción de software basada en equipos, en la cual se realiza un estudio previo con el fin de determinar el nivel actual de los estudiantes de Ingeniería de Sistemas de noveno semestre de la Universidad de Nariño. Posteriormente, se conformará una comunidad virtual de práctica que contribuya en el desarrollo de las competencias en este campo. Una vez finalizada la aplicación de la comunidad virtual de práctica, se determinará el nivel de influencia en el alcance de estas competencias por parte de los participantes. Los beneficiarios, son todos los interesados en la construcción de software basada en equipos, metodologías ágiles de desarrollo y además aquellos interesados en el aprendizaje a través de comunidades de práctica desde un ámbito virtual.

1. MARCO TEÓRICO

1.1. ANTECEDENTES

A continuación, se describe algunos de los trabajos más representativos vinculados a las temáticas de la presente investigación.

Pan y otros [5]; y Jiménez-Zarco y otros [8], proponen una comunidad virtual de práctica, como un sistema independiente de gestión del conocimiento. Para esto integran redes sociales de apoyo en beneficio de comunidad de práctica virtual. Para compartir el conocimiento utilizan foros de discusión. El estudio de campo realizado en este trabajo concluye que el apoyo de las redes sociales intensifica el intercambio de conocimientos por parejas de actores, aclarando que el diálogo entre amigos incrementa la probabilidad de colaboración e intercambio de conocimientos.

Por otra parte, en Rodríguez y otros [19]; y Rogo y otros [6], describen que las comunidades de práctica desempeñan un papel importante en los procesos de innovación, pero no dejan claro cuál es su función particular. En este artículo los autores buscan diferenciar los tipos de conocimientos adquiridos con la práctica, con el fin aplicarlos en innovación empresarial de una manera colaborativa. Para lograr esto, plantean elementos deductivos que destacan diferentes impulsos de reflexividad y diferentes lógicas internas de la reflexividad. Los autores definen que mediante la introducción de una perspectiva dinámica en los dos niveles de reflexividad permiten mejorar los procesos de innovación corporativa.

En relación con el aprendizaje virtual Ratzinger-Sakel y Grayen [7] proponen propiciar las condiciones para que surja y se forme una comunidad de práctica en el estudio de la contabilidad, enfocándose sobre la profesión de auditoría, para lo cual plantean construir un compendio de lo aprendido en la profesión, investigación contable, las prácticas estudiantil y profesional, de contabilidad. Para esto integra temas de interés para la comunidad de práctica; haciendo énfasis en la visión compartida de un cuerpo de conocimiento en temas contables de auditoría.

Para Jiménez-Zarco y otros [8] describen la manera como se propicia condiciones para que se forme una comunidad de práctica en la apropiación de conceptos del sector sanitario, proponiendo un espacio de aprendizaje colaborativo, haciendo uso de una red social, donde se vinculan personas con diferentes rasgos, pero con un interés en común para intercambiar información, experiencias y contenidos relacionados con el dominio del sector sanitario. Con el propósito de que la gente

pueda crear y compartir conocimientos, y aprender colaborativamente, para lograr beneficios relacionados con el diagnóstico y el tratamiento de los pacientes en relación con el costo, realizar una gestión más eficiente, incrementar la calidad y precisión del diagnóstico. Además, se pretendió aumentar el capital social de los participantes y crear redes de personas de confianza, dado el interés en este tema. Los resultados de participación en la comunidad de práctica demuestran que una intervención activa y colaborativa impacta de manera positiva en cada uno de los miembros incrementando el interés en el tema tratado.

En el trabajo realizado por Dascalu y otros [9], se propone componer grupos de aprendizaje efectivos, donde los miembros tienen diferentes antecedentes de conocimiento en un dominio dado. El estudio define que la cuantificación de diversos indicadores, como el fondo de la diversidad y la similitud entre el tipo de interés de los participantes dentro de un grupo y entre los grupos, pueden tener un impacto positivo en brindar condiciones para que una comunidad de práctica surja y se forme, enfocándose principalmente en el aprendizaje continuo de sus miembros.

Clarke [20], plantea analizar el aprendizaje de los estudiantes que acceden a los componentes en línea de un curso creado con base en la teoría de las comunidades de práctica, el cual reconoce que el desarrollo de competencias avanza en los estudiantes de pedagogía, cuando se trabaja bajo estos lineamientos en la construcción de su propio plan de estudios. También, se examina las cuestiones tecnológicas y pedagógicas clave que afectan el aprendizaje en línea de los estudiantes. Los resultados obtenidos muestran que la gran mayoría de los miembros utiliza con frecuencia los recursos educativos colocados a disposición de la comunidad, lo que permite aumentar el interés de participación en actividades colaborativas de cada uno de sus miembros.

En el trabajo presentado por Ordoñez y Hernández [21], se plantea potencializar las competencias de los estudiantes de informática y carreras afines en el desarrollo de aplicaciones para dispositivos móviles. Esta investigación divide las competencias de los estudiantes para esta área en seis ejes, y demuestra como a través de la comunidad de práctica se logra avanzar en el desarrollo de competencias a través de la construcción de conocimiento de manera colaborativa y apoyado por casos de estudio.

En relación con los antecedentes revisados, la presente investigación tiene como similitudes, el uso de una comunidad de práctica como medio para desarrollar competencias en un área específica de conocimiento, establece las competencias y niveles de aprendizaje que se desea alcanzar y se apoya de otros recursos computacionales como las redes sociales y entornos virtuales de aprendizaje. No

obstante, en este trabajo investigativo, se presenta como novedad, en primer lugar, al clasificar las competencias en tres dimensiones (Conocimientos, habilidades y aptitudes). Además, se elaboró una planeación que parte de las competencias que se desean desarrollar, a partir de las cuáles, se definen los niveles de aprendizaje y se utiliza como estrategia los casos de estudio. Así mismo, se organizan los recursos y su forma de uso; y se posibilita que la comunidad de práctica se auto organice y auto gestione. De igual forma, aún no se ha planteado la formación de una comunidad de practica que permita a un conjunto de estudiantes de Ingeniería de Sistemas, ubicados geográficamente distantes, reunirse en un espacio virtual, dónde puedan potencializar colaborativamente su conocimiento, habilidades y actitudes en la construcción de software basada en equipos desde un enfoque ágil, a través del intercambio de experiencias, buenas prácticas y conocimientos entre todos los miembros de la comunidad.

1.2. SUPUESTOS TEÓRICOS Y DEFINICIÓN DE CONCEPTOS

El término virtual es tratado por Duart Joseph y Sangrá Albert [22], en el cual se define como una representación tecnológica del mundo real, existen interacciones que buscan el cumplimiento de objetivos específicos. Estas no dependen del espacio ni del tiempo en el cual se realizan, además se involucra y profundiza en la virtualidad como medio y herramienta para la difusión del conocimiento y el apoyo al aprendizaje en procesos educativos.

El sector educativo no es ajeno a los avances tecnológicos y los cambios que estos generan en la sociedad, por esta razón se hace necesario integrar medios virtuales, los cuales proporcionan aportes significativos en procesos de aprendizaje y también permiten transformar los ambientes de aprendizaje tradicionales.

Por otra parte, Manuel Cebrian [23] describe lo virtual en un ambiente educativo y principalmente en el Universitario, como aquello que permite a los estudiantes, mediante un entorno de redes y computadores concebir espacios innovadores para que esta comunidad desarrolle su comunicación y gestionar de manera alternativa los servicios de la universidad como la investigación, la enseñanza y aprendizaje.

Una definición de comunidad de práctica es trabajada por Etienne Wenger [24] en el cual establece a las comunidades de práctica como un contexto para el aprendizaje entre los participantes que la conformen y cuyo objetivo es el de transformar el conocimiento a través las experiencias y competencias de cada integrante, llevando el conocimiento individual a un conocimiento global o colectivo.

Además, las comunidades de práctica propician espacios idóneos de aprendizaje y son un espacio privilegiado para la apropiación y evolución del conocimiento. En la presente investigación se trabaja con un tipo específico de comunidad de práctica, que está enfocada en un ambiente virtual. Una comunidad virtual se puede definir según Howard Rheingold [25], como un ecosistema que está conformado por unas subculturas y unos grupos que se conforman de una forma espontánea y sencilla. Así como entre grupos e integrantes de diferentes culturas y países permiten generar una serie de controversias informativas que a su vez hacen de los participantes unos críticos y generadores de nuevo conocimiento a partir de combinar los diferentes puntos de vista.

Las comunidades de práctica se componen de tres dimensiones o componentes principales a lo cual Wegner, McDermott y Snyder [26], establecen estas tres dimensiones como la identidad, la comunidad y la práctica. La identidad se refiere al tema central de la comunidad o al interés común de los participantes, la comunidad se refiere a la población perteneciente y/o involucrada con la investigación, la cual puede o no estar ubicada en la misma zona geográfica, pues las interacciones pueden darse incluso en diferentes espacios de tiempo, y por último se tiene la práctica, la cual está ligada directamente con el aprendizaje.

Según Almenara Julio [27], las comunidades virtuales de práctica en la educación se refieren a comunidades de personas que, mediante herramientas de comunicación como las redes telemáticas y la web, permiten un intercambio de valores e intereses comunes. Las comunidades virtuales buscan un aspecto de interacción social y en esta medida es un espacio ideal para el sector educativo ya que el interés es el aprendizaje de un tema común para los miembros de la comunidad virtual. Además, Zoia Bozu [28] concluye después de la creación e implementación de diferentes comunidades virtuales de práctica en el campo universitario que se trata de un grupo de personas que interaccionan continuamente y que comparten una preocupación y unos problemas comunes y con una gran diversidad al estar compuesta por distintas sensibilidades y formas de entender el conocimiento pero que de todas formas comparten espacios comunes para el entendimiento. La comunidad virtual de práctica tiene como finalidad el hacer explícito la transferencia informal de conocimiento, constituyendo una estructura formal que hace posible la adquisición de conocimiento a través de las experiencias que se comparten dentro del grupo y así se cultiva una identidad hacia el grupo y se refuerza como potencial el aprendizaje mediante un proceso de participación y liderazgo colectivo o compartido.

Las metodologías ágiles de desarrollo surgen como una respuesta ante la evolución de la industria del software en los años 90 pero fue hasta el año 2001 en que se

formalizó un acuerdo entre grandes empresarios y profesionales de la industria del software que se conformó un documento en el cual se proponen las metodologías principios y valores en la construcción de software, dicho documento fue contemplado como el manifiesto ágil.

Según lo establecido por Meyer [29], la construcción de software desde un enfoque ágil es una metodología que no está ligada a los métodos tradicionales y que permiten al equipo desarrollador mantener un contacto más fluido con el cliente o con beneficiario final del producto, obteniendo un producto software que se ajusta a sus verdaderas necesidades, siendo en pocas palabras útil y de alta calidad. Además, se plantea a la construcción de software basada en equipos como una serie de procedimientos que son asignados a cada integrante del equipo de desarrollo teniendo en cuenta sus aptitudes y habilidades durante el ciclo de vida del proyecto, con el fin de generar un producto software que cumpla estándares de calidad como la funcionalidad, portabilidad, compatibilidad, entre otros.

Cabe resaltar la importancia de hacer un desarrollo colaborativo o un trabajar en equipos, pero a su vez manteniendo una motivación constante de cada integrante, puesto que según lo que plantea Pressman [30], las personas motivadas van a utilizar la colaboración de tal manera que multiplique su talento y producción creativa más allá de sus resultados promedio; Pero a su vez determina que la colaboración o trabajo en equipo no es fácil, dado que se trata con personas que tienen un entorno personal y ajeno al equipo en cuestión. Por último, se enfoca en el individualismo como aquello que genera creatividad individual y tiene un papel importante en el pensamiento colectivo.

2. METODOLOGÍA

Paradigma de la investigación

Esta investigación es de corte cuantitativo, porque según Hernández Sampieri, Fernández y Baptista (21), representa un conjunto de procesos secuenciales y probatorios, por lo tanto no se pueden saltar los pasos debido a que el orden es riguroso que parte de una idea que una vez delimitada, se deriva en objetivos y preguntas de investigación, se determinan variables y la validez de los resultados se soportarán a través de un proceso estadístico descriptivo debido a que se realizan mediciones, se utilizan técnicas de recolección y se analizan los datos procediendo de manera inductiva donde se obtendrán conclusiones empíricas.

Enfoque de la investigación

El enfoque para esta investigación es empírico-analítico (22), debido a que se basa en las experiencias propias para propiciar el desarrollo de competencias en la construcción de software basada en equipos desde un enfoque ágil mediante una comunidad virtual de práctica.

Tipo de investigación

El tipo de investigación a utilizar en el presente trabajo es descriptivo-propositivo (22), este tipo de estudio busca describir situaciones o acontecimientos en cuanto al nivel de desempeño en la construcción de software basada en equipos de los estudiantes de Ingeniería de Sistemas. Sin embargo, después de la elaboración de las descripciones, se conformará una comunidad virtual de práctica como propuesta de intervención para determinar cuál es el nivel de aporte, que esta puede lograr en el desarrollo de competencias para propiciar el desarrollo de competencias en la construcción de software basada en equipos desde un enfoque ágil.

Diseño de la investigación

Tabla 1. Proceso de investigación

Objetivos específicos	Fuente	Técnica de recolección	Instrumento	Técnica de Procesamiento	Resultado
Identificar el nivel de desempeño que tienen los estudiantes de Ingeniería de Sistemas para la construcción de software a través de equipos.	Estudiantes de Ingeniería de Sistemas.	- Encuesta - Taller	- Cuestionario - Guía del taller	Estadística Descriptiva	Documento con el nivel de desempeño alcanzado por los estudiantes de Ingeniería de Sistemas
Determinar el nivel de aporte hecho por la comunidad de práctica virtual en el desarrollo de competencias en la construcción de software basada en equipos, a los miembros de la comunidad.	Estudiantes de Ingeniería de Sistemas.	- Encuesta - Taller	- Cuestionario - Guía del taller	Estadística descriptiva	Documento con un contraste entre el nivel de desempeño alcanzado por los estudiantes de Ingeniería de Sistemas antes y después de la comunidad de práctica virtual.

Tabla 2. Variables de la investigación

Variable	Indicador	Naturaleza	Valores	Fuente	Técnica de recolección	Técnica de análisis
Etapa	Nivel de competencia alcanzado en el conocimiento y aplicación de las etapas y actividades que se realizan para hacer software a través de equipos.	Cualitativa	<ul style="list-style-type: none"> - Muy alto (Mayor o igual a 90%) - Alto (Entre 80% y 90%) - Medio (Entre 60% y 79%) - Bajo (Entre 40% y 59%) - Muy Bajo (Menor a 40%) 	Estudiantes de Ingeniería de Sistemas de la Universidad de Nariño.	<ul style="list-style-type: none"> - Encuesta - Taller 	Estadística descriptiva
Rol	Nivel de competencia alcanzado en el conocimiento y aplicación de los roles que se desempeñan para hacer software a través de equipos.	Cualitativa	<ul style="list-style-type: none"> - Muy alto (Mayor o igual a 90%) - Alto (Entre 80% y 90%) - Medio (Entre 60% y 79%) - Bajo (Entre 40% y 59%) - Muy Bajo (Menor a 40%) 	Estudiantes de Ingeniería de Sistemas de la Universidad de Nariño.	<ul style="list-style-type: none"> - Encuesta - Taller 	Estadística descriptiva
Artefacto	Nivel de competencia alcanzado en el conocimiento y aplicación de los artefactos que se elaboran para hacer software a través de equipos.	Cualitativa	<ul style="list-style-type: none"> - Muy alto (Mayor o igual a 90%) - Alto (Entre 80% y 90%) - Medio (Entre 60% y 79%) - Bajo (Entre 40% y 59%) - Muy Bajo (Menor a 40%) 	Estudiantes de Ingeniería de Sistemas de la Universidad de Nariño.	<ul style="list-style-type: none"> - Encuesta - Taller 	Estadística descriptiva

Lineamiento	Nivel de competencia alcanzado en el conocimiento y aplicación de los principios que sirven para tomar decisiones al hacer software a través de equipos.	Cualitativa	<ul style="list-style-type: none"> - Muy alto (Mayor o igual a 90%) - Alto (Entre 80% y 90%) - Medio (Entre 60% y 79%) - Bajo (Entre 40% y 59%) - Muy Bajo (Menor a 40%) 	Estudiantes de Ingeniería de Sistemas de la Universidad de Nariño.	<ul style="list-style-type: none"> - Encuesta - Taller 	Estadística descriptiva
-------------	----------------------------------------------------------------------------------------------------------------------------------------------------------	-------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------	--------------------------------------------------------------------------------	-------------------------

Universo y/o muestra

La población corresponde a los estudiantes de Ingeniería de Sistemas de la Universidad de Nariño. El muestreo es no probabilístico de tipo intencional, debido a que, para el estudio, se trabajará con los estudiantes de noveno semestre que estén cursando la materia de Ingeniería de Software aplicada.

Instrumentos de recolección de la información

En esta investigación se aplicaron instrumentos de recolección de información como la encuesta, la cual fue utilizada para obtener datos sociodemográficos de la población y datos de tipo académico.

Por otra parte, se utilizó el taller de aplicación, el cual permitió realizar una observación directa del trabajo en equipo de la población y finalmente documentar los resultados para su posterior análisis.

Validez y confiabilidad de los instrumentos

La validación de los instrumentos se efectuó a través de juicio de expertos, los cuales realizaron las observaciones y correcciones pertinentes previas a la aplicación de los mismos.

Para garantizar la confiabilidad de los instrumentos, se realizaron pruebas piloto con cinco estudiantes egresados del programa de Ingeniería de Sistemas en la Universidad de Nariño. Esto permitió realizar una retroalimentación para verificar y

corregir posibles errores de formulación, asegurar que los instrumentos sean lo más claros posibles y observar el tipo de resultados obtenidos en las pruebas.

Procesamiento de la información

El procesamiento de la información se llevó a cabo mediante estadística descriptiva, haciendo uso del cálculo de frecuencias, medidas de tendencia central y distribución para indicadores continuos.

3. RESULTADOS DE LA INVESTIGACIÓN

3.1. IDENTIFICAR EL NIVEL DE DESEMPEÑO QUE TIENEN LOS ESTUDIANTES DE INGENIERÍA DE SISTEMAS EN LA CONSTRUCCIÓN DE SOFTWARE BASADA EN EQUIPOS.

En esta sección se describe cómo se identificó el nivel de desempeño que tienen los estudiantes de Ingeniería de Sistemas en la construcción de software basada en equipos. Para alcanzar este objetivo, se tuvo como fuente de información a los estudiantes del programa de Ingeniería de Sistemas que cursaron el noveno semestre en el periodo A de 2017. Las técnicas que se utilizaron para la recolección de información fueron: la encuesta y el taller. Para el análisis de la información se utilizó como técnica la estadística descriptiva. Las variables analizadas fueron: etapa, rol, artefacto, lineamiento y herramienta.

Una vez presentado el contexto y la forma en cómo se realizó el análisis cuantitativo de los datos, se procede a mostrar los resultados encontrados.

Los resultados que a continuación se presentan, describen inicialmente a la población de informantes. Luego, se detalla las percepciones de la población de acuerdo con las variables, después se realiza un contraste de técnicas entre la encuesta y el taller. Finalmente, se hace una síntesis de los resultados.

3.1.1. Descripción sociodemográfica de la población. La población encuestada estuvo conformada por 30 estudiantes del programa de Ingeniería de Sistemas de la Universidad de Nariño, que cursaron la materia de Ingeniería de Software Aplicada en el periodo A de 2017.

Como se puede observar en la tabla 3, el 80% de los estudiantes pertenecen al género masculino y el 20% al género femenino.

Tabla 3. Descripción sociodemográfica – Género

Categoría	FO - Frecuencia Observada	Frecuencia Observada (%)
Masculino	24	80%
Femenino	6	20%
Total	30	100%

El mayor porcentaje de los estudiantes encuestados se encuentran entre los 21 y 22 años con un 47%, como se puede observar en la Tabla 4. Así mismo, el promedio de edad corresponde a 22.7 años, con una mediana de 22.5 años. La dispersión de los datos es de 1.7 años, lo que indica que las edades tienden a concentrarse a la media. Finalmente, se puede observar que existe una asimetría positiva lo que indica que la mayor concentración de las edades está a la izquierda de la media. Los datos anteriores, permiten inferir que el 50% de la población objeto de estudio, no ha perdido materias en la línea de software. También puede suceder que aquellos estudiantes que superan la media iniciaron a estudiar con una edad superior a los 18 años o han perdido alguna materia que le ha retrasado el proceso de formación.

Tabla 4. Descripción sociodemográfica - Edad

Rango	FO - Frecuencia Observada	Frecuencia Observada (%)
21 y 22	14	47%
23 y 24	11	37%
> 25	4	13%
< 21	1	3%
Total	30	100%

Como se puede observar en la Tabla 5, la mayoría de los estudiantes son originarios de la ciudad de Pasto con un porcentaje del 53% con respecto al total, se observa que solo dos estudiantes vienen de un Departamento diferente y que todos los demás con un 40% del total, son de diferentes Municipios de Nariño.

Tabla 5. Descripción sociodemográfica - Lugar de origen

Categoría	FO - Frecuencia Observada	Frecuencia Observada (%)
Pasto	16	53%
Tangua	2	7%
Arboleda-Berruecos	1	3%
Cali	1	3%
Cuaspud	1	3%
Cumbal	1	3%
El tambo N	1	3%
Guachucal	1	3%
Guaitarilla	1	3%

Ipiales	1	3%
La unión Nariño	1	3%
Samaniego	1	3%
Sibundoy- Putumayo	1	3%
Tuquerres	1	3%
Total	30	100%

El mayor porcentaje de los estudiantes se encuentran en los estratos 1 y 2 con un 43% y 37% respectivamente, La dispersión de los datos es de 0,91. Además el coeficiente de asimetría es positivo así que los datos están concentrados a la izquierda de la media. Los datos anteriores, que se pueden observar en la Tabla 6, permiten inferir que el 80% de la población objeto de estudio pertenecen a estratos bajos de la ciudad de pasto, o que provienen de municipios del departamento de Nariño.

Tabla 6. Descripción sociodemográfica - Estrato

Estrato	FO - Frecuencia Observada	Frecuencia Observada (%)
1	13	43%
2	11	37%
3	4	13%
4	2	7%
Total	30	100%

En cuanto a las actividades que los estudiantes se proyectan a realizar en cinco años, en la Tabla 7 se destacan actividades relacionadas con el desarrollo de software con un 30% del total de estudiantes y actividades en el campo empresarial, principalmente destinados a la creación de empresa con un 37%. El 33% restante corresponde a actividades de estudios de postgrado y de proyectos de investigación, además de diferentes actividades laborales en el área de las TIC.

Tabla 7. Descripción sociodemográfica - Proyección del estudiante

Actividad	FO - Frecuencia Observada	Frecuencia Observada (%)
Emprendimiento	11	37%
Desarrollo de Software	9	30%
Tecnología de la Información	7	23%
Investigación y estudios de postgrado	3	10%
Total	30	100%

3.1.2. Descripción de las variables de análisis. Con el fin de determinar el nivel de desempeño y las competencias en construcción de Software por equipos, Se han establecido cinco variables a ser analizadas a partir de los resultados de la encuesta y el taller. A continuación, se presenta la definición de cada una de las variables utilizadas.

- **ETAPA:** una etapa es un elemento el cual reúne a un conjunto de actividades que se realizan para para alcanzar un fin.
- **ROL:** un rol es la función o papel que una persona desempeña en un equipo de trabajo.
- **ARTEFACTO:** un artefacto o entregable es un producto que se realiza y que agrega valor a la construcción de un producto software.
- **LINEAMIENTO:** un lineamiento es una tendencia, una dirección o un rasgo que posibilita la toma de decisiones.
- **HERRAMIENTA:** una herramienta es un elemento que permite soportar y facilitar procesos de desarrollo de software.

Encuesta

Etapa

Se observa en la tabla 8, que el 60% de los estudiantes consideran tener una BUENA afinidad en cuanto al desarrollo de software, mientras que el 26.7% de ellos consideran que es REGULAR, por último el 10% y 3.3% tienen una MUY BUENA y DEFICIENTE afinidad respectivamente.

Tabla 8. Competencias preliminares – Afinidad con el desarrollo de software

Categoría	FO - Frecuencia Observada	Frecuencia Observada (%)
MUY BUENA	3	10.0%
BUENA	18	60.0%
REGULAR	8	26.7%
DEFICIENTE	1	3.3%
TOTAL	30	100.0%

En la tabla 9, el 73.3% de los encuestados afirman conocer una metodología para construir software a través de equipos de trabajo mientras que solo el 26.7% no conoce metodologías. De quienes afirman conocer una metodología, la mayoría seleccionó SCRUM como la más conocida, las demás metodologías que se tienen como referencia son: PSP, XP, RUP, TSP, RAD, en cascada, clásica, espiral y metodologías ágiles.

Tabla 9. Competencias preliminares - Conocimiento de metodologías

Categoría	FO - Frecuencia Observada	Frecuencia Observada (%)
SI	22	73.3%
NO	8	26.7%
TOTAL	30	100.0%

En cuanto a la etapa inicial y de qué se trata, el 33.3% de los estudiantes expresan explícitamente que se trabaja la gestión de requisitos, pero la totalidad de los encuestados mencionan como parte de esta etapa los requisitos, gestión o elicitación de requerimientos. Por otra parte, el 43.3% de los encuestados mencionan el proceso de Análisis, el 23.3% restantes expresan que se trabajan aspectos como la definición y necesidades a cumplir por el software y sus

delimitaciones, del mismo modo una asignación de roles y planeación de las actividades a desarrollar. Observado en la Tabla 10.

Tabla 10. Competencias preliminares – Etapa, etapa inicial

Categoría	FO - Frecuencia Observada	Frecuencia Observada (%)
Proceso de Análisis	13	43.3%
Gestión de requisitos	10	33.3%
Definición de roles, necesidades y delimitación de software.	7	23.3%
TOTAL	30	100%

En cuanto a la etapa de ejecución, con respecto a la Tabla 11, el 53% de los estudiantes denotaron que se trabaja el desarrollo del proyecto o producto, el 27% de ellos expresa que esta etapa se relaciona con la codificación del producto software. El 20% restante, manifiesta que en esta etapa se trabajan aspectos como el registro de los resultados finales de las actividades, al igual que implementar pruebas y/o retroalimentación e inclusive hacer una socialización de las necesidades con el equipo de trabajo, definir las actividades y hacer un seguimiento y control; sin dejar de lado el análisis de la información.

Tabla 11. Competencias preliminares – Etapa, etapa de ejecución

Categoría	FO - Frecuencia Observada	Frecuencia Observada (%)
Desarrollo del proyecto o producto software	16	53.3%
Codificación y/o producción del proyecto o entrega ágil del mismo	8	26.7%
Registro de resultados finales de las actividades, testing y/o retroalimentación.	6	20%
TOTAL	30	100%

Con respecto a la etapa de cierre, el 50% de los estudiantes encuestados, expresan que esta etapa es para generar la entrega definitiva del producto desarrollado,

dejando en claro una verificación de calidad, una retroalimentación y un posterior mantenimiento de este para el cliente. EL 30% de ellos dicen que en esta etapa se trabaja la parte de pruebas al producto, con el fin de garantizar el correcto funcionamiento y la calidad garantizada del mismo. El 20% restante presentan esta etapa como aquella en la que en la que se produce la documentación del producto y/o proyecto, análisis de los resultados y del mismo modo donde se debe hacer una socialización de los resultados obtenidos y así generar un informe final. Observado en la Tabla 12.

Tabla 12. Competencias preliminares – Etapa, etapa de cierre

Categoría	FO - Frecuencia Observada	Frecuencia Observada (%)
Entrega definitiva del producto en desarrollo o ya desarrollado	15	50%
Pruebas y/o testing del producto	9	30%
Documentación del producto y/o proyecto, análisis de los resultados y socialización.	6	20%
TOTAL	30	100%

Rol

Según la Tabla 13, más de la mitad de la población tiene la percepción de que la principal actividad de los clientes en un proyecto de construcción de Software es el de suministrar la información del contexto y del problema a resolver mediante un producto Software.

Tabla 13. Competencias preliminares - Rol, cliente

Acciones desempeñadas por el Cliente o Interesado	Frecuencia Observada	FO %
Suministrar Información del contexto y del problema para solucionar con un producto software	16	53%
Solicitar un producto software	6	20%
Entrega los requerimientos del producto Software	6	20%
Gestiona el proceso de desarrollo	2	7%
Total	30	100%

La mitad de la población afirmó que la actividad que define a un ingeniero de requisitos es la de definir los requerimientos del software a construir; mientras que

el 33% dijo que el Ingeniero de requisitos es aquel que establece los requisitos de un producto Software, pero de acuerdo con las necesidades del cliente. Fundamentado en la tabla 14.

Tabla 14. Competencias preliminares - Rol, Ingeniero de requisitos

Acciones desempeñadas por el Ingeniero de Requisitos	Frecuencia Observada	FO %
Definir los requerimientos del producto Software a construir.	15	50%
Establecer los requerimientos del producto software a partir de las necesidades del cliente.	10	33%
Recolección de información y análisis del mundo del problema.	3	10%
Determinar acciones viables para el sistema	2	7%
Total	30	100%

En la Tabla 15, el 50% de la población afirma que las acciones del Analista de Software en un proceso de construcción de un producto Software son las de análisis y planificación de todos los elementos que intervienen en el proceso de desarrollo a partir de los requerimientos del producto software.

Tabla 15. Competencias preliminares - Rol, Analista de software

Acciones desempeñadas por el Analista de Software	Frecuencia Observada	FO %
Analizar y planificar los elementos que intervienen en el proceso de desarrollo a partir de los requerimientos del producto software.	15	50%
Implementar la arquitectura del software.	2	7%
Otros	13	43%
Total	30	100%

Con respecto a la Tabla 16, se determina que el 30% de la población especifica que la realización de los diseños de la arquitectura de software a partir de los resultados entregados por el analista de Software corresponde a las acciones desempeñadas

por el Diseñador de Software, mientras que el 27% tiene una visión general, en el cual atribuyen actividades de diseño.

Tabla 16. Competencias preliminares - Rol, Diseñador de software

Acciones desempeñadas por el Diseñador de Software	Frecuencia Observada	FO %
Realizar los diseños de la arquitectura del software a partir de los resultados del analista	9	30%
Realizar diseños para el proyecto software	8	27%
Realizar el modelado del software a través de Herramientas como UML	5	17%
Otros	8	26%
Total	30	100%

En cuanto al rol de desarrollador, el 57% de la población tiene el concepto de desarrollador como aquel encargado de la construcción del código fuente del producto software, mientras que el 40% tiene una definición más específica en la cual establecen que el encargado de desarrollar el software, lo hace de forma modular y a partir de los diseños que le son entregados. Referido de la Tabla 17.

Tabla 17. Competencias preliminares - Rol, Desarrollador

Acciones desempeñadas por el Desarrollador	Frecuencia Observada	FO %
Encargado de construir el código fuente del producto software	17	57%
Encargado de desarrollar o construir módulos de software, a partir de los diseños que le son entregados.	12	40%
Otros	1	3%
Total	30	100%

El 43% de la población describe al jefe de calidad de un proyecto de construcción de software como a la persona encargada de gestionar la calidad del software durante todo el proceso de desarrollo, otro 43% clasifica las actividades del jefe de calidad en supervisión o verificación de la calidad del producto software, como se puede observar en la Tabla 18.

Tabla 18. Competencias preliminares - Rol, Jefe de calidad

Acciones desempeñadas por el Jefe de Calidad	Frecuencia Observada	FO %
Gestionar la calidad del producto software durante todo el proceso de desarrollo	13	43%
Verificar la calidad del software	7	23%
Supervisar la calidad del software	6	20%
Otros	4	13%
Total	30	100%

En la Tabla 19, el 60% de la población afirma que el jefe de pruebas ejecuta pruebas de software con el fin de buscar posibles errores, mientras que el 23% lo enmarca en la gestión de pruebas de software y la evaluación de resultados. Finalmente, el 17% de los estudiantes tiene una percepción más clara al afirmar que las acciones principales del jefe de pruebas son las de diseñar y ejecutar pruebas de software, con el fin de asegurar que se cumplan todas las condiciones de calidad y funcionalidad.

Tabla 19. Competencias preliminares - Rol, jefe de pruebas

Acciones desempeñadas por el Jefe de Pruebas	Frecuencia Observada	FO %
Ejecutar pruebas de software para buscar errores en el software.	18	60%
Gestión de pruebas de software y evaluación de resultados	7	23%
Diseñar y ejecutar pruebas de software con el propósito de que todos los parámetros de calidad y funcionalidad se cumplan.	5	17%
Total	30	100%

El rol de líder de equipo en un proyecto de construcción de software es clasificado, en mayor medida, como aquel encargado de hacer posible la gestión de las actividades, recursos, equipo y del proceso general de desarrollo con un 83% del total de la población. Observado en la Tabla 20.

Tabla 20. Competencias preliminares - Rol, Líder de equipo

Acciones desempeñadas por el líder de equipo	Frecuencia Observada	FO %
Gestionar las actividades y recursos del equipo y del proceso de desarrollo	25	83%
Otros	5	17%
Total	30	100%

En relación con la Tabla 21, el 40% de la población clasifica al rol de líder de soporte como aquel encargado de gestionar el soporte técnico durante todo el proceso de desarrollo con el fin de prevenir y corregir problemas que entorpezcan el proceso de desarrollo. El 23% de la población también afirma que la principal actividad de un líder de soporte es la de planificar y ejecutar el soporte del producto software posterior a la entrega del producto al cliente.

Tabla 21. Competencias preliminares - Rol, Líder de soporte

4.9. Acciones desempeñadas por el líder de soporte	Frecuencia Observada	FO %
Gestionar el soporte técnico, tecnológico, logístico necesario de manera permanente para prevenir y corregir inconvenientes con el proceso de desarrollo.	12	40%
Planificar y ejecutar actividades de soporte al software, después de ser entregado al cliente.	7	23%
Otros	11	37%
Total	30	100%

Artefacto

Para más de la mitad de los estudiantes y participantes, específicamente el 70% de ellos dicen que un artefacto en una etapa inicial corresponde o tiene que ver con los documentos de requerimientos del proyecto, un 13% expresa que tiene que ver con la documentación del proyecto. Para el 10% un artefacto tiene que ver con el prototipo del diseño, fichas de usuarios y un entregable de cada objetivo, mientras que el 7% restante, desconoce el significado de artefacto. Observado de la Tabla 22.

Tabla 22. Competencias preliminares - Artefacto, etapa inicial

Artefactos	Frecuencia Observada	Frecuencia Observada (%)
Documentos de requerimientos del proyecto	21	70%
Documentación general del proyecto	4	13%
Prototipo de diseño, ficha de usuario y/o entregable	3	10%
Desconoce	2	7%
Total	30	100%

Según la Tabla 23, en la etapa de ejecución, la mitad de los estudiantes están de acuerdo con que un artefacto en este momento tiene que ver con la documentación como modelado, requisitos y desarrollo, para el 13% corresponde a unos prototipos funcionales mientras que para un 10% tiene que ver con diseño e implementación del proyecto, otro 10% están involucrados los manuales de usuarios. Un 7% manifiesta que un artefacto en este momento tiene que ver con el trabajo en equipo para lograr el objetivo y finalmente el 10% desconoce este tema.

Tabla 23. Competencias preliminares - Artefacto, etapa ejecución

Artefactos	Frecuencia Observada	Frecuencia Observada (%)
Documentación de modelado, requisitos y desarrollo.	15	50%
Diseño e implementación	3	10%
Manual de usuarios	3	10%
Prototipos funcionales	4	13%
Trabajo en equipo para lograr el objetivo	2	7%
Desconoce	3	10%
Total	30	100%

A partir de la Tabla 24, se determina que, en el momento de cierre, el 70% de los estudiantes o encuestados argumentan que un artefacto tiene que ver o implica los manuales de usuario, software o mantenimiento de este; el 17% expresan que es una etapa de pruebas y entrega al usuario, al igual que de mantenimiento, mientras que un 7% dice que es la etapa donde se lleva a cabo el desarrollo y prototipos del proyecto y finalmente otro 7% de los encuestados lo desconocen.

Tabla 24. Competencias preliminares - Artefacto, etapa cierre

Artefactos	Frecuencia Observada	Frecuencia Observada (%)
Manuales (Usuario, Software, Mantenimiento del software)	21	70%
Etapa de Pruebas, entrega al usuario y mantenimiento	5	17%
Etapa de desarrollo y prototipos	2	7%
Desconoce	2	7%
Total	30	100%

Lineamiento

En relación con los lineamientos y en su etapa de inicio, en la Tabla 25, el 67% de los estudiantes consideran que un lineamiento, corresponde o tiene que ver con requerimientos, el análisis, la metodología (ISO, SCRUM, PSP, UML) o herramientas a usar, al igual que relacionado con el cronograma a desarrollar. 10% opina que el lineamiento para este momento se relaciona con la recopilación de todo tipo de información que posibilite un valor agregado que permita el análisis de la misma respecto a problemática que se intenta discutir, en menor porcentaje 2%, aseguran que se trata de un modelo del sistema o análisis de los riesgos, mientras que un 17% desconoce de lo que se trata un lineamiento para este momento.

Tabla 25. Competencias preliminares - Lineamiento, etapa inicial

Lineamiento	Frecuencia Observada	Frecuencia Observada (%)
Tiene que ver con todo el proyecto (Requerimientos, análisis, metodología, herramientas, cronograma)	20	67%
Recopilación de información que posibilite un valor agregado para el análisis de la problemática.	3	10%
Modelado del sistema y análisis de riesgos	2	7%
Desconoce	5	17%
Total	30	100%

Para la etapa de ejecución y en relación con los lineamientos, de la Tabla 26, el 40% de los encuestados dan a conocer que para ellos el lineamiento tiene que ver con el análisis de la información, un 17% sustenta que se relaciona con las herramientas para mejorar la eficiencia en el desarrollo del software y otro 17% que corresponde a temas como el modelado, la metodología o el diseño. También para otros estudiantes, en un 10%, se relaciona con las pruebas y/o fallas, y en igual

porcentaje (10%) tiene que ver con la recolección de datos para el proyecto; simultáneamente en igual porcentaje de 3% una parte dice que un lineamiento en su etapa de ejecución se relaciona con los documentos de las actividades conexas al progreso del proyecto, mientras que el otro 3% no sabe del tema.

Tabla 26. Competencias preliminares - Lineamiento, etapa ejecución

Lineamiento	Frecuencia Observada	Frecuencia Observada (%)
Análisis de información	12	40%
Herramientas para mejorar la eficiencia en el desarrollo de software	5	17%
Modelado, metodologías y diseño	5	17%
Pruebas y fallas	3	10%
Recolección de datos	3	10%
Documento de actividades realizadas y progreso del proyecto	1	3%
Desconoce	1	3%
Total	30	100%

Finalmente, para la etapa de cierre, en relación con los lineamientos, en la Tabla 27 se observa que, la mitad de los encuestados dicen que el lineamiento trae implicado el resultado de las pruebas finales, la retroalimentación organizacional y del total del proyecto, el 23% piensa que está relacionado con las herramientas para el testeado de software. Un 10% de ellos sustenta que los lineamientos en la etapa de cierre corresponden o implican un seguimiento al proyecto, cuando se llega al límite de la entrega, se cumple con un nivel de satisfacción del usuario, se ve el verdadero coste del proyecto y las reglas de este. Y para un 17% de los encuestados es desconocido este tema.

Tabla 27. Competencias preliminares - Lineamiento, etapa cierre

Lineamiento	Frecuencia Observada	Frecuencia Observada (%)
Resultado de pruebas finales, retroalimentación organizacional y del total del proyecto	15	50%
Herramientas para el Testeo del Software	7	23%
Seguimiento, límite de entrega, costo producto y reglas.	3	10%
Desconoce	5	17%
Total	30	100%

Herramienta

A partir de los distintos tipos de herramientas que permiten soportar el proceso de construcción de software basado en equipos, el 26% de la población afirma conocer sobre herramientas de gestión de proyectos software, por otra parte, los grupos correspondientes al 34% de la población, aseguran tener un conocimiento en herramientas para las actividades de diseño, codificación y testing, según lo observado en la Tabla 28.

Tabla 28. Competencias preliminares - Herramientas proceso construcción de software

Herramientas que permitan soportar el proceso de construcción de software por equipos.	Frecuencia Observada	FO %
Gestión de proyectos	8	26%
Herramientas de Diseño y Codificación.	5	17%
Herramientas de Testing.	5	17%

Herramientas de Diseño, Codificación y Testing.	4	13%
Herramientas de Codificación	3	10%
Otros	5	17%
TOTAL	30	100%

La población nombra diferentes tipos de herramientas en las distintas etapas. Entre los cuales se destacan herramientas para la gestión de proyectos de software como JIRA, herramientas para el diseño y codificación como compiladores, herramientas CASE, programas para diagramación en UML entre otros.

Taller

Dentro del Marco del Saber Hacer, donde los participantes de la comunidad virtual de práctica deben transformar y demostrar los conocimientos del Saber o conocimientos teóricos que han adquirido; se realizan una serie de talleres prácticos que determinan y fortalecen las competencias de los estudiantes en cuanto a construcción de software basada en equipos.

Descripción y desarrollo del taller

Con el fin de determinar los niveles de desempeño y las competencias en construcción de software basada en equipos que poseen los estudiantes de ingeniería de sistemas en la Universidad de Nariño, se formuló un taller, en el cual se planteó un problema y una serie de requerimientos mínimos a ser entregados en equipos de trabajo, con la premisa de que cada equipo debía decidir que artefactos entregar al finalizar el proceso, el cual se desarrolló en cuatro horas distribuidas en dos sesiones.

El contexto del problema consiste en la necesidad de gestionar la información sobre la división política colombiana (DIVIPOLA), en la cual se necesitan datos básicos sobre los departamentos y municipios de Colombia.

Se especificó un requerimiento funcional principal, describiendo el requerimiento, y el actor que interviene con dicho requerimiento de la siguiente manera:

- Construir un producto software funcional que debe incluir la necesidad “Como Auxiliar Administrativo quiero registrar los habitantes para un municipio de un año específico”.

Adicionalmente, se especificaron tres requisitos de calidad, los cuales consisten en que el producto software a entregar sea usable, sus datos perduren en el tiempo y sea una aplicación orientada a la web.

Al finalizar cada sesión, se recolectaron datos mediante una guía de observación para cada equipo de trabajo, en la cual se obtuvo como resultado: el número de integrantes por equipo, una justificación al elegir el número de integrantes, los roles desempeñados dentro del equipo y los factores para elegir dichos roles, las actividades realizadas por cada sesión, las herramientas y tecnologías utilizadas para desarrollar el taller y por último el nivel en el que cada equipo de trabajo cumple el objetivo principal de la actividad.

RESULTADOS OBTENIDOS

Los siguientes resultados expuestos en la Tabla 29, muestran las actividades entregadas por parte de los equipos de trabajo en el taller. Y reflejadas de manera resumida en la Figura 1.

Tabla 29. Actividades entregadas en el taller N° 1

EQUIPO	Actividad							
	Especificación de roles	Ingeniería de Requisitos	Definición de casos de uso	Especificación de metodología de desarrollo/trabajo a utilizar	Diseño de MockUps	Diseño de Base de datos	Codificación	Documentación en código
1	NO	NO	NO	NO	NO	SI	SI	NO
2	NO	SI	SI	NO	NO	SI	SI	NO
3	NO	NO	NO	NO	NO	NO	SI	NO
4	SI	SI	SI	SI	NO	SI	SI	NO
5	NO	NO	NO	NO	NO	SI	SI	NO
6	NO	SI	NO	NO	SI	SI	SI	NO
7	SI	SI	SI	NO	SI	NO	SI	NO
8	NO	NO	NO	NO	SI	SI	SI	NO
9	NO	NO	NO	NO	NO	SI	SI	NO
TOTAL	2	4	3	1	3	7	9	0
PORCENTAJE	22%	44%	33%	11%	33%	78%	100%	0%

Figura 1. Actividades entregadas Taller N° 1



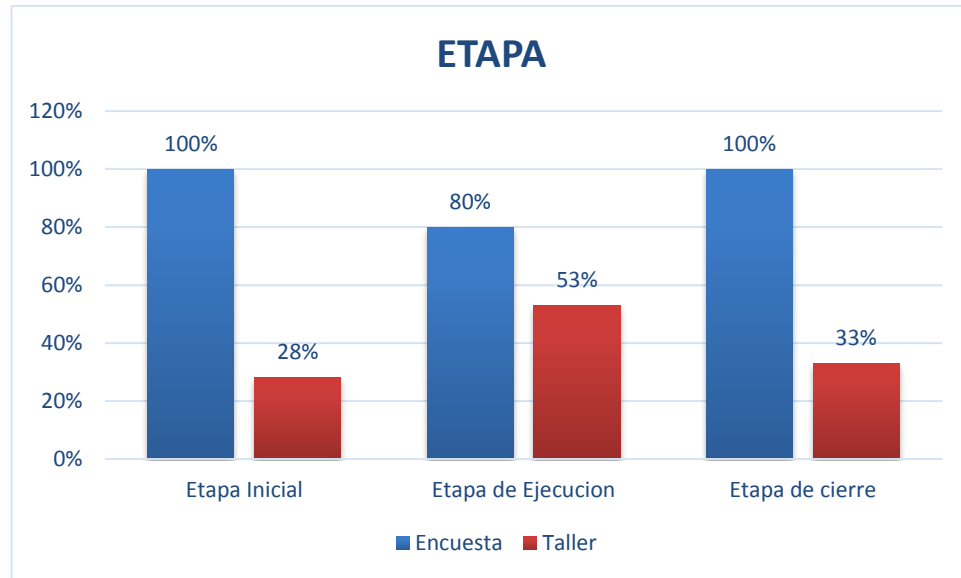
CONTRASTE DE TÉCNICAS

Después de ejecutar el taller con la población objeto de estudio y de analizar los datos obtenidos en la guía de observación, se procedió a contrastar dichos resultados con la información resultante de la encuesta. Al finalizar el contraste de los resultados, se obtuvo los siguientes resultados.

- Etapa: a partir de los resultados de la encuesta, se evidenció que el mayor porcentaje de la población conoce e identifica cada una de las etapas de desarrollo de software con sus respectivas actividades; sin embargo, en los resultados del taller se puede observar que menos de la mitad de los equipos de trabajo, definió etapas y actividades, por ejemplo como se puede apreciar en la figura 2, el 100% de la población afirmó conocer el concepto de etapa inicial de desarrollo y sus actividades, sin embargo, en promedio el 28% de las actividades realizadas por los equipos pertenecen a una etapa inicial en

el taller. En la etapa de cierre los estudiantes tienen un concepto preciso de las actividades que comúnmente se realizan para agregar valor un producto software, además en el taller se evidenció que todos los equipos de trabajo en la etapa de cierre realizaron la entrega de un producto software, teniendo en cuenta que las actividades realizadas en esta etapa corresponden al 33% de las actividades identificadas en la encuesta.

Figura 2. Contraste resultados conocimientos previos variable Etapa



- Rol: con relación a la figura 3. La mayoría de los estudiantes afirma conocer y diferenciar los tipos de roles asociados a un proyecto de desarrollo de software, además de identificar las funciones asociadas a cada rol, sin embargo, en el taller y como se observa en la figura 4, únicamente el 22% de los equipos de trabajo definieron roles de trabajo para desarrollar el taller.

Figura 3. Resultados conocimientos previos variable Rol

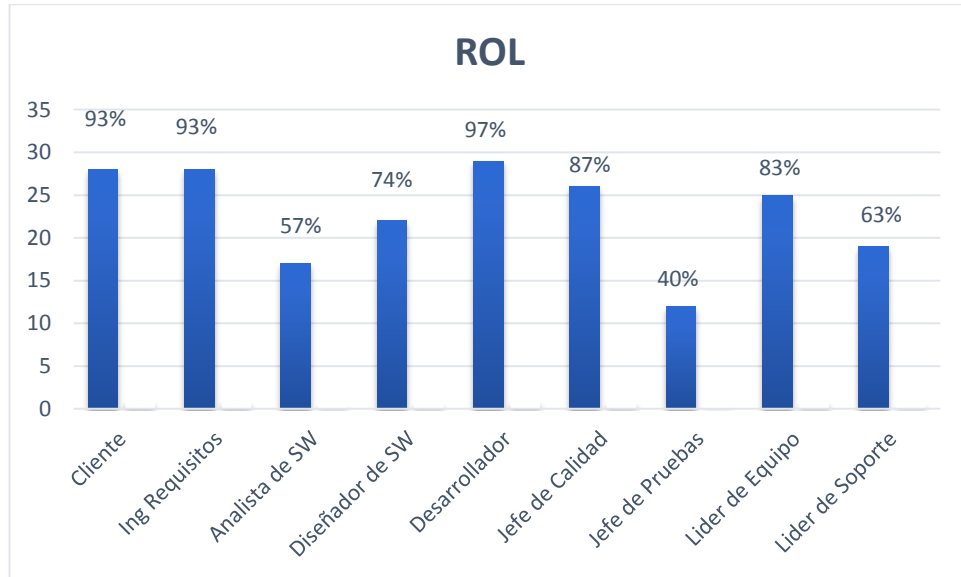
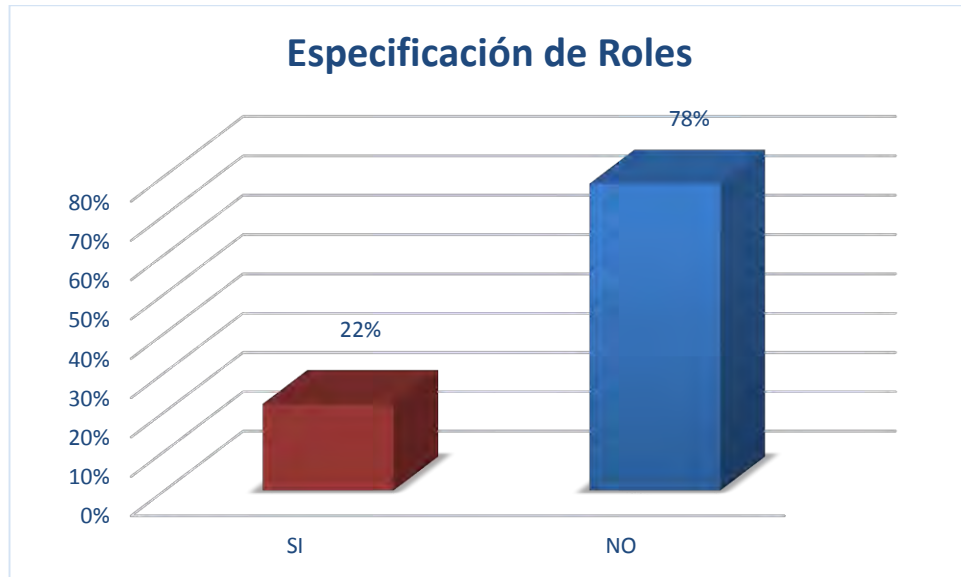


Figura 4. Contraste resultados conocimientos previos variable Rol

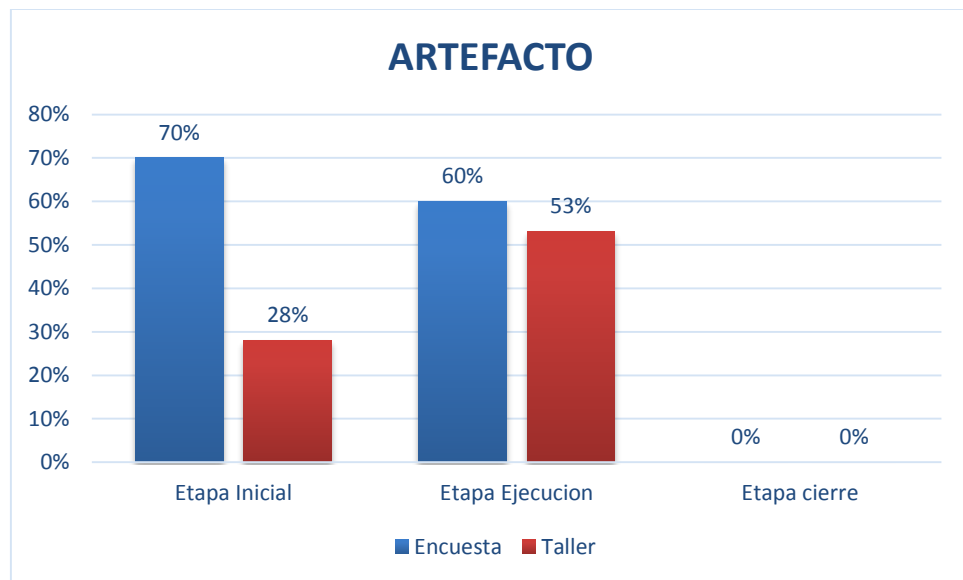


- Artefacto: en cuanto a la encuesta, se observa que la mayor parte de la población conoce e identifica el concepto de artefacto dentro de un proyecto de desarrollo de software en las etapas iniciales y de ejecución con un 70% y 60% respectivamente, como se observa en la figura 5. Pero en la etapa de

cierre los estudiantes no tienen clara la definición de artefacto y en cuanto a que tipo de artefactos se deben entregar en esta etapa de desarrollo.

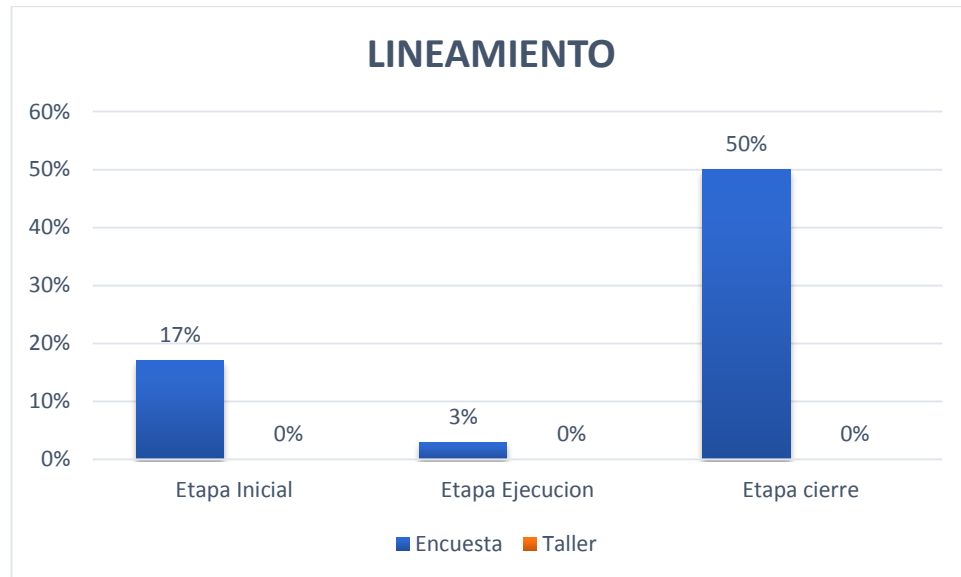
Adicionalmente, se observa que en la etapa inicial del taller, solo el 28% de los equipos de trabajo entregaron los artefactos relacionados con esta etapa. En la etapa de ejecución el 53% realizó los artefactos pertinentes, y por último ningún equipo realizó la entrega de un artefacto que respalde la etapa de cierre.

Figura 5. Contraste resultados conocimientos previos variable Artefacto



- Lineamiento: se puede observar en la figura 6, la mayoría de los estudiantes tienen un concepto erróneo al de un lineamiento en construcción de software, siendo la etapa de ejecución la que presenta el porcentaje más bajo con un 3%, por otra parte, la mitad de la población tiene un concepto cercano en cuanto a lineamiento se refiere en la etapa de cierre. En cuanto a los resultados del taller, se observa que ninguno de los equipos de trabajo se basó en un lineamiento para la construcción del producto software propuesto.

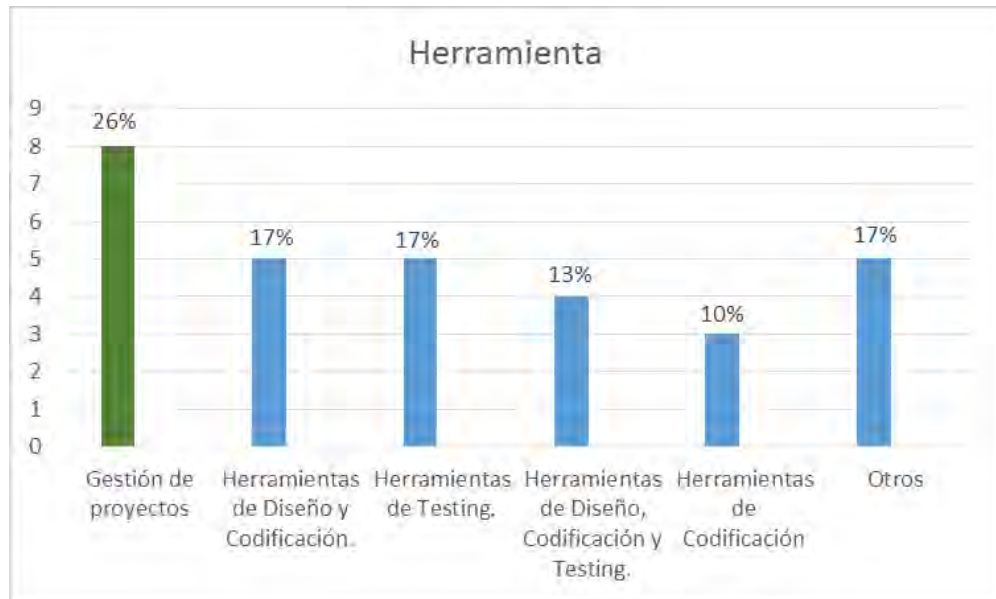
Figura 6. Contraste resultados conocimientos previos variable Lineamiento



- Herramienta: la población objeto de estudio conoce diferentes herramientas para utilizar en cada fase y actividad del proceso de desarrollo de software. En el taller los equipos utilizaron principalmente herramientas para el diseño y codificación del producto software. Centrándose, en la etapa de ejecución, y solo una minoría utilizó herramientas que corresponden al soporte de otras fases del proceso de desarrollo.

Además, como se observa en la figura 7, el 26% de la población afirma conocer sobre herramientas de gestión de proyectos de software, como por ejemplo JIRA. Sin embargo, ninguno de los equipos utilizó este tipo de herramientas en el desarrollo del taller. En cuanto a los tipos de herramientas utilizados en el taller, todos los grupos utilizaron herramientas de diseño y codificación.

Figura 7. Resultado de conocimientos previos variable Herramienta



3.2. CONFORMAR UNA COMUNIDAD DE PRÁCTICA VIRTUAL PARA EL DESARROLLO DE COMPETENCIAS EN CONSTRUCCIÓN DE SOFTWARE BASADA EN EQUIPOS

3.2.1. Descripción general de la comunidad virtual de práctica. En esta sección se describe el proceso de conformación de la comunidad virtual de práctica, sus dimensiones (identidad, comunidad, práctica) y las actividades e interacciones realizadas con el fin desarrollar las competencias en construcción de software basada en equipos de acuerdo con el análisis de resultados del primer objetivo. Adicionalmente, se realizó una encuesta y un taller en el que se refleja el nivel de competencias actuales de la población, y en el que posteriormente se podrá evidenciar el nivel de aporte hecho por la comunidad de práctica a través de un contraste.

3.2.2. Comunidad de práctica propuesta.

Identidad

Según la Real Academia española [31], se puede definir la identidad como, el conjunto de rasgos propios que posee una persona o individuo dentro de una colectividad que los caracterizan frente a los demás, es así cómo los integrantes de la Comunidad Virtual de práctica adquieren un conjunto de rasgos, que en este caso se definen con una serie de competencias dirigidas específicamente a la construcción de software basada en equipos y apoyándose en las metodologías que aportan un enfoque ágil a esta actividad, dejando como principal propósito que los participantes sean dueños de su propia evolución y aporten a los demás conocimiento de manera colectiva.

Práctica

Dentro de la comunidad virtual de práctica, ya estando completamente conformada, se plantearon una serie de actividades que permitían a los participantes apoderarse del conocimiento de una manera interactiva y que a su vez este fortaleciera sus competencias para desarrollar software como parte de un equipo y enfocados principalmente en metodologías ágiles.

Hicieron parte de estas actividades, foros que, planteando diferentes escenarios y temáticas referidas con enfoques ágiles, se logró obtener del estudiante un aspecto crítico y colaborativo en cuanto a su participación y a la de sus compañeros, aportando procesos constructivos desde criterios diferentes. Además, se suministraba una serie de documentos, que aportaban también al empoderamiento de conocimientos de cada participante, los casos de estudio que se plantearon para los talleres prácticos, fueron tomados de problemáticas reales, que le permitieran a los participantes desempeñarse en construcción de software, como un equipo de trabajo real; se propusieron unas dinámicas que permitieron centrar la atención de los participantes en la comunidad virtual de práctica, a modo de motivación, es así como después algunas de las actividades, se sorteaban entradas para eventos académicos como el tercer Congreso Andino de Computación e Informática (CACIED) realizado en San Juan de Pasto en Noviembre de 2017 y un evento de tipo social perteneciente a la programación de la semana de ingeniería en la Universidad de Nariño. También, se generó un material de apoyo visual, como son un conjunto de videos, que ayudaban a los participantes a conocer e identificar los roles que se desempeñan en un equipo de desarrollo basado en SCRUM; el apoyo más grande para la comunicación entre toda la Comunidad virtual de práctica, fueron las redes sociales, como Facebook, que generaban un acercamiento tanto a

los demás participantes, como a la comunidad, debido a lo cotidiano de esta red social. En cuanto a los talleres realizados, se apoyó con la técnica Kanban, para que los estudiantes fortalezcan las competencias colaborativas dentro de un equipo de desarrollo y así puedan entender la importancia de la planeación. Además, se apoyó en la herramienta Kunagi, la cual permite soportar el proceso de Scrum, facilitando la visualización del flujo de trabajo y afianzando las competencias que tienen que ver con la planeación de un determinado proyecto dentro de un equipo de desarrollo, dado que tiene incluido el tablero Kanban. Un proceso importante que se realizó, fueron las encuestas de conocimiento, con las cuales se pretende establecer si los participantes se apoderaron realmente de la comunidad virtual de práctica y si esta les aportó en el enriquecimiento y fortalecimiento de las competencias de desarrollo de software basado en equipos y enfocados en metodologías ágiles.

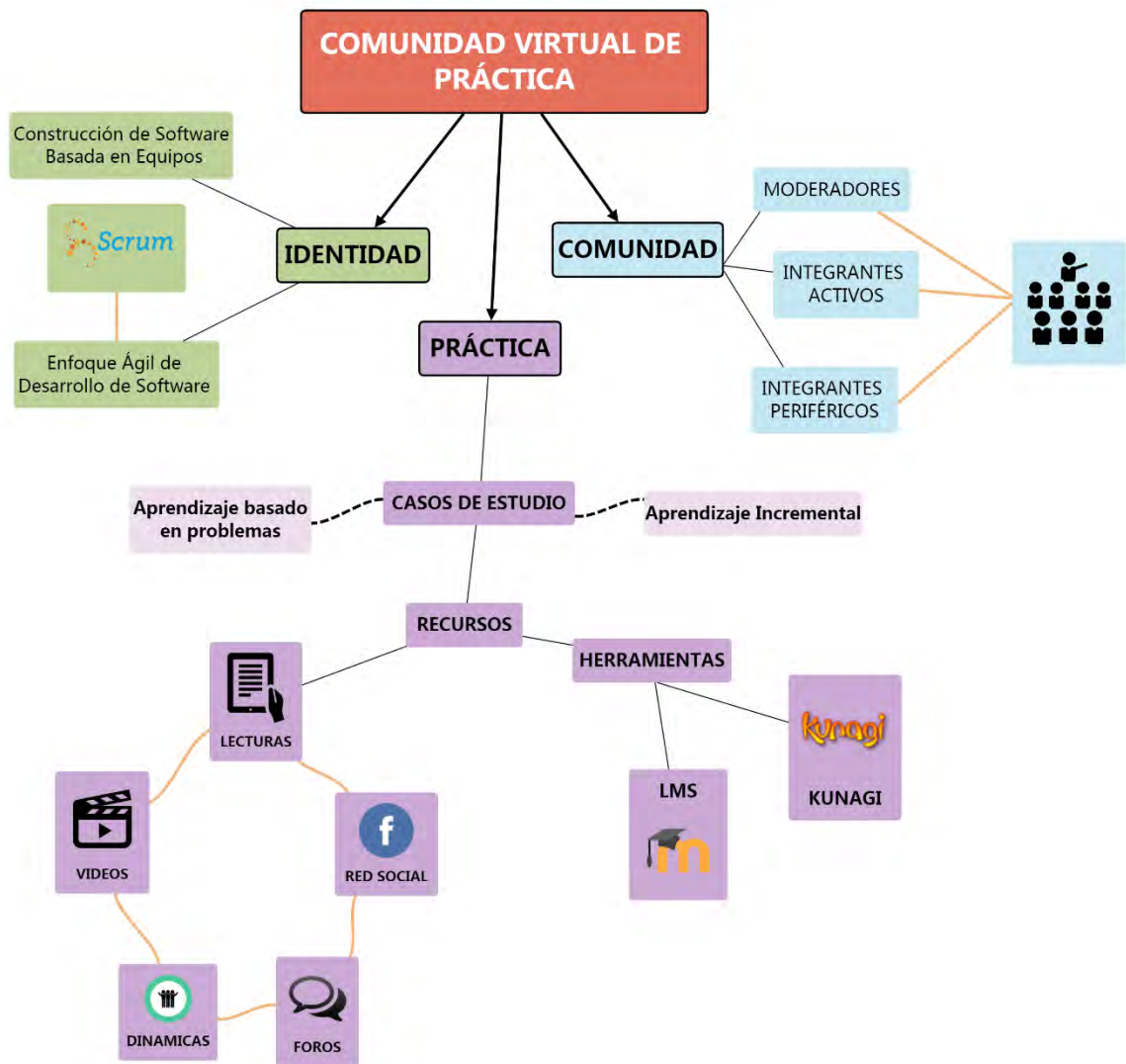
Comunidad

La comunidad virtual de práctica, está conformada por una población de individuos (Ver Figura 8) que conlleva a generar un proceso selectivo de estos, según la participación que tienen dentro de la comunidad virtual de práctica; es así como se pueden encontrar a los participantes activos, que en este caso, son todos los estudiantes de cursaban Noveno semestre de Ingeniería de sistemas en el periodo A de 2017, de la Universidad de Nariño, quienes fueron los que participaban directamente del proceso. También, los participantes periféricos, que son aquellos que están pendientes de la evolución de la comunidad, de las actividades propuestas, pero simplemente observan, para este caso es un número indeterminado, debido a que se trabajaba con ayuda de redes sociales, y por último se identifican a los moderadores, que son aquellos que plantean el material intelectual, procesos y actividades, para que los demás participantes lo utilicen.

Arquitectura

La comunidad virtual de práctica propuesta se compone de tres dimensiones, la identidad, la comunidad y la práctica, así mismo cada dimensión está compuesta de elementos como se muestra en la figura 8.

Figura 8. Arquitectura Comunidad virtual de práctica



A continuación, se describe cada uno de los elementos pertenecientes a cada dimensión.

Identidad:

- Construcción de software basada en equipos: es el tema de interés general y sobre el cual se desarrollan las competencias entre los participantes de la comunidad virtual de práctica.

- **Enfoque ágil de desarrollo de software:** en un tema específico dentro del proceso de aprendizaje en la comunidad virtual de práctica, el cual permite guiar las competencias a través de sus conceptos y aplicaciones. En este caso se utiliza un enfoque ágil empleando la metodología Scrum.

Comunidad:

- **Moderadores:** son participantes de la comunidad de práctica que se encargan de guiar el proceso de manera activa, mediante la formulación de actividades y recursos para la consecución de los objetivos de la comunidad.
- **Integrantes:** son los participantes de la comunidad virtual de practica los cuales buscan fortalecer sus competencias en construcción de software desde un enfoque ágil. Existen dos categorías de integrantes.
 - **Integrantes activos:** integrantes de la comunidad virtual de práctica que participan en todas las actividades propuestas y finalizan el proceso de aprendizaje.
 - **Integrantes periféricos:** integrantes que pueden visualizar las temáticas y actividades, pero no participan de manera directa.

Práctica:

- **Casos de estudio:** conjunto de actividades que permiten desarrollar las competencias en construcción de software en todos los participantes de la comunidad virtual de práctica a través de recursos y herramientas.
- **Recursos:** son todos los materiales que permiten soportar las actividades realizadas en la comunidad virtual de práctica, como lecturas, videos, dinámicas, foros y el uso de una red social, en la sección de anexos se encuentran evidencias en cuanto a los recursos utilizados.
- **Herramientas:** son elementos de apoyo para el funcionamiento de la comunidad de práctica como es el caso de un LMS o una plataforma de gestión de recursos de aprendizaje, también permiten apoyar las actividades y potencializar el desempeño de los participantes en competencias específicas, como por ejemplo la herramienta Kunagi, la cual permite realizar la gestión en proyectos de construcción de software basado en equipos involucrando la metodología SCRUM.

3.2.3. Desarrollo. Una vez finalizada la participación de la población en la comunidad virtual de práctica para fortalecer las competencias en construcción de software en las variables establecidas, se procedió a realizar una encuesta y un taller para determinar el nivel de competencias al finalizar las actividades dentro de la comunidad.

Encuesta

Se observa en la Tabla 30, que más del 80% de la población considera poseer una afinidad BUENA y MUY BUENA en cuanto al desarrollo de software, mientras que una minoría afirma tener una afinidad regular.

Tabla 30. Conocimientos posteriores - afinidad con el desarrollo de software

Categoría	FO- Frecuencia Observada	Frecuencia Observada (%)
MUY BUENA	9	30,00
BUENA	16	53,33
REGULAR	5	16,67
TOTAL	30	100,00

La mayoría de la población, según la Tabla 31, correspondiente al 90% afirma conocer al menos una metodología de desarrollo de software con enfoques ágiles, entre las que se destaca SCRUM, XP y RUP.

Tabla 31. Conocimientos posteriores - Conocimiento de metodologías

Categoría	FO- Frecuencia Observada	Frecuencia Observada (%)
SI	27	90
NO	3	10
TOTAL	30	100

Etapa

En cuanto a las actividades correspondientes a la etapa inicial de un proceso de construcción de software, la mayoría de los estudiantes, correspondiente al 63,33% relaciona dichas actividades con la planificación de recursos, organización y asignación de roles y actividades a desempeñar durante el sprint. El porcentaje restante está distribuido entre aquellos estudiantes que reconocen la gestión de requisitos y actividades de análisis y diseño en la etapa inicial, según lo que se observa en la Tabla 32.

Tabla 32. Conocimientos posteriores - Variable Etapa, Etapa inicial

Categoría	FO-Frecuencia Observada	Frecuencia Observada (%)
Planificación, Asignación de roles y actividades	19	63,33
Gestión de requisitos	5	16,67
Actividades de análisis y diseño	6	20,00
TOTAL	30	100,00

La población objeto de estudio identifica, según la Tabla 33 en un 80% que, dentro de las actividades inmersas en la etapa de ejecución, se centran en la construcción del producto software, es decir en aquellas actividades que permiten agregar valor al producto, entre ellas se destacan los reportes de análisis, diseños, codificación y testing. La población restante se enfoca específicamente en la codificación como actividad principal en la etapa de ejecución

Tabla 33. Conocimientos posteriores - Variable Etapa, Etapa ejecución

Categoría	FO-Frecuencia Observada	Frecuencia Observada (%)
Construcción del Producto Software	24	80,00
Codificación	6	20,00
TOTAL	30	100,00

En cuanto a las actividades pertenecientes a la etapa de cierre y observado de la Tabla 34, más de la mitad de los estudiantes reconoce como actividad principal la entrega de un producto software funcional bien sea parcial o total al finalizar una iteración, una pequeña parte de la población afirma que las actividades se destinan a la revisión de reportes y retroalimentación de la iteración, finalmente un 30% de la población afirma que en esta etapa se deben realizar las pruebas al producto software realizado.

Tabla 34. Conocimientos posteriores - Variable Etapa, Etapa cierre

Categoría	FO-Frecuencia Observada	Frecuencia Observada (%)
Entrega parcial o total de producto Software funcional	17	56,67
Pruebas/Testing del avance o producto Software	9	30,00
Reportes y retroalimentación de la iteración	4	13,33
TOTAL	30	100,00

Rol

Los roles en un proceso de construcción de software identificados por la población están generalizados en dos categorías, el 60% de la población establece que los roles son aquellos integrantes del SCRUM TEAM (scrum master, product owner y todos los roles derivados del development team), mientras que la población restante vincula los roles, específicamente con el development team por ejemplo desarrolladores, diseñadores de software, testers, entre otros. Observado en la Tabla 35.

Tabla 35. Conocimientos posteriores - Variable Rol

Categoría	FO- Frecuencia Observada	Frecuencia Observada (%)
SCRUM TEAM	18	60
ROLES ASOCIADOS AL DEVELOPMENT TEAM	12	40
TOTAL	30	100

Artefacto

En cuanto a los tipos de artefactos generados en la etapa inicial, se encuentra en la Tabla 36, que la mayoría de la población correspondiente al 83,33% identifica al documento de requisitos como el artefacto primordial en esta etapa. La población restante identifica documentos de análisis y diseños y reportes de encuestas realizadas a usuarios finales.

Tabla 36. Conocimientos posteriores - Variable Artefacto, Etapa inicial

Categoría	FO- Frecuencia Observada	Frecuencia Observada (%)
Documento de requisitos	25	83,33
Documento de análisis y diseño	3	10,00
Reportes de encuestas	2	6,67
TOTAL	30	100,00

Entre los artefactos en la fase de ejecución se destaca según la tabla 37, a los diseños de la arquitectura del software con un 40% del total de la población, posteriormente el 33,33% de la población estableció que la entrega de módulos funcionales de software son el principal artefacto en esta etapa, finalmente el 26,66% de la población identifica que los prototipos y reportes de pruebas son artefactos importantes en la etapa de ejecución.

Tabla 37. Conocimientos posteriores - Variable Artefacto, Etapa ejecución

Categoría	FO-Frecuencia Observada	Frecuencia Observada (%)
Diseños de la arquitectura del Software	12	40,00
Prototipos	4	13,33
Módulos de Software funcionales	10	33,33
Reporte de pruebas	4	13,33
TOTAL	30	100,00

En la etapa o fase de cierre, la población objeto de estudio identifica en un 46,67% que el artefacto más importante es un producto software funcional. Otra parte de la población perteneciente al 33,33% establece que los artefactos son la documentación y manuales de usuario, mientras que la población restante menciona el reporte de resultados de pruebas del producto software como insumo de retroalimentación, como se observa en la Tabla 38.

Tabla 38. Conocimientos posteriores - Variable Artefacto, Etapa cierre

Categoría	FO-Frecuencia Observada	Frecuencia Observada (%)
Producto software Funcional	14	46,67
Documentación y Manuales de usuario	10	33,33
Reporte de pruebas	6	20,00
TOTAL	30	100,00

Lineamiento

En la etapa inicial de un proceso de construcción de software, los estudiantes identifican, con relación a la Tabla 39, tres categorías de lineamientos con porcentajes equilibrados, la categoría con el mayor porcentaje es la de los requerimientos de Software con un 40%, el 60% restante está repartido

equitativamente en las necesidades del cliente y antecedentes en procesos de construcción de software similares dentro del mismo equipo de desarrollo o externo.

Tabla 39. Conocimientos posteriores - Variable Lineamiento, Etapa inicial

Categoría	FO- Frecuencia Observada	Frecuencia Observada (%)
Necesidades del Cliente	9	30,00
Requerimientos de Software	12	40,00
Antecedentes	9	30,00
TOTAL	30	100,00

En cuanto a los lineamientos identificados en la etapa de ejecución por parte de los encuestados, se encuentra en la Tabla 40, que la mayoría de la población establece que las historias de usuario son el principal lineamiento por seguir para la toma de decisiones dentro del proceso, la población restante afirma que los reportes dentro de la etapa de ejecución permiten tomar decisiones eficaces y pertinentes que guíen la dirección del proceso de construcción de software.

Tabla 40. Conocimientos posteriores - Variable Lineamientos, Etapa ejecución

Categoría	FO- Frecuencia Observada	Frecuencia Observada (%)
Historias de Usuario	23	76,67
Reportes	7	23,33
TOTAL	30	100,00

Para la etapa de cierre, la población identifica que los lineamientos son los reportes de pruebas y la retroalimentación entre iteraciones con un porcentaje de 36,67%, 43%,33% respectivamente. La población restante respondió con una categoría alejada a la definición de lineamiento, según lo observado en la Tabla 41.

Tabla 41. Conocimientos posteriores - Variable Lineamientos, Etapa cierre

Categoría	FO- Frecuencia Observada	Frecuencia Observada (%)
Reportes de pruebas	11	36,67
Retroalimentación	13	43,33
Otros	6	20,00
TOTAL	30	100,00

Herramienta

Como se observa en la Tabla 42, entre las herramientas conocidas para soportar procesos de construcción de Software por equipos, la población en su mayoría establece herramientas para la planificación, gestión de recursos y actividades, en la que se evidencie el flujo de trabajo del equipo, en este tipo de herramientas, los estudiantes en mayor medida destacan a kunagi, como la herramienta más completa en las etapas inicial, de ejecución y cierre, otras herramientas nombradas son Jira y Test-Link, herramientas para la gestión de proyectos de Software.

Tabla 42. Conocimientos posteriores - Variable Herramienta

Categoría	FO- Frecuencia Observada	Frecuencia Observada (%)
Herramientas para la gestión de recursos, actividades y toma de decisiones en el proceso de construcción de Software	26	86,67
Herramientas para soportar actividades específicas en la construcción de Software	4	13,33
TOTAL	30	100,00

Taller

Descripción y desarrollo del taller

El taller planteado consistió en un problema y una serie de requerimientos mínimos a ser entregados en equipos de trabajo, los cuales tenían la autonomía de organizarse y de entregar los artefactos que consideren necesarios al finalizar la actividad, la cual se desarrolló en una sesión de dos horas y media. Adicionalmente se exigió utilizar la herramienta kunagi para realizar la planificación de actividades, asignación de roles y visualización del flujo de trabajo en cada sprint.

El contexto del problema consiste en la necesidad de gestionar la información sobre la vinculación de profesionales en artes escénicas, al cual se denominó proyecto VIPAES, en el cual se requieren los datos básicos del profesional a vincular y la profesión a la que pertenece según el área artística respectiva.

Se especificó un requerimiento funcional principal, describiendo el requerimiento y el actor que interviene, de la siguiente forma:

- Construir un producto software funcional que debe incluir la necesidad “Como Auxiliar Administrativo quiero registrar los profesionales de artes escénicas, vinculándolos con su respectiva área y profesión”.

Adicionalmente se especificaron tres requisitos de calidad, los cuales consisten en que el producto software a entregar sea usable, sus datos perduren en el tiempo y sea una aplicación orientada a la web.

Al finalizar la actividad, se recolectaron datos mediante una guía de observación para cada equipo de trabajo, en la cual se obtuvo como resultado: el número de integrantes, los roles desempeñados, actividades realizadas, las herramientas utilizadas en el proceso de desarrollo de software y se estableció el nivel de cumplimiento de cada equipo sobre el objetivo principal de la actividad.

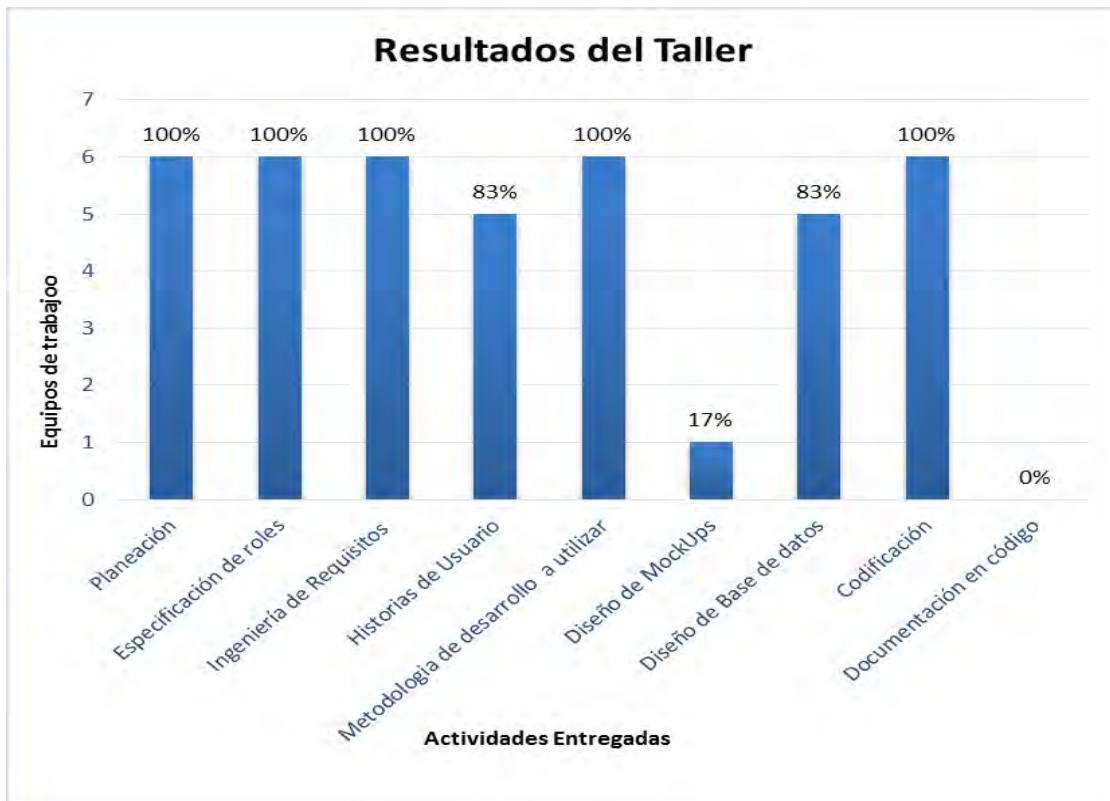
Resultados obtenidos

Los siguientes resultados en la Tabla 43 reflejan las actividades entregadas por parte de los equipos de trabajo en el taller y se resumen en la figura 9.

Tabla 43. Conocimientos posteriores – Actividades entregadas

EQUIPO	ACTIVIDAD								
	Planeación	Especificación de roles	Ingeniería de Requisitos	Definición de Historias de Usuario	Especificación de metodología de desarrollo a utilizar	Diseño de MockUps	Diseño de Base de datos	Codificación	Documentación en código
1	SI	SI	SI	SI	SI	SI	SI	SI	NO
2	SI	SI	SI	NO	SI	NO	NO	SI	NO
3	SI	SI	SI	SI	SI	NO	SI	SI	NO
4	SI	SI	SI	SI	SI	NO	SI	SI	NO
5	SI	SI	SI	SI	SI	NO	SI	SI	NO
6	SI	SI	SI	SI	SI	NO	SI	SI	NO
Total	6	6	6	5	6	1	5	6	0
Porcentaje	100%	100%	100%	83%	100%	17%	83%	100%	0%

Figura 9. Conocimientos posteriores - Actividades entregadas

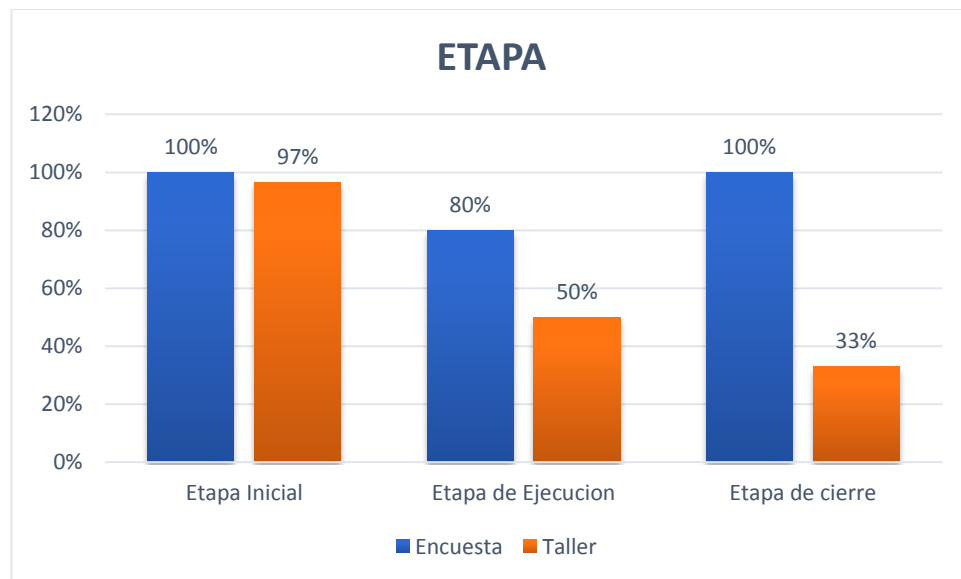


Contraste de técnicas

Después de ejecutar la segunda versión de taller con la población objeto de estudio y de analizar los datos obtenidos en la guía de observación, se procedió a contrastar dichos resultados con la información resultante de la segunda encuesta de conocimientos. Al finalizar el contraste de los resultados, se obtuvo los siguientes resultados:

- **Etapa:** en la figura 10, los resultados de la encuesta reflejan que en general, la población posee un alto nivel de conocimientos y dominio sobre los elementos y actividades vinculados a la variable etapa en las tres etapas o fases definidas. En cuanto a los datos obtenidos en el taller, se encuentra que en promedio los equipos de trabajo obtuvieron un desempeño alto en la etapa inicial, puesto que realizaron actividades centradas en esta etapa, destacando la planificación de la actividad y la asignación de roles dentro del equipo. Por otra parte, se encuentra que, en promedio, los equipos de trabajo realizaron el 50% de las actividades pertinentes a la etapa de ejecución. Finalmente, los equipos de trabajo realizaron actividades en la etapa de cierre en un 33%.

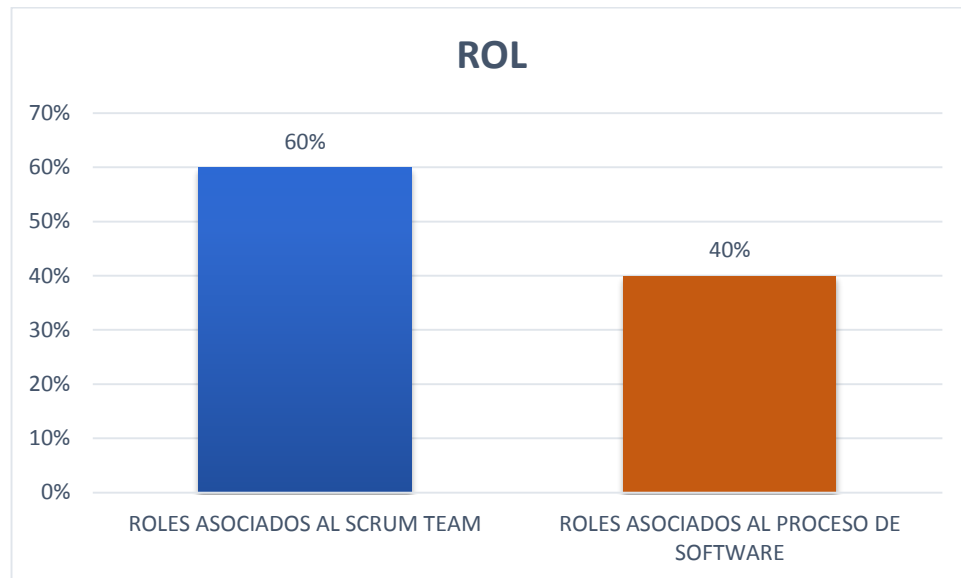
Figura 10. Contraste resultados conocimientos posteriores variable etapa



- **Rol:** en esta variable, se puede identificar que los estudiantes reconocen distintos tipos de roles y sus funciones dentro de un equipo de desarrollo de software, en el cual el 60% de los encuestados menciono roles asociados al

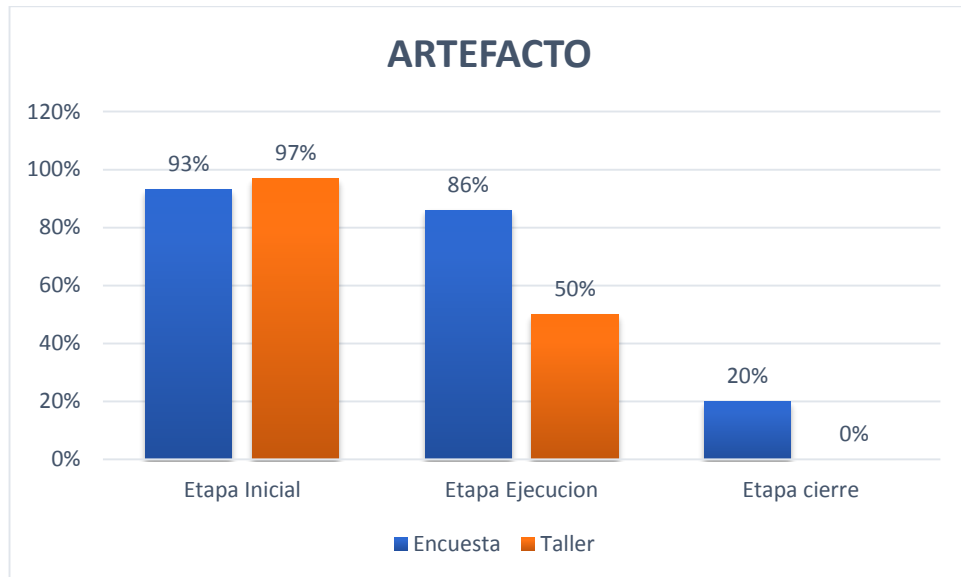
Scrum Team (Scrum Master, Product Owner, Development Team), mientras que el porcentaje restante se enfocó en los roles asociados al proceso de software (analistas, diseñadores, desarrolladores, testers, jefes de soporte, jefes de pruebas, entre otros). En el desarrollo de la actividad práctica, todos los equipos de trabajo definieron roles de software basados en la metodología Scrum. Observado en la figura 11.

Figura 11. Contraste resultados conocimientos posteriores variable Rol



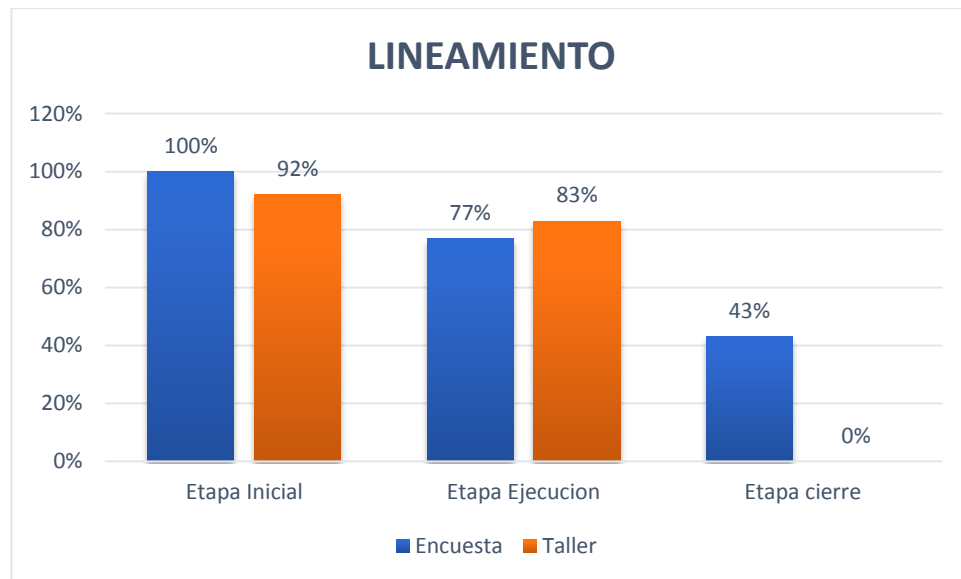
- Artefacto: los resultados de la encuesta y el taller, permiten evidenciar de la figura 12 que, en la etapa inicial, existe una relación proporcional entre el nivel de conocimiento y la aplicación de estos en cuanto a artefactos que agreguen valor al producto o al proceso con un nivel alto. El nivel de conocimientos en la etapa de ejecución también se considera alto, pero en la parte práctica, el nivel de cumplimiento en cuanto a artefactos se refiere es del 50%. Finalmente, la etapa de cierre tiene un porcentaje bajo en cuanto a conocimientos y estos no son aplicados en el taller.

Figura 12. Contraste resultados conocimientos posteriores variable Artefacto



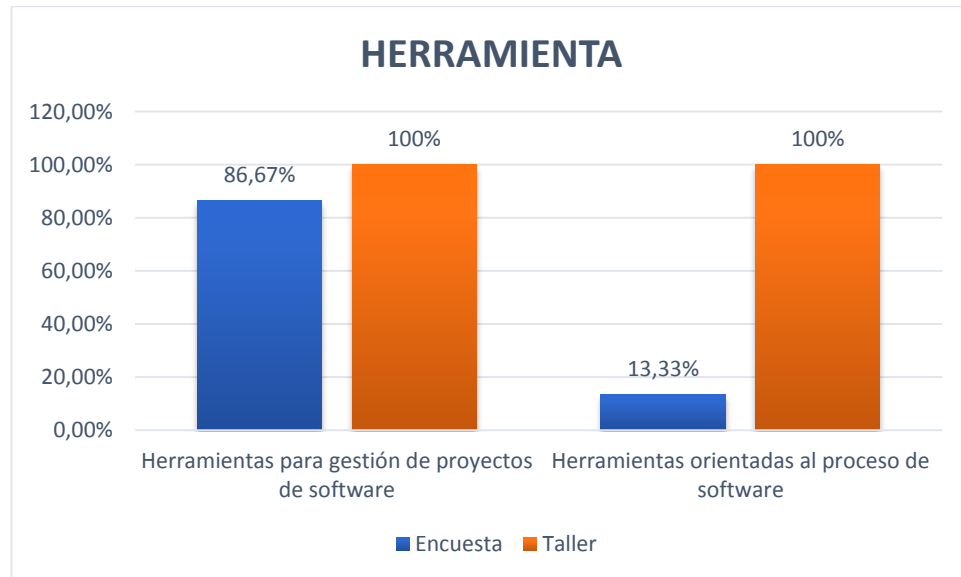
- Lineamiento: los resultados de la encuesta reflejan en la figura 13, que la población en general tiene un concepto claro sobre lineamientos, o elementos que posibiliten tomar decisiones de manera acertada en las etapas inicial y de ejecución, sin embargo, menos de la mitad de los estudiantes tiene un concepto claro sobre esta variable en la etapa de cierre. Además, los resultados del taller son proporcionales a la encuesta, es decir que, con excepción a la etapa de cierre, los equipos de trabajo demostraron un nivel de aplicación alto en cuanto a lineamiento.

Figura 13. Contraste resultados conocimientos posteriores variable Lineamiento



- Herramienta: en esta variable, se establecieron dos categorías de herramientas, de las cuales, en los resultados de la encuesta, el 86,67% de la población menciono herramientas para la gestión de proyectos de software, destacando a KUNAGI como la herramienta principal, por su influencia dentro de la comunidad virtual de práctica, el 13,33% identificó herramientas orientadas al proceso de software como herramientas para análisis, diseño, codificación, pruebas, calidad, entre otras. En cuanto a la aplicación de herramientas en el taller, todos los equipos de trabajo emplearon la herramienta KUNAGI como parte del proceso de aprendizaje de la comunidad de práctica, así como también utilizaron diferentes tipos de herramientas para soportar actividades relacionadas al proceso de software en las diferentes etapas. Observado de la figura 14.

Figura 14. Contraste resultados conocimientos posteriores variable Herramienta



3.3. DETERMINAR EL NIVEL DE APORTE HECHO POR LA COMUNIDAD DE PRÁCTICA VIRTUAL EN EL DESARROLLO DE COMPETENCIAS EN LA CONSTRUCCIÓN DE SOFTWARE BASADA EN EQUIPOS, A LOS MIEMBROS DE LA COMUNIDAD

En este capítulo, se realiza un análisis de la evolución de las competencias en construcción de software basada en equipos a los miembros de la comunidad virtual de práctica, basándose en una comparación y análisis de los resultados obtenidos en la encuesta de conocimientos y el taller práctico realizados antes de iniciar el proceso de la comunidad virtual de práctica y después de finalizar las actividades de aprendizaje dentro de la comunidad. Esto con el fin de determinar el nivel de aporte que significó la comunidad virtual de práctica.

3.3.1. Comparación de competencias

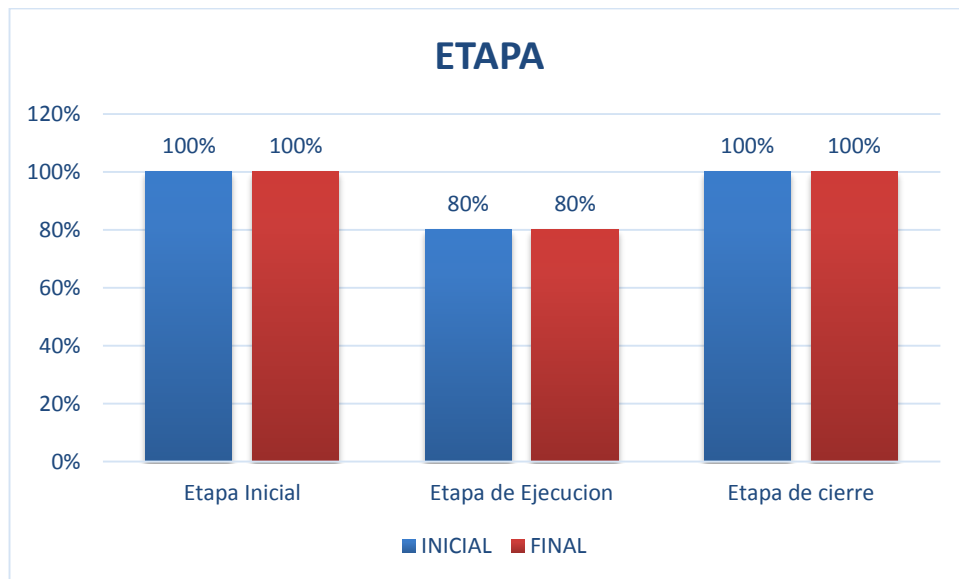
Encuesta

En esta sección se realiza un contraste en cuanto a las competencias iniciales y finales en la dimensión del saber a partir de los resultados de las encuestas de conocimiento para cada variable.

Etapa

Se evidencia en la figura 15, que los niveles de conocimiento para esta variable permanecen estables, es decir que, la población mantuvo un nivel muy alto tanto al inicio, como al finalizar el proceso de la comunidad virtual de práctica en cuanto a la inicial y de cierre y en un nivel alto en la etapa de ejecución con respecto a la claridad de los elementos y actividades que intervienen las fases iniciales, de ejecución y de cierre.

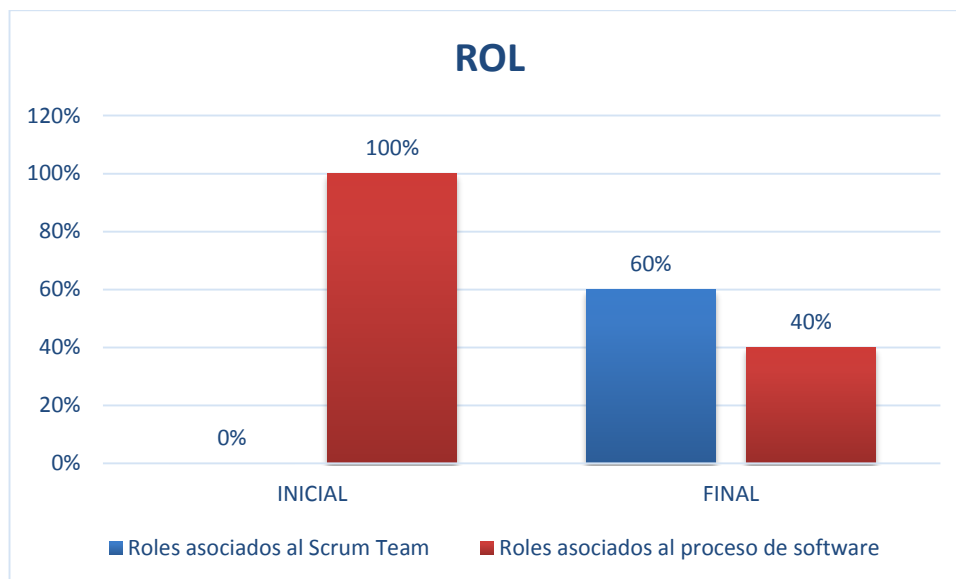
Figura 15. Comparación de competencias - Variable Etapa



Rol

Se observa que la comunidad virtual de práctica tuvo influencia en cuanto a la orientación conceptual de los roles asociados a procesos de construcción de software, ya que como se observa en la figura 16, en la encuesta de conocimientos inicial, la población se enfoca en roles asociados al proceso de software en su totalidad; mientras que, en la encuesta final, aparecen roles asociados al Scrum Team como roles predominantes, y se observa en la figura 16.

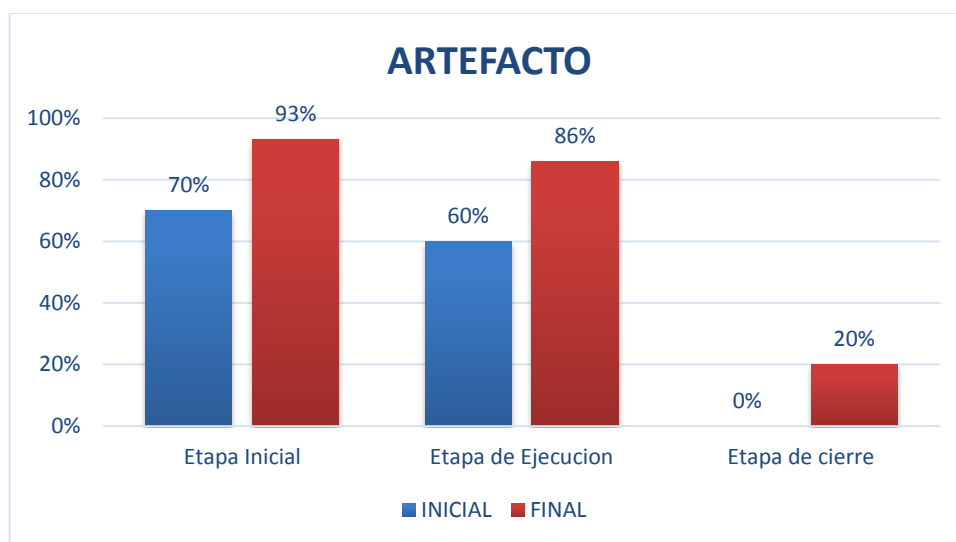
Figura 16. Comparación de competencias - Variable Rol



Artefacto

A partir de la figura 17, la dimensión del saber perteneciente a la variable artefacto, mejoró después de la aplicación de la comunidad virtual de práctica; sin embargo, en la etapa de cierre, los conceptos se mantienen en un nivel muy bajo.

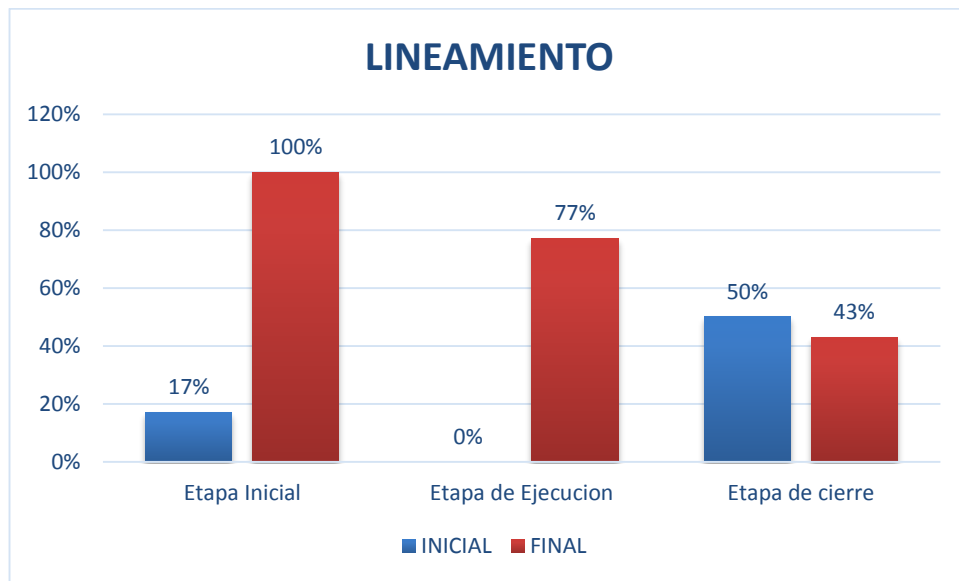
Figura 17. Comparación de competencias - Variable Artefacto



Lineamiento

Los conocimientos en cuanto a la variable lineamiento según la figura 18 aumentaron de manera significativa al finalizar el proceso de la comunidad virtual de práctica, sobre todo en las etapas inicial y de ejecución, siendo la etapa de cierre en la que se encuentra una depreciación con respecto a la encuesta inicial, cual lo mantiene en un nivel bajo.

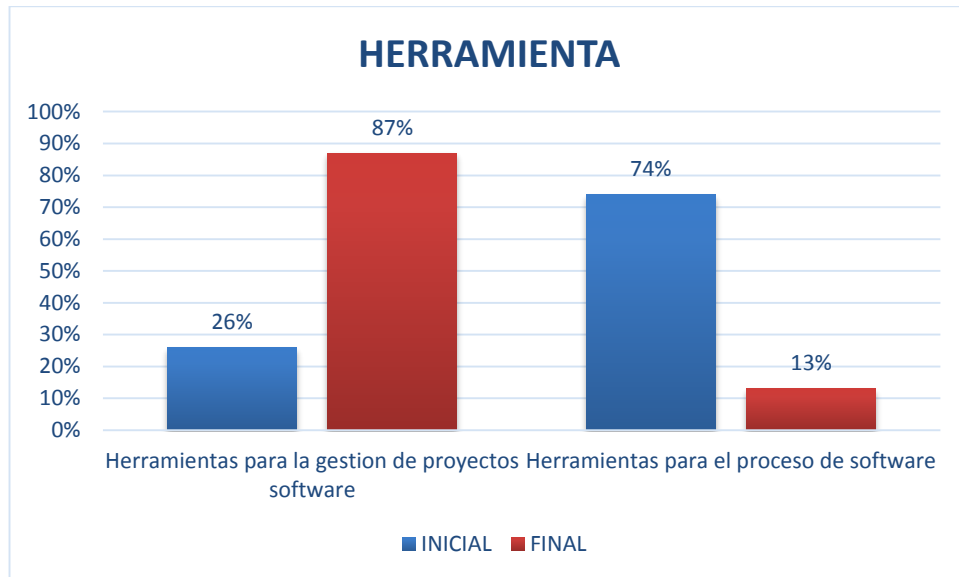
Figura 18. Comparación de competencias - Variable Lineamiento



Herramienta

El contraste de las encuestas inicial y final, muestra en la figura 19 un cambio de enfoque en la población, en cuanto al tipo de herramientas utilizadas en procesos de construcción de software. Inicialmente los estudiantes se inclinaron por herramientas para soportar el proceso de software, mientras que, al terminar la comunidad de práctica, predominaron las herramientas para la gestión de proyectos de software. Lo anterior, influenciado por la comunidad virtual de práctica al incluir la utilización de la herramienta Kunagi como parte del proceso de aprendizaje.

Figura 19. Comparación de competencias - Variable Herramienta



Taller

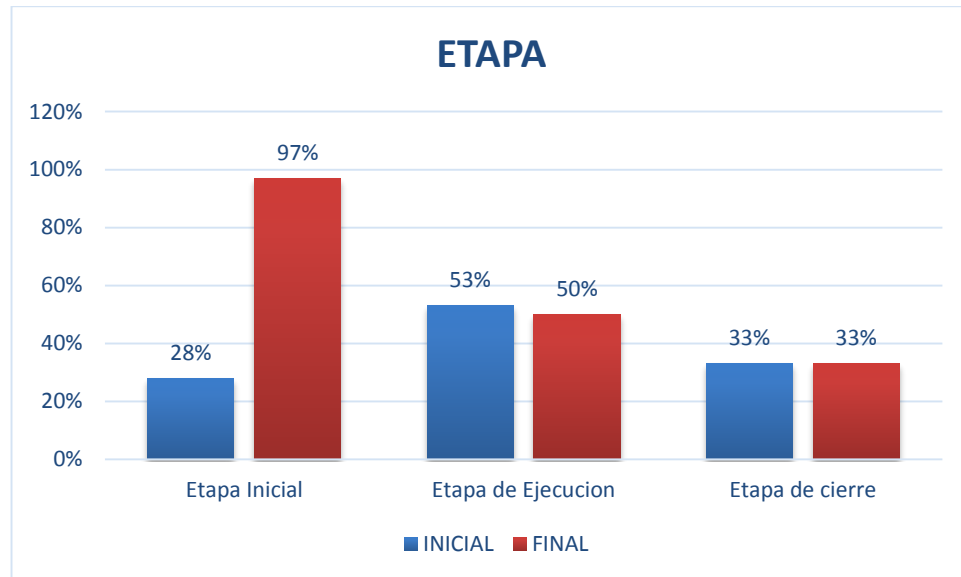
En esta sección se realiza un contraste en cuanto a las competencias iniciales y finales en la dimensión del saber hacer a partir de los resultados del taller para cada variable.

Etapas

Esta variable, presenta una evolución significativa en la etapa inicial, debido a que como se muestra en la figura 20, en el primer taller, la población obtuvo un rendimiento bajo en la aplicación de elementos y actividades pertenecientes a etapa; mientras que, en el taller final, dicho rendimiento fue muy alto.

Por otra parte, la dimensión del saber hacer se mantuvo constante en las etapas de ejecución y cierre.

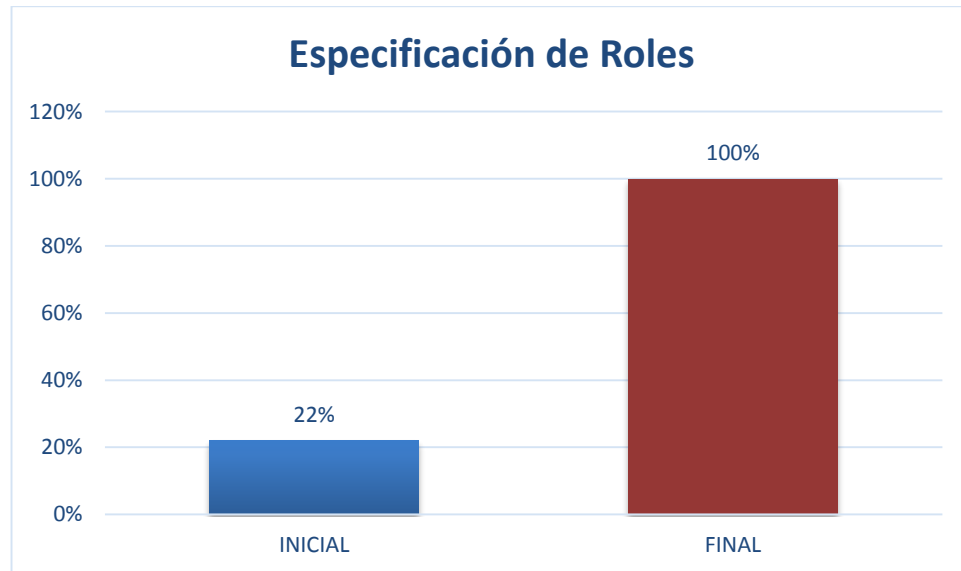
Figura 20. Comparación de competencias en Taller - Variable Etapa



Rol

La comunidad virtual de práctica incidió positivamente en cuanto a la especificación de roles dentro de los equipos de trabajo en los talleres prácticos. Según la figura 21, en el momento inicial, solo el 22% de los equipos realizaron una especificación de roles; mientras que, en el taller final, la totalidad de los equipos de trabajo realizó una especificación de roles basada en la metodología SCRUM y con ayuda de la herramienta Kunagi.

Figura 21. Comparación de competencias en Taller - Variable Rol

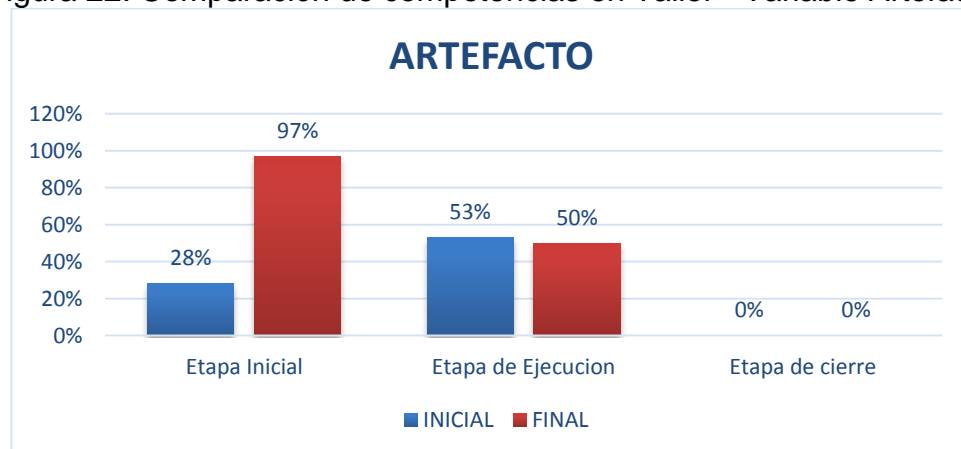


Artefacto

Se observa un cambio importante en el contraste de la etapa inicial en cuanto a la elaboración y entrega de artefactos que aumenten valor al producto o al proceso de software, pasando de un rendimiento bajo en el primer momento, a uno muy alto como se observa en la figura 22.

Las etapas de ejecución y de cierre se mantienen constantes, en la aplicación de los conceptos teóricos.

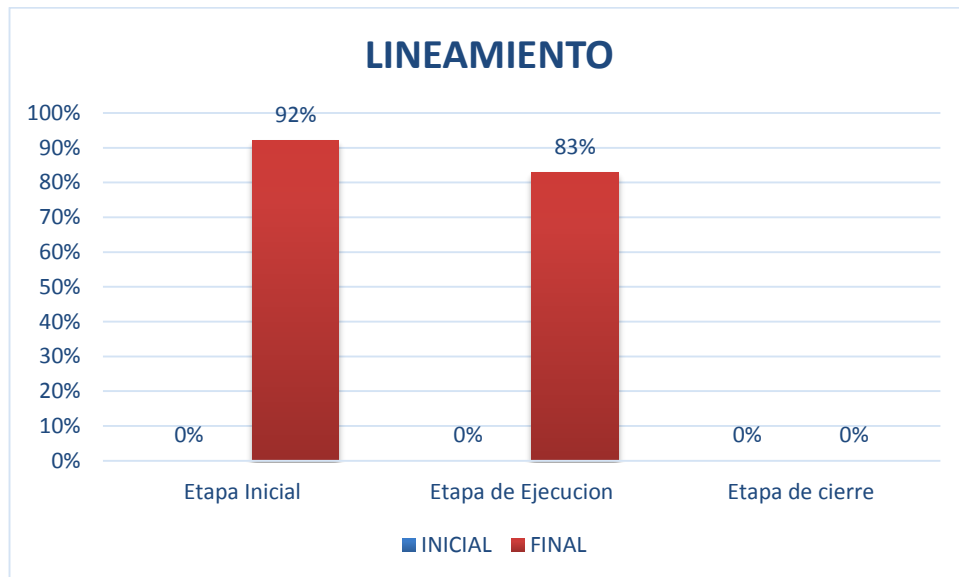
Figura 22. Comparación de competencias en Taller - Variable Artefacto



Lineamiento

En la figura 23, la aplicación de lineamientos, como rasgos que posibiliten la toma de decisiones dentro de proyectos de construcción de software, fue nula según los resultados del taller inicial. Sin embargo, el nivel de aplicación de lineamientos por parte de los equipos de trabajo mejoro significativamente en la etapa inicial y de ejecución del taller realizado al finalizar el proceso de la comunidad virtual de práctica.

Figura 23. Comparación de competencias en Taller - Variable Lineamiento

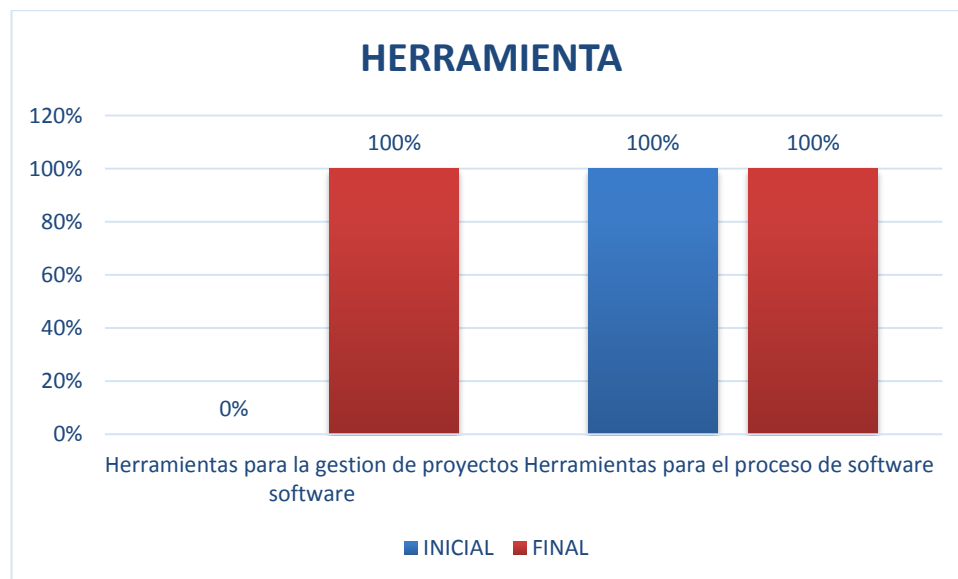


Herramienta

Los equipos de trabajo utilizaron herramientas en los talleres propuestos para diferentes actividades tales como modelado UML, diseño de Mockups, diseño de bases de datos, codificación, utilización de Frameworks de desarrollo, entre otros.

La diferencia más sobresaliente en el taller final fue la utilización de la herramienta Kunagi, orientada a la gestión de proyectos de desarrollo de software mediante la metodología SCRUM por parte de todos los equipos de trabajo, quedando en evidencia en la figura 24.

Figura 24. Comparación de competencias en Taller - Variable Herramienta



3.3.2. **Resultados globales.** En esta sección se evidencia de manera resumida, la evolución en las competencias en construcción de software basada en equipos en un estado inicial y final, en cuanto a las dimensiones del saber y el saber hacer.

En la figura 25, está plasmado el saber inicial y en la figura 26 el saber final, con lo cual se puede apreciar las competencias en la dimensión del saber, mejoraron considerablemente en las variables de lineamiento y artefacto, las demás se mantuvieron en un nivel alto.

Figura 25. Resultados globales - Saber inicial

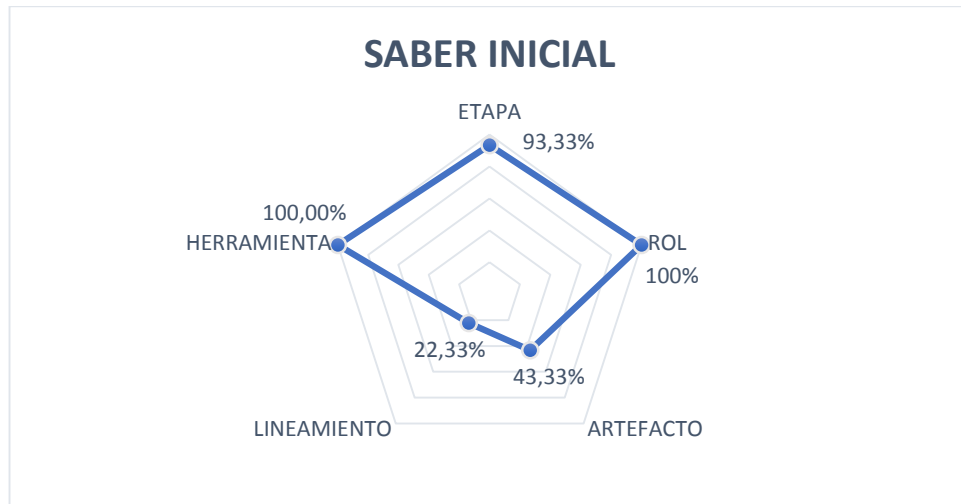
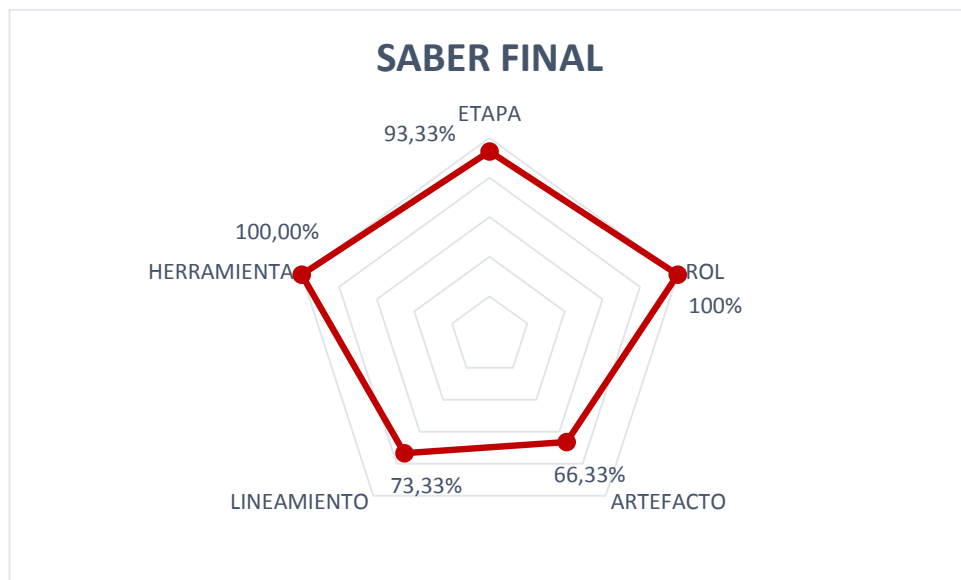


Figura 26. Resultados globales - Saber final



Por otra parte, y en relación con el Saber hacer inicial (figura 27) y el Saber hacer final (figura 28), las competencias en la dimensión del saber hacer, es decir, la capacidad de llevar a la práctica los conocimientos en construcción de software basada en equipos, mejoraron su rendimiento en cada una de las variables. Lo anterior permite observar que los miembros de la comunidad virtual de práctica finalizaron con una coherencia entre el nivel de conocimientos y su aplicación.

Figura 27. Resultados globales - Saber hacer inicial

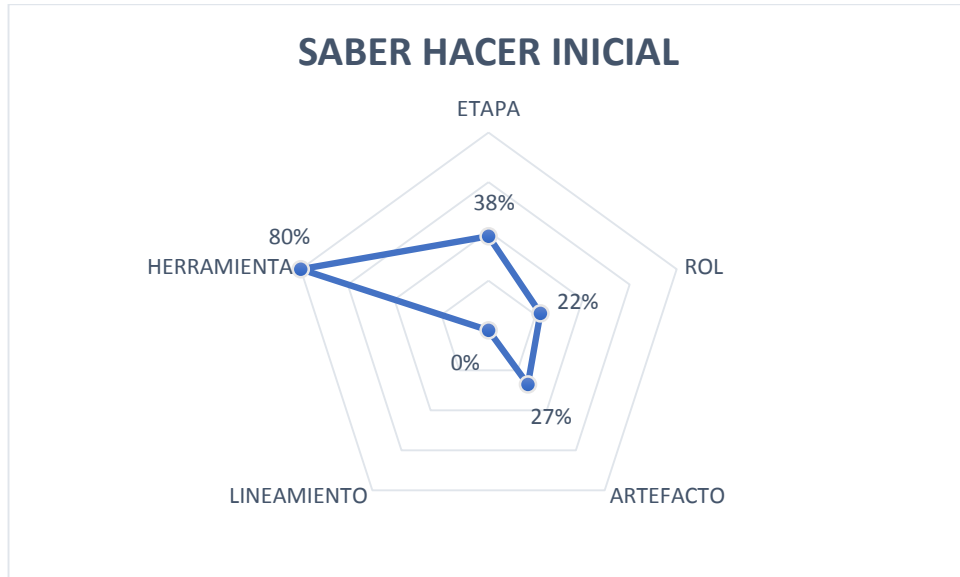
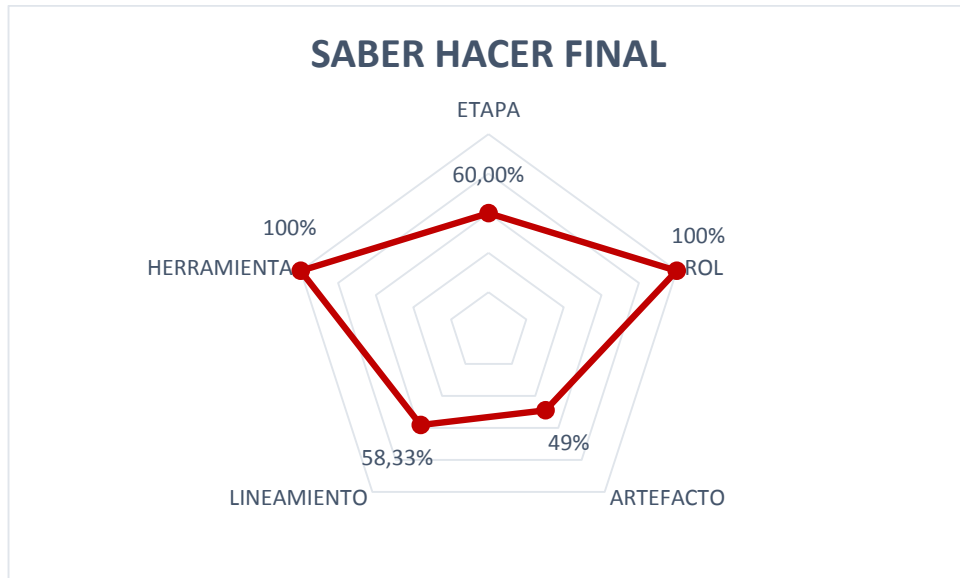


Figura 28. Resultados globales - Saber hacer final



4. ANÁLISIS Y DISCUSIÓN DE LOS RESULTADOS

Después de obtener los resultados de la investigación, se hizo un análisis de cada capítulo culminado y se obtuvo la siguiente síntesis.

En el capítulo uno, donde se deseaba identificar el nivel de desempeño que tienen los estudiantes de Ingeniería de Sistemas en la construcción de software basada en equipos. A partir de los resultados obtenidos en la encuesta, el taller de conocimientos previos y a partir del contraste de técnicas en el cual se evalúa cada variable establecida, se puede observar que desde un punto de vista global, los estudiantes objeto de estudio tienen un nivel conceptual e intelectual elevado en cuanto al proceso de construcción de software, debido a que identifican las etapas de desarrollo de software y las actividades correspondientes para cada una, además los estudiantes reconocen y argumentan que es un rol y cuáles son las funciones o responsabilidades para cada uno de ellos, de igual manera se observó que desde un aspecto conceptual los estudiantes asemejan que es un artefacto, también con relación a las herramientas ellos conocen diversas herramientas que pueden ser utilizadas en las diferentes etapas del proceso de desarrollo y así garantizar la calidad del producto en cada una de ellas. Sin embargo, al poner en práctica en el taller que se les realizó lo que anteriormente se sustenta, se detallaron grandes fallas en cada grupo. Para la parte de etapa, los estudiantes se centran en la etapa de ejecución, en menor grado en la inicial y con poca incidencia, la etapa de cierre, debido a que solo entregan una parte de lo que deberían entregar (Producto funcional), en cuanto a rol únicamente dos grupos de trabajo hicieron evidente su poder conceptual e identificaron y plasmaron en documento los diferentes roles que identificaron y proyectaron. En cuanto la variable artefacto, se centraron en la etapa de ejecución. Demostrando que carecen de conocimiento conceptual y práctico para la reconocer y plasmar artefactos en la etapa de cierre; tienen un concepto de artefacto muy alejado del real, en cuanto a lineamiento, los estudiantes tienen una considerable falencia para identificar y reconocer lo que es un lineamiento y esto fue reflejado en el taller, dado que no agregaron aspectos que le dieran un valor agregado al producto funcional en cuestión, por último, en la utilización de herramientas, los estudiantes se centraron en la etapa ejecución, dejando de lado la gestión del proyecto, que corresponde a una parte importante en el proceso o herramientas para un posterior análisis y una retroalimentación del trabajo realizado.

Para el capítulo dos, la conformación de la comunidad virtual de práctica y la participación de la población en el conjunto de actividades y experiencias

permitieron un impacto sobre las competencias en construcción de software basada en equipos desde un enfoque ágil. Los resultados obtenidos en cada variable, tanto en la encuesta de conocimientos, como en el taller y adicionalmente en el contraste de técnicas indican que en primer lugar, los estudiantes pertenecientes a la comunidad virtual de práctica poseen en general, un nivel de conocimiento alto en las etapas iniciales y de ejecución vinculadas a las variables de etapa, lineamiento y artefacto; siendo en la etapa de cierre, donde se encontraron dificultades en cuanto a los conceptos teóricos relacionados a cada variable.

En segundo lugar, las variables, rol y herramienta, fueron altamente influenciadas por el enfoque y los contenidos de la comunidad virtual de práctica, lo cual se ve reflejado en los resultados de la encuesta y el taller, en los cuales se evidencian conceptos teóricos y su aplicación, específicamente en los enfoques ágiles de desarrollo de software, la metodología de desarrollo SCRUM y la aplicación de herramientas para la gestión de proyectos de software, como el caso de Kunagi. Obteniendo un desempeño alto en las etapas inicial y de ejecución en la aplicación práctica de los conceptos teóricos en el taller final; la etapa de cierre presenta un nivel bajo, siendo proporcional a los resultados de la encuesta.

Con el fin de determinar el nivel de aporte hecho por la comunidad virtual de práctica, dirigido a las competencias en construcción de software basada en equipos en la dimensión del saber y el saber hacer, se realizó un análisis en cuanto a los resultados obtenidos de la encuesta de conocimientos y el taller. El análisis consistió en un contraste del nivel inicial y final de desempeño obtenido para cada una de las variables definidas (Etapa, rol, artefacto, lineamiento, herramienta).

Al finalizar el análisis, se encuentra que, la dimensión del saber, la cual tenía un nivel de desempeño alto en un primer momento, se mantuvo estable en las variables de etapa, rol y herramienta. Además, incremento el desempeño en las variables restantes en las cuales existían dificultades inicialmente.

En la dimensión del saber hacer, inicialmente se observa una incoherencia entre los conocimientos y el nivel de aplicación en entornos de construcción de software basado en equipos. Sin embargo, esta situación tiene una mejoría significativa según los resultados obtenidos en el taller final.

A pesar de la mejoría en la mayoría de los elementos involucrados en las competencias en construcción de software basada en equipos, aún se encuentran dificultades en la etapa de cierre ya que se observa un nivel de desempeño bajo en las variables etapa, lineamiento y herramienta. Siendo la etapa de cierre un aspecto a tener en cuenta para abordar en futuras experiencias y trabajos.

Finalmente, al concluir el proceso de la comunidad virtual de práctica, se obtiene que las competencias de los estudiantes son proporcionales entre la dimensión del saber y su respectiva aplicación en un espacio de construcción de software, lo cual indica que en general, la comunidad virtual de práctica tuvo una influencia positiva en su rol de desarrollar dichas competencias en sus integrantes.

5. CONCLUSIONES

Al finalizar el proceso de investigación, teniendo en cuenta la fundamentación teórica, las experiencias y resultados obtenidos, se obtuvieron las siguientes conclusiones.

- La conformación de una comunidad virtual de práctica permite crear un espacio alternativo y complementario a la formación que se ofrece en el programa de Ingeniería de Sistemas de la Universidad de Nariño potencializar las competencias adquiridas con el aprendizaje tradicional.
- Al identificar el nivel de competencias iniciales de la población, se encuentra que existe una brecha entre los conocimientos en construcción de software basada en equipos y su respectiva aplicación en un contexto basado en la realidad.
- La comunidad virtual de práctica tuvo un impacto positivo en las competencias en construcción de software en la dimensión del saber y el saber hacer, debido a que se observó una evolución significativa al finalizar el proceso de aprendizaje.
- Los contenidos y actividades dentro de la comunidad virtual de práctica influenciaron directamente en la evolución de los conocimientos y aplicación de enfoques y metodologías ágiles de desarrollo de software, así como de la adaptación de buenas prácticas en la construcción de software basada en equipos.
- La inclusión de una herramienta orientada a la gestión de proyectos de software como kunagi, permite potencializar de manera directa el desempeño de competencias en construcción de software.
- A pesar de que en general se mejoraron las competencias de la población, aún se presentan dificultades en la etapa de cierre de proyectos de construcción de software.
- Al finalizar el proceso de la comunidad virtual de práctica, las competencias en construcción de software son proporcionales en cuanto a la dimensión del saber y el saber hacer. Esto permite establecer que existe una coherencia entre los conocimientos y su aplicación.

- La motivación es un factor fundamental durante el desarrollo de una comunidad virtual de práctica, en el caso específico, se concluye que se esperaba una cooperación más activa a la obtenida, a pesar de las estrategias empleadas para incentivar la participación.

6. RECOMENDACIONES

Usar una herramienta orientada a la gestión de proyectos de software como Kunagi, impacta de manera positiva y propicia el desarrollo de competencias en construcción de software basada en equipos. Por esta razón, se recomienda utilizar una herramienta para la gestión de proyectos en las materias relacionadas con la construcción de software en el currículo del programa de Ingeniería de Sistemas de la Universidad de Nariño.

Conformar una comunidad virtual de práctica que permita crear un espacio complementario de formación en el programa de Ingeniería de Sistemas de la Universidad de Nariño para potencializar las competencias adquiridas con el aprendizaje tradicional y debido a que se obtiene un impacto positivo en las competencias en construcción de software en la dimensión del saber y el saber hacer, se recomienda continuar con proyectos de investigación de este tipo, con el fin de aprovechar el potencial que tienen las comunidades virtuales de práctica y de explorarlas en otros campos de conocimiento de la Ingeniería de Sistemas.

Utilizar contenidos y actividades de la comunidad virtual de práctica influye directamente en la evolución de los conocimientos, aplicación de enfoques, metodologías ágiles de desarrollo de software y adaptación de buenas prácticas en la construcción de software basada en equipos, se recomienda utilizar las actividades involucradas en la comunidad virtual de práctica en las diferentes estrategias de las materias de la línea de desarrollo de software, asociados al programa de Ingeniería de Sistemas de la Universidad de Nariño.

Propiciar espacios académicos para desarrollar las competencias en construcción de software basada en equipos debido a que existen dificultades con respecto a la etapa de cierre en la dimensión del saber y el saber hacer. Esto se puede realizar bien sea a través de asignaturas del pensum del programa de Ingeniería de sistemas vinculadas a la línea de software o mediante la utilización de una comunidad virtual de práctica.

BIBLIOGRAFÍA

- [1] R. Kummitha y S. M, «Dynamic curriculum development of social entrepreneurship - A case of study of TISS.,» *The International Journal of Management Education*, vol. 13, 2015.
- [2] D. Dumitru y V. Enăchescu, «Communities of Praticce as a mean for Decentralization,» *Proedia-Social and Behavioral Sciences*, vol. 187, 2015.
- [3] D. Lupu y A. Lauretiu, «Using new Communication and Information Technologies in Preschool Education.,» *Procedia-Social and Behavioral Sciences*, vol. 187, 2015.
- [4] H. Kim, I. Kim y H. Lee, «Third-Party mobile app developers continued participation in plattform-centric ecosystems: An empirical investigation of two different mechanisms.,» *International Journal of Information Management.*, vol. 36, 2016.
- [5] Y. Pan, Y. Xu, X. Wang, C. Zhang, H. Ling y J. Lin, «Integrating social networking support for dyadic knowledge exchange: a study in a vidual commuity of practice.,» *Information and Management.*, vol. 52, 2015.
- [6] F. Rogo, L. Cricelli y M. Grimaldi, «Assesing the performance of open innovation practices: A case study of a community of innovation.,» *Technology in Society*, vol. 38, 2014.
- [7] N. Ratzinger-Sakel y G. Gray, «Moving toward a learned profession and purposeful integration: Quantifying the gap between the academic and practice communities in auditing and identifying new research opportunities.,» *Journal of Accounting Literature.*, vol. 35, 2015.
- [8] A. Jiménez-Zarco, I. González González, F. Saigí-Rubió y J. Torrent-Sellens, «The co-learning process in healthcare professionals: Assessing user satisfaction in virtual communities of practice.,» *Computers in Human Behaviour*, vol. 51, 2015.
- [9] M. Dascalu, C. Bodea, M. Lytras, P. De Pablos y A. Burlacu, «Improving e-learning communities through optimal composition of multidisciplinary learning groups.,» *Computers in human behavior.*, vol. 30, 2015.

- [10 C. Robertson, K. Gilley, V. Crittenden y W. Crittenden, «An analysis of the predictors of software piracy within Latin America.,» Journal of Bussiness Research, vol. 61, 2008.
- [11 S. Narayanan, V. Jayaraman, Y. Luo y J. Swaminathan, «The antecedents of process integration in bussiness process outsourcing and its effects on firm oerformance,» Journal of Operations Management, vol. 29, 2011.
- [12 MINTIC y COLCIENCIAS, Visión estratégica del sector de software y servicios asociados plan de mercadeo y ventas regionalizado del sector en Colombia, Colombia, 2013.
- [13 S. McConnell, Software estimation: Demystifying the black art, Washington: Redmond, Microsoft Press, 2006.
- [14 T. DeMarco, Slack getting past burnout busywork and the myth of total efficiency., New York, USA: Boradway Books, 2001.
- [15 F. Brooks, The Mythical Man-Month: Essays in Software Engineering., Addison-Wesley, 1995.
- [16 C. Parnin y S. Rugaber, «Resumption strategies for interrupted programming.,» Software Quality Journal, vol. 19, 2010.
- [17 A. Cockburn, «Alistair Cockburn,» 19 06 2008. [En línea]. Available: <http://alistair.cockburn.us/Effectiveness+of+different+modes+of+communication+graph.gif>. [Último acceso: 17 01 2017].
- [18 S. Tobon, «La formación basada en competencias en la educación superior: el enfoque complejo.,» Universidad Autónoma de Guadalajara, 2008.
- [19 D. Rodriguez, C. Busco y R. Flores, «Information Technology within society's evolution.,» Technology in Society, vol. 40, 2015.
- [20 L. Clarke, «The POD model: Using communities of practice theory to conceptualise student teachers professional learning online.,» vol. 52, n° 3, 2009.
- [21 C. Ordoñez, G. Hernández, H. Ordoñez, A. Ordoñez y C. Cobos, «Virtual community of practice to potencie knowledge and skills for building mobile applications in computer science students.,» de 2016 8th Euro American Conference on Telematics and Information Systems (EATIS), Cartagena, Colombia, IEEE, 2016.

- [22 J. Duart y A. Sangrá, Aprender en la virtualidad., Barcelona: Gedisa, 2000.
]
- [23 M. Cebrián, Enseñanza virtual para la innovación universitaria., Narcea Ediciones, 2003.
- [24 E. Wenger, Communities of practice: Learning, meaning and identity,
] Cambridge: Cambridge University Press, 1998.
- [25 H. Rheingold y A. Weeks, Net smart: How to thrive online., Mit Press, 2012.
]
- [26 E. Wenger, R. A. McDermott y W. Snyder, Cultivating communities of practice:
] A guide to managing knowledge., Harvard Business press, 2002.
- [27 J. Almenara, «Comunidades virtuales para el aprendizaje. Su utilización en la
] enseñanza,» Edutec. Revista electronica de tecnologia educativa., nº 20, 2006.
- [28 Z. Bozu y F. Muñoz, «Creando comunidades de práctica y conocimiento en la
] universidad: Una experiencia de trabajo entre las universidades de lengua catalana.,» RUSC. Universities and knowledge society journal., vol. 6, nº 1, 2009.
- [29 B. Meyer, Object-Oriented software construction, New York: Prentice Hall,
] 1988.
- [30 R. Pressman, Software Engineering. A partitioner's guide., New York: McGraw-Hill, 2002.
- [31 R. A. Española, «Real Academia Española,» [En línea]. Available:
] <http://www.rae.es/>. [Último acceso: 15 12 2017].
- [32 R. Hernandez Sampieri, C. Fernández y P. Baptista, Metodología de la
] investigación, MCGRAW-HILL, 1997.
- [33 R. Landeau, Elaboración de trabajos de investigación, ALFA, 2007.
]

ANEXOS

Anexo A. Formulario información sociodemográfica

INFORMACIÓN SOCIODEMOGRÁFICA

1. Edad: _____
2. Género: M: _____ F _____
3. Lugar de origen: _____
4. Estrato socioeconómico:

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.

5. ¿Eres beneficiario de algún tipo de apoyo económico?

SI: _____ NO: _____

¿Cuál?: _____

6. ¿De qué manera financia sus estudios?

- Recursos propios.
- Apoyo de padres o tutor.
- Ambos.
- ICETEX.
- Beca.

7. ¿Con quién comparte Su residencia?

- Padres.
- Hermanos(as).
- Primos.
- Abuelos.
- Tíos.
- Otra. ¿Cuál?: _____

8. Número de hermanos: _____

9. Estado civil

- Soltero
- Casado
- Viudo
- Divorciado
- Unión Libre

10. ¿Tiene personas a cargo, que dependan económicamente de usted?:

SI: ____ NO: ____

¿Cuántas?: _____

11. Semestre y año de ingreso a la Universidad al programa de Ingeniería de Sistemas:

Semestre: A ____ B ____ Año: _____

12. En qué sede de la Universidad ingreso a estudiar la carrera de Ingeniería de Sistemas

- Pasto
- Ipiales
- Tumaco
- Túquerres
- Samaniego
- La Unión

13. ¿Ingresó a una carrera diferente antes de cursar la actual?

SI: ____ NO: ____

¿Cuál(es)?:

14. ¿Ha repetido alguna(s) materia(s) en el transcurso de la Carrera?

SI: ____ NO: ____

¿Cuál(es)?:

15. ¿Ha tenido que repetir más de una vez una materia?

SI: ____ NO: ____

¿Cuál(es)?:

16. ¿Haces parte del Programa de Formación Vive Digital?

SI: ____ NO: ____

Año de ingreso al programa: _____

17. ¿Qué lo motivó a estudiar Ingeniería de Sistemas?

18. ¿Qué actividad se visualiza realizando dentro de 5 años?

Anexo B. Formulario Competencias preliminares

El propósito de este cuestionario es Identificar el nivel de desempeño que tienen los estudiantes de Ingeniería de Sistemas de la Universidad de Nariño, para la construcción de software a través de equipos de trabajo.

ETAPA

1. ¿Conoce una metodología para construir software a través de equipos de trabajo?

SI: ____ NO: ____

2. Si la respuesta anterior fue afirmativa, escriba cuál(es) metodología(s) conoce, de lo contrario pase a la pregunta siguiente.

3. Una etapa es un elemento que reúne a un conjunto de actividades que se realizan para para alcanzar un fin. Con base en la definición anterior registre las actividades que se desarrollan por cada etapa

Etapa	Actividades
Inicio	
Ejecución	
Cierre	

ROL

4. Un rol es la función o papel que una persona desempeña en un equipo de trabajo. Con base en la definición anterior registre cuáles son las acciones que desempeña cada rol en el proceso de construcción de software.

Rol	Acciones
Cliente o Interesado	
Ingeniero de Requisitos	
Analista de Software	
Diseñador De Software	
Desarrollador	
Jefe de Calidad	
Jefe de Pruebas	
Líder de equipo	
Líder de soporte	
Otro rol?	

ARTEFACTO

5. Un artefacto o entregable es un producto que se realiza y que agrega valor a la construcción de un producto software. Con base en la definición anterior registre qué artefactos, se deben elaborar el proceso de construcción de software por cada etapa.

Etapa	Artefactos
Inicio	
Ejecución	
Cierre	

LINEAMIENTO

6. Un lineamiento es una tendencia, una dirección o un rasgo que posibilita la toma de decisiones. Con base en la definición anterior registre qué datos, se deben recopilar durante el proceso de construcción de software por cada etapa.

HERRAMIENTA

7. Una herramienta es un elemento que permite soportar y facilitar procesos de desarrollo de software. Con base en la definición anterior registre qué herramientas conoce, que le permitan apoyar o soportar el proceso de construcción de software en equipo.

8. Como opinión personal, en qué categoría(s) siente mayor interés y le gustaría profundizar.

- Etapa
- Rol
- Artefacto
- Lineamiento
- Herramienta

9. De una justificación de acuerdo con su respuesta anterior.

Anexo C. Actividad del Taller N° 1.

Descripción del problema

A continuación, se presenta una síntesis de las actas de reunión con las ideas más importantes de las entrevistas realizadas a los diversos actores del Departamento Administrativo Nacional de Estadística (DANE), que está sistematizando una parte de sus procesos con la ayuda de una empresa de base tecnológica de la que Ustedes hacen parte.

Proyecto DIVIPOLA Cliente: DANE

Síntesis actas de reunión con los interlocutores del cliente

En las reuniones con el Auxiliar Administrativo, el Jefe de Proyectos y el Director de la Unidad de Análisis de Datos del DANE, se determinó que para el proceso de gestión de información sobre la división política de Colombia, se requiere gestionar los datos sobre la población que existe en el país. Colombia, se encuentra dividida en departamentos, de los cuales se manejan los datos: nombre y año de fundación. Cada departamento tiene municipios, de los cuales, se tiene los datos: nombre y el año de creación. Para cada municipio se debe registrar el número total de habitantes que tiene por cada año. Además, se requiere que el producto software sea usable, los datos perduren en el tiempo y que la aplicación sea web.

Actividades

Con base en la descripción de la síntesis presentada, se debe:

Construir un producto software funcional que debe incluir la necesidad “Como Auxiliar Administrativo quiero registrar los habitantes para un municipio de un año específico”.

Adicionalmente subir a la plataforma los siguientes elementos:

1. Todos los artefactos realizados y datos recopilados para poder construir el producto software.
2. Diligenciar la Tabla 1, donde se debe consignar las dificultades presentadas en el desarrollo del taller y las temáticas que le gustaría se reforzaran en una futura actividad.

<i>Dificultades presentadas en el desarrollo de la actividad</i>	<i>Elementos que le gustaría fueran reforzados</i>

- Definición de Artefacto: Un artefacto o entregable es un producto que se realiza y que agrega valor a la construcción de un producto software.

Anexo D. Guía de observación

EQUIPO N°	ROLES DESEMPEÑADOS	OBJETIVO DEFINIDO PARA EL EQUIPO DE TRABAJO	ACTIVIDADES REALIZADAS	HERRAMIENTAS UTILIZADAS	TIEMPO UTILIZADO	
					Inicio	Fin
		<ul style="list-style-type: none"> • MUY ALTO • ALTO • MEDIO • BAJO • MUY BAJO 				

Anexo E. Actividad KANBAN

ACTIVIDAD KANBAN

QUE ES KANBAN

Kanban es un sistema de administración de tareas y flujos de trabajo. Surgió en la década de los 50 en la compañía de vehículos TOYOTA, la cual diseñó este sistema para el manejo de flujo de materiales en una línea de ensamble de manera eficaz y eficiente.

Desde sus orígenes kanban ha sido implementado en un sin número de organizaciones, permitiendo aumentar la eficacia, productividad y calidad de productos y servicios. Actualmente se utiliza ampliamente en procesos de construcción de software y específicamente en metodologías de desarrollo ágiles. Esto debido a que sus principios (calidad, reducción del desperdicio, mejora continua, flexibilidad) se acoplan muy bien a enfoques ágiles de desarrollo de software. Kanban es una palabra japonesa que significa tarjetas visuales en la cual se asigna una actividad en una tarjeta y se ubica en una de tres categorías desde el inicio de la actividad, la ejecución y su finalización. De esta manera todos los integrantes del equipo pueden visualizar y entender de manera práctica el flujo de trabajo del equipo.

DESCRIPCIÓN DE LA ACTIVIDAD

La actividad consiste en resolver un reto propuesto a realizar en equipos de trabajo. En el cual cada equipo debe entregar un producto resultante a partir de unos parámetros entregados, además, cada equipo debe utilizar el sistema “kanban” para la planificación de actividades entre los miembros del equipo de trabajo, repartición de tareas, medición de tiempos de trabajo y en general la visualización del flujo de trabajo durante la actividad. Al finalizar se entrega como evidencia el tablero kanban con las tarjetas visuales que contienen las actividades realizadas, el responsable de la actividad y la hora de inicio y fin de la actividad, además de un listado resumen de actividades.

OBJETIVOS

- Conocer aspectos generales sobre kanban
- Aplicar kanban en un reto basado en equipos
- Desarrollar habilidades de planificación, medición y trabajo en equipo
- Evidenciar la eficiencia del trabajo realizado en cada etapa

MATERIALES

Para el desarrollo de la actividad se utilizaron los siguientes materiales:

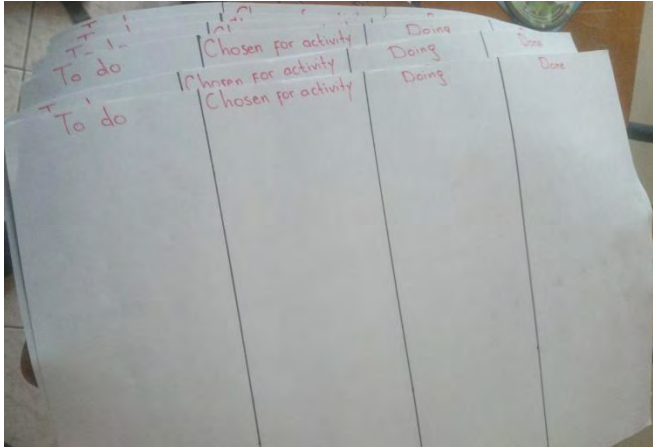
- Lápices:

Figura 29. Material actividad Kanban, Lápices



- Tableros kanban (cartulinas):

Figura 30. Material actividad Kanban, Tableros Kanban



- Tarjetas visuales (Notas adhesivas):

Figura 31. Material actividad Kanban, tarjetas visuales



DESARROLLO

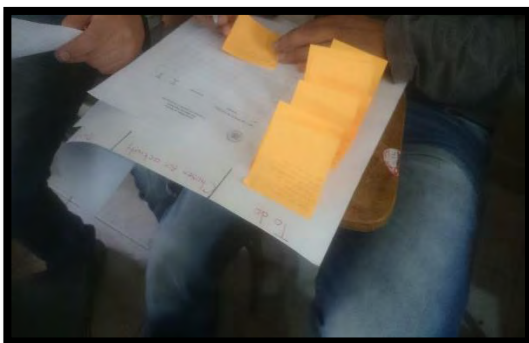
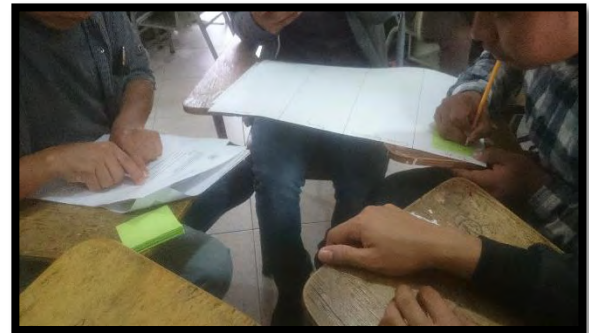
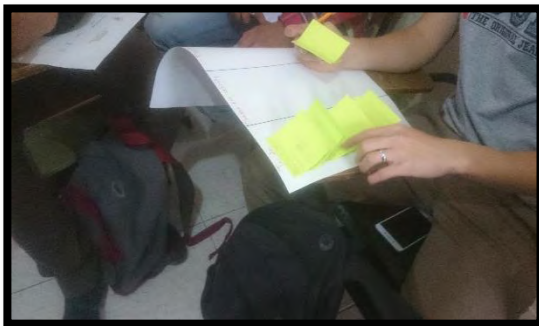
En primer lugar, se realizó una contextualización de la actividad, se explicaron las reglas de juego, se organizaron los equipos de trabajo y se entregaron los materiales. Una vez los equipos comenzaron la actividad se registró la hora de inicio. Posteriormente los equipos realizaron la planificación y distribución de actividades en el tablero kanban en el cual asignaban actividades a cada integrante

del equipo, registrando el nombre de la actividad, responsable de la actividad, hora de inicio y finalización en la correspondiente nota adhesiva (tarjeta visual).

Finalmente, los equipos de trabajo realizaron la entrega del producto generado, a partir de las premisas proporcionadas, además, de la entrega del tablero kanban debidamente diligenciado.

EVIDENCIAS

Figura 32.Evidencias - Actividad Kanban





RESULTADOS

Figura 33. Actividades encontradas Actividad Kanban

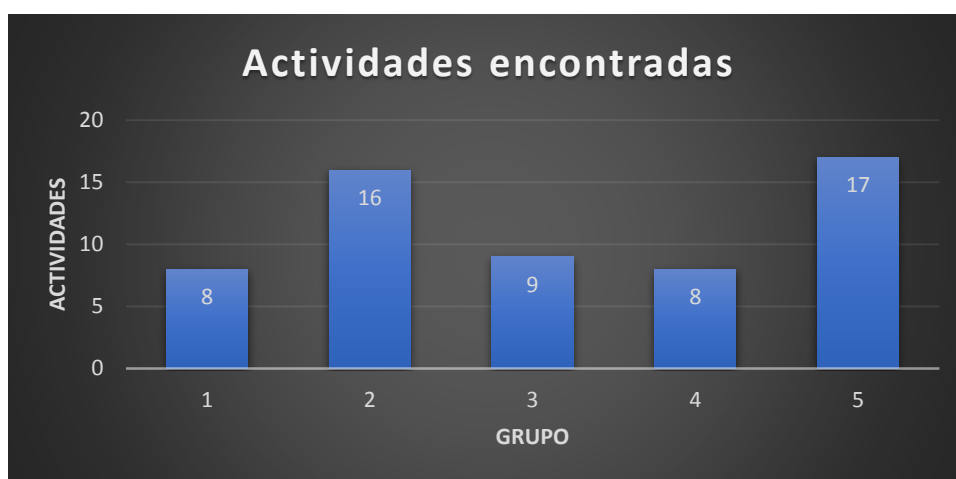
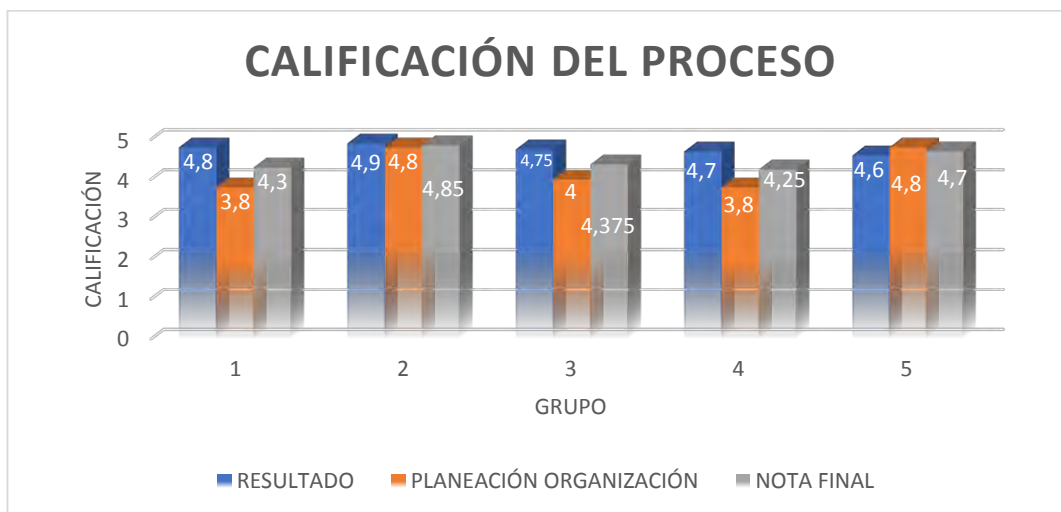


Figura 34. Calificaciones del proceso Actividad Kanban



Anexo F. Material Audio Visual

- **DEVELOPMENT TEAM:**
<https://www.youtube.com/watch?v=vj3xwRv45Ns>
- **PRODUCT OWNER:**
<https://www.youtube.com/watch?v=-6gaCtnbv2s>
- **SCRUM MASTER:**
<https://www.youtube.com/watch?v=MyvVbl6tfhw>
- **SCRUM TEAM:**
<https://www.youtube.com/watch?v=DLnfata7Q7Q>
- **ENFOQUE ÁGIL:**
<https://www.youtube.com/watch?v=-QZ6wKnFmCM>
- **SCRUM INTRODUCCIÓN:**
<https://www.youtube.com/watch?v=IG8ujC3Xx6U>

Anexo G. Actividad del Taller N° 2

Descripción del problema

A continuación, se presenta una síntesis de las actas de reunión con las ideas más importantes de las entrevistas realizadas a los diversos actores del Ministerio de Cultura de Colombia, que está sistematizando una parte de sus procesos con el soporte tecnológico de una empresa de desarrollo a la cual ustedes pertenecen.

Proyecto VIPAES (vinculación de profesionales en artes escénicas)

Cliente: Ministerio de Cultura

Síntesis actas de reunión con los interlocutores del cliente

En las reuniones con el Auxiliar Administrativo, el Jefe de Proyectos y el Director del Departamento de artes escénicas, se determinó que para el proceso de gestión de información sobre el proceso de vinculación de profesionales en artes escénicas, se requiere de una solución tecnológica a través de un producto software. Existen cuatro (4) áreas artísticas de las cuales cada una posee diferentes tipos de profesión. Se requiere el registro de los profesionales con sus datos básicos (Cedula, nombres, apellidos, fecha de nacimiento, sexo), Se elija el área de arte escénica y de acuerdo a su área la profesión.

Además, se requiere que el producto software sea usable, los datos perduren en el tiempo y que la aplicación sea web.

ÁREA	PROFESIONES
Cine/Teatro	Actor, Actriz, director, Doble, Extra
Moda	Estilista, Fotógrafo, Maquillador, Modelo
Música	Bailarín, Cantante, Coreógrafo, Músico
Circense	Acróbata, Mimo, Titiritero, Contorsionista

Actividades

Con base en la descripción de la síntesis presentada, se debe:

Construir un producto software funcional que debe incluir las siguientes necesidades:

- “Como Auxiliar Administrativo quiero registrar Profesiones pertenecientes a áreas de artes escénicas”.
- “Como Auxiliar Administrativo quiero registrar los profesionales de artes escénicas, vinculándolos con su respectiva área y profesión.

Adicionalmente subir a la plataforma los siguientes elementos:

Todos los artefactos realizados y datos recopilados para poder construir el producto software.

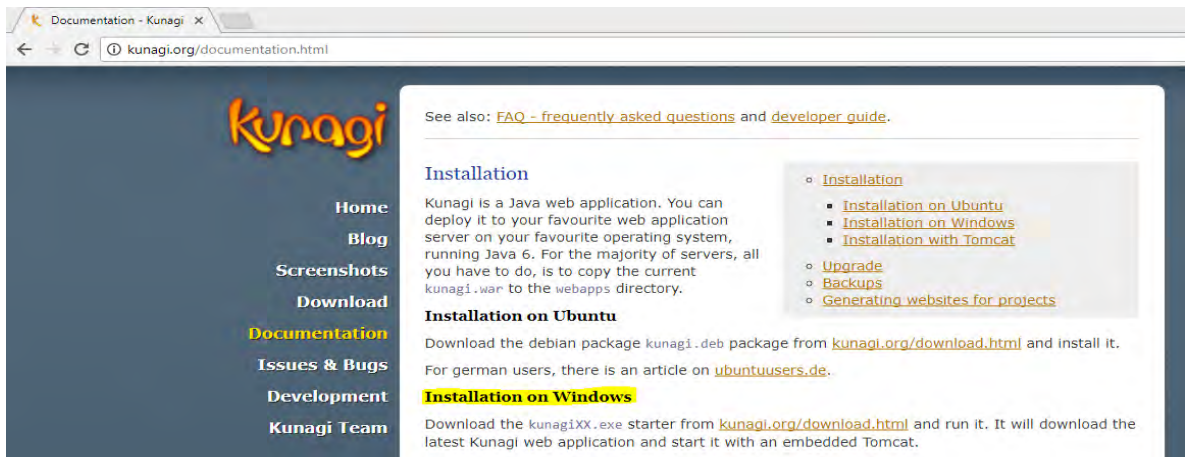
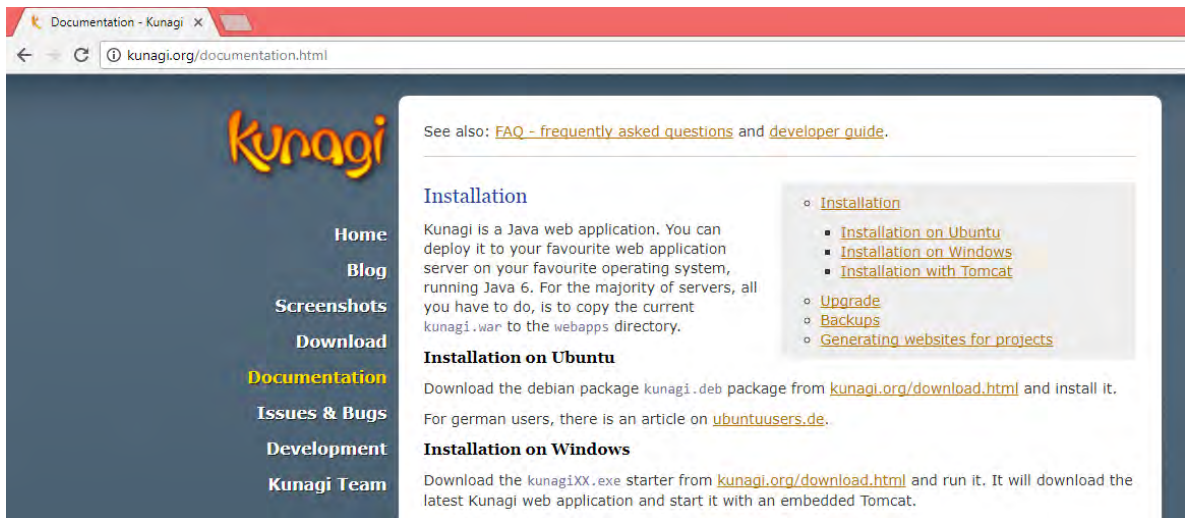
Anexo H. Guía de instalación herramienta KUNAGI



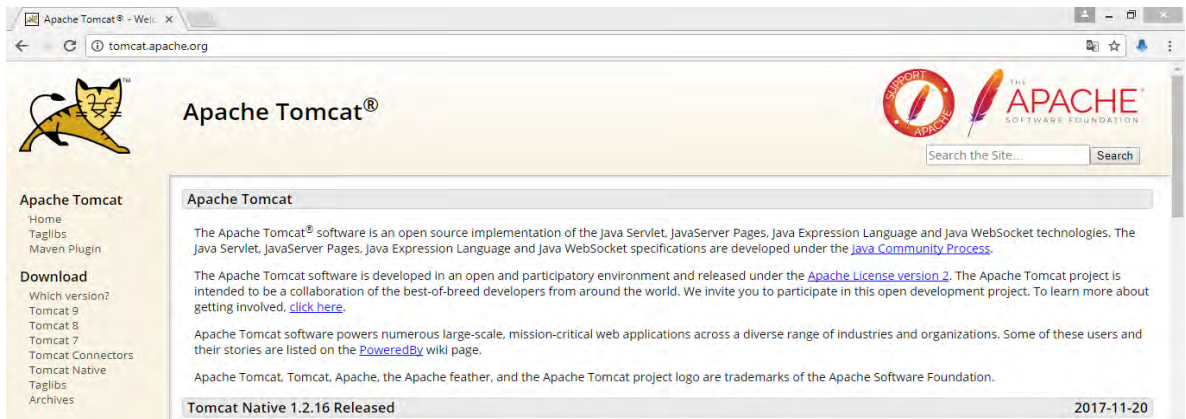
Ingresar a kunagi.org



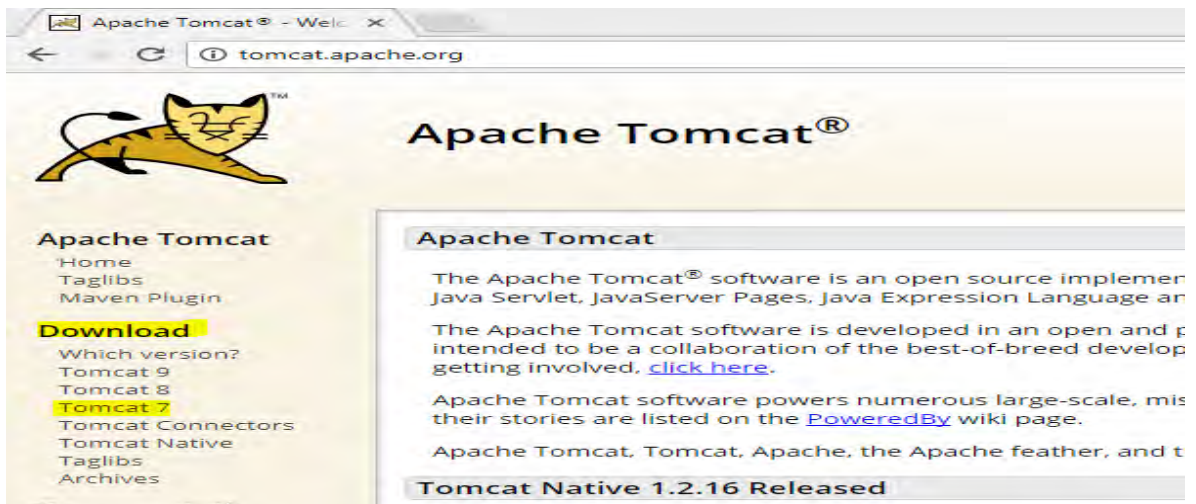
Ir a la sección de documentación para obtener las instrucciones de acuerdo con el Sistema operativo, en este caso Windows.



Para que kunagi funcione correctamente, primero debemos instalar Apache Tomcat, vamos a la página tomcat.apache.org.



Nos dirigimos a la sección de descargas y elegimos una versión, en este caso Tomcat 7.



Elegimos el repositorio.

Apache Tomcat®

Tomcat 7 Software Downloads

Welcome to the Apache Tomcat® 7.x software download page. This page provides download links for obtaining the latest version of Tomcat 7.0.x software, as well as links to the archives of older releases.

Quick Navigation

[KEYS](#) | [7.0.82](#) | [Browse](#) | [Archives](#)

Release Integrity

You **must verify** the integrity of the downloaded files. We provide OpenPGP signatures for every release file. This signature should be matched against the [KEYS](#) file which contains the OpenPGP keys of Tomcat's Release Managers. We also provide MD5 and SHA-1 checksums for every release file. After you download the file, you should calculate a checksum for your download, and make sure it is the same as ours.

Mirrors

You are currently using <http://apache.uniminuto.edu/>. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are *backup* mirrors (at the end of the mirrors list) that should be available.

Other mirrors: <http://apache.uniminuto.edu/>

7.0.82

Descargamos la versión para Windows de 32 y 64 bits.

Migration Guide
Presentations

Problems?

Security Reports
Find help
FAQ
Mailing Lists
Bug Database
IRC

Get Involved

Overview
Source code
Buildbot
Tools

Media

Twitter
YouTube
Blog

Misc

Who We Are
Heritage
Apache Home
Resources
Contact
Legal
Support Apache
Sponsorship
Thanks

Please see the [README](#) file for packaging information. It explains what every distribution contains.

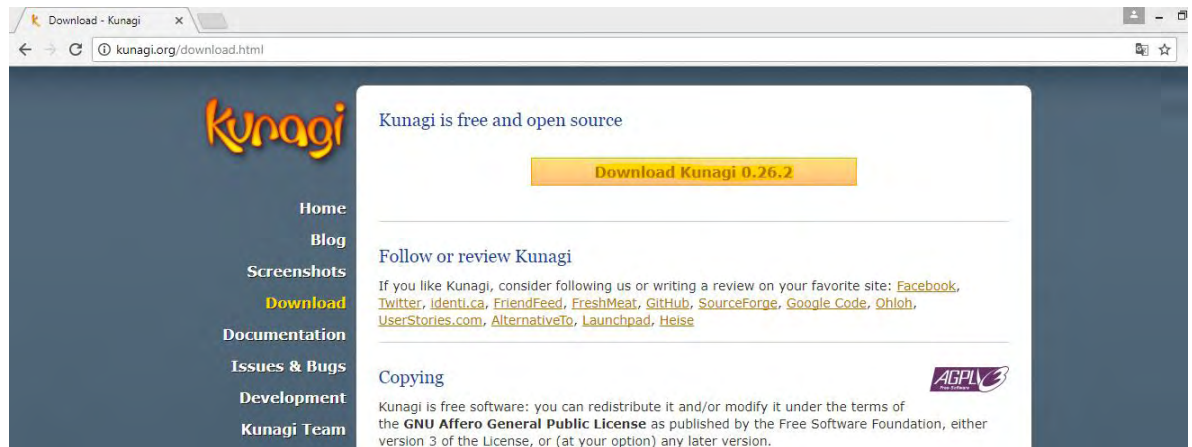
Binary Distributions

- Core:
 - [zip \(pgp, md5, sha1\)](#)
 - [tar.gz \(pgp, md5, sha1\)](#)
 - [32-bit Windows zip \(pgp, md5, sha1\)](#)
 - [64-bit Windows zip \(pgp, md5, sha1\)](#)
 - [32-bit/64-bit Windows Service Installer \(pgp, md5, sha1\)](#)
- Full documentation:
 - [tar.gz \(pgp, md5, sha1\)](#)
- Deployer:
 - [zip \(pgp, md5, sha1\)](#)
 - [tar.gz \(pgp, md5, sha1\)](#)
- Extras:
 - [JMX Remote jar \(pgp, md5, sha1\)](#)
 - [Web services jar \(pgp, md5, sha1\)](#)
 - [JULI adapters jar \(pgp, md5, sha1\)](#)
 - [JULI log4j jar \(pgp, md5, sha1\)](#)
- Embedded:
 - [tar.gz \(pgp, md5, sha1\)](#)
 - [zip \(pgp, md5, sha1\)](#)

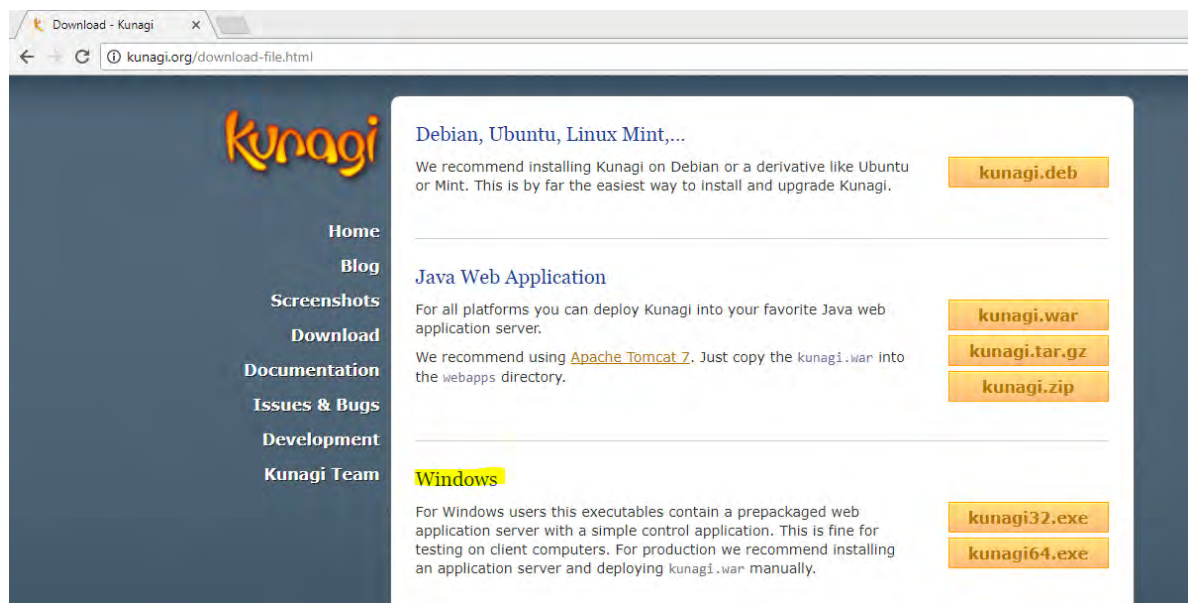
Source Code Distributions

- [tar.gz \(pgp, md5, sha1\)](#)
- [zip \(pgp, md5, sha1\)](#)

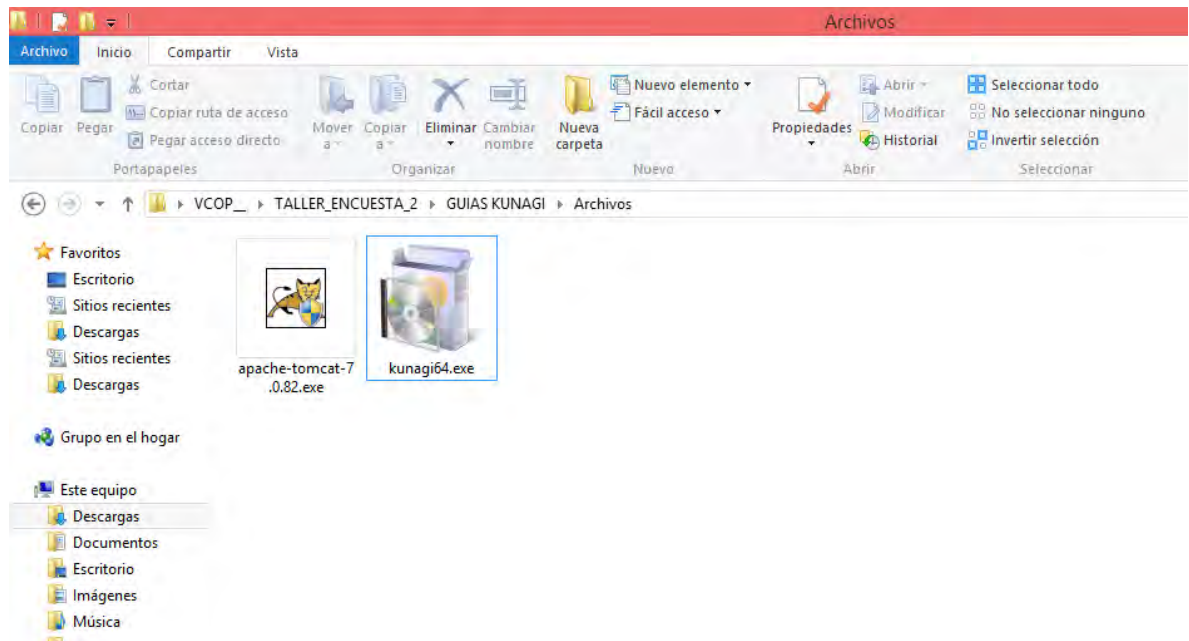
Ahora debemos descargar Kunagi, vamos a la página de Descargas y descargamos la versión más reciente.



Elegimos la versión de Windows compatible para nuestro computador.

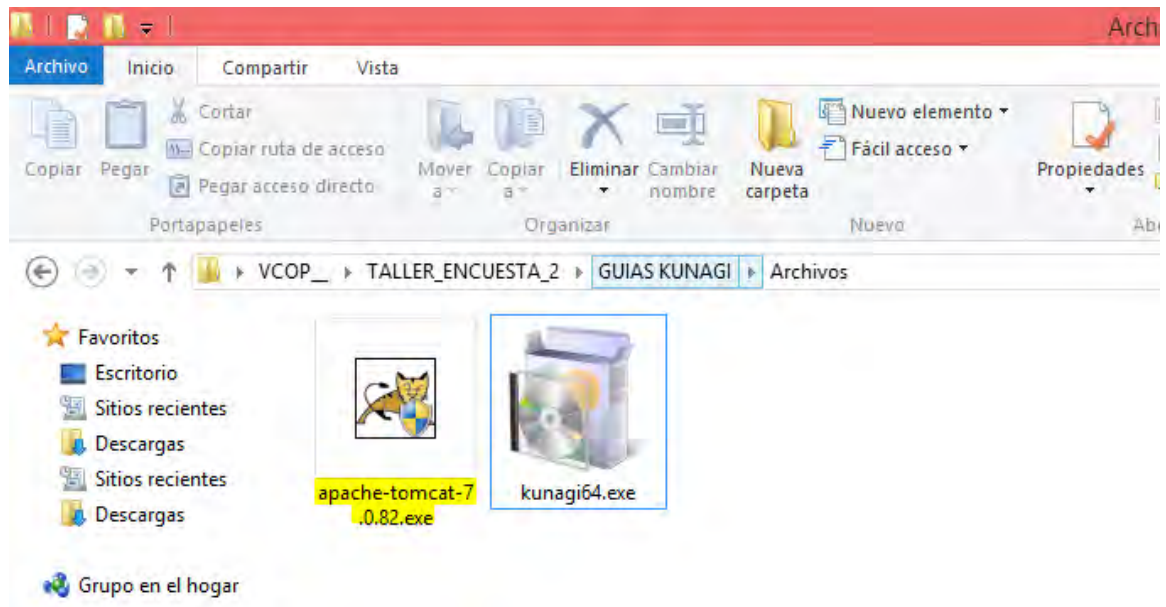


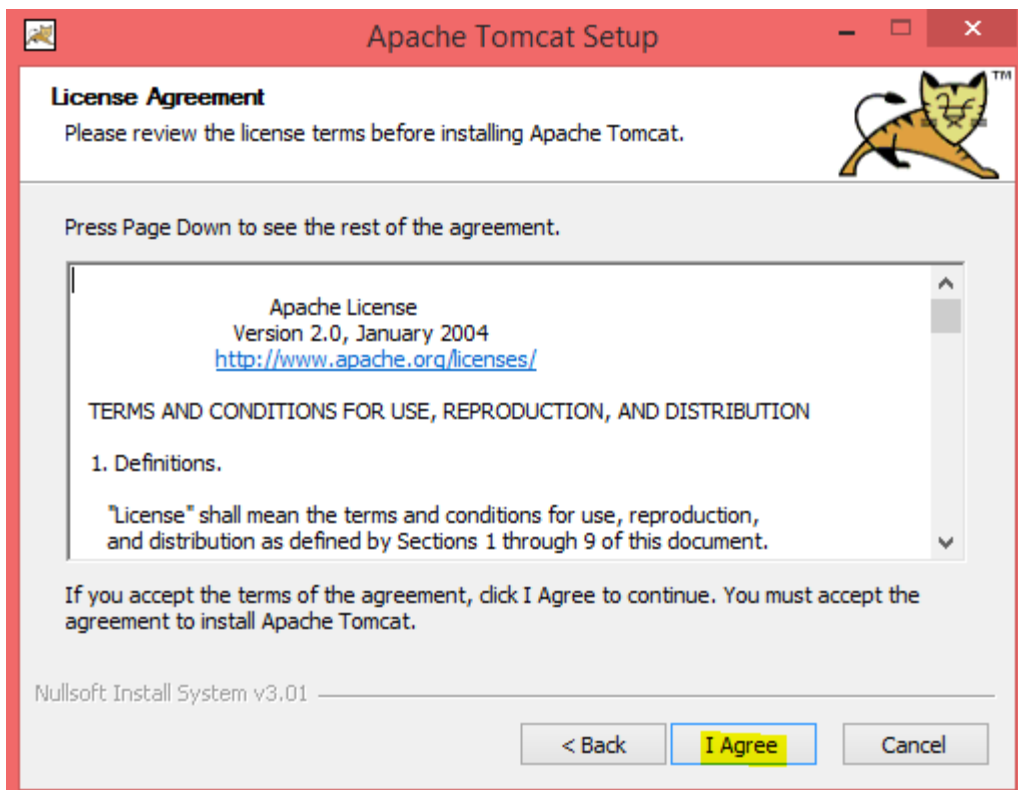
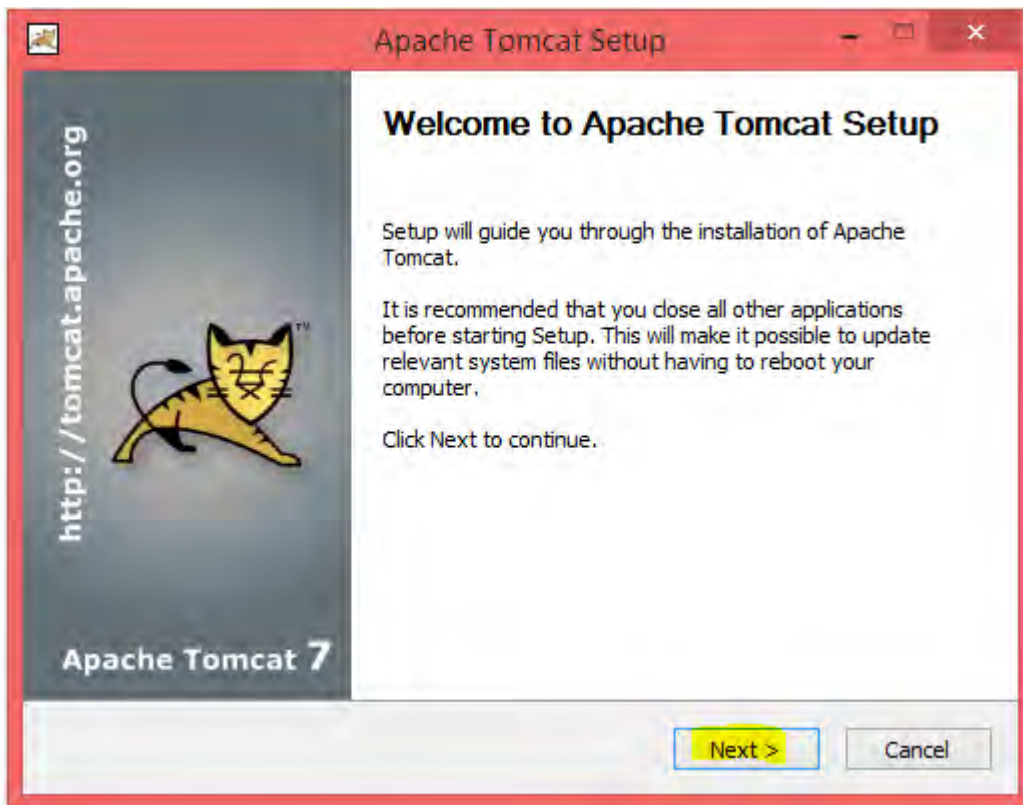
Una vez tengamos los archivos descargados, procedemos con la instalación.

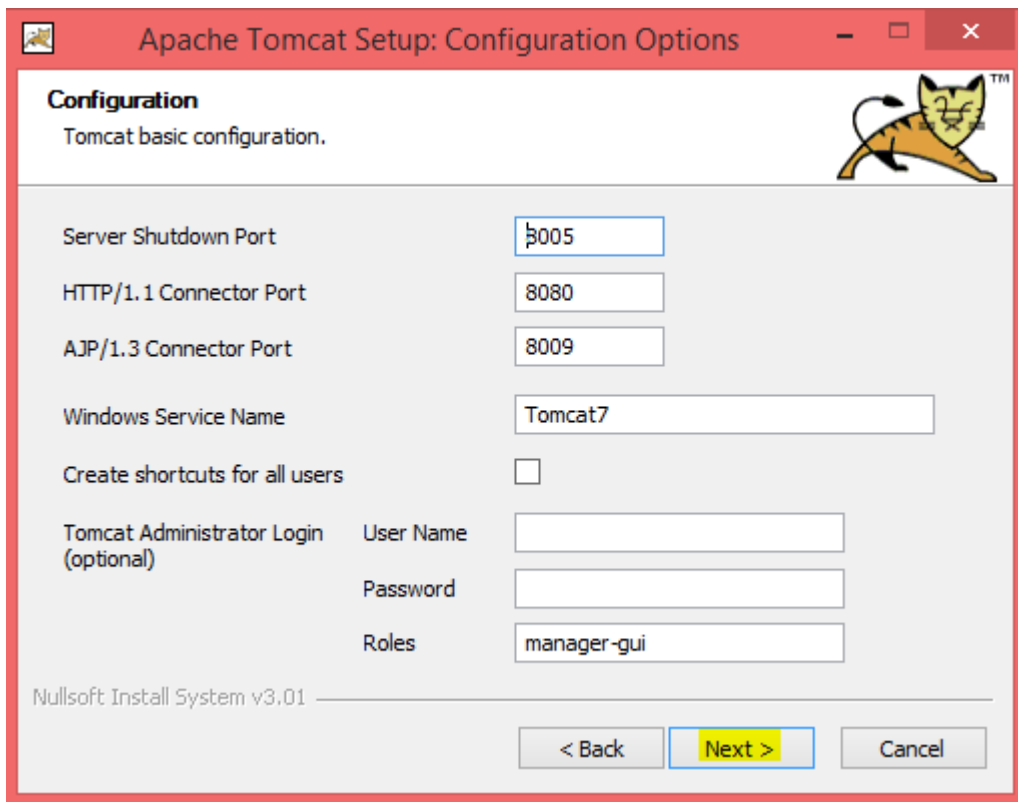
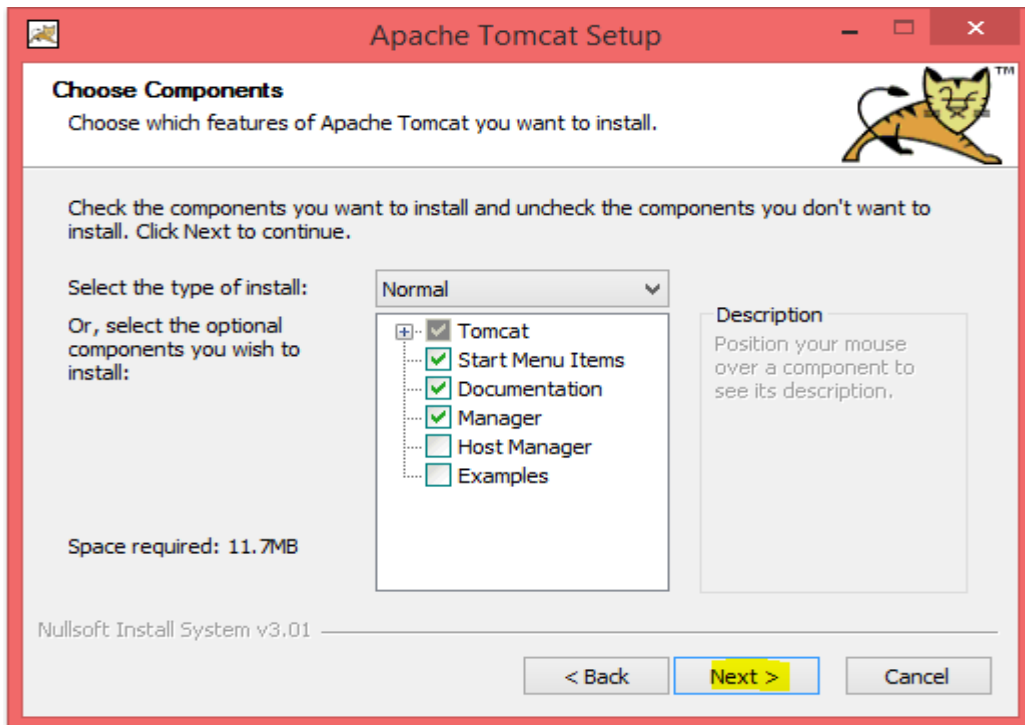


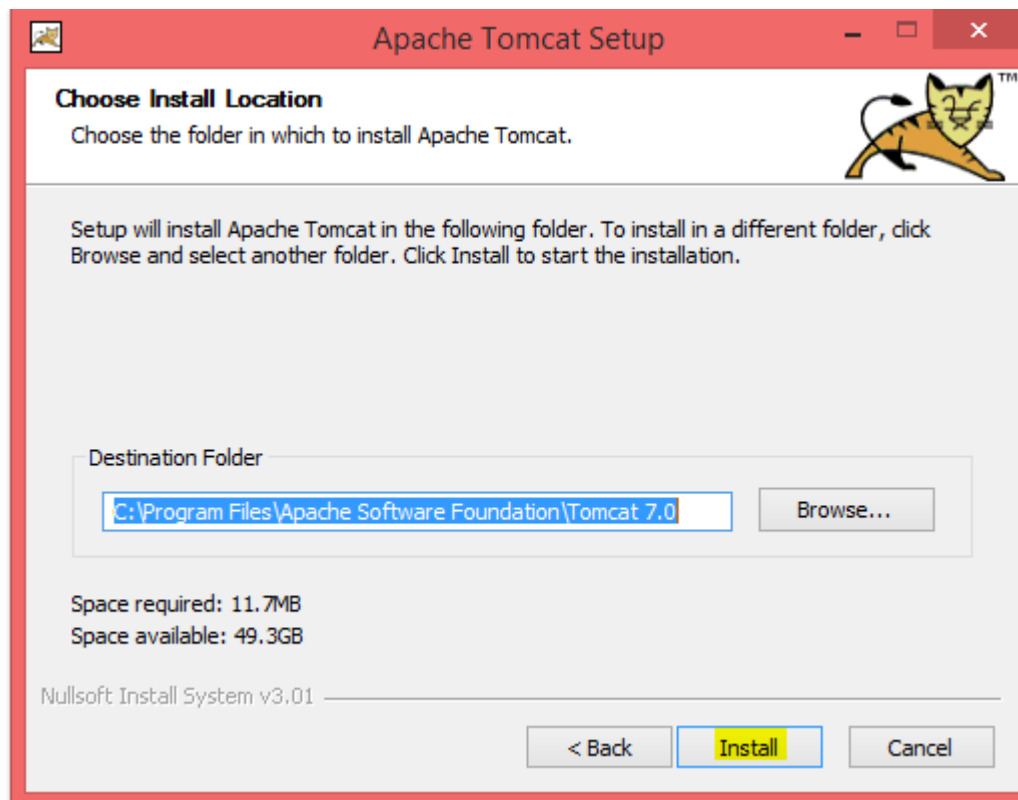
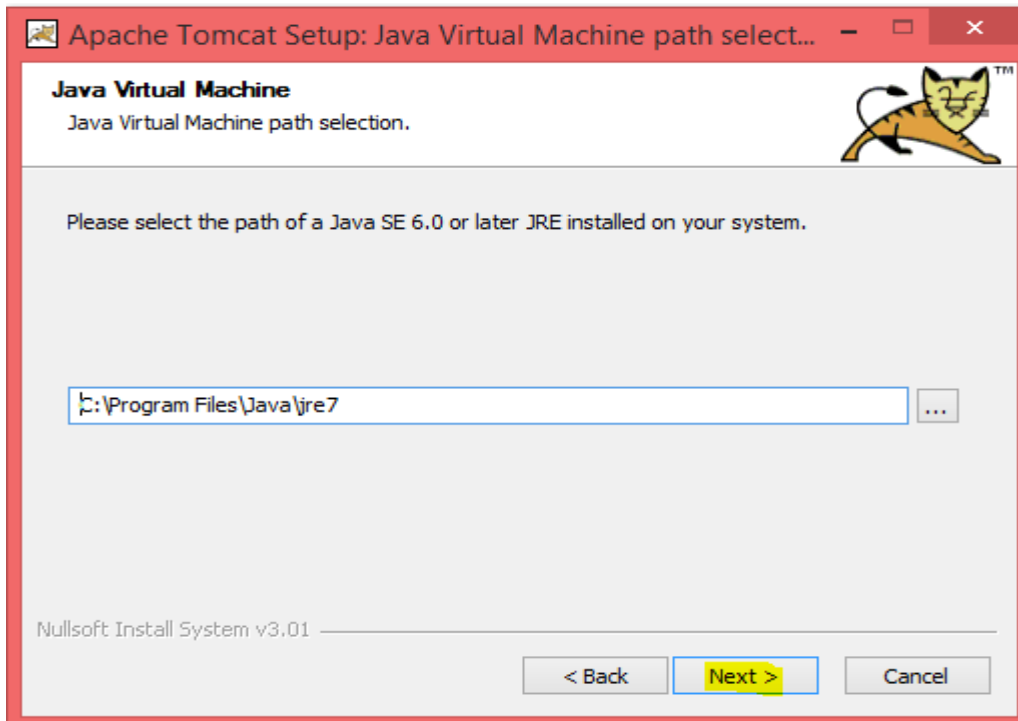
Instalación

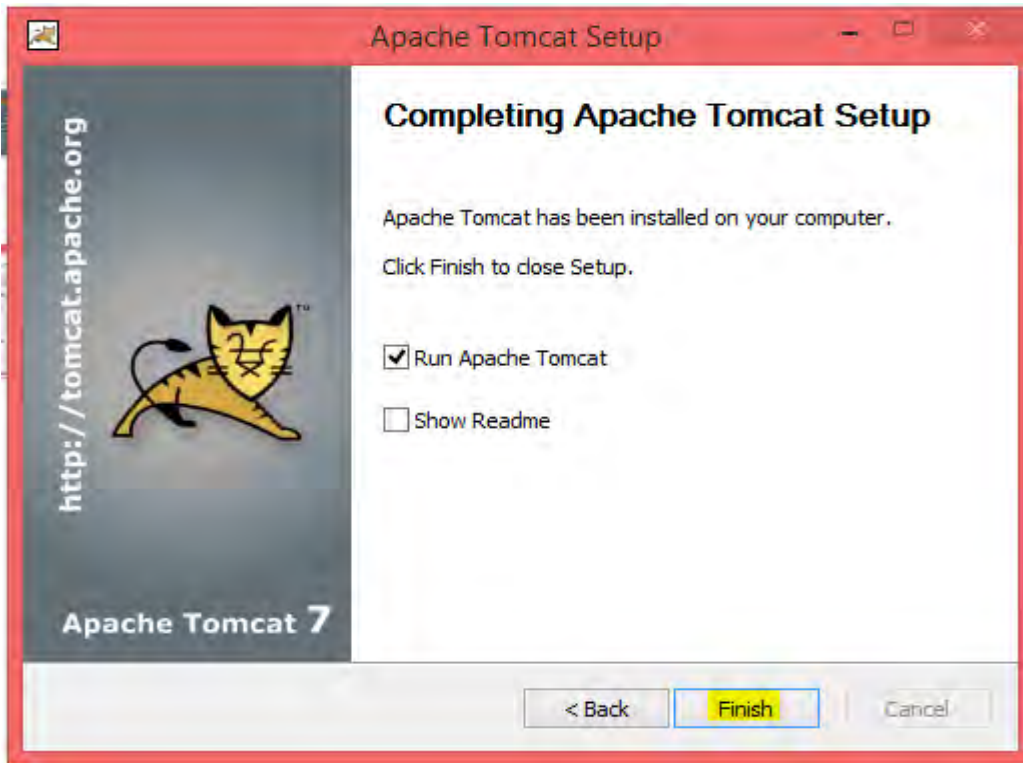
Instalamos Tomcat.



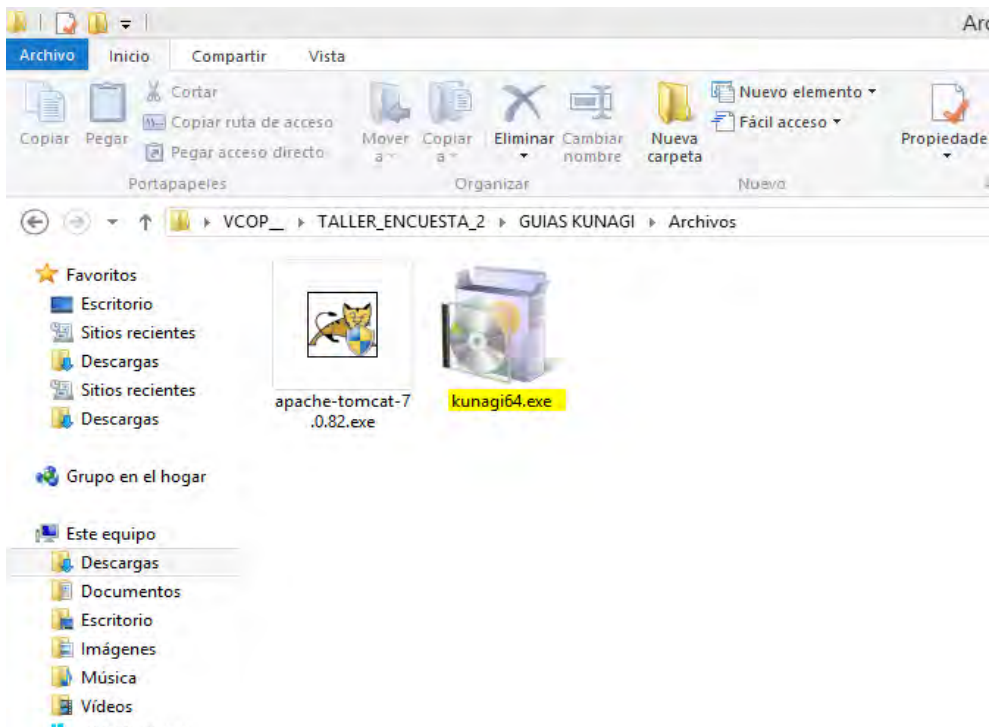




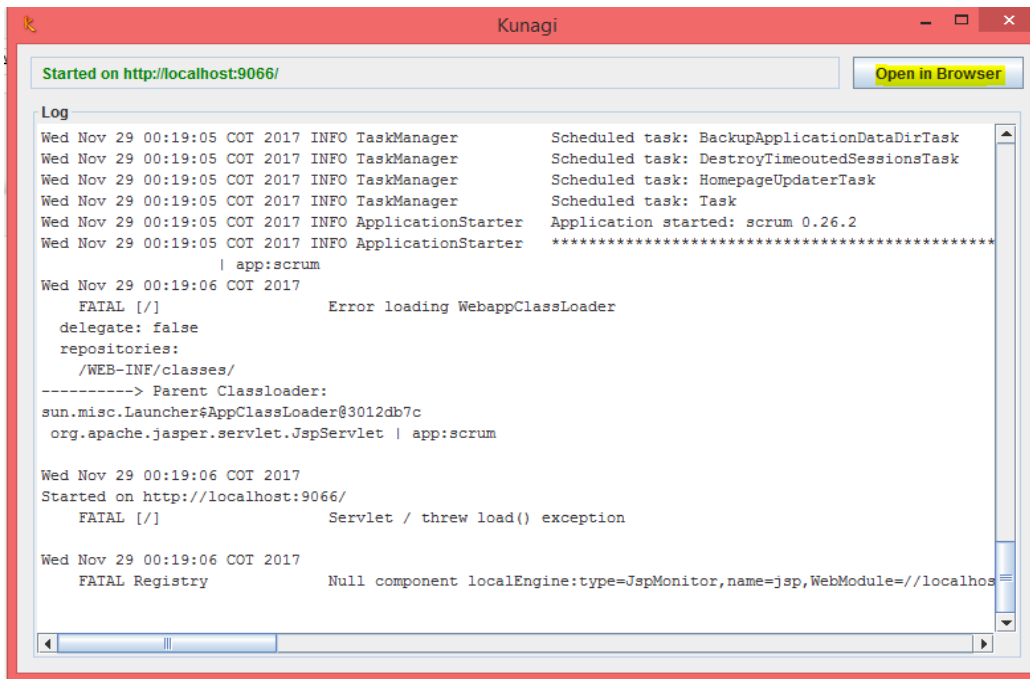




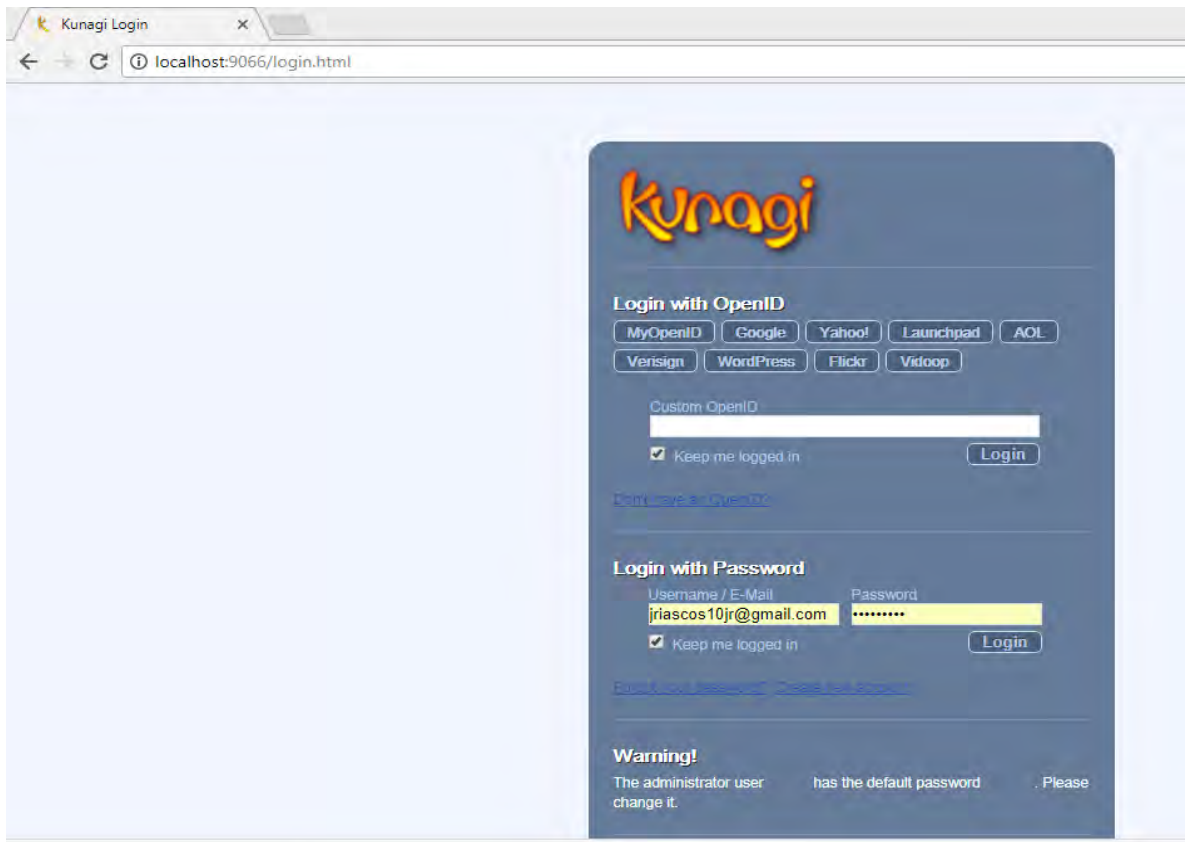
Ahora hacemos clic sobre el ejecutable de kunagi, esto tendremos que hacerlo cada vez que queramos utilizar kunagi ya que la herramienta no se instala.



La primera vez que ejecutemos kunagi tomara de 3 a 4 minutos en realizar la configuración, posteriormente hacemos clic en abrir en navegador.



Ahora kunagi está listo para ser utilizado.



Anexo I. Guía de uso Herramienta KUNAGI

The logo for Kunagi is written in a stylized, rounded font. The letters are filled with a gradient from orange to yellow, and each letter has a soft, white-to-yellow glow around it, giving it a 3D effect.

GUIA DE USO

Para crear un nuevo usuario hacemos clic en la opción create new account.



Diligenciamos los datos, solo es necesario un nombre de usuario y contraseña.



Create account

Username

E-Mail

Password

[Create account](#)

[Back to Login](#)

Kunagi 0.26.2 | [Kunagi.com](#)

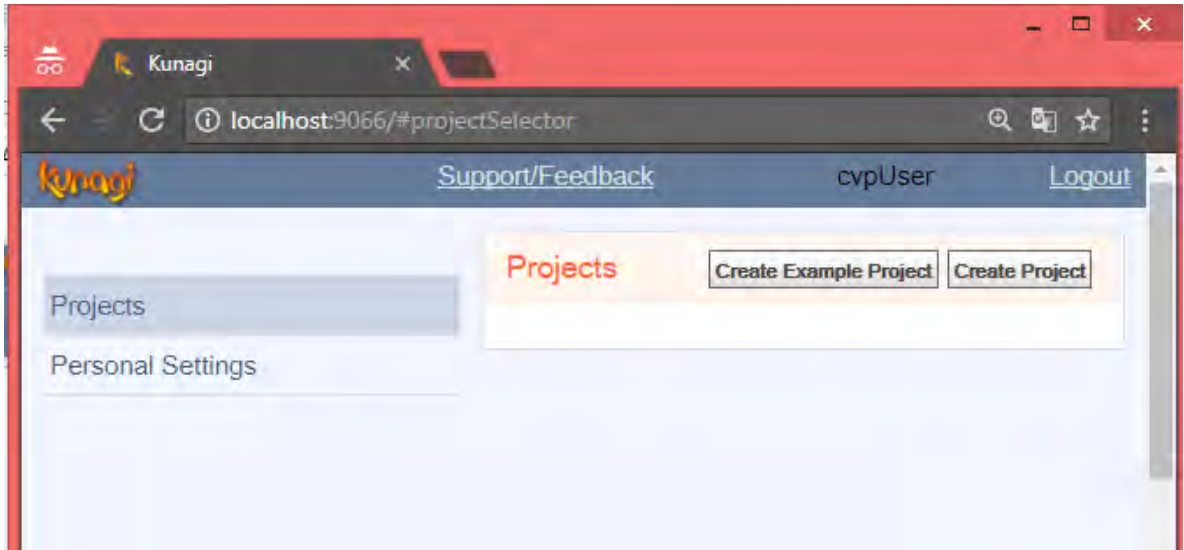
INICIO DE SESIÓN

Para iniciar sesión, simplemente ingresamos los datos del registro y clic en login.



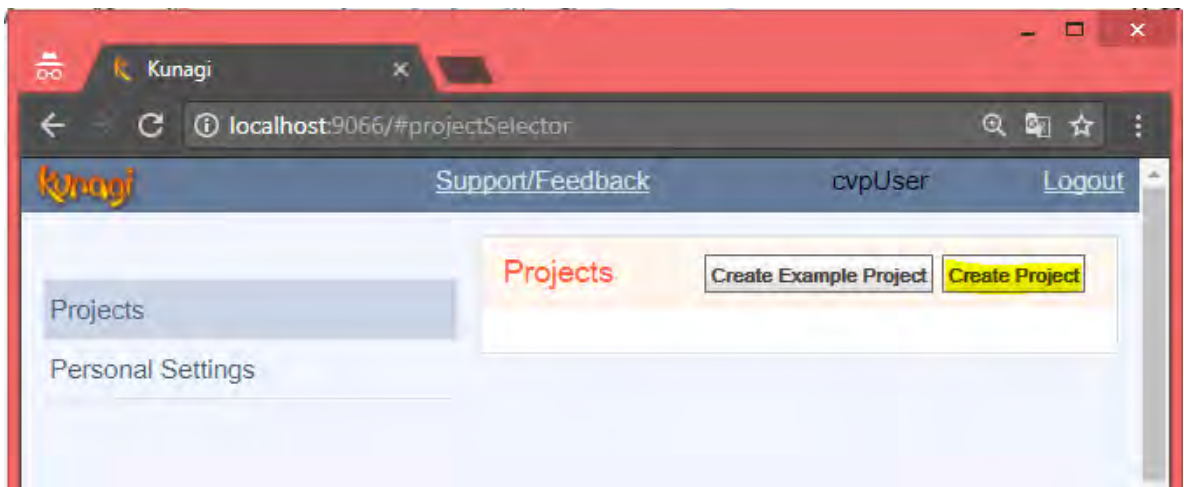
The image shows the login interface for Kunagi. At the top left is the Kunagi logo in a stylized orange and yellow font. Below the logo, there are two main login sections. The first section is titled "Login with OpenID" and contains several buttons for different OpenID providers: MyOpenID, Google, Yahoo!, Launchpad, AOL, Verisign, WordPress, Flickr, and Vidoop. Below these buttons is a text input field labeled "Custom OpenID". Underneath the input field is a checkbox labeled "Keep me logged in" which is checked, and a "Login" button. Below this section is a link that says "Don't have an account?". The second section is titled "Login with Password" and contains two input fields: "Username / E-Mail" with the text "cvpUser" and "Password" with masked characters "*****". Below these fields is another checkbox labeled "Keep me logged in" which is checked, and a "Login" button. At the bottom of this section are two links: "Forgot your password?" and "Create new account".

Accedemos a la primera interfaz de kunagi para gestionar proyectos.

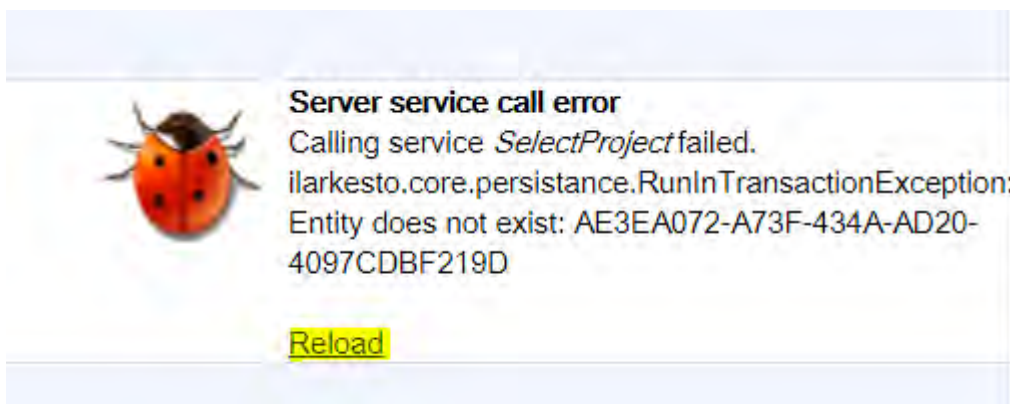


CREAR UN PROYECTO

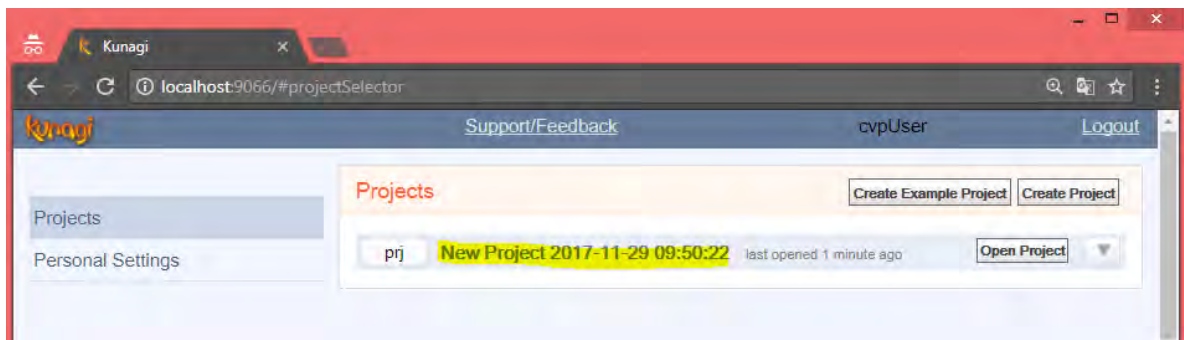
Hacemos clic en crear proyecto.



En ocasiones suele aparecer este error, simplemente hacemos clic en reload.

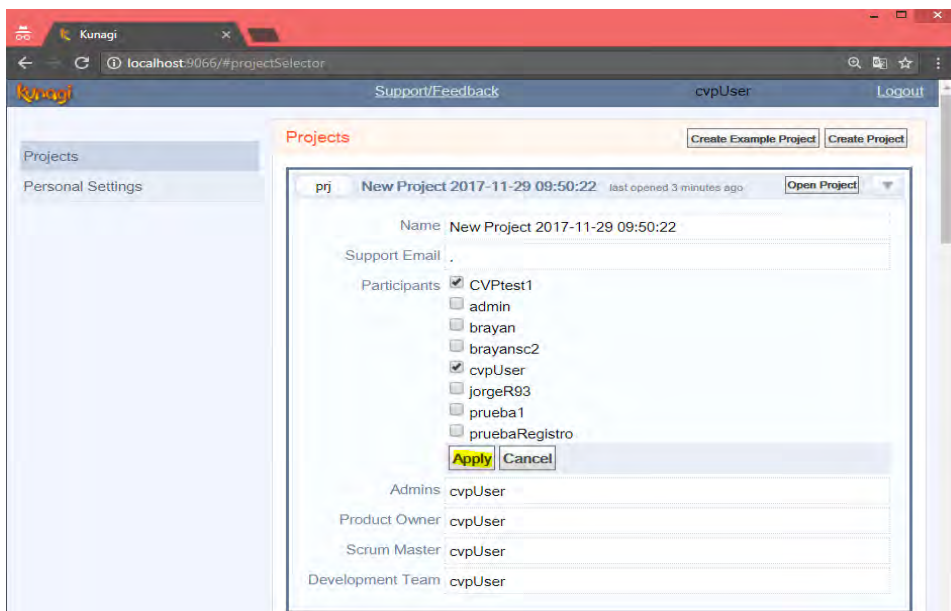


Podemos ver nuestro proyecto, listo para configurar.

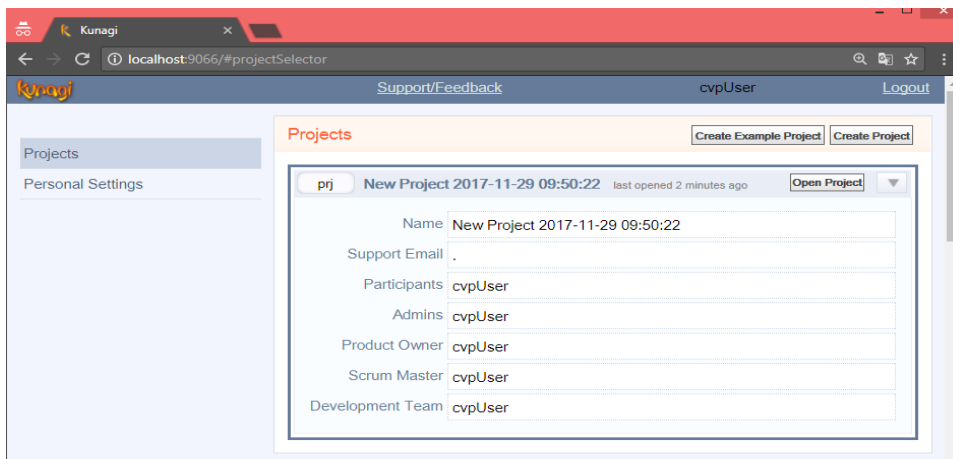


AGREGAR USUARIOS Y ROLES

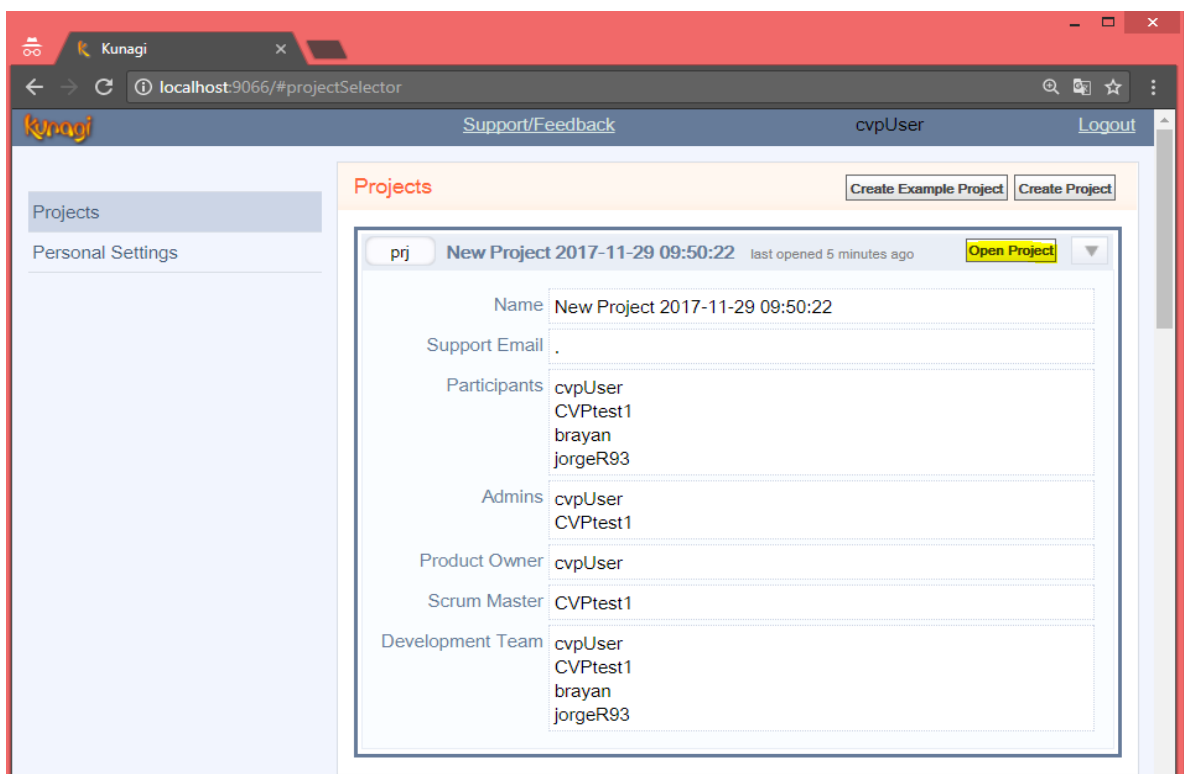
Primero debemos agregar a los participantes del proyecto, podremos elegir entre los usuarios que se han registrado previamente.

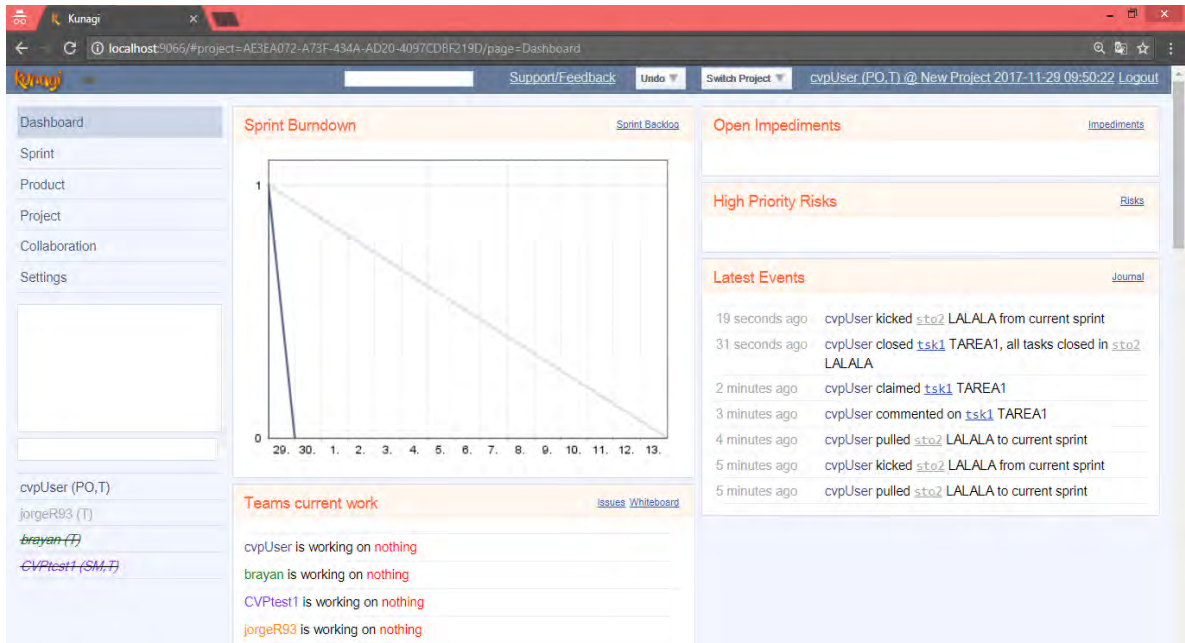


Elegimos los usuarios que harán parte del proyecto y clic en Apply.



Una vez elegidos los participantes del proyecto, elegimos a los administradores, al product owner, scrum master y a los miembros del development team. Para iniciar el proyecto hacemos clic en open Project.





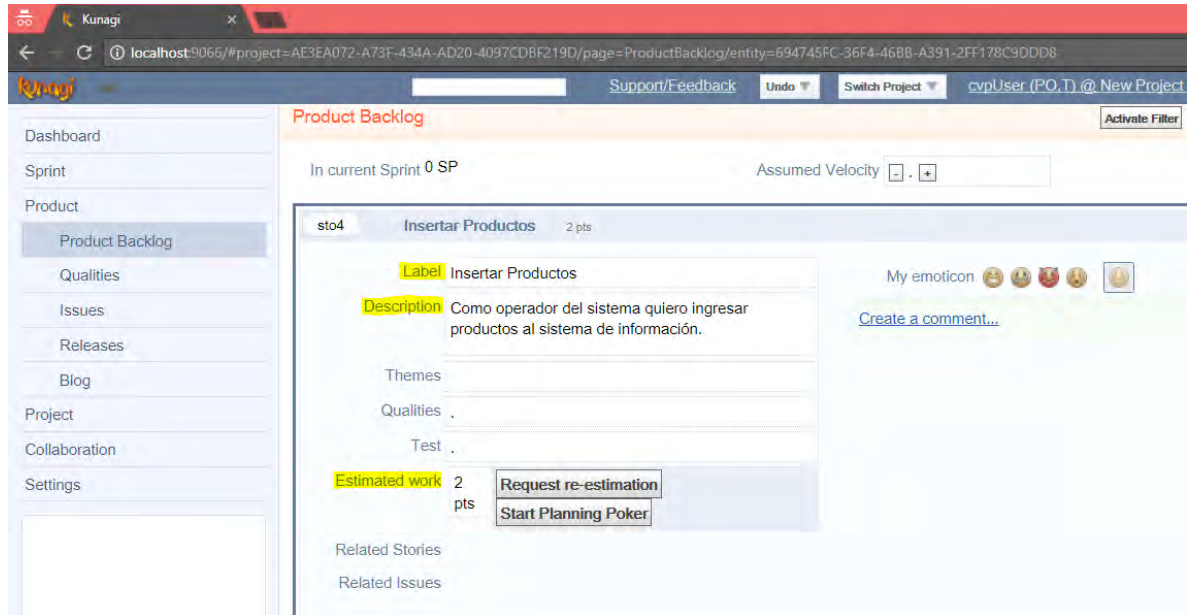
AGREGAR UNA HISTORIA DE USUARIO

Para crear una historia de usuario hacemos clic en create story.

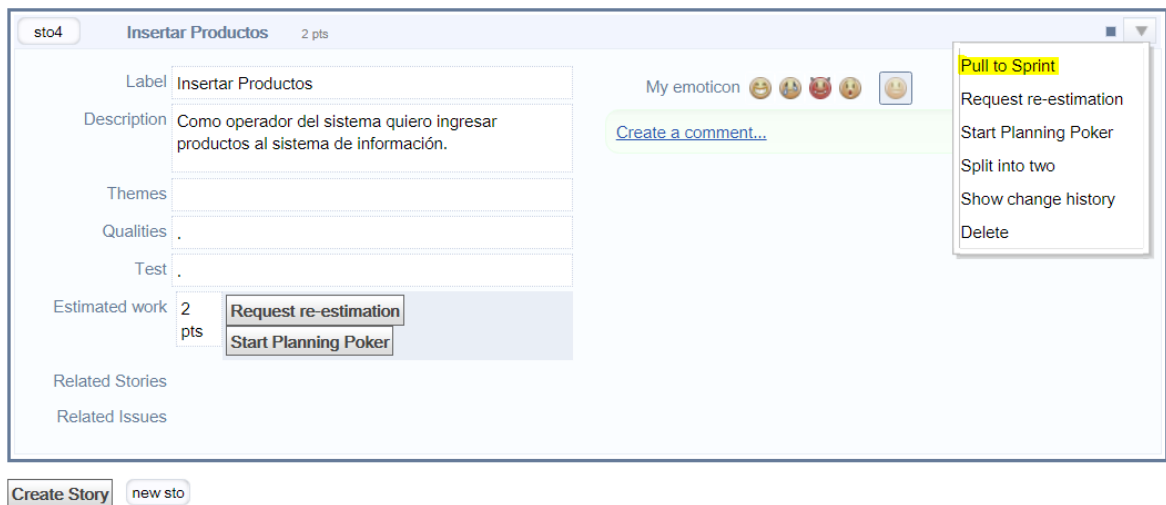
The screenshot shows the Kunagi Product Backlog page with the following components:

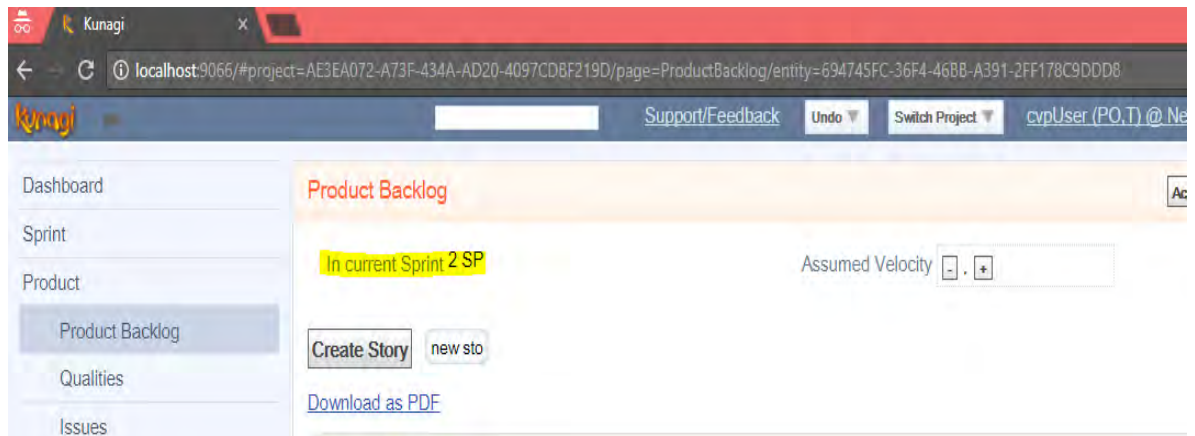
- Product Backlog:** The main section for managing the product backlog.
- In current Sprint:** 0 SP
- Assumed Velocity:** A control with minus and plus buttons.
- Create Story:** A prominent yellow button for creating new stories.
- new sto:** A smaller button for creating new stories.
- Download as PDF:** A link to download the backlog as a PDF.
- User Guide:** A section providing information about the Product Backlog:
 - The Product Backlog is a prioritized list of desired product features.
 - Product Owner creates and prioritizes Product Backlog**
 - The Product Owner is responsible for creating the Product Backlog and keeping it up to date. It contains Stories that describe desired product features. Furthermore, the Stories are prioritized, the more important the
 - presumably take to be possible, th and explanation.
 - not be changed
 - be present for e
 - the life descriptio

Ingresamos un título, una descripción de la historia de usuario incluyendo al actor y la funcionalidad a cumplir, y se hace una estimación de la carga de trabajo para el actual objetivo.



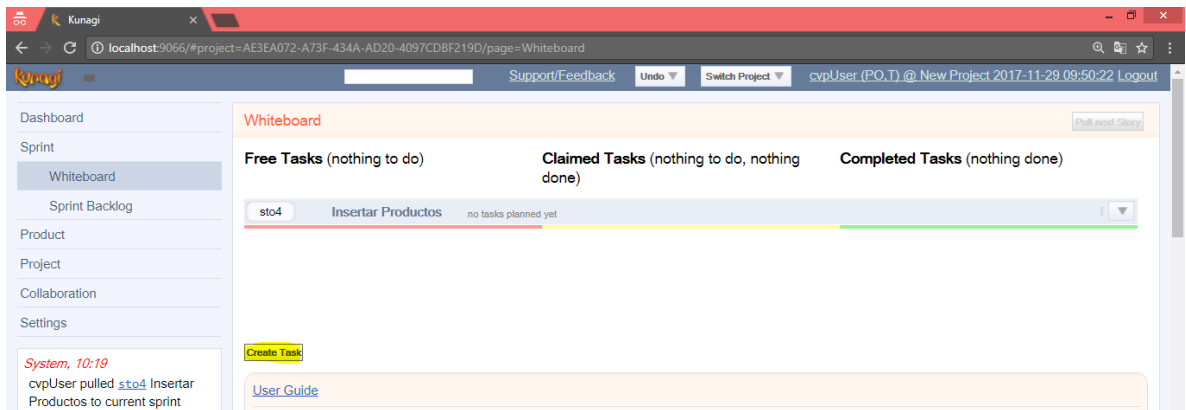
Cuando la historia de usuario este lista, podemos enviarla al Sprint.



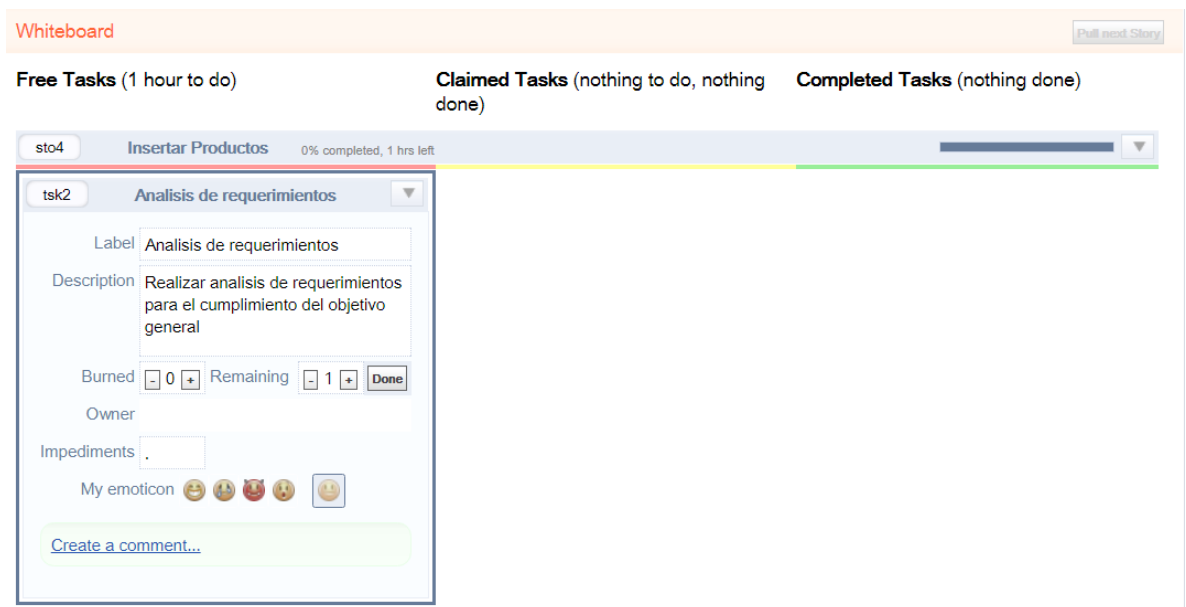


SPRINT Y TAREAS

En el whiteboard podremos administrar el sprint, las historias de usuario definidas, y además podemos agregar tareas a cada historia de usuario para cumplir con la funcionalidad. Para crear una tarea clic en create task.



Agregamos título, descripción y el tiempo requerido para realizar la actividad.



Podemos establecer el número de tareas que consideremos necesarias y podemos administrarlas en el tablero kanban.

Existen tres columnas Free tasks, claimed tasks, completed tasks correspondientes a tareas por realizar, Tareas en proceso, y tareas finalizadas respectivamente.

Whiteboard

Free Tasks (2 hours to do) Claimed Tasks (nothing to do, nothing done) Completed Tasks (nothing done)

sto4 Insertar Productos 0% completed, 2 hrs left

tsk2 Analisis de requerimientos

tsk3 Diseño de Base de Datos

Label: Diseño de Base de Datos

Description: Realizar diseño de Base de datos

Burned: 0 Remaining: 1 Done

Owner:

Impediments:

My emoticon: 😊 😐 😡 😞 😴

[Create a comment...](#)

cvpUser 0 millis ago

Responsible: Jorge Riascos

sto4 Insertar Productos 0% completed, 3 hrs left

tsk2 Analisis de requerimientos

tsk3 Diseño de Base de Datos

tsk4 Codificar

Label: Codificar

Description: Realizar la codificación del modulo de registro de productos

Burned: 0 Remaining: 1 Done

Owner:

Impediments:

My emoticon: 😊 😐 😡 😞 😴

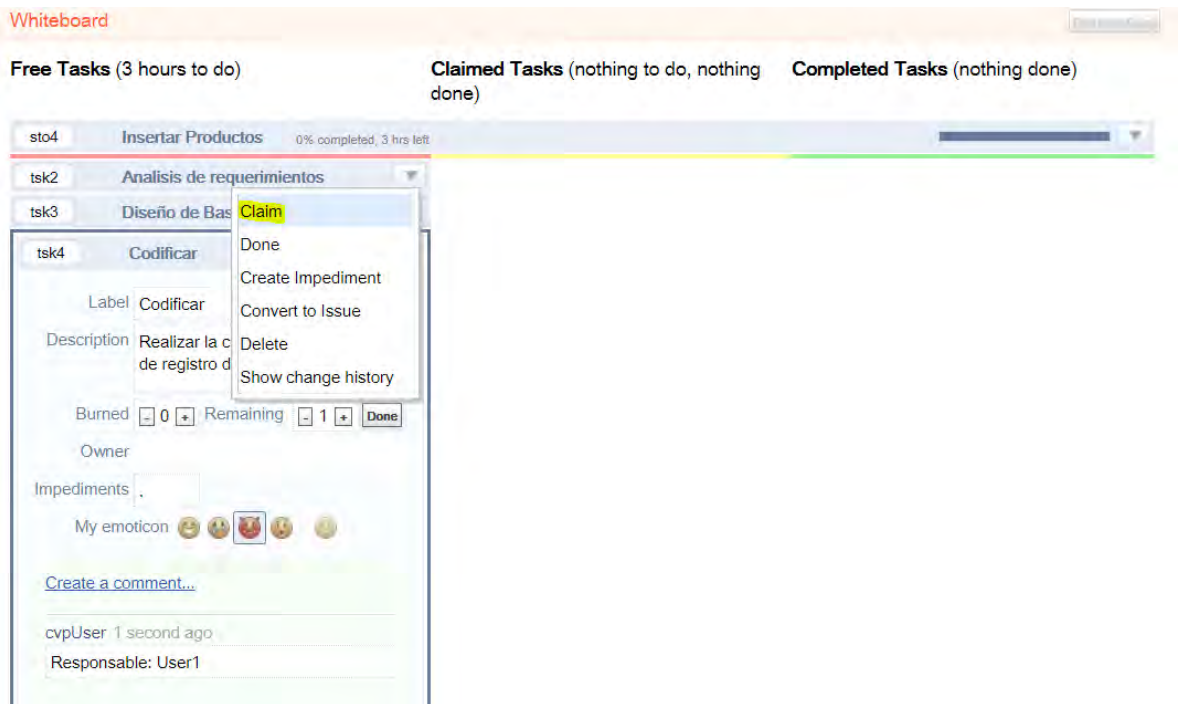
[Create a comment...](#)

cvpUser 1 second ago

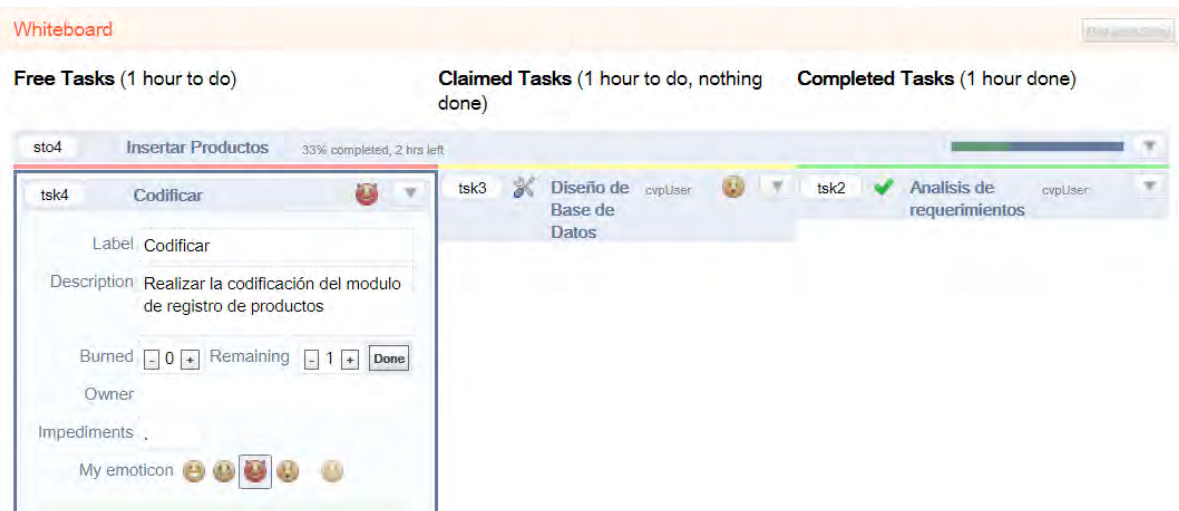
Responsible: User1

Create Task

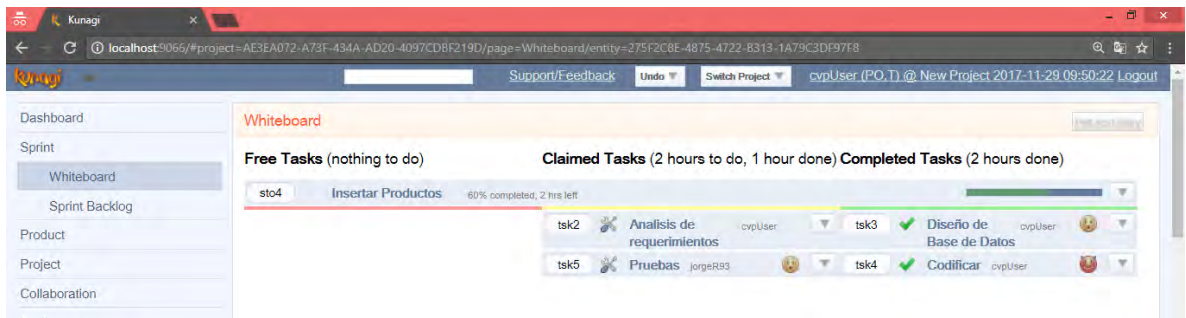
Para iniciar a trabajar una tarea hacemos clic sobre la flecha de cada tarea y hacemos clic en claim.



Si la actividad, ha sido finalizada hacemos clic en done.



Podemos observar el flujo de trabajo de todos los integrantes con tareas dentro del sprint.



Finalmente, en dashboard, podemos observar gráficas, eventos, entre otras características para evaluar el desempeño del equipo de desarrollo.

