



**“Aplicación de estrategias de desarrollo basado en pruebas e integración
continua en una aplicación web, para mejorar la calidad del software durante la etapa de
mantenimiento”**

Arias Alvarez, Nicole Noemi y Cueva Cuenca, Rodrigo Javier

Departamento de Ciencias de la Computación

Carrera de Ingeniería de Sistemas e Informática

Trabajo de titulación, previo a la obtención del título de Ingeniero en Sistemas e
Informática

Ing. Raura Ruiz, Jorge Geovanny

23 de mayo del 2021

Reporte de Similitud de Contenidos



Proyecto de Titulación Arias-Cueva.pdf

Scanned on: 19:53 August 15, 2022 UTC



Overall Similarity Score



Results Found



Total Words in Text

Identical Words	362
Words with Minor Changes	56
Paraphrased Words	59
Omitted Words	3374



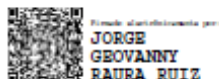
Departamento de Ciencias de la Computación

Carrera de Ingeniería de Sistemas e Informática

Certificación

Certifico que el trabajo de titulación, “Aplicación de estrategias de desarrollo basado en pruebas e integración continua en una aplicación web, para mejorar la calidad del software durante la etapa de mantenimiento”, fue realizado por los señores Arias Álvarez, Nicole Noemí y Cueva Cuenca, Rodrigo Javier; el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Sangolquí 10 de agosto del 2022



Ing. Raura Ruiz, Jorge Geovanny

C.C.: 0501773063





Departamento de Ciencias de la Computación
Carrera de Ingeniería de Sistemas e Informática

Responsabilidad de Autoría

Nosotros, Arias Álvarez, Nicole Noemi y Cueva Cuenca, Rodrigo Javier con cédulas de ciudadanía N° 1724005952 y 1719069237, declaramos que el contenido, ideas y criterios del trabajo de titulación: **“Aplicación de estrategias de desarrollo basado en pruebas e integración continua en una aplicación web, para mejorar la calidad del software durante la etapa de mantenimiento”** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí 10 de agosto del 2022


Arias Álvarez, Nicole Noemi
C.C.: 1724005952


Cueva Cuenca Rodrigo Javier
C.C.: 1719069237



Departamento de Ciencias de la Computación
Carrera de Ingeniería de Sistemas e Informática

Autorización De Publicación

Nosotros, Arias Álvarez, Nicole Noemi y Cueva Cuenca, Rodrigo Javier con cédulas de ciudadanía N° 1724005952 y 1719069237, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: Título: "Aplicación de estrategias de desarrollo basado en pruebas e integración continua en una aplicación web, para mejorar la calidad del software durante la etapa de mantenimiento" en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Sangolquí 10 de agosto del 2022

Arias Álvarez Nicole Noemi

C.C.: 1724005952

Cueva Cuenca Rodrigo Javier

C.C.: 1719069237

Dedicatoria

El presente proyecto de titulación va dedicado especialmente a mi madre, por su apoyo incondicional durante toda esta etapa, por impulsarme siempre a seguir adelante y alcanzar mi objetivo.

A mis hermanas que de una u otra forma han sido parte y testigos de los buenos y malos momentos que pase durante este viaje.

Para mis amigos más cercanos que siempre han estado pendientes.

Nicole Noemi Arias Álvarez

Para mi abuelita Pastora de Jesús Marín Cuenca que siempre represento lo más puro y bueno de la vida, que siempre estuvo para mantenerme y hacerme sentir feliz, querido, protegido e importante. Quien supo llenarme de alegría y buenos consejos, ella que fue y será el pilar central a lo largo de mi crecimiento personal y profesional.

Para mi madre Alba Cuenca, hermana Verónica Cueva y sobrinas Valentina y Victoria que han sido parte fundamental en mi vida y me han impulsado para alcanzar los objetivos que me he planteado, siendo ellas, una gran fuente de inspiración y compromiso.

Para mi tío Marco Cuenca que me ha brindado apoyo y que siempre ha estado pendiente por el bienestar de mí y de toda mi familia, siendo un gran ejemplo de persona altruista.

Para mis amigos y familiares que siempre han esperado que triunfe en lo que me propongo y que han ido aportando con un granito de arena para que pueda cumplir este objetivo tan importante.

Rodrigo Javier Cueva Cuenca

Agradecimiento

Agradezco a las personas que he conocido en este trayecto y que me han ayudado a formarme como profesional y mejorar como persona.

A los docentes que no solo han compartido sus conocimientos en una asignatura, sino que aportaron con experiencias de vida y consejos que me acompañaran siempre.

Al Ing. Geovanny Raura por su apoyo y guía en el proceso de titulación.

A mis compañeros, que desde que entre a la Universidad he conocido personas que se han convertido en amigos de vida, Bryan Coloma, Alejandra Punguil, Kevin Guayasamín y Luy Radrigan.

En especial a mi amiga Cristina Díaz por su apoyo incondicional y por compartir conmigo momentos buenos y malos que se han presentado durante este proceso.

A mi amigo Rodrigo Cuenca por la dedicación puesta en este trabajo y por los momentos compartidos desde que entramos a la universidad.

Y en general a las personas que, aunque han sido pasajeras, en su momento fueron mi apoyo para poder cumplir con esta meta.

Nicole Noemi Arias Álvarez

Agradezco al ingeniero Geovanny Raura que nos ha guiado de la manera más eficaz posible en este camino para culminar el proceso de titulación y que nos apoyado en todo lo necesario.

Agradezco a todos los profesores y maestros que han impartido sus conocimientos desde la escuela hasta la universidad, cada uno de ellos han sido importantes y me han dejado una enseñanza valiosa para afrontar la vida profesional.

Un agradecimiento especial para mi tía Germania Cuenca, que ha estado acompañándome en todo momento y que siempre ha tratado de aconsejarme y guiarme por un buen camino.

Agradezco a mi amiga Nicole Arias por su apoyo y trabajo en todo este ciclo y fase final para poder alcanzar el objetivo de ser profesionales.

Agradezco a la universidad y sus departamentos que han estado prestos y dispuestos a colaborar en las necesidades que se iban presentando en el desarrollo del presente trabajo.

Rodrigo Javier Cueva Cuenca

Índice de Contenidos

Reporte de Similitud de Contenidos	2
Dedicatoria.....	6
Agradecimiento.....	8
Índice de Contenidos	10
Índice de Tablas	13
Índice de Figuras	17
Resumen	20
Abstract	21
CAPÍTULO I: Introducción.....	22
Antecedentes.....	22
Planteamiento Del Problema	23
Justificación	24
Objetivos.....	26
Hipótesis.....	29
CAPÍTULO II: Marco Teórico	30
Metodología de desarrollo.....	30
Red de Categorías	33
Variables Dependientes	33
Ingeniería de software	33
Desarrollo de software	34

Verificación y Validación del Software.....	35
Pruebas de Software	36
Mantenimiento de Software	37
Integración Continua.....	38
Red de Categorías Variables Independientes	39
Calidad del Software.....	40
Aseguramiento de la Calidad del Software	40
Medición de la Calidad del Software	41
Calidad Externa y Productividad del Software.....	42
Estado Del Arte	45
Características del Estado del Arte	56
CAPÍTULO III: Análisis y diseño.....	57
Introducción.....	57
Metodología de desarrollo.....	57
Alcance.....	58
Personal involucrado.	59
Diagrama de Casos de Uso	63
Descripción de los Casos de Uso	64
Sprint Backlog.....	89
Diseño de la Base de Datos.....	94
Modelo Arquitectónico de Sistema GPI.....	98
Método de desarrollo para análisis de resultados:	99
Implementación de Integración Continua	101

CAPÍTULO IV: Desarrollo y Pruebas.....	102
Definición del Product Backlog.....	102
Desarrollo de Sprints.....	112
Desarrollo del Sprint 1	113
Entregables realizados	133
Desarrollo del Sprint 2	137
CAPÍTULO V: Validación y resultados	149
Introducción.....	149
Conclusiones	182
Recomendaciones	183
Bibliografía.....	185

Índice de Tablas

Tabla 1 Preguntas de Investigación	28
Tabla 2 Calidad Interna y Externa ISO/IEC 9126	43
Tabla 3 Criterios de inclusión y exclusión.	46
Tabla 4 Artículos que conforman el Grupo de Control.....	47
Tabla 5 Estudios Primarios	50
Tabla 6 Personal involucrado, miembro 1	59
Tabla 7 Personal involucrado, miembro 2	60
Tabla 8 Personal involucrado, miembro 3	60
Tabla 9 Personal involucrado, miembro 4	61
Tabla 10 Personal involucrado, miembro 5	61
Tabla 11 Personal involucrado, miembro 6	62
Tabla 12 Personal involucrado, miembro 7	62
Tabla 13 Caso de uso Gestionar Actividades.....	64
Tabla 14 Caso de uso Gestionar Escritura y Publicación de Libros.	66
Tabla 15 Caso de uso Gestionar Congresos.....	71

Tabla 16 Caso de Uso Evaluar Congresos	79
Tabla 17 Caso de Uso visualizar Reportes	86
Tabla 18 Lista de Features	102
Tabla 19 Tareas de cada feature	104
Tabla 20 Estimación y prioridad por historia de usuario HU1	114
Tabla 21 Estimación y prioridad por historia de usuario HU2	114
Tabla 22 Estimación y prioridad por historia de usuario HU3	115
Tabla 23 Estimación y prioridad por historia de usuario HU4	115
Tabla 24 Estimación y prioridad por historia de usuario HU5	116
Tabla 25 Estimación y prioridad por historia de usuario HU6	116
Tabla 26 Estimación y prioridad por historia de usuario HU7	117
Tabla 27 Estimación y prioridad por historia de usuario HU8	117
Tabla 28 Estimación y prioridad por historia de usuario HU9	118
Tabla 29 Estimación y prioridad por historia de usuario HU10	119
Tabla 30 Estimación y prioridad por historia de usuario HU11	120
Tabla 31 Estimación y prioridad por historia de usuario HU12	120

Tabla 32 Estimación y prioridad por historia de usuario HU13	121
Tabla 33 Estimación y prioridad por historia de usuario HU14	121
Tabla 34 Estimación y prioridad por historia de usuario HU15	122
Tabla 35 Estimación y prioridad por historia de usuario HU16	122
Tabla 36 Estimación y prioridad por historia de usuario HU17	123
Tabla 37 Estimación y prioridad por historia de usuario HU18	123
Tabla 38 Estimación y prioridad por historia de usuario HU19	124
Tabla 39 Estimación y prioridad por historia de usuario HU20	124
Tabla 40 Estimación y prioridad por historia de usuario HU21	125
Tabla 41 Sprint Backlog S1.....	126
Tabla 42 Ajustes del Sprint	132
Tabla 43 Features Sprint 2.....	137
Tabla 44 Features Sprint 3.....	146
Tabla 45 Particiones de Equivalencia de Congresos	151
Tabla 46 Escenarios de casos de prueba	161
Tabla 47 Caso de Prueba Congreso P1.....	162

Tabla 48 Caso de prueba Libro P1	169
Tabla 49 Resultados Casos de Prueba.....	176

Índice de Figuras

Figura 1 Metodología de desarrollo tradicional.....	30
Figura 2 Metodología de desarrollo con TDD e integración continua	30
Figura 3 Cascada de la red de categoría variable dependiente.....	33
Figura 4 Escenario típico de Integración Continua.....	39
Figura 5 Cascada de la Red de categoría Variable Independiente.....	39
Figura 6 Diagrama de casos de uso.....	63
Figura 7 Sprint Backlog del Desarrollo del Módulo.....	89
Figura 8 Historias de Usuario Sprint 1 parte 1.....	90
Figura 9 Historias de Usuario Sprint 1 parte 2.....	90
Figura 10 Historias de Usuario Sprint 2 parte 1.....	91
Figura 11 Historias de Usuario Sprint 2 parte 2.....	92
Figura 12 Historias de Usuario Sprint 2 parte 3.....	92
Figura 13 Historias de Usuario Sprint 3.....	93
Figura 14 Modelo Conceptual de la BD.....	95
Figura 15 Modelo Lógico de la BD.....	96

Figura 16 Figura Modelo Físico de la BD	97
Figura 17 Modelo arquitectónico del sistema GPI	98
Figura 18 Plan de desarrollo parte 1	99
Figura 19 Plan de desarrollo parte 2	100
Figura 20 Arquitectura de Integración Continua de la UFA “ESPE”	101
Figura 21 Historias de Usuario realizadas con el Método Tradicional	113
Figura 22 Historias de Usuario TDD y CI	118
Figura 23 Visualización de Gestionar Planificación 1	133
Figura 24 Visualización de Gestionar Planificación 2	134
Figura 25 Visualización de Gestionar Publicaciones 1	135
Figura 26 Visualización de Gestionar Publicaciones 2	135
Figura 27 Visualización de Gestionar Libros 1	136
Figura 28 Visualización de Gestionar Libros 2	137
Figura 29 Integración Continua del módulo.....	149
Figura 30 Comparación Casos de Prueba Fase 1.....	159
Figura 31 Comparación Casos de Prueba Fase 2.....	160

Figura 32 BurnDown Chart Sprint 1 178

Figura 33 BurnDown Chart Sprint 2 180

Resumen

El desarrollo basado en pruebas TDD junto con la Integración continua son técnicas utilizadas para la construcción de software, que buscan mejorar la calidad del código y la productividad de los programadores. Existen diversos estudios que buscan analizar la efectividad de estas técnicas, sin embargo, los resultados encontrados son ambiguos, en unos casos muestran que se mejora la calidad y productividad y en otros, se encuentra que retrasan la entrega de proyectos y aumenta el uso de recursos. Por otro lado, existe poca literatura que analiza la aplicación de TDD y CI durante la etapa de mantenimiento del software.

El presente trabajo busca aplicar tanto TDD como CI en el desarrollo de un módulo de software para determinar si las técnicas mencionadas contribuyen a mejorar la calidad externa y la productividad durante la etapa de mantenimiento.

Como metodología se aplicó un estudio de caso que consistió en el desarrollo de un módulo para el sistema denominado GPI de la Universidad de las Fuerzas Armadas "ESPE". Además, se utilizó la metodología SCRUM aplicando en cada iteración el desarrollo tradicional y las técnicas TDD y CI de manera alternada.

Como resultado se obtuvo que las historias de usuario desarrolladas utilizando TDD y CI arrojaron más casos de prueba exitosos, también se observó que el dominio en la aplicación de estas herramientas mejora la productividad del software y calidad externa del software en la etapa de mantenimiento.

Palabras Clave: Integración Continua, Desarrollo basado en pruebas, Calidad del Software, Productividad.

Abstract

Test-driven development TDD together with Continuous Integration are techniques used to build software, which seek to improve code quality and the productivity of programmers. There are several studies that seek to analyze the effectiveness of these techniques, however, the results found are ambiguous, in some cases they show that quality and productivity are improved and in other cases it is found that they delay the delivery of projects and increase the use of resources. On the other hand, there is little literature that analyzes the application of TDD and CI during the software maintenance stage.

The present work seeks to apply both TDD and CI in the development of a software module to determine if the techniques contribute to improving external quality and productivity during the maintenance stage.

As a methodology, a case study was applied that consisted of the development a module part of a system called GPI of the Universidad de las Fuerzas Armadas "ESPE". In addition, the SCRUM methodology was used, applying the traditional development and the TDD and CI techniques alternately in each iteration.

As a result, it was obtained that the user stories developed using TDD and CI yielded more successful test cases, it was also observed that the expertise in the application of these tools improves the productivity and the external quality of the software in the maintenance stage.

Keys Words: Continuous Integration, Test-Driven-Development, Software Quality, Productivity.

CAPÍTULO I: Introducción

Antecedentes

En la actualidad las organizaciones necesitan mejorar continuamente sus sistemas informáticos, ya sea para adaptarse a la demanda tecnológica a la que nos enfrentamos día a día o para cubrir necesidades y requerimientos de las empresas.

Cuando un software se somete a cambios constantes se pueden presentar dificultades durante la etapa de mantenimiento, reduciendo su calidad externa, la misma que es definida por la ISO/IEC 9126 como la evaluación del comportamiento del sistema en un entorno determinado y en su fase final del ciclo de vida de desarrollo, obteniendo como resultado la funcionalidad esperada y la satisfacción del usuario en cuanto a los requerimientos establecidos.

Uno de los inconvenientes presentados durante el mantenimiento del software es la disminución de la productividad, según (Vasallo, 2019) la Integración Continua permite mejorar este aspecto al momento del desarrollo y asegurar su calidad interna y externa.

(Mojtaba Shahin, 2017) define la Integración Continua como prácticas o técnicas de la industria del desarrollo que permiten a las organizaciones lanzar con frecuencia y de forma fiable nuevas funciones y productos, pero esta implementación no es fácil ya que requiere de conocimientos sólidos, comprensión total de cómo el software debe ser rediseñado para obtener los resultados esperados.

Por último, (Roldan 2011), dice que para asegurar la calidad del software existen metodologías de desarrollo ágil como TDD (Test-Driven Development) que consiste en escribir primero las pruebas, refactorizar e implementar el código. El objetivo es minimizar los errores, obtener la funcionalidad justa que el usuario necesita y producir un software modular altamente reutilizable y preparado para el cambio.

Actualmente, por la demanda tecnológica, tenemos que adaptarnos a los cambios y buscar técnicas efectivas que nos ayuden a mitigar errores y reducir riesgos en los sistemas informáticos. En el presente estudio se pretende combinar las técnicas de TDD y CI con la finalidad de mejorar la calidad externa y la productividad del software en la etapa de mantenimiento, es decir que los cambios y nuevas funciones definidas por el usuario se adapten al sistema y de esta manera se encuentre siempre operativo habiendo cumplido los requerimientos satisfactoriamente.

Planteamiento Del Problema

Según (Lai, S.-T.,2019), el desarrollo de las aplicaciones de software generalmente se enfrenta a cambios, innovaciones e integraciones de nuevas funcionalidades, por lo tanto, las aplicaciones deben mitigar los errores que se presentan en el mantenimiento del sistema informático.

La integración de sistemas según (Rivadeneira, 2012), representa la unión o modificación de los distintos módulos que integran el software. Este proceso abarca varios eventos como la corrección de errores, revisión de la especificación de requisitos, evolución del

entorno, recursos y tecnologías. Estos cambios en el sistema originan también modificaciones en el plan y ejecución de un proyecto, lo que ocasiona que los documentos y el código fuente no se pueden revisar, probar e integrar de manera efectiva, aumentando el riesgo de introducir defectos en los aplicativos.

(Torres, et al 2017) menciona que Integrar sistemas, puede causar efectos adversos que disminuyen la calidad del software en general, y de manera particular en aplicaciones con tecnología web, la que más desarrollo ha tenido en los últimos años y en donde la necesidad de integración ha sido una constante.

Según (M.M. Lehman, 2002), el mantenimiento de software es realmente un desarrollo evolutivo y las decisiones que se toman están basadas en la comprensión de lo que sucede en el sistema a lo largo del tiempo y cómo van evolucionando constantemente. A medida que evolucionan, se vuelven más complejos a menos que se tomen algunas medidas como la refactorización del código para reducir su complejidad.

El mantenimiento del software se ha representado tradicionalmente con la figura de un iceberg, la parte visible corresponde a las actividades de desarrollo que tienen un costo conocido y previsible: análisis, diseño e implementación. La sección sumergida, que puede llegar a ser el 80% del total, conforma el mantenimiento del sistema, donde se oculta una gran cantidad de problemas que tienen asociados un costo por todos los recursos implicados.

Justificación

La etapa de mantenimiento de un software abarca entre un 60 a 80% el ciclo de vida de desarrollo, donde se encuentra gran cantidad de problemas, ya sea, en cuanto al rendimiento del sistema, el tiempo empleado por los desarrolladores el cual puede extenderse y utilizar más

recursos de lo previsto y el gasto que en algunos casos es elevado y no cumple con la relación costo-beneficio, dando como resultado un cliente insatisfecho.

(Gallaba, 2019) menciona que las organizaciones deben mantener un ritmo rápido y adaptarse a los cambios que se presentan en sus sistemas, para que de esta manera se pueda construir, probar y lanzar mejoras que solventen los nuevos requerimientos.

Se han realizado varios estudios que buscan disminuir los errores y mitigar el riesgo de fallo de un sistema, por ejemplo, (Gustavo Sizilio, 2019) realizó un estudio en la universidad de Otago, Nueva Zelanda, consistió en evaluar los resultados de 82 proyectos implementando CI (Integración Continua) y la misma cantidad con NOCI (Sin Integración Continua).

Se determinó que en el 40.2% de la muestra que usó CI tuvo los resultados esperados y solo el 17.2% con NOCI llegaron a la funcionalidad correcta, en conclusión, nueve de cada diez proyectos en los que se implementa CI culminan el sistema exitosamente, mientras que, cinco de cada diez proyectos con NOCI no lo hacen.

También se han realizado estudios de la metodología de desarrollo ágil Test Driven Development (TDD), según (Beck, y otros, 2001), TDD es una técnica iterativa de diseño de software orientado a objetos que forman parte de la metodología XP (Programación Extrema), la cual fue definida como su núcleo.

En el estado del arte identificamos algunos estudios en los que se han aplicado TDD en los sistemas informáticos, algunos autores llegan a la conclusión que esta metodología mejora la calidad del software, sin embargo, son muy escasos los estudios que se han realizado en la etapa de mantenimiento.

Este proyecto pretende aportar a la comunidad científica más datos acerca del uso de la metodología TDD y la técnica de CI combinadas en la fase de mantenimiento del software y de esta manera demostrar que realmente mejoran tanto la calidad como la productividad de las aplicaciones web, dando como resultado el cumplimiento satisfactorio de los requerimientos proporcionados por los usuarios.

Objetivos

Objetivo General

Aplicar la metodología de desarrollo basado en pruebas y la técnica de integración continua tendiente a mejorar la calidad externa y productividad del software durante la etapa de mantenimiento, tomando como caso de estudio el desarrollo de una aplicación web.

Objetivos Específicos

- Realizar una revisión sistemática de literatura relacionada con la integración continua de sistemas informáticos para conocer metodologías y técnicas que mejoran la calidad del desarrollo de software.
- Aplicar la metodología de desarrollo basado en pruebas y la técnica de integración continua para incrementar la funcionalidad de un aplicativo web existente.
- Validar la metodología y técnica de desarrollo utilizadas en el caso de estudio para determinar su efectividad en la etapa de mantenimiento del software.

Alcance

En el presente proyecto de investigación se pretende aplicar las técnicas de TDD y CI, estudiadas en la revisión sistemática de literatura, a un caso de estudio el cual consiste en el desarrollo de un módulo de una aplicación web. En la Universidad de las Fuerzas Armadas “ESPE” existe un Sistema al cual fue denominado GPI (Gestión de Proyectos de Investigación), está conformado por varios módulos, por ejemplo, el registro de proyectos de Investigación de la institución, en este Sistema se requiere incluir un nuevo módulo “Planificación de Actividades de Investigación de Docencia”, trabajaremos en su desarrollo implementando las técnicas mencionadas y de esta manera probar nuestra hipótesis planteada más adelante.

Para mejorar la calidad del software se utilizará la metodología SCRUM la cual nos ayudará a controlar el tiempo y a dividir el desarrollo e implementación del módulo en historias de usuario, las mismas que serán divididas en tareas y asignadas a cada uno de los tesisistas.

Las historias de usuario serán divididas en dos grupos, el grupo que se desarrollará con el método tradicional y el grupo en el que utilizaremos TDD y CI, posteriormente realizaremos la respectiva comparación entre historias de usuario similares.

Para probar nuestra hipótesis aplicaremos particiones de equivalencia que nos permitirán realizar casos de prueba tanto para los dos grupos de historias de usuario mencionados, se aplicarán los mismos escenarios de prueba en cada una de las historias y de esta manera se obtendrá resultados comparables en su totalidad.

Una vez que el módulo sea integrado al sistema se pasará a realizar pruebas con los usuarios que se pueden ver en el diagrama de casos de uso.

Para delinear de forma adecuada el alcance de la investigación planteada, se formulan las preguntas desagregadas de cada objetivo específico, así como se muestra en la Tabla 1.

Tabla 1

Preguntas de Investigación

Objetivos Específicos	Preguntas de Investigación
<p>OE1: Realizar una revisión sistemática de literatura relacionada con la integración continua de sistemas informáticos para conocer metodologías y técnicas que mejoran la calidad del desarrollo de software.</p>	<ol style="list-style-type: none"> 1. ¿Cuáles son las metodologías que se aplican en los sistemas informáticos que se someten a cambios constantemente? 2. ¿Cómo reducir o mitigar errores en la etapa de mantenimiento de un sistema informático?
<p>OE2: Aplicar la metodología de desarrollo basado en pruebas y la técnica de integración continua para incrementar la calidad y productividad en un aplicativo web existente.</p>	<ol style="list-style-type: none"> 1. ¿Cuáles son las fases en el proceso de la implementación de un módulo en un sistema informático que se somete a cambios constantemente? 2. ¿Cómo mejorar la calidad y productividad de un sistema informático que está siendo modificado constantemente?

Objetivos Específicos	Preguntas de Investigación
<p>OE3: Validar la metodología y técnica de desarrollo utilizadas en el caso de estudio para determinar su efectividad en la etapa de mantenimiento del software tomado como caso de estudio.</p>	<ol style="list-style-type: none"><li data-bbox="805 453 1414 617">1. ¿Las metodologías y técnicas aplicadas realmente mejoran la productividad y calidad externa de un sistema informático?<li data-bbox="805 653 1414 821">2. ¿Con la implementación de las metodologías y técnicas de desarrollo se obtuvieron los resultados esperados?

Hipótesis

La técnica de integración continua y la metodología de desarrollo basado en pruebas mejora la calidad externa y la productividad del software en la etapa de mantenimiento.

CAPÍTULO II: Marco Teórico

Metodología de desarrollo

Se implementarán dos metodologías de desarrollo:

Metodología de desarrollo tradicional

Figura 1

Metodología de desarrollo tradicional

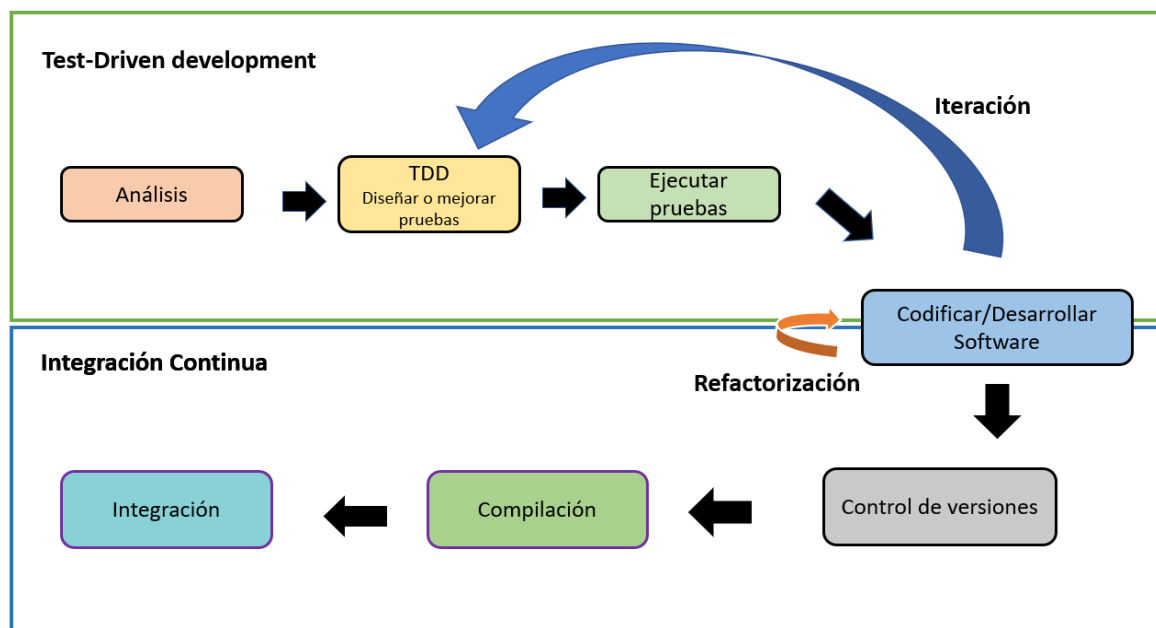


Nota. El gráfico representa las etapas a seguir en la metodología de desarrollo tradicional.

Metodología de desarrollo con TDD e Integración continua

Figura 2

Metodología de desarrollo con TDD e integración continua



Nota. El gráfico representa las etapas y el procedimiento a seguir con la metodología.

Las dos metodologías se las implementarán a un contexto académico y caso de estudio.

- **Contexto académico:**

En la Universidad de las Fuerzas Armadas “ESPE” para el vicerrectorado de investigaciones. Se desarrollará un módulo de un sistema de software aplicando la técnica TDD y CI que será comparada con el desarrollo tradicional.

- **Caso de estudio:**

Para el caso de estudio se desarrollará un nuevo módulo que pertenecerá al Sistema GPI (Gestión de Proyectos de Investigación) que será denominado “Sistema

de Planificación de Actividades de Investigación de docencia” en la Universidad de las Fuerzas Armadas “ESPE”

Se implementará la metodología TDD en conjunto con la técnica de Integración continua en comparación con la metodología tradicional, utilizaremos la metodología SCRUM y en base a nuestras historias de usuario se aplicarán pruebas de caja negra para de esta manera validar la calidad externa y productividad del Software.

Marco Teórico

Planteamiento del Marco Teórico

Para desarrollar el Marco teórico de la presente investigación se tuvo en cuenta la hipótesis planteada de donde se obtuvo las variables dependientes e independientes, las cuales de detalla a continuación

Variables dependientes

- Técnica de integración continua
- Metodología de desarrollo basado en pruebas

Variables independientes

- Calidad del software
- Productividad del software

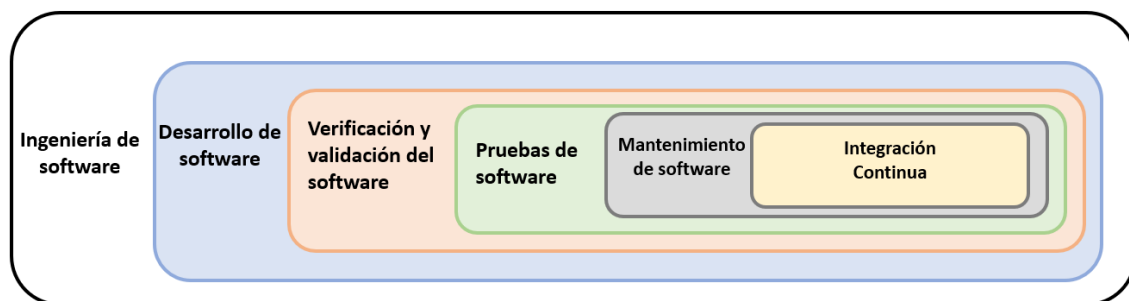
Red de Categorías

Para determinar la red de categorías se ha desglosado los temas desde lo general a lo más específico de las variables dependientes e independientes.

Variables Dependientes

Figura 3

Cascada de la red de categoría variable dependiente



Nota. El gráfico se presenta los conceptos desde el más general al más específico.

Ingeniería de software

Una de las ramas de las Ciencias de la Computación es la ingeniería de software que se enfoca en resolver problemas de automatización y de esta manera ofrecer un producto confiable y de buena calidad. Para cumplir con las expectativas del software se integra estándares de calidad como son ISO/IEC/IEEE 12207, ISO 9001, ISO/ IEC/IEEE 26511 cada

una de ellas con características importantes que aportan con buenas prácticas y conceptos en el desarrollo de sistemas de software.

Para empezar con el sistema informático debemos alinear la estrategia de desarrollo del sistema con la estrategia de la organización, en esto están incluidos los stakeholders del proyecto tanto de parte de la organización como del grupo de desarrollo. Según (ISO 26511, 2018) la organización debe contener planes estratégicos en los que se implican factores como la madurez del producto, el mercado, la industria, una parte importante es los controles regulatorios o legales para mantenerse en lo establecido por la ley y a largo plazo no represente inconvenientes que obliguen al gerente a tomar caminos alternos que alteren los tiempos establecidos en el cronograma del proyecto.

Desarrollo de software

Para desarrollar software de calidad se deben considerar las diferentes etapas establecidas por los estándares internacionales los cuales aportan características fundamentales que deben cumplir tanto los involucrados en el sistema como el sistema en general.

El coordinador del software juega un papel importante ya que él debe tener en cuenta factores esenciales para empezar con el proceso de desarrollo, (ISO 26511, 2018) manifiesta que los involucrados deben seguir una estrategia de desarrollo, se puede dividir el equipo en grupos orientados a tareas específicas, por ejemplo:

- Grupos orientados al cliente (Marketing, soporte técnico)
- Grupo de desarrollo de software que se ocupen del software, hardware, seguridad, etc.

- Grupos orientados a verificar la calidad por ejemplo desarrolladores de pruebas de software, soporte técnico, asuntos regulatorios, etc.

Los Sistemas Automatizados deben seguir ciertas etapas ya hemos hablado de la primera que es organizar el grupo de personas que se van a involucrar con el software, ahora hablemos del ciclo que tiene que cumplir un software durante su etapa de elaboración.

Según (ISO12207, 2020) las principales etapas del ciclo de vida de un sistema son: Concepto, Desarrollo, Producción, Utilización, Soporte y Retiro. A su vez se pueden incluir otras etapas según la necesidad del sistema, por ejemplo: Operación, Mantenimiento, Exploración, entre otros. Para asegurar que cada etapa se concluya con satisfacción se puede realizar procesos iterativos e incrementales así se puede realizar un seguimiento y evaluación de cada parte del proceso.

Verificación y Validación del Software

Durante la etapa del desarrollo de software tenemos la fase de verificación y validación del software en la cual se deben cumplir ciertos procesos de comprobación y análisis, de esta forma aseguráramos que el sistema cumpla con las especificaciones y necesidades expresadas por los clientes.

La función de la verificación es comprobar si el sistema se está desarrollando de forma correcta, en otras palabras, si el sistema cumple con los requisitos funcionales y no funcionales del software.

La validación del software es visto desde un concepto más general, en este se comprueba si el sistema cumple con lo que el usuario espera que haga el sistema en concreto.

Estos dos conceptos se pueden confundir durante la fase de desarrollo ya que van de la mano y las dos requiere de la satisfacción del usuario.

A continuación, se listará características que según (IEEE A. S., 2017) cumplen estas fases del software

Verificación:

- El software debe cumplir con los requisitos especificados y asegurar la integridad, consistencia, corrección y precisión de estos.
- Se debe implementar estándares y prácticas que aseguren la calidad y confiabilidad del producto obtenido.
- Seguir el proceso de ciclo de vida del desarrollo de software y hacer seguimiento en cada una de las actividades y fases para asegurar el funcionamiento del sistema.

Validación:

- Comprobar al final de cada fase del ciclo de vida si los requisitos establecidos se están cumpliendo de manera satisfactoria.
- Cumplir con las necesidades del usuario validando si el uso del producto final es el esperado.

Pruebas de Software

La prueba de Software es una de las acciones más importantes en el desarrollo, ya que gracias a esto es posible detectar defectos y reducir riesgos asociados, esto es clave para la satisfacción del cliente y tiene un impacto directo en el costo y el desarrollo del producto.

Tradicionalmente se ha llevado a cabo al final del proceso, cuándo el producto está terminado. Actualmente por la complejidad del Software se exige que las pruebas se realicen de forma paralela al desarrollo, para que los errores sean encontrados a tiempo y reducir costos. “La automatización de las pruebas surgió como una alternativa para agilizar su ejecución, a la vez que para mejorar la fiabilidad del producto y su calidad” (Serna, Martinez, & Tamayo, 2021).

Con la automatización, las actividades de pruebas manuales son reemplazadas por procesos automatizados, lo que es igual a no requerir la participación humana para generar una prueba, si se implementa bien esto ayudaría a eliminar o minimizar tareas repetitivas.

Mantenimiento de Software

El mantenimiento es un elemento clave para el ciclo de vida de cualquier sistema de Información. “Este proceso está compuesto por todas aquellas actividades que, se realizan desde el momento en el que el software está operativo hasta que es retirado y que están destinadas a corregir sus errores, adaptarlo o mejorar su rendimiento” (Silvera & Vargas, 2010), actividades como detección de errores, cambios o programación son muy importantes en este proceso.

El mantenimiento de Software permite alcanzar varios beneficios, en cuanto a la realización de actividades y tareas para que sean más rápidas y eficientes. A continuación, se describe las categorías de mantenimiento definidas por (Lientz & Swanson, 1980).

- **Mantenimiento Correctivo:** Aquí se realizan correcciones de fallos detectados en el diseño y código. Este punto es muy importante para que la aplicación funcione.

- **Mantenimiento Adaptativo:** En este punto se adapta la aplicación a los requerimientos de los usuarios.
- **Mantenimiento Perfectivo:** Se intenta mejorar el rendimiento del Software, su eficiencia, eficacia y sostenibilidad.
- **Mantenimiento Preventivo:** Se anticipa y descubre problemas potenciales reduciendo el riesgo de fallos serios y minimizar consecuencias.

Integración Continua

La integración de Software es un problema complejo, sobre todo si en su desarrollo se involucran diferentes códigos de varias personas, por esta razón la Integración Continua ofrece un esquema que permite realizar integraciones a medida que se lleva a cabo el desarrollo generando incrementos pequeños y mostrando los resultados obtenidos.

“La integración continua es una práctica que comienza con la organización de los proyectos en una estructura de directorios adecuada para establecer el orden de ejecución de sus componentes (incluyendo casos de prueba), y de esta manera facilitar la construcción correcta del software “. (Salamón, y otros, 2014)

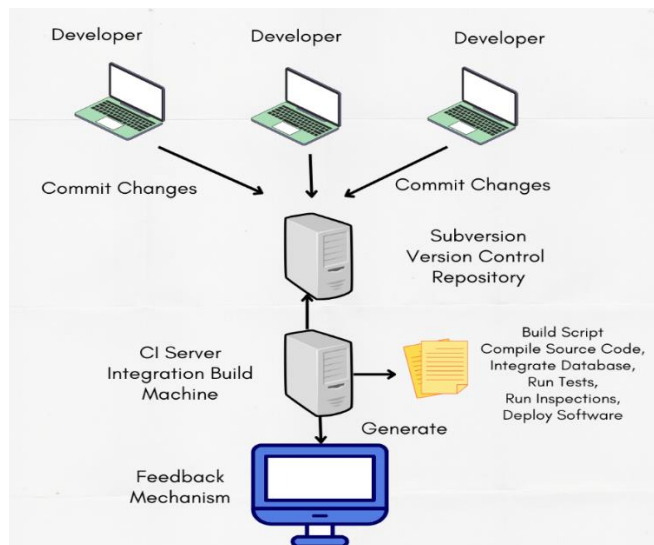
Como siguiente punto se presenta un escenario típico de Integración continua según el trabajo investigativo de (Salamón, y otros, 2014) usando como referencia a (Duvall, Steve, & Glover, 2007)

1. El desarrollador realiza un commit de su trabajo al repositorio de control
2. El servidor de Integración Continua verifica cambios en el repositorio cada 5 minutos.

3. El mismo detecta un cambio y extrae el último commit ejecutando una build script que se encarga de integrar los distintos componentes del software en desarrollo.
4. El servidor de integración continua genera feedback con los resultados del proceso de building, el cual es enviado a los miembros que se especifica del proyecto.
5. Y de forma cíclica el El servidor de integración continúa revisando cambios en el repositorio de control.

Figura 4

Escenario típico de Integración Continua

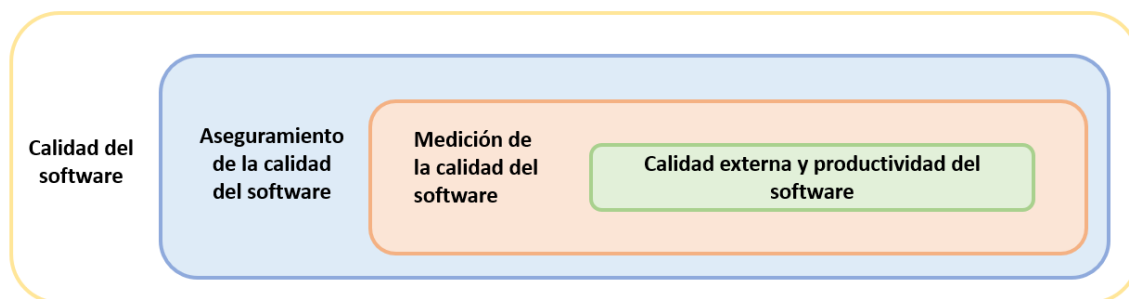


Nota. La imagen muestra el funcionamiento de CI.

Red de Categorías Variables Independientes

Figura 5

Cascada de la Red de categoría Variable Independiente



Calidad del Software

Actualmente el software se encuentra en diferentes y diversos campos de la actividad humana, por esto es importante su calidad para satisfacer las necesidades de los usuarios. Hay que diferenciar entre la calidad del software y la calidad del desarrollo, aunque ambos van a estar entrelazados ya que la calidad del proceso del desarrollo va a depender mucho de la calidad del producto. La calidad se obtiene mejorando día a día el proceso de producción, mantenimiento y gestión de software

“La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”
(IEEE I. o., 1990)

Aseguramiento de la Calidad del Software

En este punto es necesario aplicar métodos, herramientas y técnicas que permiten gestionar la calidad en el desarrollo de un producto de software. Según la publicación del

artículo científico de (Carrizo & Alfaro, 2018) existe un método de Aseguramiento de Calidad de Software que consta de tres componentes:

1. El equipo de trabajo debe entender el concepto de calidad
2. Las herramientas para el control de la calidad del proyecto del Software
3. Métricas que midan resultados y procesos

Así mismo, (Pressman R. S., 2010) en su libro Ingeniería del Software realiza una propuesta de Aseguramiento de Calidad de Software de 6 pasos.

1. Diseño del proceso.
2. Tareas específicas de aseguramiento de la calidad y control.
3. Prácticas eficientes de ingeniería de software (métodos y herramientas).
4. Control de todos los productos del trabajo de software y de los cambios que sufren.
5. Procedimiento para garantizar el cumplimiento de los estándares del desarrollo de software
6. Mecanismos de medición y reporte.

Medición de la Calidad del Software

La medición es importante en cualquier proceso de ingeniería para comprender mejor los modelos que se crean. Las métricas pueden ser directas o indirectas y se calculan en base a una fórmula. “Una métrica contiene la definición de un método de medición o un método de cálculo y la escala asociada” (Rivero, Madariago, Toledo, Lamothe, & Hechavarría, 2016). Esto constituyen la base para detectar las desviaciones del rendimiento aceptable en los procesos y

producto de software, las oportunidades de mejora, identificar y priorizar las principales preocupaciones, dar seguimiento a la solución y mejorar la calidad del producto.

Medir la calidad del Software ha adquirido mucha relevancia en las organizaciones ya que cada vez el mercado es más competitivo, “Esta no sólo tiene influencia en los costos finales, sino que es también un factor clave para mejorar la imagen frente a los clientes y como elemento diferenciador de la competencia” (Irrazábal, 2015)

Actualmente se observa una tendencia de institucionalización de buenas prácticas de calidad en organizaciones de desarrollo software que busca alinear sus procedimientos con modelos de referencias conocidos, como CMMI-DEV o ISO/IEC, estas métricas muestran detalles sobre que normas usar y cuál es la más adecuada.

Calidad Externa y Productividad del Software

Para la medición externa de la calidad y productividad del software existen varios modelos y estándares considerados eficientes, confiables y funcionales como lo es ISO/IEC 9126, donde se describe la normativa para un modelo de calidad y su uso como marco para la evaluación del producto del software. Este modelo se define por medio de características generales de software y con atributos medibles, cuyos valores son calculado usando alguna métrica.

Al igual que la calidad, existen métricas que miden la productividad como la entrega frecuente y temprana del software, y el valor que agregan las tareas al producto del software, estas se orientan al desempeño organizacional y al proyecto. “Para mejorar la productividad de un equipo, es necesario conocer cómo se comporta de forma regular en unos intervalos de

tiempo". (Hernández, Jimenez, Martínez, & Jiménez, 2019), para esto es importante recopilar y analizar los datos obtenidos para generar información sólida y tomar decisiones.

A continuación, se presenta un modelo de calidad interna y externa de la ISO/IEC 9126.

Tabla 2

Calidad Interna y Externa ISO/IEC 9126

Características	Subcaracterísticas
Funcionalidad	Idoneidad
	Precisión
	Interoperabilidad
	Seguridad
	Funcionalidad
	cumplimiento
Fiabilidad	Madurez
	Tolerancia a fallos

Características	Subcaracterísticas
	Recuperabilidad
	Fiabilidad
	cumplimiento de
	Comprensibilidad
Usabilidad	Comprensibilidad
	capacidad de aprendizaje
	Operabilidad
	Atractivo
	Usabilidad
	cumplimiento
Eficiencia	Comportamiento del tiempo
	Eficiencia en la utilización de los recursos
	Cumplimiento
Mantenibilidad	Analizable

Características	Subcaracterísticas
	Posibilidad de cambiar
	Estabilidad
	Comprobabilidad
	Cumplimiento
Portabilidad	Adaptabilidad
	Instalabilidad
	Coexistencia
	Reemplazabilidad
	Cumplimiento

Estado Del Arte

Planteamiento de la revisión de literatura preliminar

En esta fase se realizó una breve descripción del problema de investigación para proporcionar un contexto para la búsqueda de estudios científicos; posteriormente se

procedió a definir un objetivo de búsqueda y plantear preguntas de investigación para alinear la búsqueda en relación con el problema de investigación y finalmente se definieron los criterios de inclusión y exclusión.

Objetivo de la Búsqueda

Realizar una revisión de literatura preliminar acerca de la integración y modificación de sistemas informáticos con el propósito de conocer las estrategias que se aplican en la actualidad para mejorar la calidad interna y externa en la etapa de mantenimiento.

Preguntas de Investigación

RQ1 ¿Cómo se puede mitigar o reducir los errores en los sistemas informáticos cuando se agrega un módulo durante el desarrollo?

RQ2 ¿Cómo implementar metodologías y técnicas de desarrollo de un sistema informático que se somete a cambios constantes?

5.2 Criterios de inclusión y exclusión

Tabla 3.

Criterios de inclusión y exclusión

Criterios de Inclusión	Criterios de Exclusión
Estudios enfocados en los procesos de mejora continua de un sistema informático en su etapa de mantenimiento.	Estudios que no apliquen metodologías o técnicas para la integración continua de sistemas.
Reducción de errores en un sistema en la etapa de mantenimiento.	Estudios en los que su enfoque sea desarrollar un software que no se someta a cambios constantes.
Metodologías que se aplican en la actualidad para la integración y modificación de sistemas.	

Grupo de control

En esta fase se delimitaron los artículos que se consideran relevantes en la investigación, eliminando a aquellos que no profundizan o no nos dan el contenido que esperamos, para realizar este estudio participamos dos investigadores los cuales una vez obtenidos los estudios validamos la información y obtuvimos el siguiente grupo de control:

Tabla 4.

Artículos que conforman el Grupo de Control

Código	Título	Términos relevantes
EC1	Applying continuous integration for increasing the maintenance quality and efficiency of web app	Continuous Integration, agile process, Web app, Integration test, maintenance quality and efficiency
EC2	Enabling Continuous Improvement of a Continuous Integration Process	Continuous Integration, Build Failures, Antipatterns, Best Practices
EC3	Effects of Test-Driven Development: A Comparative Analysis of Empirical Studies.	Test-driven development; test-first programming; software testing; software verification; software engineering; empirical study.
EC4	Software Maintainability and Usability in Agile Environment	Test-driven development, meta-analysis, code quality, programmer

Código	Título	Términos relevantes
		productivity, agile software development

5.4 Cadena de búsqueda

Las cadenas de búsqueda según los estudios analizados y tomando en cuenta los términos de inclusión y exclusión son las siguientes:

- Esta cadena nos va a ayudar a encontrar artículos relacionados a la integración continua de sistemas y mantenimiento de software.

((“continuous integration” OR “integration test”) AND (“web app” OR “software”) AND (“maintenance quality” OR “code quality” OR “best practices”) AND (“agile process” OR “agile software development”))

- Esta cadena nos va a ayudar a encontrar artículos relacionados al desarrollo basado en pruebas durante la etapa de mantenimiento del software.

((“TDD” OR “Test driven development” OR “Test-driven development” OR “Test driven design” or “Test-driven design” OR “Integration driven development” OR “Integration-driven development”) AND (“Productivity” OR “Maintainability”))

Proceso de selección

Al aplicar la cadena de búsqueda en la base digital IEEE Explore se obtuvo alrededor de 120 artículos relacionados con el tema el cual se consideró un número de artículos manejable.

Para seleccionar los estudios hemos escogido los que tienen vigencia a partir del 2011; los cuales se detallan en la siguiente tabla.

Tabla 5

Estudios Primarios

Código	Título	Cita
EP1	An Empirical Study of the Relationship between Continuous Integration and Test Code Evolution	(Gustavo Sizilio, 2019)
EP2	Improving the Robustness and Efficiency of Continuous Integration and Deployment	(Gallaba, 2019)
EP3	Applying continuous integration for increasing the maintenance quality and efficiency of web app	(Lai, 2019)

Código	Título	Cita
EP4	The effectiveness of test-driven development: an industrial case study	(Tomaz Dogsa, 2011)
EP5	Effectiveness of Test-Driven Development and Continuous Integration	(Chintan Amrit, 2018)

Síntesis y Resultados.

EP1 (Gustavo Sizilio, 2019): An Empirical Study of the Relationship between Continuous Integration and Test Code Evolution

En esta investigación han estudiado 82 proyectos con integración continua (CI) y 82 proyectos sin integración continua (NOCI), el objetivo fue investigar tendencias crecientes en lo que se refiere a test code-ratio (una métrica que nos ayuda a saber la relación entre prueba-código; el número de *líneas de código de producción (LPC)* frente al número de *líneas de código de prueba (LTC)*).

Las pruebas automatizadas juegan un papel fundamental al combinarlas con CI, ya que nos ayudan a detectar errores a tiempo y mejorar la calidad del código.

En este estudio abordan la calidad analizando las tendencias en su código, usaron el tamaño de la prueba para comparar la métrica de la proporción que es el número de líneas de código de prueba automatizadas sobre las líneas totales del sistema analizado.

Para encontrar los resultados de la investigación se asignó proyectos para trabajar con CI Y NOCI, pero también se continuó el desarrollo de cambios y modificaciones de softwares que ya fueron establecidos con anterioridad, aunque en los resultados no especifica los sistemas que empezaron desde cero a los que continuaron con su etapa de mantenimiento.

La conclusión es que IC mejora la calidad del código por medio de la métrica test code-ratio, el 40.2 % de los proyectos con IC la mejoraron en relación con el 17% con NOIC.

EP2 (Gallaba, 2020): Improving the Robustness and Efficiency of Continuous Integration and Deployment

En este estudio usaron el método CI/CD (Integración Continua y Distribución Continua), primero se implementa CI en los proyectos que serán integrados en repositorios ascendentes después de haber sido construidos y verificados por un flujo de trabajo automatizado, CD sigue su proceso asegurando que el software puede ser confiable lanzado en cualquier momento.

El objetivo es analizar las características al implementar CI/CD, para esto tomaron una muestra de 9312 proyectos alojados en GITHUB (plataforma de

alojamiento del código para el control de versiones y colaboración) a su vez adoptaron el servicio TRAVIS CI (servicio de integración continua que se utiliza para crear y probar proyectos software alojados en github), como resultados obtuvieron que los desarrolladores a menudo encuentran inconvenientes al usar CI/CD como por ejemplo la mala configuración de entornos, mal interpretación de resultados y mal uso de los recursos disponibles, en este artículo no se especifica si los proyectos fueron evaluados en la etapa de mantenimiento pero sin duda nos da una pauta para el control de versiones con el método (CI/CD).

EP3 (Lai, 2019): Applying continuous integration for increasing the maintenance quality and efficiency of web app

La propuesta de esta investigación es el uso de IID (Desarrollo incremental iterativo) y el continuo mantenimiento de aplicaciones web basado en la integración para mejorar la calidad y eficiencia en cuanto a las aplicaciones web.

Se habla de los problemas que pueden presentar al momento de dar mantenimiento a una web app, las pruebas unitarias son uno de ellos ya que en ocasiones están incompletas y esto causa retraso con el desarrollo y defectos en el resultado, una de las formas de mitigarlos es el uso de pruebas que sean claras, definidas y automatizadas, uno de los componentes de clase de alta calidad es la implementación de TDD para generar el código de una prueba cuando pasa, esta investigación propone seguir cinco pasos para mejorar la calidad del software que son:

1. Pruebas unitarias automatizadas
2. Gestión de la configuración y control de versiones

3. Integración e implementación automatizadas
4. Integración y validación
5. Mejora continua

Este proceso asegura mejorar de manera eficiente la calidad y la productividad de las aplicaciones web para beneficiar la experiencia del usuario y depurar defectos y errores.

EP4 (Tomaz Dogsa, 2011) The effectiveness of test-driven development: an industrial case study

En esta investigación se realizaron 3 proyectos dos de los cuales realizó su desarrollo con las técnicas tradicionales y el último utilizando la metodología TDD. Partiendo de la premisa de que el objetivo del TDD es escribir un código limpio y funcional que provea beneficios al sistema con el fin de representar mayor impacto en la etapa de mantenimiento de software.

Los estudios seleccionados fueron elegidos de tal manera que en lo posible sean iguales, esto se logró ya que los 3 se desarrollaron para el mismo cliente, además los programadores tenían un perfil similar en cuanto a las necesidades de los proyectos. También se hizo uso de las métricas recomendadas por la norma ISO 91267, para medidas de calidad internas, externas y de uso.

Las métricas de mantenibilidad se definieron de acuerdo con la norma ISO 9126 como el esfuerzo expresado en las horas de personal necesarias para corregir fallas,

mejorar el desempeño o adaptarse a un entorno cambiante durante la fase de mantenimiento.

Es así como se concluyó que el uso de TDD en la fase de mantenimiento reduce el esfuerzo necesario para depurar y realizar correcciones al código fuente mejorando también la calidad del software, adicionalmente se dice que se necesita de más estudios en esta etapa de desarrollo para poder fortalecer o refutar los resultados encontrados.

EP5 (Chintan Amrit, 2018) Effectiveness of Test-Driven Development and Continuous Integration

La integración continua y TDD tienen un fin en común, el cual es el de mejorar la calidad de la aplicación disminuyendo la cantidad de tiempo que se necesita para realizar una corrección en el código, así como el de reducir los costos de esta actividad. Pero para que esto tenga validez o sea aceptado por una empresa se debe presentar información o estudios empíricos que la respalden ya que en un principio utilizar dicha metodología y técnica de desarrollo conlleva un gasto adicional para las empresas.

Entonces se han hecho estudios empíricos que indican la efectividad de las implementaciones de TDD en entornos académicos e industriales los cuales han dado como resultado un aumento en la calidad del código, pero también se reflejó una baja en la productividad producida por el esfuerzo que conlleva al escribir las pruebas.

Por otro lado, se puede argumentar que TDD podría mejorar la calidad del código interno y externo, lo que lleva a una menor generación de defectos, mejorando así la productividad. Si bien en esta investigación se hizo uso de TDD como CI.

Se encontró de manera general que se produjo un aumento en la calidad del software y le proporcionó a la empresa una infraestructura más robusta para evaluar de mejor forma los procesos del sistema, se necesita de varias investigaciones en la etapa de mantenimiento para que respalden estos resultados.

Características del Estado del Arte

Se han encontrado 5 estudios empíricos que analizan el uso de metodologías ágiles, así como la integración continua y el desarrollo basado en pruebas. De los cuales uno de ellos, en concreto el estudio realizado por (Chintan Amrit, 2018), el cual habla acerca del uso en conjunto de esta metodología y técnica en la etapa de mantenimiento, donde afirma que la calidad externa del software tuvo un aumento al igual que la robustez del código, dando resultados satisfactorios y una mejor respuesta ante fallos.

Como punto en común, la mayoría de los estudios dicen que existe muy poca investigación de estas técnicas en la etapa de mantenimiento y dejan abierta la posibilidad de realizarla, para con ello refutar o validar los resultados que encontraron en cada una de sus respectivas investigaciones.

Así bien, nuestro estudio pretende ampliar el conocimiento que se tiene acerca de estas 2 técnicas en la etapa de mantenimiento y con ello verificar si los resultados encontrados mejoran o no la calidad externa del software.

CAPÍTULO III: Análisis y diseño

Introducción

En el presente capítulo se detalla el proceso a seguir previo al desarrollo del módulo del Sistema GPI (Gestión de Proyectos de Investigación) de la Universidad de las Fuerzas Armadas “ESPE”, el cual estará basado en la utilización de metodologías y técnicas de desarrollo de Software como estrategias para la obtención del producto final esperado.

Metodología de desarrollo.

Para el desarrollo del módulo de “Planificación de Actividades de Investigación de los docentes” del Sistema GPI, se utilizará la metodología SCRUM ya que nos permitirá trabajar en equipo y tener una comunicación efectiva con el usuario (Unidad de Gestión de Proyectos de la UFA “ESPE”).

En la metodología mencionada empezamos con el desarrollo de nuestro Product Backlog, que es la obtención de requisitos y necesidades que el sistema debe cubrir, se realizará un documento ERS (Especificación de Requerimientos del Sistema) en el cual se determinarán los actores/usuarios que intervendrán, así como, sus respectivos casos de uso.

Para nuestro Sprint Backlog dividiremos el trabajo a realizar en tareas específicas y con un tiempo determinado, cada sprint durará dos meses y estará compuesto de historias de usuario, las cuales estarán basadas en los casos de uso del ERS que tendremos que

desarrollarlas según el tiempo establecido y con las estrategias de desarrollo seleccionadas (TDD Y CI).

Alcance

El alcance lo definirá el documento de Especificación de Requerimientos del Sistema ya que estarán establecidas las tareas a desarrollar (Casos de uso) y los usuarios que intervendrán en el sistema.

El Módulo del Sistema GPI “Planificación de las Actividades de Investigación de Docencia” pretende que el docente planifique sus horas de investigación de forma semestral o anual, una vez registrada en el sistema la carga horaria, esta pasará al coordinador de Investigación y director de departamento para validar y aceptar la información ingresada y de esta manera tener un control de todas las actividades de investigación realizadas por los docentes de la Universidad, el analista de la Unidad de Gestión de Investigación (Vicerrectorado de la Universidad) tendrá acceso a esta información, la cual será presentada mediante reportes.

Las actividades que constan en el documento de carga horaria de los docentes son:

- Publicaciones Científicas
- Capítulos de Libros
- Escritura y Publicación de Libros
- Congresos
- Otras Actividades (Dirección de proyectos de investigación, Participación de proyectos de investigación, Participación en Revistas, etc.).

Se realizará una sección de reportes

- Los docentes podrán visualizar el estado de sus actividades planificadas.
- Los directores y coordinadores de departamento podrán visualizar las actividades planificadas por los docentes pertenecientes a su departamento.
- Personal administrativo seleccionado de la UGI (Unidad de Gestión de Investigación) podrá visualizar los reportes de todas las sedes y departamentos de la Universidad.

Personal involucrado.

A continuación, se detallará el equipo de desarrollo que se encargó de la planificación, desarrollo, correcciones y pruebas del sistema acentuando además su rol dentro de scrum.

Tabla 6

Personal involucrado, miembro 1

Nombre	Nicole Noemi Arias Álvarez
Rol	Analista, diseñador de Interfaces y programador
Categoría profesional	Tesista de la carrera de Ingeniería de Sistemas e Informática.
Responsabilidades	Documentación, scrum máster y desarrollador del sistema
Información de contacto	nnarias@espe.edu.ec

Tabla 7*Personal involucrado, miembro 2*

Nombre	Rodrigo Javier Cueva Cuenca
Rol	Modelador de Base de Datos, programador
Categoría profesional	Tesista de la carrera de Ingeniería de Sistemas e Informática.
Responsabilidades	Base de Datos, manejo del servidor, desarrollador del sistema
Información de contacto	rjcueva@espe.edu.ec

Tabla 8*Personal involucrado, miembro 3*

Nombre	Luis Gonzalo Rocha Hoyos
Rol	Analista de sistema
Categoría profesional	Ingeniero en Sistemas e Informática.
Responsabilidades	Dirección del proyecto, diseñador de la arquitectura

Información de contactogrocha@espe.edu.ec**Tabla 9***Personal involucrado, miembro 4*

Nombre	Nelly Oliva Cevallos Mejía
Rol	Especialista de aplicaciones
Categoría profesional	Ingeniera en Informática.
Responsabilidades	Revisión del proyecto, analista de base de datos.
Información de contacto	nocevallos@espe.edu.ec

Tabla 10*Personal involucrado, miembro 5*

Nombre	Wendy Castillo
Rol	Analista de difusión de la ciencia
Categoría profesional	Lcda. En Comunicación Corporativa
Responsabilidades	Revisión de requerimientos área publicaciones (artículos y libros).

Validación funcional del sistema área específica.

Información de contacto

wecastillo1@espe.edu.ec

Tabla 11

Personal involucrado, miembro 6

Nombre	Tannia Mejía Campaña
Rol	Analista de Gestión de Proyectos de Investigación
Categoría profesional	Ingeniera en Sistemas
Responsabilidades	Planteamiento y revisión de requerimientos área proyectos - área proyectos. Validación funcional del sistema Evaluación de actividades.
Información de contacto	tpmejia@espe.edu.ec

Nota. La tabla muestra el personal 6 involucrado en el desarrollo del módulo de planificación del sistema GPI.

Tabla 12

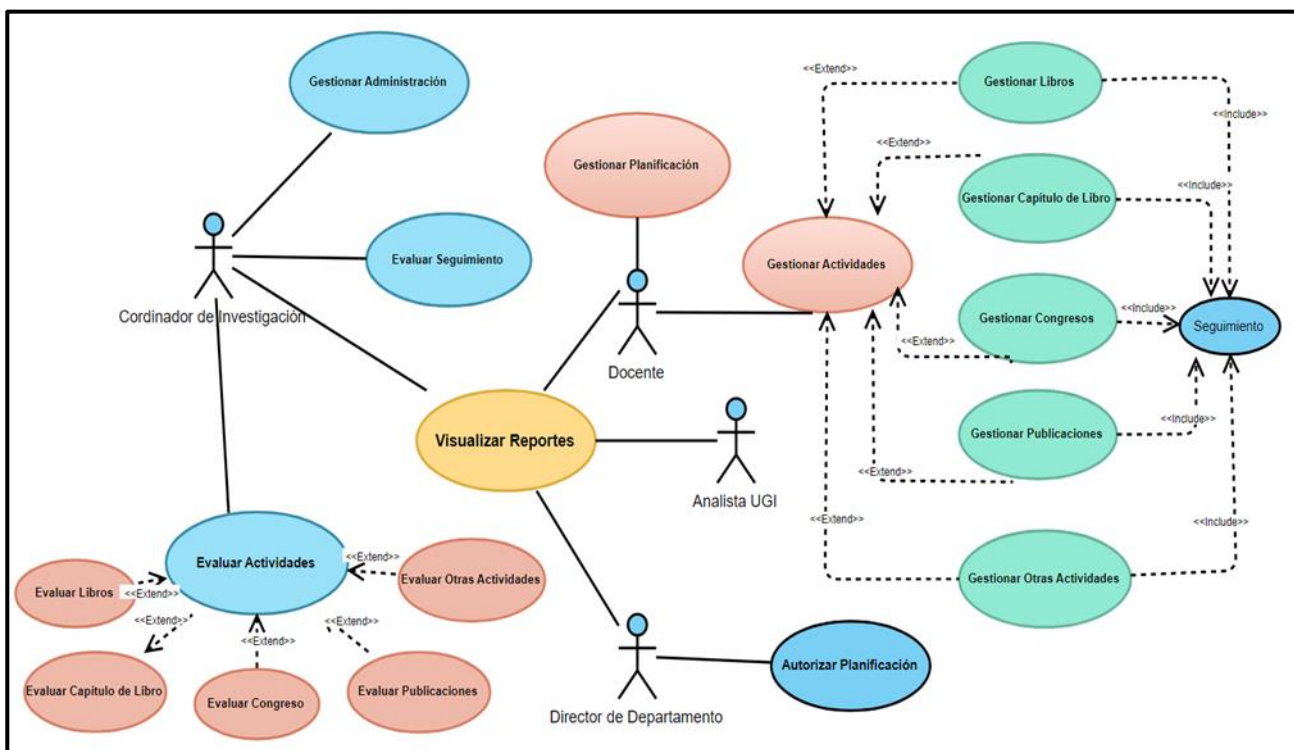
Personal involucrado, miembro 7

Nombre	Lilí Carolina Salcedo Vallejo
Rol	Docente de Apoyo de la Unidad de Gestión de Investigación
Categoría profesional	Economista
Responsabilidades	<ul style="list-style-type: none">• Seguimiento a la implementación funcional del proyecto.• Validación funcional del sistema• Evaluación de actividades. de Investigación
Información de contacto	lcsalcedo@espe.edu.ec

Diagrama de Casos de Uso

Figura 6

Diagrama de casos de uso



Nota. La figura muestra las funcionalidades de cada Usuario/Actor.

Descripción de los Casos de Uso

Tabla 13

Caso de uso Gestionar Actividades

Nombre del caso de uso:	Gestionar Actividades.
--------------------------------	------------------------

Módulo:	Planificación
----------------	---------------

Descripción: Se visualiza las opciones habilitadas de las actividades registradas por el usuario según su planificación.

Actores: Docente

Precondiciones: El actor debe estar autenticado en el sistema

Paso	Acción
	El sistema desplegará un Menú, en donde el actor
1.	deberá seleccionar las opciones: “Planificación”, “Gestionar Planificación”.
Secuencia normal:	
	El sistema desplegará las opciones habilitadas de las actividades registradas en la planificación, las
2.	actividades que no haya escogido serán bloqueadas y no se podrá ingresar datos.

El proceso de planificación ha resultado exitoso.

Postcondiciones: El sistema desplegará la ventana principal de “Planificación” donde se podrá visualizar la lista de actividades escogidas.

Paso	Acción
Excepciones:	
4,7	Si el actor no ha ingresado ninguna actividad.

E.1 El sistema mostrará deshabilitadas las actividades.

Nota. La tabla muestra todos los campos y variables que contienen el caso de uso Gestionar Actividades.

Tabla 14

Caso de uso Gestionar Escritura y Publicación de Libros.

Nombre del caso de uso:	Gestionar Escritura y Publicación de Libros.
Módulo:	Planificación
Descripción:	El usuario gestiona la información de la actividad Libros.
Actores:	Docente
Precondiciones:	El actor debe estar autenticado en el sistema
	Paso Acción
Secuencia normal:	El sistema desplegará un menú, en donde el actor deberá 1. seleccionar las opciones: “Planificación”, “Gestionar Planificación” y “Libros”

Nombre del caso Gestionar Escritura y Publicación de Libros.

de uso:

El sistema desplegará una pantalla con los campos a registrar:

- Código del Libro
- Código de la Planificación
- Código IES
- Tipo de la Publicación
- Código de La Publicación
- Título del Libro
- Código ISBN (contiene una lista desplegable parametrizable en la sección de Administración)
 2.
 - Físico
 - Digital
- Fecha de Publicación
- Campo Amplio (Lista desplegable según los datos correspondientes en el anexo)
- Campo Detallado (Lista desplegable según los datos correspondientes en el anexo)
- Campo Específico (Lista desplegable según los datos correspondientes en el anexo)
- Revisión por Pares

Nombre del caso Gestionar Escritura y Publicación de Libros.
de uso:

- Filiación UFA ESPE
- Participación
- Duración
- Completado
- Aprobación
- Observaciones

3. Si el actor pulsa sobre la opción: “Modificar Libro”

El sistema presentará una tabla con la información del libro la cual es:

- Código del libro
- Código IES
- Tipo de la publicación
- Código de la publicación
- Título del libro
- Código ISBN (contiene una lista desplegable parametrizable en la sección de Administración)
 - Físico

Nombre del caso Gestionar Escritura y Publicación de Libros.

de uso:

○ Digital

- Fecha de publicación
- Campo Amplio (Lista desplegable según los datos correspondientes en el anexo)
- Campo Detallado (Lista desplegable según los datos correspondientes en el anexo)
- Campo Específico (Lista desplegable según los datos correspondientes en el anexo)
- Revisión por pares
- Filiación UFA ESPE
- Participación
- Duración.

4. El actor podrá modificar las actividades ingresadas anteriormente.

5. El actor deberá pulsar en “Modificar Libro”, el sistema guardará los cambios en la base de datos.

6. **Si el actor pulsa sobre la opción: “Visualizar Libro”**

Nombre del caso Gestionar Escritura y Publicación de Libros.

de uso:

El sistema presentará una pantalla en donde se mostrará la información del libro la cual es:

- Código del libro
- Código IES
- Tipo de la publicación
- Código de la publicación
- Título del libro
- Código Identificación (contiene una lista desplegable)
- Código ISBN (contiene una lista desplegable parametrizable en la sección de Administración)
 - Físico
 - Digital
- Fecha de publicación,
- Revisión por pares
- Filiación UFA ESPE
- Participación
- Duración
- Aprobación
- Observación

Nombre del caso de uso: Gestionar Escritura y Publicación de Libros.

El proceso de gestión de libro ha resultado exitoso.

Postcondiciones: El sistema desplegará la ventana principal de “Libro” donde se podrá visualizar la lista de Libros con su información.

	Paso	Acción
Excepciones:	4,7	Si el actor no ha ingresado los campos requeridos correctamente.
	E.1	El sistema no habilitará la opción guardar los datos de la Libro en la base de datos.
	4,7	Si el evaluador no aprueba la planificación del docente

Nota. La tabla muestra todos los campos y variables que contienen el caso de uso Gestionar Libros.

Tabla 15

Caso de uso Gestionar Congresos

Nombre del caso Gestionar Congresos.

de uso:

Módulo: Planificación

Descripción: El usuario gestiona la información de la actividad Congresos.

Actores: Docente

Precondiciones: El actor debe estar autenticado en el sistema

Paso Acción

El sistema desplegará un menú, en donde el actor

1. deberá seleccionar las opciones: “Planificación”, “Gestionar Planificación” y “Congresos”

El sistema desplegará una pantalla con los campos registrados:

- Código del congreso
- Código de la planificación
- Código IES
- Tipo de congreso
- Medio de publicación (Contiene lista desplegable)
- Memoria
- Capítulo

Nombre del caso Gestionar Congresos.

de uso:

- Revista
- Otro
- Campo para especificar
- Código de la publicación (DOI)
- Nombre de ponencia
- Nombre del evento
- Edición del evento
- Organizador el evento
- País
- Ciudad
- Fecha de publicación
- Detalle de publicación del congreso
- Campo Amplio (Lista desplegable según los datos correspondientes en el anexo)
- Campo Detallado (Lista desplegable según los datos correspondientes en el anexo)
- Campo Específico (Lista desplegable según los datos correspondientes en el anexo)
- Participación
- Autor

Nombre del caso Gestionar Congresos.

de uso:

- Coautor

El sistema permitirá ingresar un autor y máximo tres coautores

De cada uno se ingresará

- Identificación del participante
- Participación
- Nombres Completos
- Departamento
- Sede
- Duración
- Completado
- Aprobación
- Observaciones
- Acciones para Visualizar y Modificar congreso.

2. **Si el actor pulsa sobre la opción: “Modificar Congreso”**

El sistema presentará una tabla con la información del congreso la cual es:

Nombre del caso Gestionar Congresos.

de uso:

- Código del congreso
- Código de la planificación
- Código IES
- Tipo de congreso
- Medio de publicación (Contiene lista desplegable)
 - Memoria
 - Capítulo
 - Revista
 - Otro
 - Campo para especificar
- Código de la publicación (DOI)
- Nombre de Ponencia
- Nombre del evento
- Edición del evento
- Organizador el evento
- País
- Ciudad
- Fecha de publicación
- Detalle de publicación del congreso

Nombre del caso Gestionar Congresos.

de uso:

- Participación
 - Autor
 - Coautor

El sistema permitirá ingresar un autor y máximo tres coautores

De cada uno se ingresará

- Identificación del participante
- Participación
- Nombres Completos
- Departamento
- Sede
- Duración
- Completado
- Aprobación
- Observaciones

3. El actor podrá modificar la información ingresada anteriormente.

4. El actor deberá pulsar en “Modificar Congreso”, el sistema guardará los cambios en la base de datos.

Nombre del caso Gestionar Congresos.

de uso:

5. Si el actor pulsa sobre la opción: “Visualizar Congreso”

El sistema presentará una pantalla en donde se mostrará la información del congreso la cual es:

- Código del congreso
- Código de la planificación
- Código IES
- Tipo de congreso
- Medio de publicación (Contiene lista desplegable)
 - Memoria
 - Capítulo
 - Revista
 - Otro

Campo para especificar

- Código de la publicación (DOI)
- Nombre de Ponencia
- Nombre del evento
- Edición del evento

Nombre del caso Gestionar Congresos.

de uso:

- Organizador el evento
- País
- Ciudad
- Fecha de publicación
- Detalle de publicación del congreso
- Participación
 - Autor
 - Coautor

El sistema permitirá ingresar un autor y máximo tres coautores

De cada uno se ingresará

- Identificación del participante
- Participación
- Nombres Completos
- Departamento
- Sede
- Duración
- Completado
- Aprobación
- Observaciones

Nombre del caso de uso:	Gestionar Congresos.
--------------------------------	----------------------

Postcondiciones:	<p>El proceso de gestión de Congresos ha resultado exitoso.</p> <p>El sistema desplegará la ventana principal de “Congreso” donde se podrá visualizar la lista de Congresos con su información.</p>
-------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Paso	Acción
4,7	<p>Si el actor no ha ingresado los campos requeridos correctamente.</p> <p>E.1 El sistema no habilitará la opción guardar los datos del Congreso en la base de datos.</p>
4,7	<p>Si el evaluador no aprueba la planificación del docente</p> <p>E1 El sistema no permitirá ingresar la información de la actividad.</p>

Nota. La tabla muestra todos los campos y variables que contienen el caso de uso Gestionar Congresos.

Tabla 16

Caso de Uso Evaluar Congresos

Nombre del caso Evaluar Congresos.

de uso:

Módulo: Planificación

Descripción: El actor evalúa la información ingresada en la actividad Congresos.

Actores: Coordinador de Investigación

Precondiciones: El actor debe estar autenticado en el sistema

Paso Acción

El sistema desplegará un menú, en donde el actor

1. deberá seleccionar las opciones: “Planificación”, “Evaluar Planificación” y “Evaluar Congresos”

El sistema desplegará una pantalla con los campos registrados:

Secuencia normal:

- Código del congreso
- Código IES
- Tipo de congreso
- Medio de publicación (Contiene lista desplegable)
 - Memoria
 - Capítulo

Nombre del caso Evaluar Congresos.

de uso:

- Revista
- Otro
- Campo para especificar
- Código de la publicación (DOI)
- Nombre de ponencia
- Nombre del evento
- Edición del evento
- Organizador el evento
- País
- Ciudad
- Fecha de publicación
- Campo Amplio (Lista desplegable según los datos correspondientes en el anexo)
- Campo Específico (Lista desplegable según los datos correspondientes en el anexo)
- Campo Detallado (Lista desplegable según los datos correspondientes en el anexo)
- Participación
 - Autor
 - Coautor

Nombre del caso Evaluar Congresos.

de uso:

- Identificación del participante
- Participación
- Nombres Completos
- Departamento
- Sede
- Duración
- Completado
- Aprobación
- Observaciones
- Estado se Seguimiento
- Acciones para Visualizar y Modificar congreso.

2. Si el actor pulsa sobre la opción: “Modificar Libro”

Se podrá únicamente visualizar los siguientes campos:

- Código del congreso
- Código IES
- Tipo de congreso
- Medio de publicación
- Código de la publicación (DOI)

Nombre del caso Evaluar Congresos.

de uso:

- Nombre de Ponencia
- Nombre del evento
- Edición del evento
- Organizador el evento
- País
- Ciudad
- Fecha de publicación
- Participación (Autores y coautores)
- Duración
- Nombre del proyecto Asociado (En caso de haber proyecto)

Se podrá modificar:

- Decisión del Evaluador (Se despliega una lista)
 - Aprobada
 - Pendiente
 - No aprobada
- Observaciones

3. El actor deberá pulsar en “Modificar Congreso”, el sistema guardará los cambios en la base de datos.

Nombre del caso Evaluar Congresos.

de uso:

4. Si el actor pulsa sobre la opción: “Visualizar Congreso”

El sistema presentará una pantalla en donde se mostrará la información del congreso la cual es:

- Código del congreso
- Código de la planificación
- Código IES
- Tipo de congreso
- Medio de publicación
- Código de la publicación (DOI)
- Nombre de Ponencia
- Nombre del evento
- Edición del evento
- Organizador el evento
- País
- Ciudad
- Fecha de publicación
- Detalle de publicación del congreso
- Participación (Autores y Coautores)

Nombre del caso Evaluar Congresos.

de uso:

- Duración
- Documentos de seguimiento
 - Documento de respaldo
 - Documento Acta/DEP
 - Documento informe seguimiento
 - Porcentaje de avance de seguimiento
- Sección Entregables
 - Porcentaje de avance de entregables
- Documentos de cierre
 - Documento Acta/PUB
 - Documento Matriz
 - Porcentaje de avance de cierre
 - Porcentaje de avance total
- Decisión del Evaluador (Se puede visualizar)
 - Aprobada
 - Pendiente
 - No aprobada
- Observaciones

Nombre del caso de uso:	Evaluar Congresos.
--------------------------------	--------------------

Postcondiciones:	<p>El proceso de gestión de Congresos ha resultado exitoso.</p> <p>El sistema desplegará la ventana principal de “Congresos” donde se podrá visualizar la lista de Congresos con su información.</p>
-------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Paso	Acción
4,7	<p>Si el actor no ha ingresado los campos requeridos correctamente.</p> <p>E.1 El sistema no habilitará la opción guardar los datos del Congreso en la base de datos.</p>
4,7	<p>Si el actor ingresa otro tipo de datos en los campos requeridos.</p> <p>E.1 El sistema no habilitará la opción guardar Congresos en la base de datos.</p>

Nota. La tabla muestra todos los campos y variables que contienen el caso de uso Evaluar Congresos.

Tabla 17

Caso de Uso visualizar Reportes

Nombre del caso de uso:	Visualizar reportes.
--------------------------------	----------------------

Módulo:	Planificación
Descripción:	El usuario podrá visualizar los reportes de las actividades ingresadas.
Actores:	Coordinador de Investigación, docente, director de departamento, Analista UGI.
Precondiciones:	El actor debe estar autenticado en el sistema

Paso	Acción
1	El sistema desplegará un menú, en donde el actor deberá seleccionar las opciones: “planificación”, “reportes planificación” y “reportes”
2	El sistema desplegará una pantalla con el icono y la acción de “Seleccionar”

Secuencia normal:	<p>Si el actor pulsa sobre la opción “Visualizar reporte”</p> <p>El sistema presentará una tabla con los datos de la planificación:</p> <ul style="list-style-type: none"> • Código de la planificación, • Fecha de registro, • Número de horas planificadas,
--------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Nombre del Visualizar reportes.

caso de uso:

- Número de horas planificadas completas,
 - Número de horas planificadas incompletas,
 - Cantidad de actividades planificadas,
 - Una tabla en donde se muestra las actividades con los siguientes ítems:
 - Tipo de investigación
 - Duración
 - Título de la investigación
 - Función
 - Horas semanales por actividad
 - Horas semanales declaradas por el docente
 - Estado de desarrollo, estado de aprobación.
 - Se visualizará un diagrama que muestre el porcentaje de avance de las actividades.
 - Se podrá filtrar por actividad, actividades completas, actividades incompletas.
-

Si desea ver todos los casos de uso, puede visitar los siguientes repositorios, usted encontrará a detalle los requisitos levantados por la Unidad de Gestión de Investigación <https://gitlab.espe.edu.ec/Tesis/Inventigacion/backend/GestionInvestigacionBackend> y <https://gitlab.espe.edu.ec/Tesis/Inventigacion/fontend/GestionInvestigacionFrontend>

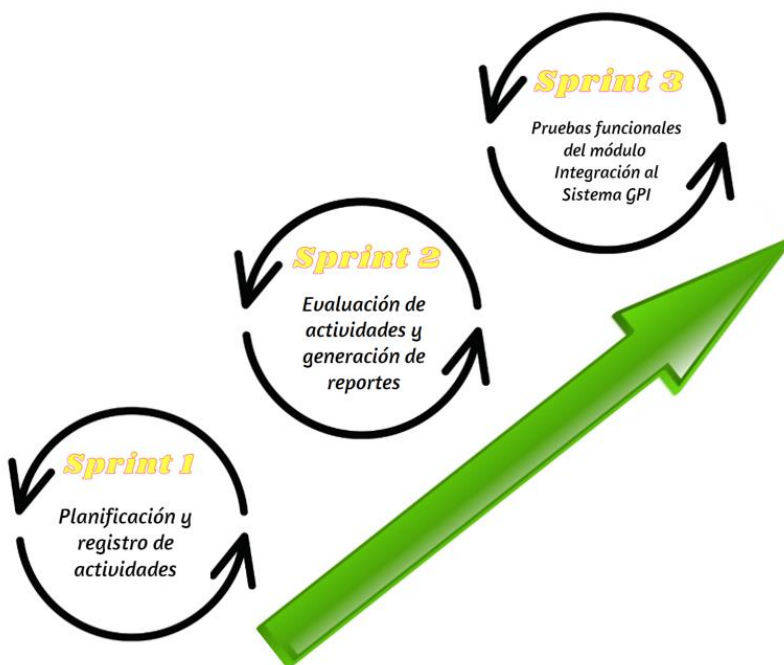
Sprint Backlog

Una vez recopilados los requisitos en el ERS y analizados los actores/usuarios presentados en la figura 6 (diagrama de casos de uso) de este documento, procedemos a dividir las tareas en intervalos de tiempo, cada Sprint durará dos meses y estará compuesto por las historias de usuario que fueron tomadas de los casos de uso.

A continuación, se explica como estarán conformados los Sprint para el desarrollo del módulo.

Figura 7

Sprint Backlog del Desarrollo del Módulo.

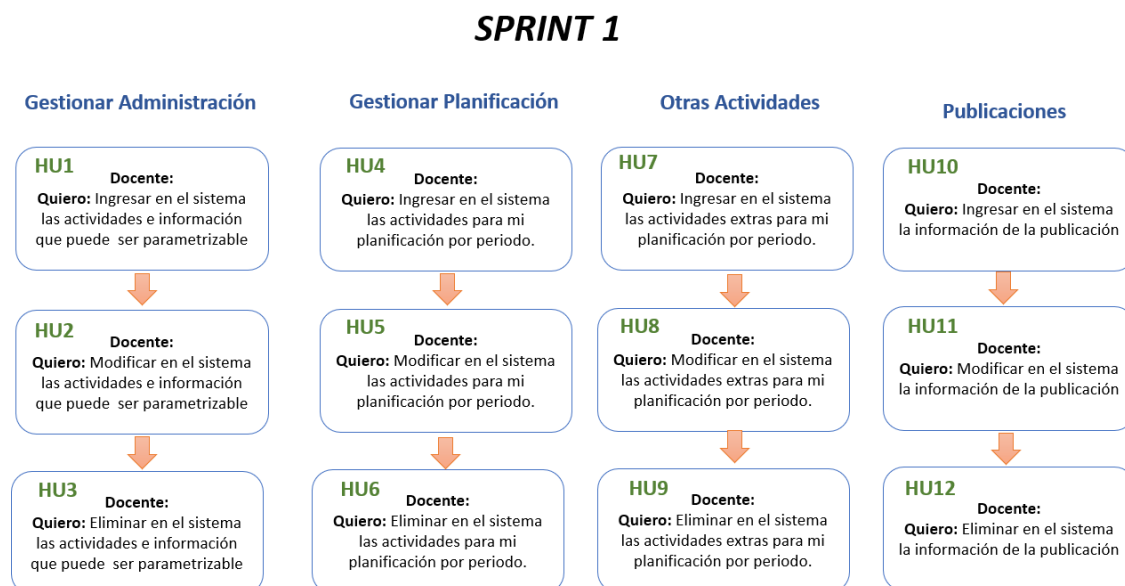


Nota. Se presenta los tres Sprints que conforman el proyecto.

El primer Sprint estará conformado por las siguientes historias de usuario

Figura 8

Historias de Usuario Sprint 1 parte 1

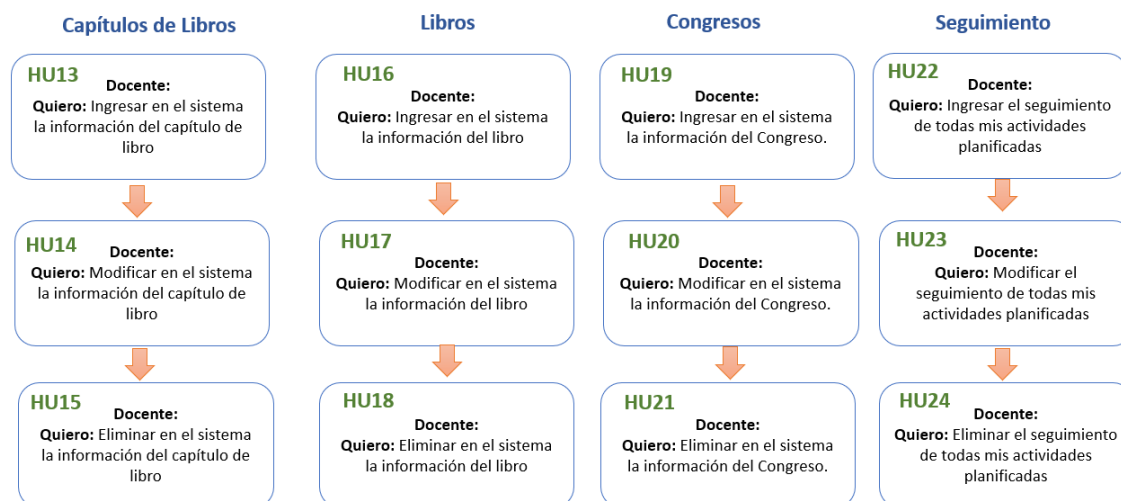


Nota. En la figura se muestra las historias de usuario por cada caso de uso.

Figura 9

Historias de Usuario Sprint 1 parte 2

SPRINT 1

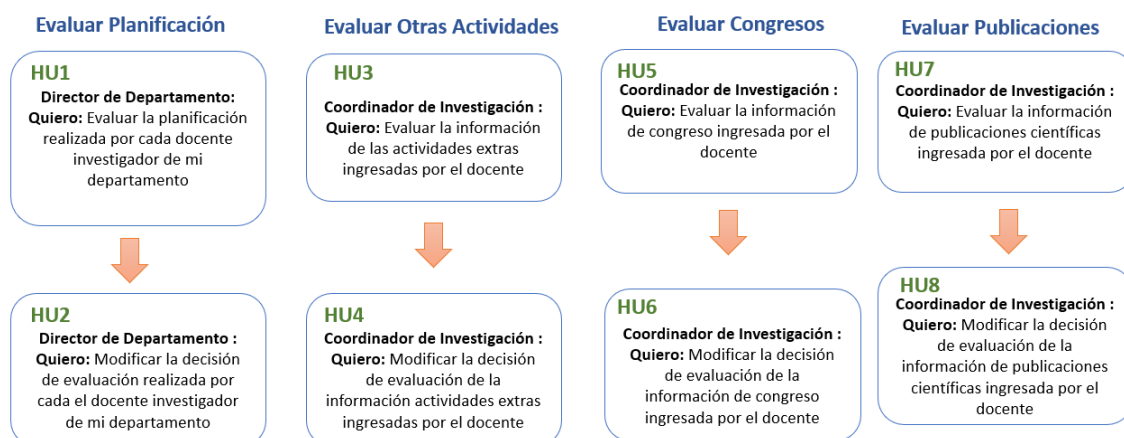


Nota. En la figura se muestra las historias de usuario por cada caso de uso.

Figura 10

Historias de Usuario Sprint 2 parte 1

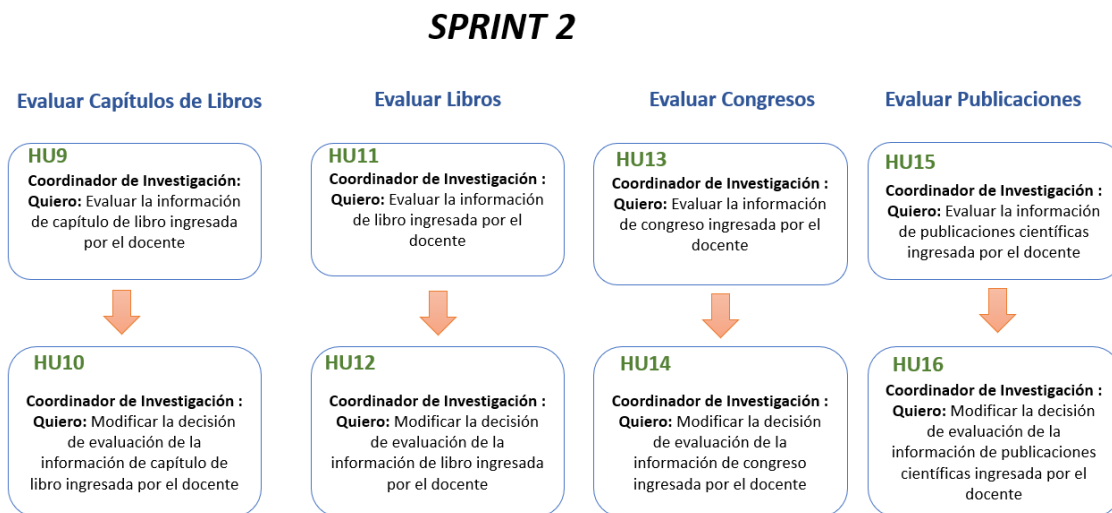
SPRINT 2



Nota. En la figura se muestra las historias de usuario por cada caso de uso.

Figura 11

Historias de Usuario Sprint 2 parte 2

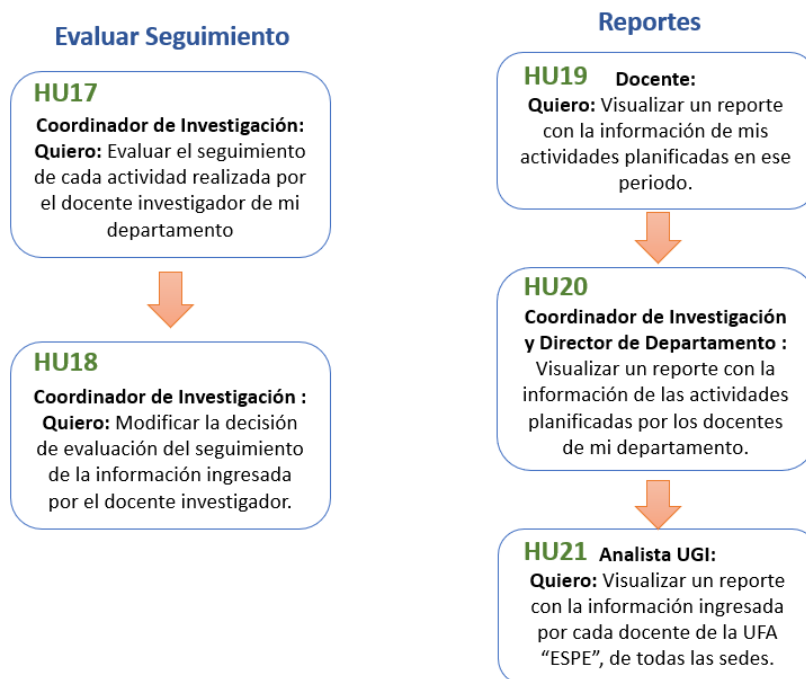


Nota. En la figura se muestra las historias de usuario por cada caso de uso.

Figura 12

Historias de Usuario Sprint 2 parte 3

SPRINT 2

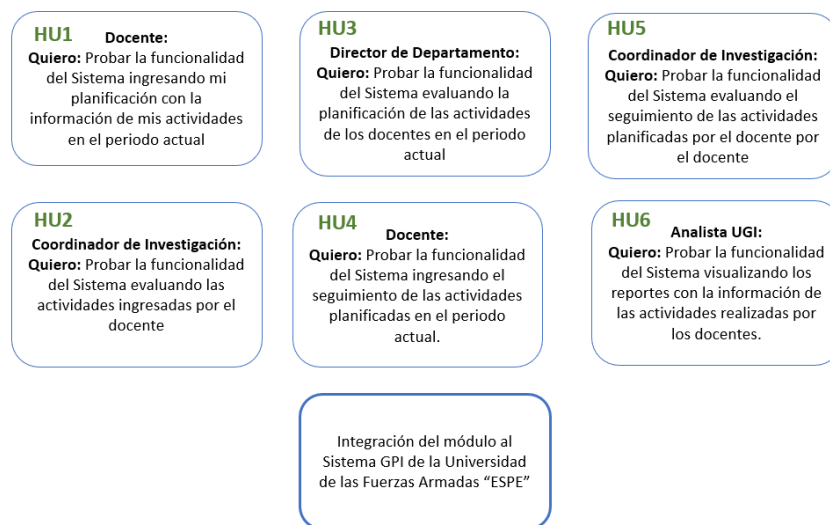


Nota. En la figura se muestra las historias de usuario por cada caso de uso.

Figura 13

Historias de Usuario Sprint 3

SPRINT 3



Nota. En la figura se muestra las historias de usuario para el sprint 3.

Diseño de la Base de Datos

Para la administración de los datos se ha seleccionado un modelo de base de datos relacional por las propiedades con las que cuenta: ACID (atomicity, consistency, isolation, and durability). Estas características proporcionan seguridad y confiabilidad al momento de realizar transacciones en nuestros registros. Como motor de base de datos usaremos Oracle versión 12c ya que esta deberá coincidir con la versión utilizada por UTICS (Unidad de Tecnología de Información y Comunicación) de la Universidad de las Fuerzas Armadas "ESPE".

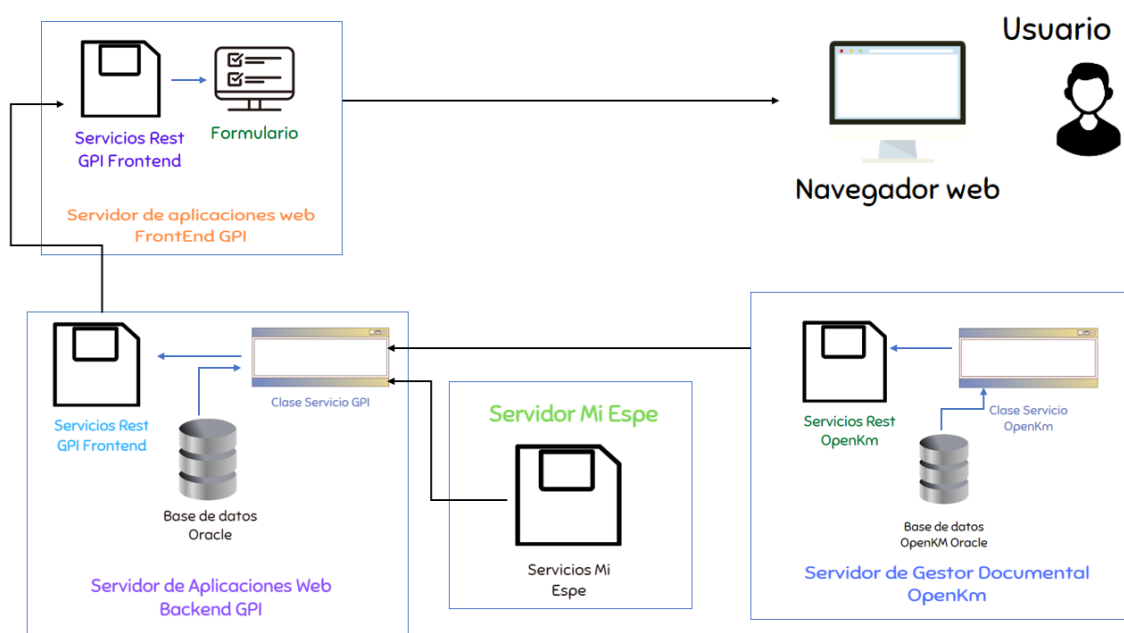
Para elaborar nuestro diseño de base de datos se utilizó la herramienta Power Designer de donde obtuvimos el modelo conceptual, físico y lógico como se presenta a continuación.

Modelo Arquitectónico de Sistema GPI

En la Universidad de las Fuerzas Armadas “ESPE” se cuenta con dos principales servidores, un servidor de aplicaciones Web del lado del Back End en donde se encuentra la lógica del negocio e intervine el motor de base de datos Oracle 12C. Además, utilizaremos “Spring boot” que es una tecnología que nos facilita el desarrollo de software y nos permite crear aplicaciones autocontenidas. Por el lado del otro Servidor tenemos el Front End, para lo cual utilizaremos “Angular 10” en donde almacenaremos los formularios y desarrollaremos las interfaces que interactúan con el usuario. Se requiere también almacenar documentos para su visualización y su descarga, para esto utilizaremos OpenKm que es un gestor de aplicaciones.

Figura 17

modelo arquitectónico del sistema GPI



Nota. Figura modelo Arquitectónico del servidor de Aplicaciones del GP.

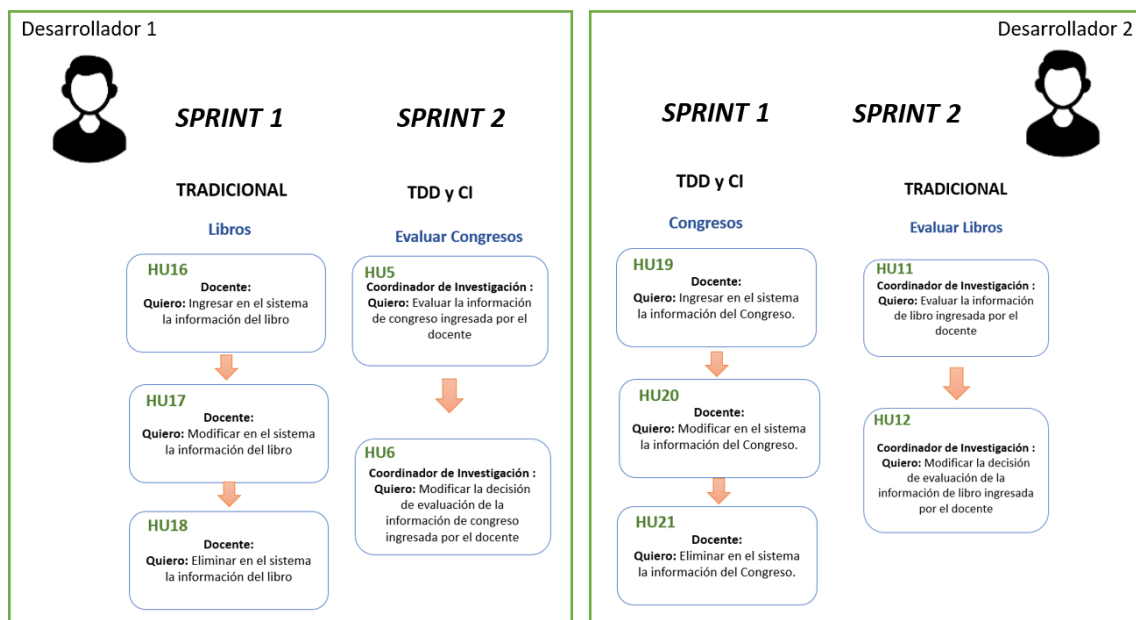
Método de desarrollo para análisis de resultados:

Para probar los objetivos que nos hemos planteado en este proyecto, elaboramos un plan de desarrollo, escogimos las historias de usuario que se van a realizar con el método de desarrollo tradicional y las historias de usuario en donde se aplicará la metodología Test Driven Development en conjunto con la técnica de Integración Continua. Las historias de usuario escogidas son muy similares entre sí, de tal manera que se apliquen los mismos escenarios en los casos de prueba.

Dos desarrolladores realizaremos las tareas correspondientes con el Sprint 1 y Sprint 2, alternaremos el desarrollo con las diferentes metodologías y técnicas para obtener resultados comparables en su totalidad. En la siguiente figura se muestra las historias de usuario asignadas a cada desarrollador y la metodología que se aplicará.

Figura 18

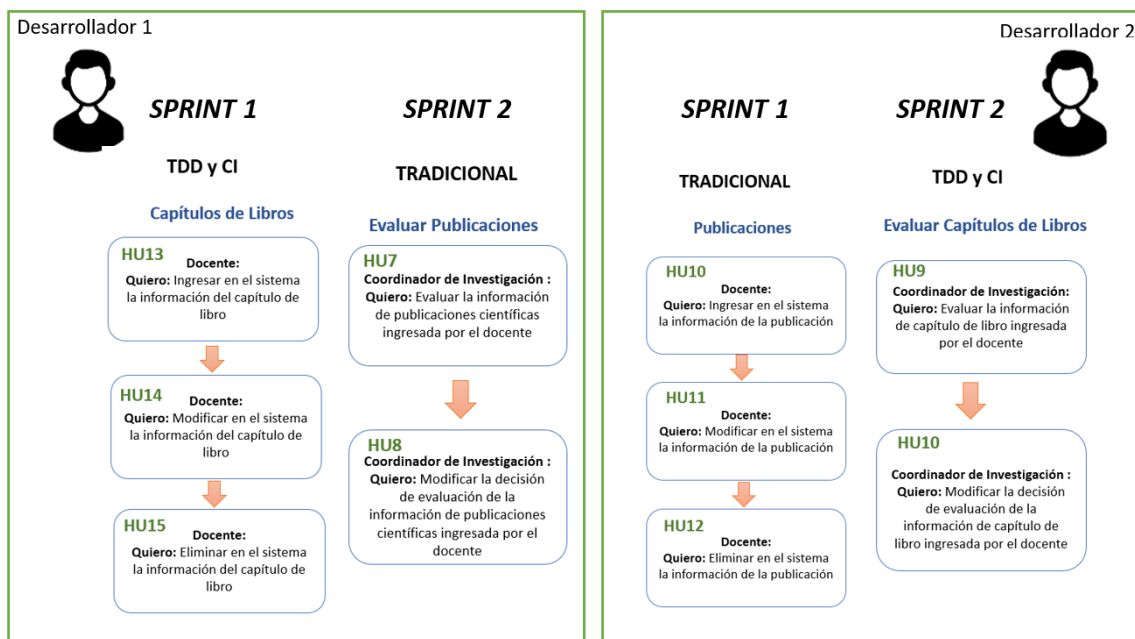
Plan de desarrollo parte 1



Nota. En la figura se muestra las historias de usuario realizadas por cada desarrollador en la parte 1.

Figura 19

plan de desarrollo parte 2



Nota. En la figura se muestra las historias de usuario realizadas por cada desarrollador en la parte 1.

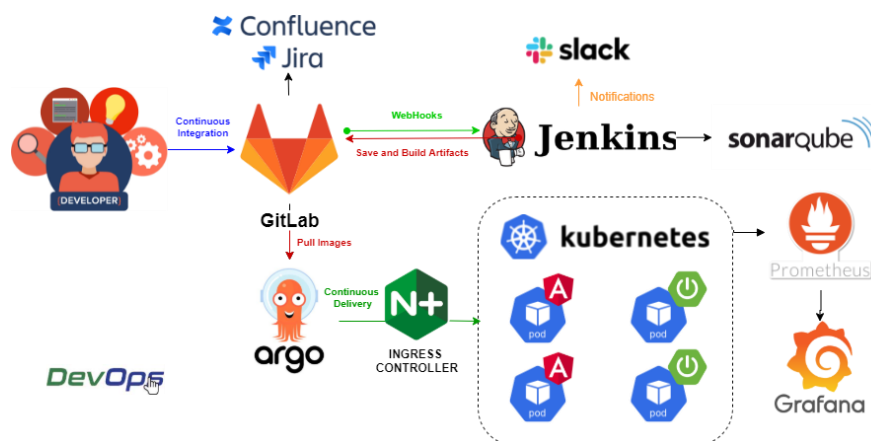
Implementación de Integración Continua

Para la implementación de la integración continua en el desarrollo del presente módulo trabajaremos en conjunto con UTIC (Unidad de Tecnología de Información y Comunicación), ya que el aplicativo debe desplegarse considerando la arquitectura de integración continua y el despliegue continuo (CI/CD) establecido para el desarrollo de proyectos de software de la Universidad.

En la siguiente imagen se observa la arquitectura de Integración Continua utilizada en la Universidad.

Figura 20

Arquitectura de Integración Continua de la UFA "ESPE"



Nota. Imagen tomada de (Espe 2018).

Usaremos Gitlab, que es un servicio web que nos ayudará con el control de versiones de un proyecto, también la herramienta Open Source Jenkins, que se utiliza para compilar y probar proyectos de forma continua a través de pipelines. El objetivo será dividir los entregables en tareas pequeñas y de esta forma lograr la integración del módulo. Los componentes restantes de la arquitectura mostrada no fueron utilizados en el presente proyecto.

CAPÍTULO IV: Desarrollo y Pruebas

Definición del Product Backlog

Una vez que se tiene la descripción de casos de uso y las historias de usuario por cada Sprint, procedemos a determinar los *features* del desarrollo del módulo como se muestra en la siguiente tabla.

Tabla 18

Lista de Features

Features

Gestionar Administración

Gestionar Planificación

Gestionar Libros

Gestionar Capítulos de Libros

Gestionar Congresos

Gestionar Publicaciones

Gestionar Otras Actividades

Autorizar Planificación

Evaluar Libros

Evaluar Capítulos de Libros

Evaluar Congresos

Evaluar Publicaciones

Evaluar Otras Actividades

Evaluar Libros

Ingresar Seguimiento

Evaluar Seguimiento de las Actividades

Visualizar Reportes

Nota. En la tabla se muestran los Features del desarrollo del Módulo.

A continuación, se determinan las tareas a realizar por cada *feature* en la siguiente tabla.

Tabla 19

Tareas de cada feature

Feature	Tarea
Gestionar Administración	Realizar el ingreso en el sistema de las actividades de investigación de los docentes.
	Modificar en el sistema las actividades de investigación de los docentes.
	Eliminar actividades en el sistema
Gestionar Planificación	Ingresar en el sistema los datos informativos para la planificación de las actividades de investigación a realizar por parte del docente.
	Modificar en el sistema la información de la planificación de las actividades.
	Eliminar actividades específicas de la planificación.
	Eliminar toda la planificación

Feature	Tarea
Gestionar Libros	Ingresar en el sistema la información de libros.
	Modificar en el sistema la información de Libros ingresada por el docente.
	Eliminar la información de Libros ingresada por el docente.
Gestionar Capítulo de Libro	Ingresar en el sistema la información de Capítulo de Libros.
	Modificar en el sistema la información de Capítulo de Libros ingresada por el docente.
	Eliminar la información de Capítulo de Libros ingresada por el docente.
	Ingresar en el sistema la información de Congresos.

Feature	Tarea
Gestionar Congresos	Modificar en el sistema la información de Congresos ingresada por el docente.
	Eliminar la información de Congresos ingresada por el docente.
	Ingresar en el sistema la información de publicaciones.
Gestionar Publicaciones	Modificar en el sistema la información de publicaciones ingresada por el docente.
	Eliminar la información de publicaciones ingresada por el docente.
	Ingresar en el sistema la información de Otras Actividades.

Feature	Tarea
Gestionar Otras Actividades	Modificar en el sistema la información de Otras Actividades ingresada por el docente.
	Eliminar la información de Otras Actividades ingresada por el docente.
Autorizar Planificación	Ingresar al sistema y visualizar las planificaciones realizadas por el docente
	Ingresar a cada planificación y visualizar la información con las actividades.
	Aprobar o No aprobar la planificación realizada por el docente.
	Ingresar al sistema y visualizar la información de libros ingresada por el docente

Feature	Tarea
Evaluar Libros	Aprobar o No aprobar la actividad de Libro realizada por el docente.
	Modificar la Aprobación o No aprobación del Libro cuando se requiera.
	Ingresar al sistema y visualizar la información de Capítulo de Libro ingresada por el docente
Evaluar Capítulo de Libro	Aprobar o No aprobar la actividad de Capítulo de Libro realizada por el docente.
	Modificar la Aprobación o No aprobación del Capítulo de Libro cuando se requiera.

Feature	Tarea
Evaluar Congreso	<p>Ingresar al sistema y visualizar la información de Congreso ingresada por el docente</p> <p>Aprobar o No aprobar la actividad de Congreso realizada por el docente.</p> <p>Modificar la Aprobación o No aprobación del Congreso cuando se requiera.</p>
Evaluar Publicaciones	<p>Ingresar al sistema y visualizar la información de Publicaciones ingresada por el docente</p> <p>Aprobar o No aprobar la actividad de Publicaciones realizada por el docente.</p> <p>Modificar la Aprobación o No aprobación del Publicaciones cuando se requiera.</p>

Feature	Tarea
Evaluar Otras Actividades	Ingresar al sistema y visualizar la información de Otras Actividades ingresada por el docente
	Aprobar o No aprobar la actividad de Otras Actividades realizada por el docente.
	Modificar la Aprobación o No aprobación del Otras Actividades cuando se requiera.
Ingresar Seguimiento por Actividad	Ingresar en el sistema el seguimiento de cada actividad planificada
	Modificar la información del seguimiento por actividad.
	Ingresar en el sistema y visualizar el seguimiento de cada actividad.

Feature	Tarea
Evaluar Seguimiento por Actividad	Aprobar o No Aprobar el seguimiento de cada actividad
Visualizar Reportes	Visualizar el reporte de cada docente con la información de las actividades planificadas en el periodo.
	Visualizar el reporte de todas las planificaciones de cada departamento.
	Visualizar el reporte de todas las planificaciones por sedes.

Desarrollo de Sprints

Para el desarrollo de cada Sprint del proyecto se estableció una duración de dos meses, se lo realizó de lunes a viernes con una estimación de cinco horas diarias.

Se realizó reuniones con los usuarios con el objetivo de solventar dudas y realizar correcciones en el documento de especificación de requisitos, en paralelo se actualizó las historias de usuario según las respuestas obtenidas.

Desarrollo del Sprint 1

Comenzamos con el desarrollo del Sprint 1 en donde escogimos las historias de usuario que se van a desarrollar con el método tradicional.

Figura 21

Historias de Usuario realizadas con el Método Tradicional



Para estimar el tiempo de duración de cada historia de usuario, se utilizó la métrica de puntos por historia considerando una escala de Likert del 1 al 5. Siendo 5 el de mayor esfuerzo y 1 el de menor esfuerzo). Cada desarrollador escogió el valor de esfuerzo, de acuerdo con su experiencia.

Para determinar la prioridad se utilizó también una escala Likert del 1 al 3, donde las historias con mayor prioridad se las calificó con 1, considerando la secuencia lógica en el proceso de desarrollo de software (crear, modificar, eliminar, consultar. A continuación, se describe cada historia de usuario con su estimación y prioridad.

Tabla 20

Estimación y prioridad por historia de usuario HU1

Como	Me gustaría	De esa manera	Prioridad	Estimación
Docente	Ingresar en el sistema la información de la publicación científica que estoy realizando.	Puedo sistematizar la información y pedir aprobación del coordinador de Investigación.	1	5

Tabla 21

Estimación y prioridad por historia de usuario HU2

Como	Me gustaría	De esa manera	Prioridad	Estimación
Docente	Modificar en el sistema la	Puedo sistematizar la información y	1	5

Como	Me gustaría	De esa manera	Prioridad	Estimación
	información de la publicación científica que estoy realizando	pedir aprobación del coordinador de Investigación.		

Tabla 22

Estimación y prioridad por historia de usuario HU3

Como	Me gustaría	De esa manera	Prioridad	Estimación
Docente	Eliminar en el sistema la información de la publicación científica ingresada.	Puedo eliminar la información de la publicación e ingresar de nuevo la información.	1	3

Tabla 23

Estimación y prioridad por historia de usuario HU4

Como	Me gustaría	De esa manera	Prioridad	Estimación
Docente	Ingresar en el sistema la información del Libro que estoy realizando.	Puedo sistematizar la información y pedir aprobación del coordinador de Investigación.	1	5

Tabla 24

Estimación y prioridad por historia de usuario HU5

Como	Me gustaría	De esa manera	Prioridad	Estimación
Docente	Modificar en el sistema la información del Libro que estoy realizando	Puedo sistematizar la información y pedir aprobación del coordinador de Investigación.	1	5

Tabla 25

Estimación y prioridad por historia de usuario HU6

Como	Me gustaría	De esa manera	Prioridad	Estimación
Docente	Eliminar en el sistema la información del libro ingresada.	Puedo eliminar la información del libro e ingresar de nuevo la información.	1	3

Tabla 26

Estimación y prioridad por historia de usuario HU7

Como	Me gustaría	De esa manera	Prioridad	Estimación
Docente	Ingresar en el sistema la información del Seguimiento de mis actividades ingresadas	Puedo sistematizar la información y pedir aprobación del coordinador de Investigación.	1	5

Tabla 27

Estimación y prioridad por historia de usuario HU8

Como	Me gustaría	De esa manera	Prioridad	Estimación
Docente	Modificar en el sistema la información del Seguimiento que estoy realizando	Puedo sistematizar la información y pedir aprobación del coordinador de Investigación.	1	5

Tabla 28

Estimación y prioridad por historia de usuario HU9

Como	Me gustaría	De esa manera	Prioridad	Estimación
Docente	Eliminar en el sistema la información del Seguimiento ingresada.	Puedo eliminar la información del libro e ingresar de nuevo la información.	1	3

De manera análoga al desarrollo tradicional, a continuación, se muestran las historias de usuario realizadas aplicando las técnicas de TDD y CI.

Figura 22

Historias de Usuario TDD y CI

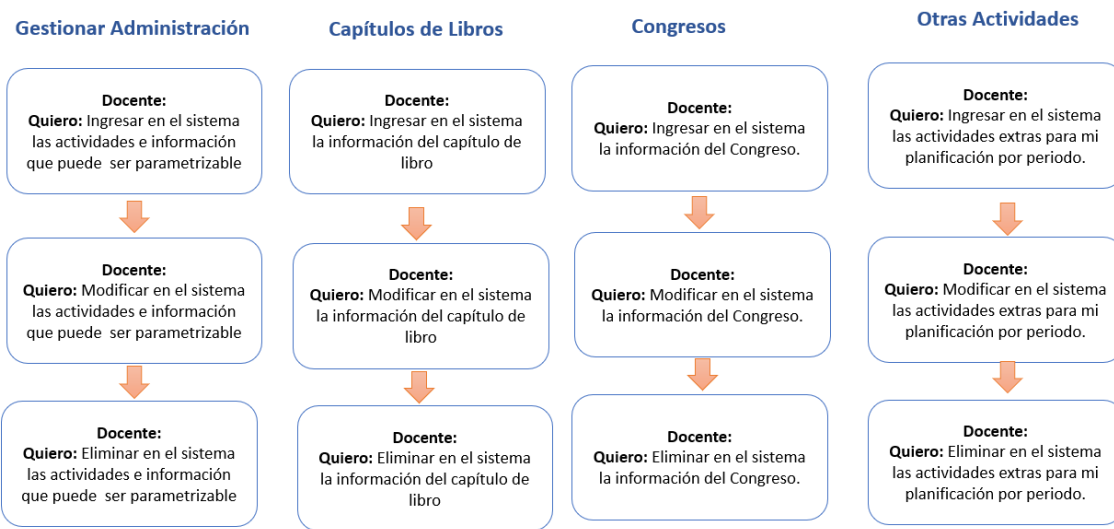
SPRINT 1**HISTORIAS DE USUARIO DESARROLLADAS CON INTEGRACIÓN CONTINUA Y TDD**

Tabla 29

Estimación y prioridad por historia de usuario HU10

Como	Me gustaría	De esa manera	Prioridad	Estimación
Docente	Ingresar en el sistema la información que puedo	Puedo parametrizar la información y hacer que el sistema se adapte a	1	5

Como	Me gustaría	De esa manera	Prioridad	Estimación
	parametrizar (Actividades, cuartiles, etc.)	los cambios establecidos cada periodo.		

Tabla 30

Estimación y prioridad por historia de usuario HU11

Como	Me gustaría	De esa manera	Prioridad	Estimación
Docente	Modificar en el sistema la información que puedo parametrizar (Actividades, cuartiles, etc.)	Puedo parametrizar la información y hacer que el sistema se adapte a los cambios establecidos cada periodo.	1	5

Tabla 31

Estimación y prioridad por historia de usuario HU12

Como	Me gustaría	De esa manera	Prioridad	Estimación
Docente	Eliminar en el sistema la información que ya no se requiere parametrizar.	Puedo eliminar la información que ya no se va a utilizar en el sistema.	1	3

Tabla 32

Estimación y prioridad por historia de usuario HU13

Como	Me gustaría	De esa manera	Prioridad	Estimación
Docente	Ingresar en el sistema la información del Capítulo de Libro de mis actividades ingresadas	Puedo sistematizar la información y pedir aprobación del coordinador de Investigación.	1	5

Tabla 33

Estimación y prioridad por historia de usuario HU14

Como	Me gustaría	De esa manera	Prioridad	Estimación
Docente	Modificar en el sistema la información del Capítulo de Libro que estoy realizando	Puedo sistematizar la información y pedir aprobación del coordinador de Investigación.	1	5

Tabla 34

Estimación y prioridad por historia de usuario HU15

Como	Me gustaría	De esa manera	Prioridad	Estimación
Docente	Eliminar en el sistema la información del Capítulo de Libro ingresada.	Puedo eliminar la información del libro e ingresar de nuevo la información.	1	3

Tabla 35

Estimación y prioridad por historia de usuario HU16

Como	Me gustaría	De esa manera	Prioridad	Estimación
Docente	Ingresar en el sistema la información del Congresos de mis actividades ingresadas	Puedo sistematizar la información y pedir aprobación del coordinador de Investigación.	1	5

Tabla 36

Estimación y prioridad por historia de usuario HU17

Como	Me gustaría	De esa manera	Prioridad	Estimación
Docente	Modificar en el sistema la información del Congreso que estoy realizando	Puedo sistematizar la información y pedir aprobación del coordinador de Investigación.	1	5

Tabla 37

Estimación y prioridad por historia de usuario HU18

Como	Me gustaría	De esa manera	Prioridad	Estimación
Docente	Eliminar en el sistema la información del Congresos ingresada.	Puedo eliminar la información del libro e ingresar de nuevo la información.	1	3

Tabla 38

Estimación y prioridad por historia de usuario HU19

Como	Me gustaría	De esa manera	Prioridad	Estimación
Docente	Ingresar en el sistema la información de Otras Actividades de mis actividades ingresadas	Puedo sistematizar la información y pedir aprobación del coordinador de Investigación.	1	5

Tabla 39

Estimación y prioridad por historia de usuario HU20

Como	Me gustaría	De esa manera	Prioridad	Estimación
Docente	Modificar en el sistema la información de Otras Actividades que estoy realizando	Puedo sistematizar la información y pedir aprobación del coordinador de Investigación.	1	5

Tabla 40

Estimación y prioridad por historia de usuario HU21

Como	Me gustaría	De esa manera	Prioridad	Estimación
Docente	Eliminar en el sistema la información de Otras Actividades ingresada.	Puedo eliminar la información del libro e ingresar de nuevo la información.	1	5

A continuación, se presenta el Sprint Backlog S1, que corresponde al primer Spring del proceso de desarrollo. Se presenta cada *feature*, con la tarea a realizar, el estado (sea por realizar, o realizado), así como la duración estimada en días y las fechas de inicio y finalización.

Tabla 41

Sprint Backlog S1

Feature	Tarea	Estado	Tiempo en días	Fecha Inicio	Fecha Fin
Gestionar Administración	Realizar el ingreso en el sistema de las actividades de investigación de los docentes.	Por Realizar	3	03/05/2021	05/05/2021
	Modificar en el sistema las actividades de investigación de los docentes.	Por Realizar	3	06/05/2021	10/05/2021

Gestionar Planificación	Eliminar actividades en el sistema	Por Realizar	1	11/05/2021	11/05/2021
	Ingresar en el sistema los datos informativos para la planificación de las actividades de investigación a realizar por parte del docente.	Por Realizar	3	12/05/2021	14/05/2021
	Modificar en el sistema la información de la planificación de las actividades.	Por Realizar	3	17/05/2021	19/05/2021
	Eliminar actividades	Por Realizar	1	20/05/2021	20/05/2021

	específicas de la planificación.				
	Eliminar toda la planificación	Por Realizar	1	21/05/2021	21/05/2021
	Ingresar en el sistema la información de libros.	Por Realizar	3	24/05/2021	26/05/2021
Gestionar Libros	Modificar en el sistema la información de Libros ingresada por el docente.	Por Realizar	3	27/05/2021	31/05/2021
	Eliminar la información de Libros ingresada por el docente.	Por Realizar	1	01/06/2021	04/06/2021
	Ingresar en el sistema la información de	Por Realizar	3	07/06/2021	09/06/2021

Gestionar Capítulo de Libro	Capítulo de Libros.				
	Modificar en el sistema la información de Capítulo de Libros ingresada por el docente.	Por Realizar	3	10/06/2021	14/06/2021
	Eliminar la información de Capítulo de Libros ingresada por el docente.	Por Realizar	1	15/06/2021	15/06/2021
Gestionar Congresos	Congresos.				
	Ingresar en el sistema la información de	Por Realizar	1	16/06/2021	16/06/2021
	Modificar en el sistema la información de Congresos	Por Realizar	1	17/06/2021	17/06/2021

	ingresada por el docente.				
	Eliminar la información de Congresos ingresada por el docente.	Por Realizar	1	18/06/2021	18/06/2021
	Ingresar en el sistema la información de publicaciones.	Por Realizar	3	19/06/2021	22/06/2021
Gestionar Publicaciones	Modificar en el sistema la información de publicaciones ingresada por el docente.	Por Realizar	3	23/06/2021	26/07/2021
	Eliminar la información de publicaciones	Por Realizar	1	27/06/2021	27/06/2021

	ingresada por el docente.				
	Ingresar en el sistema la información de	Por Realizar	1	28/06/2021	28/06/2021
Gestionar	Otras				
Otras	Actividades.				
Actividades	Modificar en el sistema la información de	Por Realizar	1	29/06/2021	29/06/2021
	Otras				
	Actividades				
	ingresada por el docente.				
	Eliminar la información de	Por Realizar	1	30/06/2021	30/06/2021
	Otras				
	Actividades				
	ingresada por el docente.				

Autorizar Planificación	Ingresar al sistema y visualizar las planificaciones realizadas por el docente	Por Realizar	1	31/06/2021	31/06/2021
	Ingresar a cada planificación y visualizar la información con las actividades.	Por Realizar	3	01/07/2021	03/07/2021
	Aprobar o No aprobar la planificación realizada por el docente.	Por Realizar	4	03/07/2021	06/07/2021

En la siguiente tabla, se muestran los ajustes que fueron considerados durante el proceso de desarrollo del Sprint, en función de las necesidades detectadas por los usuarios.

Tabla 42

Ajustes del Sprint

Ajustes

Ingresar la duración en cada actividad de Investigación

Ingresar en la pantalla de planificación el director de departamento

Subir y visualizar un archivo en la pantalla de Otras Actividades

Bloquear el registro de actividades si no ha sido aprobada la planificación

Cambiar las imágenes de la pantalla principal

Nota. La tabla describe los ajustes requeridos para el Sprint 1.

Entregables realizados

De acuerdo con los requerimientos planteados en el Sprint 1, se desarrollaron los módulos que automatizan los procesos definidos en la lista de features que fueron presentados en el apartado anterior. A continuación, se presentan algunas pantallas de la interfaz de usuario desarrollada.

Figura 23

Visualización de Gestionar Planificación 1

Crear Planificación

Datos de la Planificación:

Fecha de Registro:
10-07-2022

Estado de Aprobación:
En espera de Aprobación

Director de Departamento *
-

Período Académico *
202150 - PREGRADO S-I MAY21 - SEP21

Selección Grado Académico *
MASTER

Horas semanales de acuerdo a su grado académico *
Hasta un máximo de 6H semanales

Actividades Disponibles: CONGRESOS

Duración de la Actividad en...: 1

Actividades Seleccionadas (+1)

Actividad	Duración	Acciones
CONGRESOS	1	

Nota. La figura muestra el ingreso de una planificación.

Figura 24

Visualización de Gestionar Planificación 2

Tabla de Planificación

Buscar..

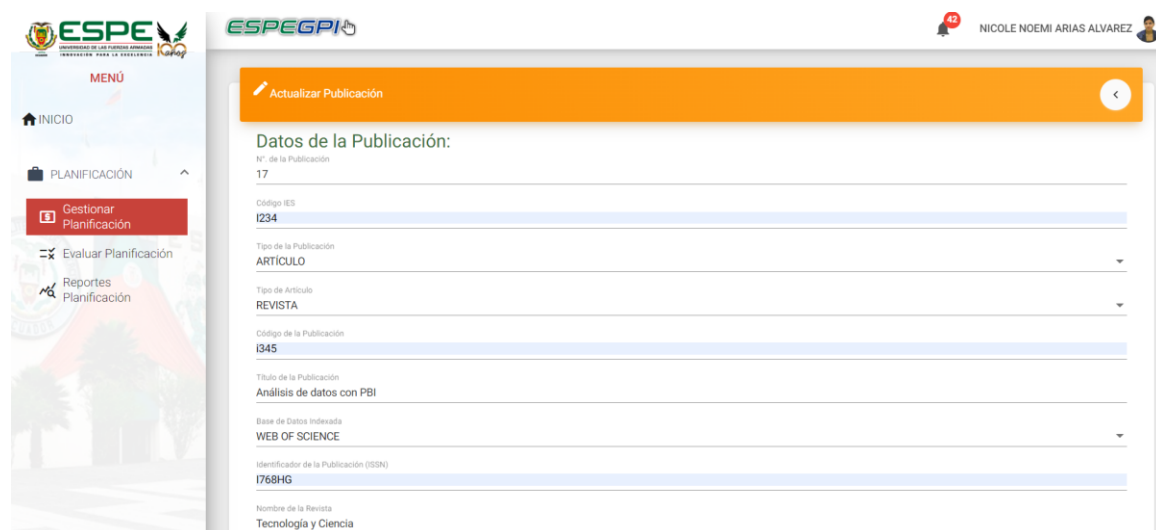
Código de Planificación	Fecha de Registro	Docente Responsable	Estado de Aprobación de Planificación	Estado de Desarrollo de Planificación
63	2022-07-10	NICOLE NOEMI ARIAS ALVAREZ	Pendiente	Completado

Duración Máxima (En horas)	Evaluador	Observaciones	Acciones
6	Magi Paul Diaz Zúñiga, Mgtr.	Ninguna	

Nota. La figura muestra la pantalla principal con la planificación registrada al lado derecho tenemos las acciones de visualizar, modificar y eliminar la planificación.

Figura 25

Visualización de Gestionar Publicaciones 1



The screenshot displays the ESPEGPI web application interface. On the left is a navigation menu with options: INICIO, PLANIFICACIÓN, Gestionar Planificación (highlighted in red), Evaluar Planificación, and Reportes Planificación. The main content area is titled 'Actualizar Publicación' and contains a form with the following fields:

Datos de la Publicación:	
N° de la Publicación	17
Código IES	I234
Tipo de la Publicación	ARTÍCULO
Tipo de Artículo	REVISTA
Código de la Publicación	I345
Título de la Publicación	Análisis de datos con PBI
Base de Datos Indexada	WEB OF SCIENCE
Identificador de la Publicación (ISSN)	I768HG
Nombre de la Revista	Tecnología y Ciencia

Nota. Ingreso de la información de publicaciones.

Figura 26

Visualización de Gestionar Publicaciones 2

The screenshot displays the ESPEGPI web application interface. On the left is a navigation menu with options: INICIO, PLANIFICACIÓN (with sub-options: Gestionar Planificación, Evaluar Planificación, Reportes Planificación), and a logo for ESPE. The main content area is titled 'Tabla de Publicaciones' and features a search bar. Below the search bar is a table with the following data:

Nº. de la Publicación	Código de Planificación	Título de la Publicación	Tipo de Publicación	Tipo de Artículo	Estado de Desarrollo de la Actividad
17	23	Análisis de datos con PBI	ARTÍCULO	REVISTA	Incompleto

Below the table is a detailed view for the selected publication, showing:

Base de Datos Indexada	Nombre de la Revista	Evaluación del Desarrollo de la Actividad	Observaciones del Evaluador	Estado de Evaluación del Seguimiento de la Actividad	Acciones
WEB OF SCIENCE	Tecnología y Ciencia	Pendiente			View, Edit

At the bottom, it indicates 'Showing 1 to 1 of 1 entries' with pagination controls.

Nota. La figura muestra la pantalla general de Publicaciones Científicas.

Figura 27

Visualización de Gestionar Libros 1

The screenshot displays the ESPEGPI web application interface for 'Visualizar Datos'. The main content area is titled 'Visualizar Datos' and contains a form for entering book information:

Datos del libro:

- Nº del libro: 61
- Código IES: _____
- Tipo de la Publicación: _____ Código de la Publicación: _____
- Título del Libro: _____
- Tipo Código ISBN: _____ Identificador del Libro ISBN: _____
- Fecha de la Publicación: _____
- Campo amplio: _____
- Campo específico: _____
- Campo Detallado: _____

Nota. La Figura muestra la información a ingresar de Libro.

Figura 28

Visualización de Gestionar Libros 2

The screenshot shows a web application interface for managing books. On the left is a sidebar menu with options like 'INICIO', 'PLANIFICACIÓN', 'Gestionar Planificación', 'Evaluar Planificación', 'Reportes', and 'Planificación'. The main area is titled 'Tabla de Libro' and contains a table with the following data:

N°. del libro	Código de Planificación	Nombre del libro	Participación	Tipo de la Publicación	Estado de Desarrollo de la Actividad
61	63				Incompleto

Below the table, there is a second table with columns for 'Fecha de la Publicación', 'Código IES', 'Evaluación del Desarrollo de la Actividad', 'Observaciones del Evaluador', 'Estado de Evaluación del Seguimiento de la Actividad', and 'Acciones'. The 'Evaluación del Desarrollo de la Actividad' column shows 'Pendiente'. At the bottom, there is a pagination control showing 'Showing 1 to 1 of 1 entries' and a dropdown menu set to '10'.

Nota. La Figura muestra la pantalla general de Libro.

Desarrollo del Sprint 2

Para el desarrollo de segundo Sprint, se siguió el mismo proceso descrito en el Sprint 1. Se ha omitido los detalles de este proceso, pero se presentan los *features* realizados.

Tabla 43

Features Sprint 2

Feature	Tarea	Estado	Tiempo en días	Fecha Inicio	Fecha Fin
Evaluar Libros	Ingresar al sistema y visualizar la información de libros ingresada por el docente	Por Realizar	1	08/07/2021	08/07/2021
	Aprobar o No aprobar la actividad de Libro realizada por el docente.	Por Realizar	1	09/07/2021	09/07/2021
	Modificar la Aprobación o No aprobación del Libro	Por Realizar	1	10/07/2021	10/07/2021

Feature	Tarea	Estado	Tiempo en días	Fecha Inicio	Fecha Fin
	cuando se requiera.				
Evaluar Capítulo de Libro	Ingresar al sistema y visualizar la información de Capítulo de Libro ingresada por el docente	Por Realizar	1	11/07/2021	11/07/2021
	Aprobar o No aprobar la actividad de Capítulo de Libro realizada por el docente.	Por Realizar	1	13/08/2021	13/08/2021

Feature	Tarea	Estado	Tiempo en días	Fecha Inicio	Fecha Fin
	Modificar la Aprobación o No aprobación del Capítulo de Libro cuando se requiera.	Por Realizar	1	14/07/2021	14/07/2021
Evaluar Congreso	Ingresar al sistema y visualizar la información de Congreso ingresada por el docente	Por Realizar	1	15/07/2021	15/07/2021
	Aprobar o No aprobar la actividad de	Por Realizar	1	16/07/2021	16/07/2021

Feature	Tarea	Estado	Tiempo en días	Fecha Inicio	Fecha Fin
	Congreso realizada por el docente.				
	Modificar la Aprobación o No aprobación del Congreso cuando se requiera.	Por Realizar	1	17/07/2021	17/07/2021
	Ingresar al sistema y visualizar la información de Publicaciones ingresada por el docente	Por Realizar	1	18/07/2021	18/07/2021
Evaluar Publicaciones					

Feature	Tarea	Estado	Tiempo en días	Fecha Inicio	Fecha Fin
	Aprobar o No aprobar la actividad de Publicaciones realizadas por el docente.	Por Realizar	1	19/07/2021	19/07/2021
	Modificar la Aprobación o No aprobación del Publicaciones cuando se requiera.	Por Realizar	1	20/07/2021	20/07/2021
	Ingresar al sistema y visualizar la información	Por Realizar	1	21/07/2021	21/07/2021

Feature	Tarea	Estado	Tiempo en días	Fecha Inicio	Fecha Fin
Evaluar Otras Actividades	de Otras Actividades ingresada por el docente	Por Realizar	1	22/07/2021	22/07/2021
	Aprobar o No aprobar la actividad de Otras Actividades realizada por el docente.	Por Realizar	1	23/07/2021	23/07/2021
	Modificar la Aprobación o No aprobación del Otras Actividades	Por Realizar	1		

Feature	Tarea	Estado	Tiempo en días	Fecha Inicio	Fecha Fin
	cuando se requiera.				
Ingresar Seguimiento por Actividad	Ingresar en el sistema el seguimiento de cada actividad planificada	Por Realizar	1	24/07/2021	24/07/2021
	Modificar la información del seguimiento por actividad.	Por Realizar	1	25/07/2021	25/07/2021
Evaluar Seguimiento por Actividad	Ingresar en el sistema y visualizar el seguimiento de cada actividad.	Por Realizar	1	26/07/2021	26/07/2021

Feature	Tarea	Estado	Tiempo en días	Fecha Inicio	Fecha Fin
Visualizar Reportes	Aprobar o No Aprobar el seguimiento de cada actividad	Por Realizar	1	27/07/2021	27/07/2021
	Visualizar el reporte de cada docente con la información de las actividades planificadas en el periodo.	Por Realizar	3	28/07/2021	28/07/2021
	Visualizar el reporte de todas las	Por Realizar	3	29/08/2021	31/08/2021

Feature	Tarea	Estado	Tiempo en días	Fecha Inicio	Fecha Fin
	planificación de cada departamento. Visualizar el reporte de todas las planificaciones por sedes.	Por Realizar	3	01/08/2021	03/08/2021

Tabla 44*Features Sprint 3*

Feature	Tarea	Estado	Tiempo en días	Fecha Inicio	Fecha Fin
Probar en el sistema el	Probar con el docente el ingreso de la planificación en un periodo seleccionado	Por realizar	4	01/08/2021	04/08/2022
Ingreso de la planificación	Modificar la planificación creada por el docente	Por realizar	4	05/08/2022	08/08/2022
	Eliminar la planificación creada por el docente.	Por realizar	4	09/08/2022	12/08/2022
Probar la evaluación de la planificación y actividades	Probar la evaluación de la planificación y la aprobación de la planificación por docente	Por realizar	5	14/08/2022	19/08/2022
	Verificar la aprobación de la planificación por docente				

Feature	Tarea	Estado	Tiempo en días	Fecha Inicio	Fecha Fin
	Verificar la información de cada actividad por docente	Por realizar	6	21/08/2022	26/08/2022
	Verificar la información del seguimiento	Por realizar	7	27/08/2022	30/08/2022
	Visualizar los reportes por docente	Por realizar	8	30/08/2022	08/09/2022
	Visualizar los reportes por coordinador de investigación	Por realizar	9	10/09/2022	22/06/2022
Visualización de reportes	Visualizar los reportes por director de carrera	Por realizar	10	23/09/2022	02/10/2022

Feature	Tarea	Estado	Tiempo en días	Fecha Inicio	Fecha Fin
	Visualizar los reportes en forma general por departamentos y sedes	Por realizar	11	03/10/2022	24/10/2022

CAPÍTULO V: Validación y resultados

Introducción

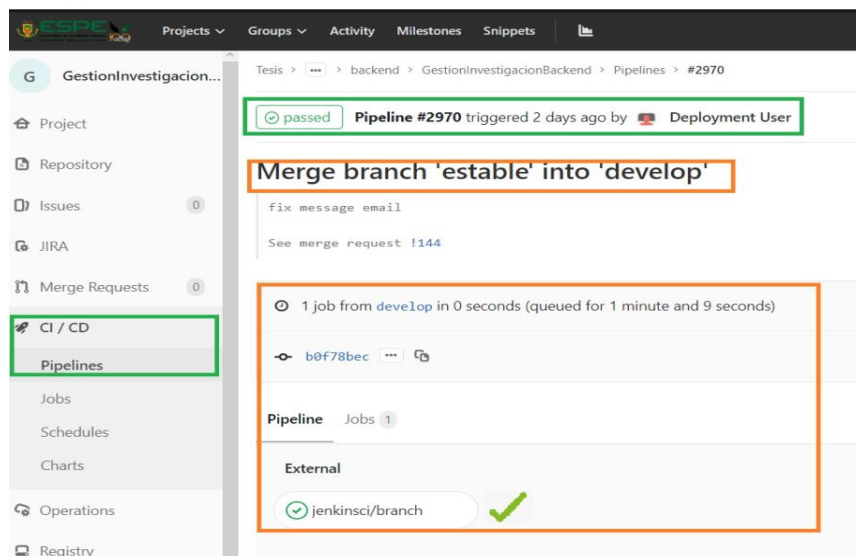
En este capítulo se realiza la validación de la aplicación de las técnicas de desarrollo basado en pruebas e integración continua. Se presentan los resultados obtenidos en función de establecer si la hipótesis planteada se cumple.

Aplicación de la técnica de integración continua CI

Mediante el uso de la arquitectura CI que está representada en la Figura 20 (Arquitectura de Integración Continua ESPE), se nos facilitó integrar nuestro módulo con el Sistema de Gestión de Proyectos de Investigación (GPI), ya que utilizamos las herramientas proporcionadas por UTICS, las cuales detectaban nuestros cambios en el código y se realizaba la Integración y Despliegue Continuo (CI/CD) automáticamente.

Figura 29

Integración Continua del módulo



Nota. En la imagen se muestra las ramas en las que se trabajó la Integración Continua.

Como se observa en la imagen, estas herramientas ayudan a optimizar los tiempos en la Integración y Despliegue Continuo, permitiendo tener el sistema actualizado en tiempo real.

Cuando las herramientas detectaban un error en la integración, se generaba una alerta en el punto específico en donde había ocurrido el fallo, lo que nos permitía resolverlo de manera inmediata.

Aplicación de la técnica de desarrollo basado en pruebas TDD

Para la aplicación de TDD se definieron los casos de prueba mediante la técnica de particiones de equivalencia. Posteriormente se diseñaron los casos y escenarios de prueba. Estos instrumentos se describen a continuación:

Particiones de Equivalencia

Para realizar las pruebas funcionales usamos la técnica de partición de equivalencia la cual nos permitió tener un conjunto de datos válidos o inválidos de cada historia de usuario y de esta manera evaluar el comportamiento de los valores con datos de entrada y salida. Además, sirvió como insumo para definir los casos de prueba necesarios para aplicar TDD.

Se ha omitido la presentación de todos los casos de prueba, sin embargo los mismos se encuentran disponibles en:

<https://gitlab.espe.edu.ec/Tesis/Investigacion/backend/GestionInvestigacionBackend> y

<https://gitlab.espe.edu.ec/Tesis/Investigacion/fontend/GestionInvestigacionFrontend>. En estos repositorios además se encuentra el ERS, el código fuente, particiones de equivalencia y casos de prueba del módulo realizado. A manera de ejemplo, se muestra la partición de equivalencia para el módulo de Congresos.

Tabla 45

Particiones de Equivalencia de Congresos

Historia de Usuario	Tipo de Dato	Clases de equivalencia válida	Clases de equivalencia inválidas
HU19	Código IES	Ingreso de 1 - 6 caracteres	Ingreso de letras
			Ingreso no mayor a 6 caracteres
		Ingreso de números	Cadena nula
			Ingreso de valores negativos
	Tipo de Congreso	Escoger opción Multidisciplinario	Cadena nula
		Escoger opción Transdisciplinario	
		Escoger opción Omnidisciplinario	
	Medio de publicación	Escoger opción Memoria	Cadena nula
		Escoger opción Capítulo de Libro	

Historia de Usuario	Tipo de Dato	Clases de equivalencia válida	Clases de equivalencia inválidas
		Escoger opción Revista	
		Escoger opción Otro (Ingreso de letras)	
	Código de publicación	Ingreso de números	cadena nula
	DOI	Ingreso de letras	Ingreso no mayor a 30 caracteres
		ingreso de caracteres especiales ". / :"	Ingreso de caracteres especiales, excepto ". / :"
	Nombre de ponencia	Ingreso de letras	Ingreso de números
			Caracteres especiales
			Ingreso no mayor a 300 caracteres
	Nombre del evento	Ingreso de letras	Ingreso de números

Historia de	Tipo de Dato	Clases de equivalencia válida	Clases de equivalencia inválidas
Usuario			Caracteres especiales
			Ingreso no mayor a 300 caracteres
Edición del evento		número entero	valores negativos
		1 < n < 100	números mayores a 100
Organizador del evento		Ingreso de letras	Cadena nula
			Cadena nula
			Ingreso de números
			Caracteres especiales
			Ingreso no mayor a 100 caracteres

Historia de Usuario	Tipo de Dato	Clases de equivalencia válida	Clases de equivalencia inválidas
	Comité organizador	Ingreso de letras	Cadena nula
			Ingreso de caracteres especiales, excepto ", - ."
		ingreso de caracteres especiales ", - ."	Ingreso de números
			Ingreso no mayor a 2000 caracteres
	País	Ingreso de letras	Ingreso de números
			Caracteres especiales
			Ingreso no mayor a 30 caracteres
			Cadena nula
	Ciudad	Ingreso de letras	Ingreso de números

Historia de	Tipo de Dato	Clases de equivalencia válida	Clases de equivalencia inválidas
Usuario			Caracteres especiales
			Ingreso no mayor a 30 caracteres
			Cadena nula
	Fecha de publicación	Ingreso de fecha mediante un calendario con formato dd/mm/aaaa	Cadena nula
	Campo amplio	Escoger opciones (Ciencias Naturales, Matemáticas, Estadística, etc.)	Cadena nula
	Campo específico	Escoger opciones (Ciencias Físicas, Medio Ambiente, etc.)	Cadena nula
	Campo detallado	Escoger opciones (Biofísica, genética, etc.)	Cadena nula

Historia de Usuario	Tipo de Dato	Clases de equivalencia válida	Clases de equivalencia inválidas
	Participación	Escoger opciones (Autor o coautor) Ingresar los coautores que participen	Cadena nula
	Autores adicionales	Identificación (Ingresar solo números)	No mayor a 10 números valores negativos letras Caracteres especiales
	Proyecto a asociar	Escoger el proyecto a asociar presentado en la lista cadena nula	No escoger más de 1 proyecto

Historia de	Tipo de Dato	Clases de equivalencia válida	Clases de equivalencia inválidas
Usuario			
	Actualizar congreso	Guardar cambios	No guardar cambios

Diseño de Casos de Pruebas

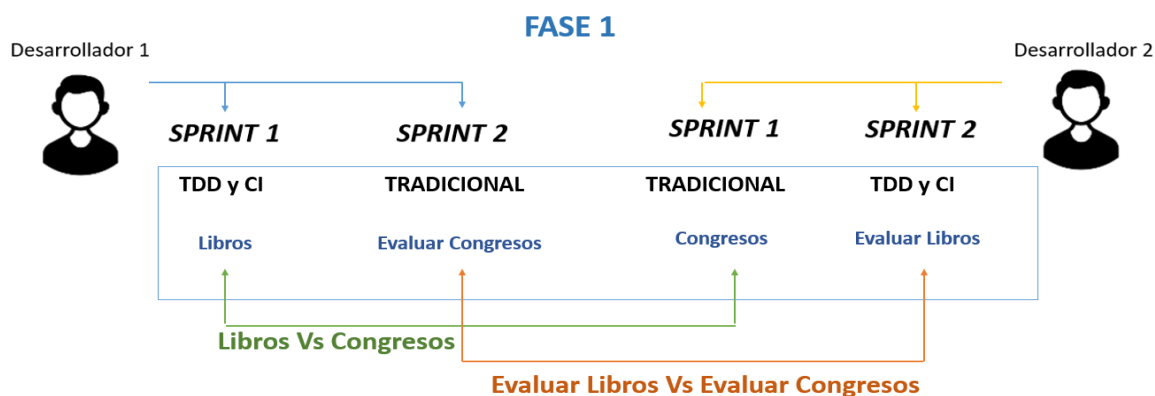
Para el diseño de casos de prueba definimos los valores condicionados de entrada de acuerdo con las particiones de equivalencia válidas e inválidas, como se muestra en la tabla 45 (Particiones de Equivalencia de Congresos). Ejecutamos distintos tipos de escenarios hasta cubrir en la totalidad las clases de equivalencia válidas e inválidas.

Para realizar los casos de prueba se compararon historias de usuario muy similares. Debido al alcance del estudio, se han escogido las historias de usuario que tienen diferencias ínfimas entre ellas para aplicar los mismos escenarios en los casos de prueba y tener un resultado comparable entre sí.

En la siguiente figura se muestra cómo se realizó la comparación entre historias de usuario de usuario.

Figura 30

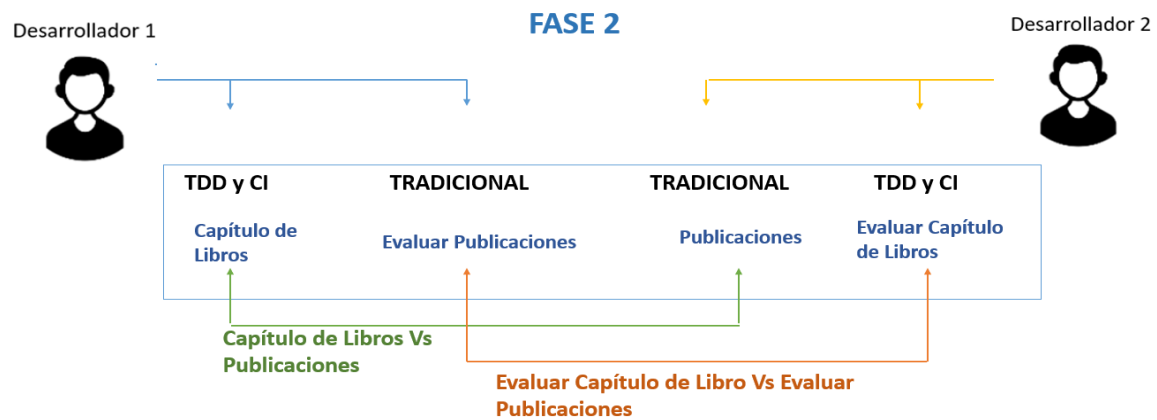
Comparación Casos de Prueba Fase 1



Nota. En la figura se muestra los casos de prueba que se compararon.

Figura 31

Comparación Casos de Prueba Fase 2



Nota. En la figura se muestra los casos de prueba que se compararon.

Escenarios de prueba

Para realizar los casos de prueba generamos escenarios iguales en los dos módulos que se compararon, dando un total de cinco casos de prueba para cada módulo, como se puede observar a continuación.

Tabla 46

Escenarios de casos de prueba

Casos de prueba Congresos VS Libros		
CP	Escenario de caso de prueba	Escenario de caso de prueba
	Congreso	Libros
CP1	Validación de actualizar Congreso y Libros con datos válidos	
CP2	Validación de actualizar Congreso y Libros enviando los campos vacíos en "campo detallado", "campo específico", "campo amplio" y los otros campos enviar válidos	
CP3	Validación de Actualizar Congreso y Libros con datos válidos e inválidos	
CP4	Validación de Actualizar congreso y Libros con datos inválidos en los campos no estáticos	
CP5	Validación de Actualizar Congreso y Libros con todos los campos vacíos	

A continuación, se muestra uno de los casos de prueba del módulo de congresos.

Tabla 47

Caso de Prueba Congreso P1

Historia de Usuario	#Caso de Prueba	Escenario del Caso de prueba	Precondición	Acción	Paso	Data	Resultado esperado	Resultado Obtenido
HU19	HU191	Validación de Actualizar congreso con datos válidos	Tener una planificación creada. Planificación aprobada por el	Hacer clic en el campo código IES e ingresar una cadena de caracteres válida	1	1079	Se visualiza el texto ingresado correctamente	Texto visualizado correctamente

Historia de Usuario	#Caso de Prueba	Escenario del Caso de prueba	Precondición	Acción	Paso	Data	Resultado esperado	Resultado Obtenido
	HU191		director de Departamento.	Hacer clic en el campo Tipo de Congreso y escoger una de las opciones existentes en el sistema.	2	Multidisciplinario	Se visualiza el texto escogido en el campo visualizado	Texto correctamente
	HU191			Hacer clic en el campo Medio de Publicación y escoger unas de las opciones existentes en el sistema.	3	Memoria	Se visualiza el texto escogido en el campo visualizado	Texto correctamente

Historia de Usuario	#Caso de Prueba	Escenario del Caso de prueba	Precondición	Acción	Paso	Data	Resultado esperado	Resultado Obtenido
	HU191			Hacer clic en el campo código de publicación DOI e ingresar el valor requerido	4	https://doi.org/10.1109/CIMPS.2018.8625625	Se visualiza el texto escogido en el campo visualizado	Texto correctamente
	HU191			Hacer clic en el campo Nombre de Ponencia e ingresar el valor requerido	5	Detection Of Behavior Patterns Through Social Networks Like Twitter	Se visualiza el texto escogido en el campo visualizado	Texto correctamente

Historia de Usuario	#Caso de Prueba	Escenario del Caso de prueba	Precondición	Acción	Paso	Data	Resultado esperado	Resultado Obtenido
	HU191			Hacer clic en el campo Nombre del evento e ingresar el valor requerido	6	Applications In Software Engineering - Proceedings	Se visualiza el texto escogido en el campo	Texto visualizado correctamente
	HU191			Hacer clic en el campo Organizador del evento e ingresar el valor requerido	7	IEEE	Se visualiza el texto escogido en el campo	Texto visualizado correctamente

Historia de Usuario	#Caso de Prueba	Escenario del Caso de prueba	Precondición	Acción	Paso	Data	Resultado esperado	Resultado Obtenido
	HU191			Hacer clic en el Fecha y escoger en el calendario el año, mes y día evento	11	28/01/2019	Se visualiza el texto escogido en el campo texto escogido	Se visualiza el texto escogido en el campo
	HU191			Hacer clic en campo amplio y escoger una de las opciones del sistema	12	Ciencias Naturales, Matemáticas Y Estadísticas	Se visualiza el texto escogido en el campo texto escogido	Se visualiza el texto escogido en el campo
	HU193			Hacer clic en campo detallado y escoger	13	Ciencias Biológicas Y Afines.	Se visualiza el texto escogido en el campo texto escogido	Se visualiza el texto escogido en el campo

Historia de Usuario	#Caso de Prueba	Escenario del Caso de prueba	Precondición	Acción	Paso	Data	Resultado esperado	Resultado Obtenido
				una de las opciones del sistema				
	HU194			Hacer clic en campo 14 específico y escoger una de las opciones del sistema		Biofarmacéutica	Se visualiza el texto escogido en el campo	Se visualiza el texto escogido en el campo
	HU195			Hacer clic en participación y escoger una de las opciones del sistema	15	Autor	Se visualiza el texto escogido en el campo	Se visualiza el texto escogido en el campo

Historia de Usuario	#Caso de Prueba	Escenario del Caso de prueba	Precondición	Acción	Paso	Data	Resultado esperado	Resultado Obtenido
	HU196			Hacer clic en Proyecto a Asociar y escoger un proyecto	16	Proyecto Bioquímica	Se visualiza el texto escogido en el campo texto escogido	Se visualiza el texto escogido en el campo
	HU197			Hacer clic en Actualizar Congreso	17	Clic en actualizar congreso	Congreso ingresado satisfactoriamente	Congreso ingresado satisfactoriamente

Nota. En la tabla se muestra los valores de cada sección que fueron tomados para realizar el caso de prueba.

De la misma forma, se muestra, uno de los casos de prueba para Libros.

Tabla 48

Caso de prueba Libro P1

Historia de Usuario	#Caso de Prueba	Escenario del Caso de prueba	Precondición	Acción	Paso	Data	Resultado esperado	Resultado Obtenido
HU16	HU161	Validación de Actualizar Libros con datos válidos	Tener una planificación creada. Planificación aprobada por el	Hacer clic en el campo código IES e ingresar una cadena de caracteres válida	1	1079	Se visualiza el texto ingresado correctamente	Texto visualizado correctamente

Historia de Usuario	#Caso de Prueba	Escenario del Caso de prueba	Precondición	Acción	Paso	Data	Resultado esperado	Resultado Obtenido
	HU161		director de Departamento.	Hacer clic en el campo Tipo de Publicación y escoger una de las opciones existentes en el sistema.	2	Multidisciplinari o	Se visualiza el texto escogido en el campo	Texto visualizado correctamente
	HU161			Hacer clic en el campo Medio de Publicación y escoger unas de las opciones	3	Memoria	Se visualiza el texto escogido en el campo	Texto visualizado correctamente

Historia de Usuario	#Caso de Prueba	Escenario del Caso de prueba	Precondición	Acción	Paso	Data	Resultado esperado	Resultado Obtenido
				existentes en el sistema.				
	HU161			Hacer clic en el campo código de publicación DOI e ingresar el valor requerido	4	L2019-UGI-01	Se visualiza el texto escogido en el campo	Texto visualizado correctamente
	HU161			Hacer clic en Título de Libro y colocar el texto	5	Termotecnia Y Máquinas Térmicas	Se visualiza el texto escogido en el campo	Texto visualizado correctamente

Historia de Usuario	#Caso de Prueba	Escenario del Caso de prueba	Precondición	Acción	Paso	Data	Resultado esperado	Resultado Obtenido
	HU161			Hacer clic en Tipo de código ISBN	6	Digital	Se visualiza el texto escogido en el campo	Texto visualizado correctamente
	HU161			Hacer clic en Identificador del libro ISBN y colocar el texto	7	978-9942-765-49-9	Se visualiza el texto escogido en el campo	Texto visualizado correctamente
	HU161			Hacer clic en el Fecha y escoger en el calendario el	8	28/01/2019	Se visualiza el texto escogido en el campo	Se visualiza el texto escogido en el campo

Historia de Usuario	#Caso de Prueba	Escenario del Caso de prueba	Precondición	Acción	Paso	Data	Resultado esperado	Resultado Obtenido
				año, mes y día evento				
	HU161			Hacer clic en campo amplio y escoger una de las opciones del sistema	9	Ingeniería, Industria Y Construcción	Se visualiza el texto escogido en el campo	Se visualiza el texto escogido en el campo
	HU161			Hacer clic en campo detallado y escoger una de	10	Ingeniería Y Profesiones Afines.	Se visualiza el texto escogido en el campo	Se visualiza el texto escogido en el campo

Historia de Usuario	#Caso de Prueba	Escenario del Caso de prueba	Precondición	Acción	Paso	Data	Resultado esperado	Resultado Obtenido
				las opciones del sistema				
	HU161			Hacer clic en campo específico y escoger una de las opciones del sistema	11	Electrónica, Automatización Y Sonido	Se visualiza el texto escogido en el campo	Se visualiza el texto escogido en el campo
	HU161			Hacer clic en participación y escoger una de	12	Autor	Se visualiza el texto escogido en el campo	Se visualiza el texto escogido en el campo

Historia de Usuario	#Caso de Prueba	Escenario del Caso de prueba	Precondición	Acción	Paso	Data	Resultado esperado	Resultado Obtenido
				las opciones del sistema				
	HU161			Hacer clic en Proyecto a Asociar y escoger un proyecto	13	Proyecto Bioquímica	Se visualiza el texto escogido en el campo	Se visualiza el texto escogido en el campo
	HU161			Hacer clic en Actualizar Libro	14	Clic en actualizar Libro	Congreso ingresado satisfactoriamente	Libro ingresado satisfactoriamente

Si se desea ver todos los casos de prueba aplicados para este análisis, se los puede encontrar en <https://gitlab.espe.edu.ec/Tesis/Inventigacion/backend/GestionInvestigacionBackend> y <https://gitlab.espe.edu.ec/Tesis/Inventigacion/fontend/GestionInvestigacionFrontend>, en estos repositorios se encuentra el ERS, el código fuente, particiones de equivalencia y casos de prueba del módulo realizado.

Resultados de la aplicación de TDD y CI

Para validar las técnicas de TDD en conjunto con CI, se optó por definir métricas para evaluar la calidad externa y la productividad.

Para evaluar la calidad externa del software ocupamos la siguiente fórmula:

$$QLTY = \frac{\text{\# de casos de prueba acertados}}{\text{\# de casos de prueba ejecutados}}$$

En donde QLTY es la calidad externa y representa el número de casos de prueba acertados sobre el número de casos de prueba ejecutados.

En la siguiente tabla se muestra los resultados de los casos de prueba ejecutados con las dos técnicas: TDD combinada con CI vs el desarrollo tradicional.

Tabla 49

Resultados Casos de Prueba

MEDICIÓN CALIDAD EXTERNA							
Método Tradicional				Método TDD y CI			
Casos de prueba	Ejecutados	Acertados	Resultado	Casos de prueba	Ejecutados	Acertados	Resultado
CP Congresos	5	2	0,40	CP Libros	5	3	0,60
CP Evaluar Congresos	5	4	0,80	CP Evaluar Libros	5	5	1,00
CP Capítulo de Libro	5	2	0,40	CP Publicaciones	5	3	0,50
CP Evaluar Publicaciones	5	3	0,50	CP Evaluar Capítulo de libro	5	5	1,00
Total			2,1/5	Total			3,1/5

Para determinar el número de casos de prueba exitosos (lo cual aparece en la columna Acertados de la tabla anterior), se siguió el siguiente protocolo:

- Si el resultado esperado es igual al resultado obtenido, en todos los pasos del escenario propuesto, entonces, el caso de prueba es exitoso.
- Si existe discrepancia en cualquiera de los pasos del escenario propuesto, entonces, el caso de prueba resultó fallido.

Los casos de prueba exitosos con el uso de TDD y CI dieron un resultado superior de 3,1 puntos en comparación a los 2,1 puntos obtenidos con la metodología tradicional.

En la tabla hemos utilizado el color amarillo para identificar los casos de prueba del Sprint 1 y el color verde para el Sprint 2, teniendo esto en cuenta observamos lo siguiente:

- En los casos de prueba que se desarrollaron con la aplicación de TDD y CI en el Sprint 1 se obtuvieron más resultados fallidos en comparación a los del Sprint 2. Esto debido a que en un comienzo no se tenía dominio de las técnicas.
- En los casos de prueba en los que se aplicó TDD y CI se obtuvieron menos resultados fallidos en comparación con la metodología tradicional.

Conforme avanzó el desarrollo íbamos perfeccionando el uso de TDD y CI, lo que condujo a obtener casos de prueba acertados en su totalidad, como es el caso de los *features* Evaluar Libro y Evaluar Capítulo de libros (Tabla 49).

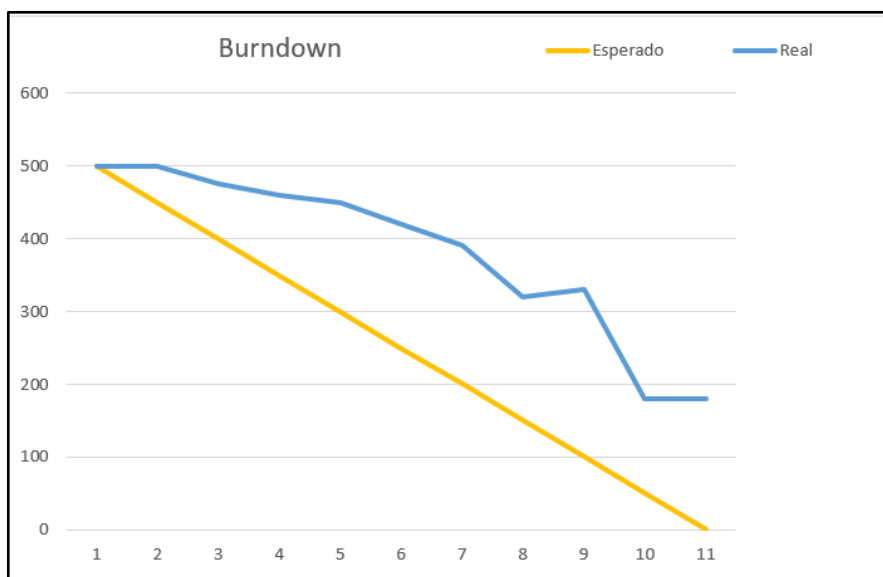
Con este análisis probamos una parte de nuestra hipótesis, la implementación de TDD y CI mejora la calidad externa del Software en comparación con la metodología tradicional.

Para evaluar la productividad usamos nuestro diagrama BurnDown Chart del Sprint uno y dos, el cual se basa en:

- El número de días que contiene cada sprint
- La estimación en puntos de las tareas esperadas por día
- Los puntos que en realidad se obtuvieron diariamente, es decir, en un día X se esperaba 50 para completar la tarea, pero en realidad solo se obtuvieron 35.

Figura 32

BurnDown Chart Sprint 1

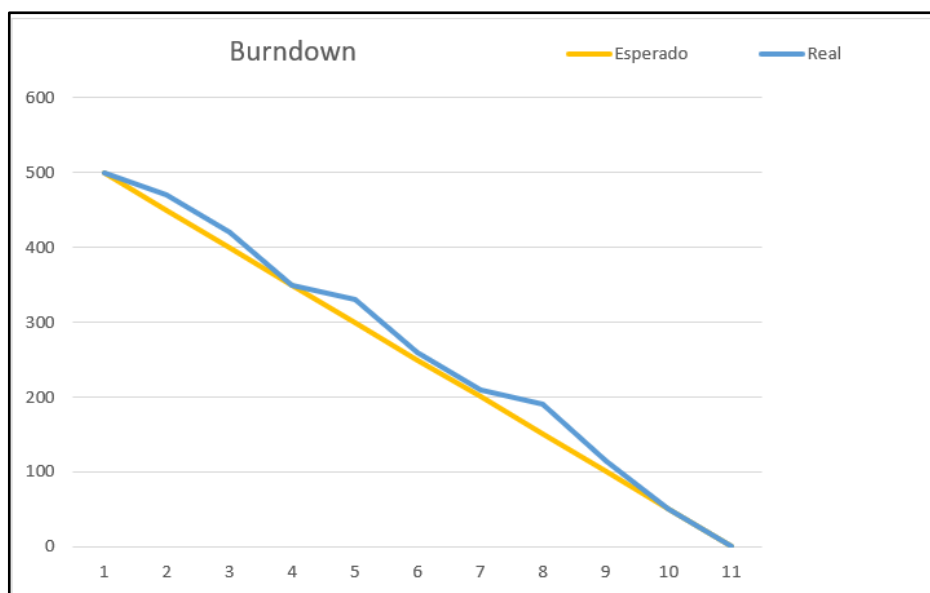


Nota. En la figura se visualiza el diagrama de tareas reales vs las esperadas para el Sprint 1.

Podemos visualizar que, para la entrega total del Sprint 1 se utilizó 5 días adicionales dado que, como se ha mencionado con anterioridad, al inicio no estábamos familiarizados con el uso de TDD y CI, por lo tanto, este tiempo adicional es un factor para tomar en cuenta cuando se implementa las técnicas por primera vez.

Adicionalmente, observamos que las historias de usuario con más correcciones a ejecutar eran en las que se aplicó el método tradicional, ya que conforme se realizó las pruebas funcionales se identificó los errores que requerían ajustes para cumplir con el requerimiento definido.

A continuación, se presenta el BurnDown Char para el Sprint 2

Figura 33*BurnDown Chart Sprint 2*

Nota. En la figura se visualiza el diagrama de tareas reales vs las esperadas para el Sprint 2.

En el Sprint 2 podemos visualizar que se cumplió con el tiempo establecido para el desarrollo de las historias de usuario. Aunque hubo historias de usuario que requirieron ajustes, con la experiencia adquirida en la aplicación de las técnicas, no existieron mayores inconvenientes en el desarrollo del aplicativo. Nuevamente, pudimos notar que las historias de usuario en donde fue necesario realizar más correcciones al aplicar los casos de prueba, fueron aquellas en las que se aplicó la metodología tradicional, pero en comparación con el Sprint 1 disminuyeron notablemente.

Después de comparar los dos BurnDown Chart observamos que, conforme íbamos adquiriendo experiencia en la implementación de las dos técnicas, mejoraban los tiempos de entrega de las historias de usuario y a su vez requerían menos ajustes, es decir, iba mejorando la calidad del software.

Con estos resultados podemos decir que, dominar la implementación de las técnicas de TDD y CI en el desarrollo de sistemas informáticos, mejora su productividad notablemente.

Amenazas a la validez

Al realizar diseños de medidas repetidas pueden presentarse amenazas a la validez del caso de estudio, sin embargo, hemos tratado de mitigar los posibles sesgos que se pueden presentar para obtener resultados válidos.

Una de las amenazas es la práctica o experiencia de los desarrolladores en las metodologías o técnicas utilizadas.

Al empezar a desarrollar el módulo no conocíamos del uso de TDD y CI, las tareas de desarrollo asignadas fueron equitativas usando las dos metodologías de desarrollo planteadas en la Figura 1 (Método tradicional) Y Figura 2 (Desarrollo combinando TDD y CI), al contrario de ser una amenaza nos sirvió para darnos cuenta de que la experiencia en la aplicación de las metodologías nos puede ayudar a obtener mejores resultados mejorando así la calidad externa y la productividad del software.

Por otra parte, por cuestión de tiempo limitado, los casos de prueba comparados no fueron totalmente iguales, fueron similares, entre ellos tenían diferencias ínfimas, lo que nos permitió plantear escenarios de casos de prueba iguales y con esto los resultados obtenidos

provinieron de una comparación equitativa, esta si pudo haber sido una amenaza a la validez de nuestro proyecto, pero logramos mitigarla en su totalidad tomando en cuenta las clases de equivalencia válidas e inválidas de cada historia de usuario.

Conclusiones

- No existen suficientes estudios que prueben que realmente TDD mejora la productividad y calidad externa del software, los experimentos que se han encontrado han tenido resultados ambiguos y no muestran con certeza las posibles mejoras que podrían obtenerse al aplicar esta técnica.
- Los resultados que se han encontrado en los estudios que aplican CI, han sido favorables en cuanto a la mejora de la productividad en el desarrollo de Software, ya que es una técnica que hace uso de herramientas que permiten actualizar de forma automática los sistemas.
- Mientras más extenso es un sistema, más compleja resulta la integración entre los equipos de desarrollo, aplicando la técnica de Integración Continua notamos que se minimiza esta problemática, dándonos como resultado un sistema actualizado en tiempo real y con una mejor productividad en la etapa de mantenimiento.
- Al aplicar Test Driven Development notamos que mejora la calidad del software, pero se requiere dominar el uso de esta técnica para obtener los resultados esperados, de lo contrario puede representar incremento en el uso de recursos y retrasos en tiempos de entrega.

- En los resultados de las pruebas de caja negra aplicadas a nuestro estudio, se obtuvo un mayor puntaje en las historias de usuario desarrolladas con la combinación de técnicas TDD y CI, frente a las realizadas con el método tradicional.
- Realizando una comparativa entre el sprint uno y dos de nuestro Burndown Chart, comprobamos que en el Sprint dos, con dominio en TDD y CI, tenemos una mejor productividad y cumplimientos en tiempos de entrega, mientras que en el Sprint uno no cumplimos con los objetivos establecidos, ya que no contábamos con los conocimientos necesarios en estas técnicas.

Recomendaciones

- Mientras más grande sea el sistema informático, más difícil es organizar la integración de código de los distintos desarrolladores que participan en el proyecto, por esta razón, una opción viable es incluir técnicas y metodologías de desarrollo ágil que nos permitirá ahorrar tiempo y recursos que pueden ser destinados a otras áreas.
- Se debe tomar en cuenta el factor tiempo que supone el aprendizaje de TDD y CI cuando se lo implementa por primera vez en un proyecto, ya que si no se lo incluye en los tiempos de desarrollo de un Sprint puede llegar a generar retrasos en su entrega.
- En la actualidad la demanda tecnológica nos obliga a obtener excelentes resultados en un corto tiempo, por esta razón se recomienda aplicar al menos una técnica o

metodología en nuestros sistemas para lograr cumplir con los objetivos propuestos por los usuarios y de esta manera ser competitivos en el mundo laboral.

Bibliografía

- About Gitlab.* (2021). Obtenido de GitLab es la plataforma DevOps: <https://about.gitlab.com/>
- Adnan Causevic, D. S. (2011). *Factors Limiting Industrial Adoption of Test Driven Development*. Västerås, Sweden.
- Aguilera Díaz, A. (2017). *I costo-beneficio como herramienta de decisión en la inversión en actividades científicas*. Recuperado el 19 de Agosto de 2021, de <http://scielo.sld.cu/pdf/cofin/v11n2/cofin22217.pdf>
- Alonso, F., Martínez , L., & Segovia, F. J. (2005). *Introducción a la ingeniería del software Modelos de desarrollo de programas* (Primera ed.). (J. B. Rubio, Ed.) Madrid, España: Delta Publicaciones Universitarias. Recuperado el 13 de Julio de 2020, de https://books.google.com.ec/books?id=rXU-WS4UatYC&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false
- Amr Noaman Abdel-Hamid, M. A.-K. (2011). *Process Increments:An Agile Approach to Software*. El Cairo, Egypto.
- Angular. (2020). *Introduction to Angular concepts*. Recuperado el 20 de Febrero de 2021, de Anular: <https://angular.io/guide/architecture>
- Arachchi, I. P. (2018). *Continuous Integration and Continuous Delivery Pipeline Automation for Agile Software Project Management* . Sri Lanka.

- Avishek Sharma Dookhun, L. N. (2019). Assessing The Effectiveness Of Test-Driven Development and Behavior-Driven Development in an Industry Setting. Réduit, Mauritius.
- Bara, M. (s.f.). *Bussines School*. Obtenido de UIC Barcelona: <https://obsbusiness.school/es/blog-investigacion/project-management/las-5-etapas-en-los-sprints-de-un-desarrollo-scrum>
- Beck, K., Beedle , M., Bennekum, A. v., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D. (Febrero de 2001). *Manifiesto por el Desarrollo Ágil de Software*. Recuperado el 19 de Agosto de 2021, de agilemanifiesto: <http://agilemanifiesto.org/iso/es/manifiesto.html>
- Blanco, P., Camarero, J., Fumero, A., Werterski, A., & Rodriguez, P. (2009). *Metodología de desarrollo ágil para sistemas móviles*. Recuperado el 19 de Agosto de 2021, de https://www.researchgate.net/profile/Antonio-Fumero/publication/267795011_Metodologia_de_desarrollo_agil_para_sistemas_moviles_Introduccion_al_desarrollo_con_Android_y_el_iPhone/links/577009d108ae842225aa444b/Metodologia-de-desarrollo-agil-para-sistemas-m
- Blum, B. (1992). *Software Engineering: A Holistic View*. (O. U. Press, Ed.) Nueva York, Estados Unidos. Recuperado el 13 de Julio de 2020
- Boehm, B. (1976). *Software Engineering*. doi:10.1109/TC.1976.1674590
- Briceño, M. (2017). *Memorias del PNFI*. Recuperado el 19 de Agosto de 2021, de http://upttmbi.edu.ve/site/Noticias/MemoriasdelPNFI_Valera.pdf#page=22
- Carrizo, D., & Alfaro, A. (2018). Método de aseguramiento de la calidad en una metodología de desarrollo de software: un enfoque práctico. *Revista Chilena de Ingeniería*.

Chintan Amrit, Y. M. (Febrero de 2018). Effectiveness of Test-Driven Development and Continuous Integration. Enschede.

Crispin, L. (2006).

Desarrollo WEB. (19 de julio de 2002). Obtenido de <https://desarrolloweb.com/articulos/840.php#:~:text=Oracle%20es%20b%C3%A1sicamente%20una%20herramienta,y%20multinacionales%2C%20por%20norma%20general.>

DoucUC . (2018). Obtenido de Investigación Aplicada: <http://www.duoc.cl/biblioteca/crai/definicion-y-proposito-de-la-investigacion-aplicada>

Duvall, P. M., Steve, M., & Glover, A. (2007). *Continuous Integration: Improving Software Quality and Reducing Risk*. Pearson Education.

ES, O. (2019). *OPENKM*. Obtenido de Gestión Documental.: <https://www.openkm.com/es/>

Estayno Marcelo, D. G. (2018). MODELOS Y MÉTRICAS PARA EVALUAR LA CALIDAD DE SOFTWARE.

Gallaba, K. (2020). Improving the Robustness and Efficiency of Continuous Integration. Canada.

Geovanny, M. (2014). Herramienta de desarrollo Netbeans. Universidad del Norte.

Guaselma, G. (15 de Enero de 2013). *Técnicas y herramientas de desarrollo de software*. Recuperado el 19 de Agosto de 2021, de <https://es.slideshare.net/gualsema/tecnicas-y-herramientas-de-desarrollo-de-software1>

- Gustavo Sizílio, D. A. (2019). An Empirical Study of the Relationship between Continuous Integration and Test Code Evolution. Brazil.
- Hernández, G., Jimenez, R., Martínez, Á., & Jiménez, F. (2019). Métricas de productividad para equipo de trabajo de desarrollo ágil de software: una revisión sistemática. *TecnoLógicas*.
- Herraz, J. I. (2011). TDD como metodología de diseño de software-Paradigma . Madrid.
- Humphrey, W. (1989). *Managing the Software Process*. (P. Education, Ed.) Recuperado el 13 de Julio de 2020
- IEEE. (1990). *IEEE Standar Glossary of Software Engineering*. Estados Unidos. doi: 10.1109/IEEESTD.1990.101064
- IEEE, A. S. (28 de Septiembre de 2017). IEEE Standard for System, Software, and Hardware Verification and Validation. New York, USA.
- IEEE, I. o. (1990). *Calidad del Software*.
- Irrazábal, E. (Octubre de 2015). *Mejora de la mantenibilidad con un modelo de medición de la calidad: resultados en una gran empresa*. Obtenido de Universidad Nacional de la Plata: http://sedici.unlp.edu.ar/bitstream/handle/10915/50334/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y
- ISO 26511, I. S. (Diciembre de 2018). Systems and software engineering — Requirements for managers of information for users of systems, software, and services. New York, USA.

- ISO12207, I. S. (18 de Diciembre de 2020). Systems and software engineering —Life cycle management. New York, USA.
- Kollanus, S. (2010). Test-Driven Development - Still a Promising. Jyväsaskylä, Finland.
- Ladrón de Guevara, J. M. (2018). *Fundamentos de programación en Java*. Recuperado el 20 de Febrero de 2021, de <http://190.57.147.202:90/jspui/bitstream/123456789/1401/1/Fundamentos%20de%20programacion%20en%20Java.pdf>
- Lai, S.-T. (2019). APPLYING CONTINUOUS INTEGRATION FOR INCREASING THE MAINTENANCE QUALITY AND EFFICIENCY OF WEB APP. Japón Taiwan.
- Lientz, B. P., & Swanson, B. E. (1980). *Software Maintenance Management: A Study of the Maintenance of Computer Application Software in 487 Data Processing Organizations*. Michigan.
- Lopez Sanz, M., Soltero Domingo, F., Sánchez Fúquene, D., Moreno Pérez, Á., Bollati, V., & Vara Mesa, J. M. (2016). *Programación Web en el Entorno Servidor*. Madrid: Grupo Editorial RA-MA. Recuperado el 19 de Agosto de 2021, de [https://books.google.com.ec/books?id=7I2fDwAAQBAJ&pg=PA30&lpg=PA30&dq="técnica+de+costo+beneficio"+"desarrollo+de+software"&source=bl&ots=NuNjkk84e8&sig=A CfU3U1_bw0A7UvVVavZX4NsY83s7M1kOg&hl=es-419&sa=X&ved=2ahUKEwihhNK0vMPyAhWZRjABHUIYAc4Q6AF6BAgvEAM#v=o](https://books.google.com.ec/books?id=7I2fDwAAQBAJ&pg=PA30&lpg=PA30&dq=)
- M.M. Lehman, J. R. (06 de Agosto de 2002). Metrics and laws of software evolution-the nineties view. Albuquerque, USA.

- Martin Brandtner, E. G. (2014). *Supporting Continuous Integration by Mashing-Up Software Quality Information*. Suiza.
- Mojtaba Shahin, M. A. (2017). *Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices*. Australia.
- Monika Agarwal, R. M. (04 de 2013). *Software Maintainability and Usability in Agile Environment*. Noida, India.
- Naur, P., & Randell, B. (1969). *Report on a conference sponsored by the NATO SCIENCE COMMITTEE Garmisch, Germany, 7th to 11th October 1968*. Recuperado el 13 de Julio de 2020, de <http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PDF>
- Navarro Cadavid, A., Fernandez Martínez, J. D., & Morales Vélez, J. (2013). *Revisión de metodologías ágiles para el desarrollo de software*. Recuperado el 19 de Agosto de 2021, de <https://www.redalyc.org/pdf/4962/496250736004.pdf>
- oauth. (s.f.). *OAuth 2.0*. Obtenido de 2020: <https://oauth.net/2/>
- Oracle. (2021). Recuperado el 19 de Agosto de 2021, de <https://www.oracle.com/es/database/technologies/>
- Pantaleo, G., & Rinaudo, L. (2016). *Ingeniería de software*. México: Alfaomega Grupo Editor, S.A. Recuperado el 13 de Julio de 2020, de https://books.google.com.ec/books?id=a8j2DQAAQBAJ&printsec=frontcover&source=gb_s_ge_summary_r&cad=0#v=onepage&q&f=false

- Paños Alvarez, A. (2000). *Influencia de las tecnologías de la información en los procesos de información y toma de decisiones de las empresas*. Murcia. Recuperado el 07 de Junio de 2020, de <https://revistas.ucm.es/index.php/CDMU/article/view/68892/4564456553254>
- Power Data*. (24 de noviembre de 2017). Obtenido de <https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/introduccion-al-sistema-oracle-database-y-oracle-big-data-cloud>
- Pressman, R. (2015). *Software Engineering: A Practitioner's Approach* (Octava ed.). (M.-H. Education, Ed.) Recuperado el 13 de Julio de 2020
- Pressman, R. S. (2010). *Ingeniería del Software. Un enfoque práctico*. México: McGraw-Hill 7ta edición.
- Quinapaxi German, P., & Viracocha Ortega, j. G. (2019). *DISEÑAR E IMPLEMENTAR UNA SOLUCIÓN DE CONTROL DE INGRESO Y SALIDA DEL*. Recuperado el 19 de Agosto de 2021, de <http://repositorio.utc.edu.ec/bitstream/27000/5748/1/T-001126.pdf>
- Rienda láñez, J. (Octubre de 2019). *Diseño e implementación de un microservicio con Spring*. Recuperado el 19 de Agosto de 2021, de <http://hdl.handle.net/10016/30515>
- Rivero, Y., Madariago, C., Toledo, A., Lamoth, L., & Hechavarría, J. (2016). *Software educativo para la enseñanza del proceso de medición de la calidad de software*. La Habana: INFOREDU.
- Salamón, A., Maller, P., Boggio, A., Mira, N., Perez, S., & Coenda, F. (2014). La Integración Continua Aplicada en el Desarrollo de Software en el Ámbito Científico – Técnico. *Trabajo de Investigación. Universidad Nacional de la Plata, 2*.

- Salazar Moncada, L. F., & Tirira Iturralde, R. E. (2016). *Guía práctica para la planificación de Auditoría de estados financieros bajo Normas Internacionales de Auditoría*. Recuperado el 19 de Agosto de 2021, de <http://201.159.223.180/bitstream/3317/5140/1/T-UCSG-PRE-ECO-CICA-214.pdf>
- Serna, E., Martinez, R., & Tamayo, P. (2021). Una revisión a la realidad de la automatización de las pruebas del software. *Scielo*.
- Silvera, S., & Vargas, L. (2010). Modelo Bidimensional de Riesgos del Mantenimiento de Sistemas Integrados de Gestión (ERP). *Investigaciones Europeas de Dirección y Economía de la Empresa*, 173 - 190.
- Sommerville, I. (2005). *Ingeniería del software*. (P. Education, Ed.) Recuperado el 13 de Julio de 2020, de https://books.google.com.ec/books?id=gQWd49zSut4C&printsec=frontcover&source=gb_s_ge_summary_r&cad=0#v=onepage&q&f=false