



**UNIVERSIDAD TÉCNICA DE COTOPAXI**  
**FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS**  
**CARRERA DE INGENIERÍA EN INFORMÁTICA Y SISTEMAS**  
**COMPUTACIONALES**

**PROYECTO DE INVESTIGACIÓN**

**“ANÁLISIS COMPARATIVO DE HERRAMIENTAS DE INTEGRACIÓN  
CONTINUA: CASO DE ESTUDIO JENKINS VS TRAVIS”**

Proyecto de Titulación presentado previo a la obtención del Título de Ingenieras en  
Informática y Sistemas Computacionales.

**AUTORAS:**

Farinango Alcásiga Paola Elizabeth

Panchi Zapata Diana Pamela

**TUTORA:**

Ing. M.Sc. Verónica del Consuelo Tapia Cerda

**LATACUNGA – ECUADOR**

**Marzo 2022**



## DECLARACIÓN DE AUTORÍA

Nosotros, **Farinango Alcásiga Paola Elizabeth** con cédula de ciudadanía No **1753097292** y **Panchi Zapata Diana Pamela** con cédula de ciudadanía No **0550239636**, declaramos ser autores del presente proyecto de investigación: **ANÁLISIS COMPARATIVO DE HERRAMIENTAS DE INTEGRACIÓN CONTINUA: CASO DE ESTUDIO JENKINS VS. TRAVIS**, siendo la **Ing. M.Sc Tapia Cerda Verónica del Consuelo**, tutora del presente trabajo; y eximimos expresamente a la Universidad Técnica de Cotopaxi y a sus representantes legales de posibles reclamos o acciones legales.

Además, certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

**Farinango Alcásiga Paola Elizabeth**  
C.I. 1753097292

**Panchi Zapata Diana Pamela**  
C.I. 0550239636



## AVAL DEL TUTOR DE PROYECTO DE TITULACIÓN

En calidad de Tutor del Trabajo de Investigación sobre el título:

**“ANÁLISIS COMPARATIVO DE HERRAMIENTAS DE INTEGRACIÓN CONTINUA: CASO DE ESTUDIO JENKINS VS. TRAVIS”**, de **FARINANGO ALCÁSIGA PAOLA ELIZABETH** y **PANCHI ZAPATA DIANA PAMELA**, de la carrera de Ingeniería en Informática y Sistemas Computacionales, considero que dicho Informe Investigativo cumple con los requerimientos metodológicos y aportes científico-técnicos suficientes para ser sometidos a la evaluación del Tribunal de Validación de Proyecto que el Consejo Directivo de la Facultad de Ciencias de la Ingeniería y Aplicadas de la Universidad Técnica de Cotopaxi designe, para su correspondiente estudio y calificación.

Latacunga, Marzo, 2022

**TUTORA DE PROYECTO DE INVESTIGACIÓN**

**Ing. M.Sc. Verónica del Consuelo Tapia Cerda**

**C.I: 0502053697**



## APROBACIÓN DEL TRIBUNAL DE TITULACIÓN

En calidad de Tribunal de Lectores, aprueban el presente Informe de Investigación de acuerdo a las disposiciones reglamentarias emitidas por la Universidad Técnica de Cotopaxi, y por la Facultad de Ciencias de Ingeniería y Aplicadas; por cuanto, los postulantes: **FARINANGO ALCÁSIGA PAOLA ELIZABETH** y **PANCHI ZAPATA DIANA PAMELA** con el título de Proyecto de titulación: “**ANÁLISIS COMPARATIVO DE HERRAMIENTAS DE INTEGRACIÓN CONTINUA: CASO DE ESTUDIO JENKINS VS. TRAVIS**” han considerado las recomendaciones emitidas oportunamente y reúne los méritos suficientes para ser sometido al acto de Sustentación de Proyecto.

Por lo antes expuesto, se autoriza realizar los empastados correspondientes, según la normativa institucional.

Latacunga, Marzo del 2022.

Para constancia firman:

Lector 1 (Presidente)

Nombre: Ph.D. Chancusig Chisag Juan Carlos  
CC: 0502275779

Lector 2

Nombre: Mg. Iza Carate Miryan Dorila  
CC: 0501957617

Lector 3

Nombre: Mg. Llano Casa Alex Christian  
CC: 0502589864

## **AGRADECIMIENTO**

En el presente proyecto de investigación tengo a bien agradecer: en primer lugar, A Dios quien ha sido mi guía y fortaleza en los momentos difíciles de mi Carrera.

En segundo lugar, a mis Padres Luis y María quienes con su amor, paciencia, confianza y esfuerzo me han permitido llegar a cumplir el día de hoy un meta más en mi vida, gracias por inculcar en mí el ejemplo y valentía de no temer las adversidades porque Dios está siempre conmigo. A mis hermanos José y Jefferson por apoyarme en cada decisión importante en mi vida.

En tercer lugar, quiero expresar un profundo agradecimiento a mi amiga y a su vez compañera de tesis por el apoyo incondicional en la trayectoria universitaria, finalmente quiero agradecer y a la vez expresar mi profundo respeto a mi tutora de tesis Ing. MSc. Verónica Tapia quien gracias a su conocimientos, enseñanza y colaboración permitió el desarrollo de este trabajo y a la Universidad Técnica de Cotopaxi por brindarme la oportunidad de formarme como profesional.

*Paola Farinango*

## **DEDICATORIA**

Dedico esta tesis a mis padres que son ejemplo de sacrificio y trabajo quienes supieron comprender y brindarme todo su apoyo moral y económico, a mis hermanos que con sus sonrisas y locuras me hacían ver de mejor manera los momentos difíciles. Por esas palabras de aliento, por su confianza y el ejemplo de lucha que cada día puedo apreciar, para ustedes quienes son mi más grande regalo.

Los amo con todo mi corazón.

*Paola Farinango*

## **AGRADECIMIENTO**

Al concluir una etapa maravillosa de mi vida quiero extender un profundo y sincero agradecimiento, a quienes hicieron posible este sueño, aquellos que junto a mí caminaron en todo momento y siempre fueron inspiración, apoyo y fortaleza. Esta mención en especial para DIOS, mis padres Nancy y Byron, mi hermana Keyli. Muchas gracias a ustedes por demostrarme que "El verdadero amor no es otra cosa que el deseo inevitable de ayudar al otro para que este se supere." Mi gratitud a las Autoridades de la Universidad Técnica de Cotopaxi, mi agradecimiento sincero a la tutora de Tesis, Ing. MSc. Verónica Tapia, gracias a cada docente quienes con su apoyo y enseñanzas constituyen la base de mi vida profesional.

*Con amor y admiración*

*Diana Panchi.*

## **DEDICATORIA**

Los resultados de este proyecto de investigación, merece expresar un profundo agradecimiento, aquellas personas que de alguna forma son parte de su culminación, quienes con su ayuda, apoyo y comprensión me alentaron a lograr esta hermosa realidad. Mi agradecimiento, va dirigido especialmente a mis padres, quienes me han apoyado día tras día. A mis profesores, quienes han impartido sus conocimientos y experiencias, para formarme como una profesional, a la ingeniera que fue mi tutora de tesis, quien supo creer en mi capacidad y orientarme sin interés alguno, para culminar con el éxito de esta investigación.

*Con cariño, agradecimiento y respeto*

*Diana Panchi.*

## ÍNDICE GENERAL

DECLARACIÓN DE AUTORÍA .....	ii
AVAL DEL TUTOR DE PROYECTO DE TITULACIÓN .....	iii
APROBACIÓN DEL TRIBUNAL DE TITULACIÓN.....	iv
AGRADECIMIENTO .....	v
DEDICATORIA.....	vi
AGRADECIMIENTO .....	vii
DEDICATORIA.....	viii
ÍNDICE GENERAL .....	ix
ÍNDICE DE TABLAS .....	xii
ÍNDICE DE ILUSTRACIONES .....	xiii
RESUMEN .....	xiv
ABSTRACT .....	xv
AVAL DE TRADUCCIÓN.....	xvi
1. INFORMACIÓN GENERAL.....	17
2. INTRODUCCIÓN: .....	19
2.1. PROBLEMA:.....	19
2.1.1. Situación Problemática: .....	19
2.1.2 Formulación del problema: .....	20
2.3 OBJETO Y CAMPO DE ACCIÓN.....	20
2.4 BENEFICIARIOS.....	20
2.5 JUSTIFICACIÓN .....	20
2.6 HIPÓTESIS.....	21
2.7 OBJETIVOS .....	21
2.7.1 General .....	21
2.7.2 Específicos .....	21
2.8 SISTEMA DE TAREAS.....	22
3. FUNDAMENTACIÓN CIENTÍFICO TÉCNICA.....	24
3.1 DESARROLLO DE SOFTWARE .....	24
3.1.1 Concepto.....	24
3.2 MODELOS Y METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	24
3.2.1 Modelos de desarrollo de software.....	24
3.2.2 Metodologías de desarrollo de software.....	26

3.3	DevOps .....	30
3.3.1	Definición de DevOps .....	30
3.4	INTEGRACIÓN CONTINUA .....	32
3.4.1	Definición.....	32
3.4.2	Prácticas de integración continua.....	33
3.4.3	Beneficios de la integración continua .....	36
3.5	HERRAMIENTAS DE INTEGRACIÓN CONTINUA.....	37
3.5.1	Jenkins.....	37
3.5.2	Travis.....	39
3.5.3	CircleCI .....	40
3.5.4	TeamCity .....	40
3.5.5	Bamboo .....	41
3.5.6	Características de las herramientas de Integración Continua.....	41
3.6	METODOLOGIAS .....	43
3.6.1	Metodologías de investigación.....	43
3.6.2	Metodologías de desarrollo .....	44
3.6.2.1	Prácticas ágiles .....	44
3.7	TÉCNICA DE VALIDACIÓN .....	45
3.7.1	Validación por expertos .....	45
3.7.2	Alfa de Cronbach .....	45
3.7.3	Interpretación del Coeficiente de Alfa de Cronbach.....	46
4	MATERIALES Y MÉTODOS .....	47
4.1	METODOLOGÍA DE INVESTIGACIÓN.....	47
4.1.1	Tipo de investigación cualitativa.....	47
4.1.2	Nivel de estudio descriptivo .....	47
4.1.3	Diseño de investigación narrativo .....	48
4.2	TÉCNICAS E INSTRUMENTOS DE INVESTIGACIÓN .....	48
4.2.1	Observación.....	48
4.2.2	Fichas bibliográficas .....	48
4.2.3	Juicio de expertos .....	48
4.2.4	Cuestionario. ....	48
4.6	METODOLOGÍA DEL DESARROLLO.....	48
4.6.1	Prácticas ágiles .....	48
5	ANÁLISIS Y DISCUSIÓN DE LOS RESULTADOS.....	50

5.1	Resultado del análisis comparativo.....	50
5.2	Práctica con la herramienta Jenkins.....	60
5.3	Práctica con la herramienta Travis CI.....	66
5.3	Validación de expertos.....	70
5.6	PRESUPUESTO.....	78
5.6.1	Costos directos.....	78
5.6.2	Costos indirectos.....	78
5.6.3	Costos totales.....	78
6	CONCLUSIONES Y RECOMENDACIONES.....	79
6.1	CONCLUSIONES.....	79
6.2	RECOMENDACIONES.....	79
7	BIBLIOGRAFÍA.....	80
8	ANEXOS.....	85

## ÍNDICE DE TABLAS

<b>Tabla 1:</b> Cuadro comparativo de las herramientas de Integración Continua: Jenkins y Travis CI.....	50
<b>Tabla 2:</b> Análisis bibliográfico de las herramientas de Integración Continua Jenkins y TravisCI.....	53
<b>Tabla 3:</b> Costos directos .....	78
<b>Tabla 4:</b> Costos indirectos .....	78
<b>Tabla 5:</b> Costos totales .....	78

## ÍNDICE DE ILUSTRACIONES

<b>Ilustración 1:</b> Modelo cascada .....	25
<b>Ilustración 2:</b> Modelo iterativa incremental .....	26
<b>Ilustración 3:</b> Comparación entre enfoque tradicional y ágil.....	27
<b>Ilustración 4:</b> Rational Unified Process.....	28
<b>Ilustración 5:</b> Modelo Scrum .....	29
<b>Ilustración 6:</b> Extreme Programming.....	30
<b>Ilustración 7:</b> Modelo básico de Integración Continua con Jenkins .....	38
<b>Ilustración 8:</b> Integración Continua con Travis CI.....	39
<b>Ilustración 9:</b> Valoración del Coeficiente de Alfa de Cronbach.....	47
<b>Ilustración 10:</b> Comparación de tendencia de las herramientas de Integración Continua: Jenkins y Travis CI en el Ecuador.....	56
<b>Ilustración 11:</b> Desglose comparativo por subregión de las herramientas de Integración Continua: Jenkins y Travis CI.....	56
<b>Ilustración 12:</b> Comparación de tendencia de las herramientas de Integración Continua: Jenkins y Travis CI en el mundo .....	57
<b>Ilustración 13:</b> Desglose comparativo por región de las herramientas de Integración Continua: Jenkins y Travis CI .....	57
<b>Ilustración 14:</b> Métricas y definiciones de visitas y ranking .....	58
<b>Ilustración 15:</b> Tráfico y participación.....	58
<b>Ilustración 16:</b> Visitas a lo largo del tiempo .....	58
<b>Ilustración 17:</b> Tráfico compartido por región geográfica. ....	59
<b>Ilustración 18:</b> Comparación de las herramientas de Integración Continua más usadas en el mundo. ....	59
<b>Ilustración 19:</b> Proyecto en el repositorio .....	60
<b>Ilustración 20:</b> Crear tarea en Jenkins .....	60
<b>Ilustración 21:</b> Configuración de Pipeline .....	61
<b>Ilustración 22:</b> Codificación Pipeline Script .....	61
<b>Ilustración 23:</b> Resultados .....	62
<b>Ilustración 24:</b> Resultados modo Consola.....	62
<b>Ilustración 25:</b> Multibranch Pipeline .....	63
<b>Ilustración 26:</b> Creación de credenciales .....	63
<b>Ilustración 27:</b> Conexión a Github.....	64
<b>Ilustración 28:</b> Crear archivo Jenkinsfile .....	64
<b>Ilustración 29:</b> Selección para ejecución del proyecto .....	65
<b>Ilustración 30:</b> Resultado de Pipeline .....	65
<b>Ilustración 31:</b> Crear archivo yml en el proyecto de Visual Studio Code .....	66
<b>Ilustración 32:</b> Activación del repositorio.....	66
<b>Ilustración 33:</b> Selección de repositorios .....	67
<b>Ilustración 34:</b> Configuración automática en Travis CI .....	68
<b>Ilustración 35:</b> Activación del repositorio.....	68
<b>Ilustración 36:</b> Configuración para activar la compilación .....	69
<b>Ilustración 37:</b> Resultado en Travis CI .....	69



## UNIVERSIDAD TÉCNICA DE COTOPAXI

### FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

**TITULO:** “ANÁLISIS COMPARATIVO DE HERRAMIENTAS DE INTEGRACIÓN CONTINUA: CASO DE ESTUDIO JENKINS VS. TRAVIS”

**Autores:**

Farinango Alcásiga Paola Elizabeth

Panchi Zapata Diana Pamela

### RESUMEN

El presente proyecto de investigación se realizó con el objetivo de describir a través de un análisis comparativo documental y práctico la eficacia de las herramientas de Integración Continua para recomendar su aplicación en proyectos de desarrollo de software. Se utilizaron diferentes tipos de metodologías de investigación como es el caso de la investigación cualitativa que ayudó a evaluar las características de las herramientas de integración continua, la investigación descriptiva que permitió enfatizar el objeto de estudio que son las herramientas Jenkins y Travis, adicionalmente se utilizó la investigación narrativa para describir las características de las herramientas y determinar cuál es la mejor alternativa para el proceso de integración continua. Para la ejecución de los casos prácticos se utilizaron enfoques de desarrollo ágil como la propiedad colectiva del código, trabajo en equipo y como herramientas clave Jenkins y Travis para la integración continua; Github como repositorio del código del proyecto y para el control de versión. De esta manera, se aplicó el proceso comparativo práctico de Integración Continua con un sistema web desarrollado en Visual Studio Code con lenguaje de programación Python. La validación del proyecto se realizó a través de la técnica del juicio de expertos utilizando el Coeficiente Alfa de Cronbach con un valor de 0,73 que equivale a una calificación de Bueno. Finalmente, se puede concluir que el objetivo del proyecto se cumplió ya que se tiene como resultado un trabajo comparativo teórico y práctico del cual se ha comprobado que la mejor herramienta de Integración Continua entre las propuestas que son Jenkins y Travis, la más eficaz es Jenkins.

**Palabras claves:** DevOps, Integración Continua, Jenkins, Travis CI, Software.



## TECHNICAL UNIVERSITY OF COTOPAXI

### ENGINEERING SCIENCES AND APPLIED FACULTY

**TOPIC:** “CONTINUOUS INTEGRATION TOOLS COMPARATIVE ANALYSIS: CASE STUDY JENKINS VS. TRAVIS”.

**Authors:**

Farinango Alcásiga Paola Elizabeth

Panchi Zapata Diana Pamela

### ABSTRACT

The actual research project was made with aim at describing, through a documentary and practical comparative analysis, the Continuous Integration tools effectiveness for recommending their application into software development projects. It was research methodologies different types, such as, qualitative research, which it helped to assess the continuous integration tools characteristics, descriptive research that allowed emphasizing the study object, which are the Jenkins and Travis tools, furthermore, it was used narrative research for describing the tools characteristics and determine, what is the best alternative for continuous integration process. For practical cases execution was used agile development approaches, such as, the code collective ownership, teamwork and Jenkins and Travis as key tools for the continuous integration; Github as a repository for the project code and for version control. This way, it was applied the Continuous Integration practical comparative process with a developed web system into Visual Studio Code with Python programming language. The project validation was made, through the expert judgment technique by using Cronbach's Alpha Coefficient with a 0.73 value, what is equivalent to a Good rating. Finally, it can be concluded, which the project aim was fulfilled, since the result is, a theoretical and practical comparative work, what it has been verified, what is the best Continuous Integration tool between the proposals, which are Jenkins and Travis, the proposal is most effective proposal Jenkins.

**Keywords:** DevOps, Continuous Integration, Jenkins, Travis CI, Software.

## AVAL DE TRADUCCIÓN



CENTRO  
DE IDIOMAS

## AVAL DE TRADUCCIÓN

En calidad de Docente del Idioma Inglés del Centro de Idiomas de la Universidad Técnica de Cotopaxi; en forma legal **CERTIFICO** que:

La traducción del resumen al idioma Inglés del trabajo de titulación cuyo título versa: **“ANÁLISIS COMPARATIVO DE HERRAMIENTAS DE INTEGRACIÓN CONTINUA: CASO DE ESTUDIO JENKINS VS. TRAVIS.”** Farinango Alcásiga Paola Elizabeth y Panchi Zapata Diana Pamela , estudiantes de la Carrera de **Ingeniería en Informática y Sistemas Computacionales** perteneciente a la **Facultad de Ciencias de la Ingeniería y Aplicadas** lo realizaron bajo mi supervisión y cumple con una correcta estructura gramatical del Idioma.

Es todo cuanto puedo certificar en honor a la verdad y autorizo a las peticionarias hacer uso del presente aval para los fines académicos legales.

Latacunga, 21 marzo del 2022

Atentamente,



CENTRO  
DE IDIOMAS

Mg. Marco Paúl Beltrán Semblantes

**DOCENTE CENTRO DE IDIOMAS-UTC**  
CI: 0502666514

## **1. INFORMACIÓN GENERAL**

**Título:**

Análisis comparativo de herramientas de Integración Continua: caso de estudio Jenkins vs Travis.

**Fecha de inicio:**

25 de Octubre del 2021.

**Fecha de finalización:**

11 de Febrero del 2022.

**Lugar de ejecución:**

(Cotopaxi/Latacunga/Universidad Técnica de Cotopaxi)

**Facultad que auspicia**

Facultad de Ciencias de la Ingeniería y Aplicadas.

**Carrera que auspicia:**

Ingeniería en Informática y Sistemas Computacionales.

**Proyecto de investigación vinculado:**

Aplicación del Modelo Iterativo- Incremental en el desarrollo de herramientas informáticas para Instituciones, organizaciones y empresas del entorno educativo, productivo y comercial de la Provincia de Cotopaxi.

**Equipo de Trabajo:**

**Investigadora 1:**

Apellidos y Nombre: Farinango Alcásiga Paola Elizabeth

Cedula de ciudadanía: 1753097292

Fecha de nacimiento: 05 de agosto del 1997

Estado Civil: Soltera

Email institucional: paola.farinango7292@utc.edu.ec

Celular: 0988962234

**Investigadora 2:**

Apellidos y Nombre: Panchi Zapata Diana Pamela

Cedula de ciudadanía: 0550239636

Fecha de nacimiento: 01 de Agosto de 1998

Estado Civil: Soltera

Email institucional: dianapanchi9636@utc.edu.ec

Celular: 0983705331

**Tutor:**

Ing. MSc. Tapia Cerda Verónica del Consuelo

Cedula de ciudadanía: 0502053697

Email institucional: veronica.tapia@utc.edu.ec

Celular: 099252383

**Área de Conocimiento:**

06/ 061 Informática y comunicación (TIC) / 0613 Software y desarrollo y análisis de aplicativos.

**Línea de investigación:**

Tecnologías de la Información y Comunicación (TICs).

**Sub líneas de investigación de la Carrera:**

Ciencias Informáticas para la modelación de Sistemas de Información a través del desarrollo de software.

## **2. INTRODUCCIÓN:**

### **2.1. PROBLEMA:**

#### **2.1.1. Situación Problemática:**

En la actualidad las herramientas de Integración Continua para la automatización de etapas en el desarrollo de software ha crecido considerablemente, pero nos encontramos en una gran brecha en donde estudios demuestran que en América latina existen una gran cantidad de proyectos de desarrollo de software en donde continúan utilizando metodologías y procedimientos obsoletos dando como resultado de productos y servicios que se entregan con retraso o fallas. Según los estudios [1] la Universidad de las Ciencias Informáticas (UCI) es una de las entidades dedicada a la producción de software en Cuba que mantiene compromisos tanto a nivel nacional como internacional, de ahí la importancia de garantizar la gestión de los proyectos y asegurar la calidad de sus productos y servicios. No practicar la Integración Continua en algunos proyectos es uno de los problemas que afectan la calidad de los productos que son liberados en la (UCI), ya que el 20% de los proyectos no utilizan esta técnica.

En el Ecuador las herramientas de Integración Continua no son utilizadas debido a que las empresas no tienen conocimiento de su existencia y es por ello que no las ponen en práctica ya que al realizar un proyecto en equipo se trabaja en espacios sincronizados por lo que se hace modificaciones o cambios de manera individual haciendo que la ejecución del proyecto genere resultados desfavorables para despliegue del software. Según [2] con la evolución del desarrollo de software en los últimos años se dio fundamental que en la Dirección de información y Comunicación de las fuerzas terrestres cuenten con herramientas de Integración Continua ya que desarrollan productos de software para sistematizar y agilizar los procesos de reclutamiento para administración del personal entre otros y al momento del despliegue existían inconvenientes en el código.

En la Universidad Técnica de Cotopaxi no se aplica la Integración Continua a nivel de desarrollo de software dentro de la carrera de Sistemas de Información porque no se conoce las herramientas de integración que ayude a identificar los errores en la compilación del proyecto. En la actualidad los procesos de despliegue de software se lo realiza de forma manual ocasionando dificultades de tiempo para los desarrolladores en la entrega de un producto.

### **2.1.2 Formulación del problema:**

¿Cómo determinar características eficientes en las herramientas de Integración Continua para los procesos de desarrollo de software?

### **2.3 OBJETO Y CAMPO DE ACCIÓN**

**Objeto de estudio:** Desarrollo de software.

**Campo de acción:** Integración Continua de software.

### **2.4 BENEFICIARIOS**

**Beneficiarios directos:** los estudiantes de la carrera de sistemas de información.

**Beneficiarios indirectos:** los usuarios de sistemas.

### **2.5 JUSTIFICACIÓN**

La siguiente investigación trata sobre la comparación del uso de los gestores de Integración Continua de Jenkins vs Travis ya que actualmente los desarrolladores de software tienen a su alcance nuevas metodologías ágiles que les permite sistematizar procesos que favorecen a la productividad de una empresa y a su vez beneficiándolas en la reducción de costos y minimizando el tiempo dedicado a corregir errores dentro del código de un software.

La investigación pretende que mediante un análisis comparativo de las herramientas de Integración Continua Jenkins vs Travis se dé a conocer las características y funcionalidades de cada una de estas, permitiendo que los desarrolladores de la carrera de Sistemas de la Información se apoyen en una de estas ya que el objetivo de estas herramientas consiste en comprobar que cada actualización del código fuente no genere problemas en la aplicación que se está desarrollando. Se podrán automatizar tareas que abarcan desde que el desarrollador modifica el código fuente del repositorio software hasta su producción.

Con el conocimiento de las funcionalidades de las herramientas de Integración Continua los usuarios de sistemas y estudiantes de la carrera de Sistemas de la Información de la Universidad Técnica de Cotopaxi podrán ver los beneficios que tiene tanto Jenkins como Travis y así poder elegir la o las herramientas que se adapte a sus necesidades.

## **2.6 HIPÓTESIS**

El análisis comparativo de las herramientas de Integración Continua permitirá identificar características eficientes para la ejecución de estos procesos.

## **2.7 OBJETIVOS**

### **2.7.1 General**

Describir a través de un análisis comparativo documental y práctico, la eficacia de las herramientas de Integración Continua para recomendar su aplicación en proyectos de desarrollo de software.

### **2.7.2 Específicos**

- Realizar la investigación bibliográfica que permita sustentar el marco teórico de esta investigación.
- Ejecutar el caso de estudio de herramientas de Integración Continua para identificar y comparar las características de las herramientas aplicadas.
- Validar la propuesta a través del método de juicios de expertos.

## 2.8 SISTEMA DE TAREAS

Objetivos específicos	Actividad (tareas)	Resultado esperados	Técnicas, Medios e Instrumentos
Realizar la investigación bibliográfica que permita sustentar el marco teórico de esta investigación.	<ol style="list-style-type: none"> <li>1. Revisión bibliografía</li> <li>2. Clasificación de la información.</li> <li>3. Estructuración de la fundamentación científico técnica.</li> </ol>	<ul style="list-style-type: none"> <li>• Selección de artículos.</li> <li>• Análisis de las diferentes fuentes.</li> <li>• Obtención de una base teórica.</li> <li>• Ideas primordiales, definiciones, conceptos.</li> </ul>	<ul style="list-style-type: none"> <li>• Fichas bibliográficas</li> <li>• Tesis</li> <li>• Revistas</li> <li>• Libros</li> </ul>
Ejecutar el caso de estudio de herramientas de Integración Continua para identificar y comparar las características de las herramientas aplicadas.	<ol style="list-style-type: none"> <li>1. Búsqueda del servicio y producto.</li> <li>2. Selección de las herramientas.</li> <li>3. Adquisición de las herramientas.</li> <li>4. Instalar las herramientas.</li> <li>5. Realizar pruebas con cada herramienta en un software web.</li> </ol>	<ul style="list-style-type: none"> <li>• Costos para la adquisición de las herramientas.</li> <li>• Características de cada herramienta.</li> <li>• Disponibilidad de la licencia.</li> <li>• Herramientas instaladas.</li> </ul>	<ul style="list-style-type: none"> <li>• Herramienta Jenkins</li> <li>• Herramienta Travis</li> <li>• Software desarrollado por las investigadoras.</li> </ul>

<p>Validar la propuesta a través del método de juicios de expertos.</p>	<ol style="list-style-type: none"> <li><b>1.</b> Selección de los expertos en cuanto al perfil referente al tema de investigación.</li> <li><b>2.</b> Entrega del instrumento para su ejecución.</li> <li><b>3.</b> Obtención de resultado</li> </ol>	<ul style="list-style-type: none"> <li>• Cuestionario llenado.</li> <li>• Instrumento llenados por el experto.</li> <li>• Validación de la información de los expertos.</li> </ul>	<ul style="list-style-type: none"> <li>• Cuestionario</li> <li>• Test de Cronbach</li> </ul>
---	---	--	--

### **3. FUNDAMENTACIÓN CIENTÍFICO TÉCNICA**

#### **3.1 DESARROLLO DE SOFTWARE**

##### **3.1.1 Concepto**

El desarrollo de software no es una tarea fácil ya que actualmente se caracteriza por la programación orientada a objetos y la separación de las diferentes etapas lógicas en nivel de presentación y además existen numerosas propuestas metodológicas que facilitan en el proceso de desarrollo [3].

Desarrollar un software significa construirlo simplemente mediante su descripción. Esta es una muy buena razón para considerar la actividad de desarrollo de software como una ingeniería. En un nivel más general, la relación existente entre un software y su entorno es clara, ya que el software es introducido en el mundo de modo de provocar ciertos efectos en el mismo.

#### **3.2 MODELOS Y METODOLOGÍAS DE DESARROLLO DE SOFTWARE**

##### **3.2.1 Modelos de desarrollo de software**

###### **3.2.1.1 Concepto**

Una de las definiciones generales de modelo según [4] dice que el modelo es una representación simplificada de la realidad en la que aparecen algunas de sus propiedades y según [5] manifiesta que un modelo es un objeto, concepto o conjunto de relaciones que se utiliza para representar y estudiar de forma simple y comprensible una porción de la realidad empírica.

De tal manera que un modelo se lo entiende como un patrón a seguir o una guía en donde se puede demostrar nuevos estudios en base a un modelo que sirve de referencia o ejemplo para que los próximos proyectos tengan un sustento de la manera en como se lo realizó.

Un modelo de procesos del software es una descripción simplificada de un proceso del software que presenta una visión de ese proceso. Estos modelos pueden incluir actividades que son partes de los procesos y productos de software y el papel de las personas involucradas en la ingeniería del software [6].

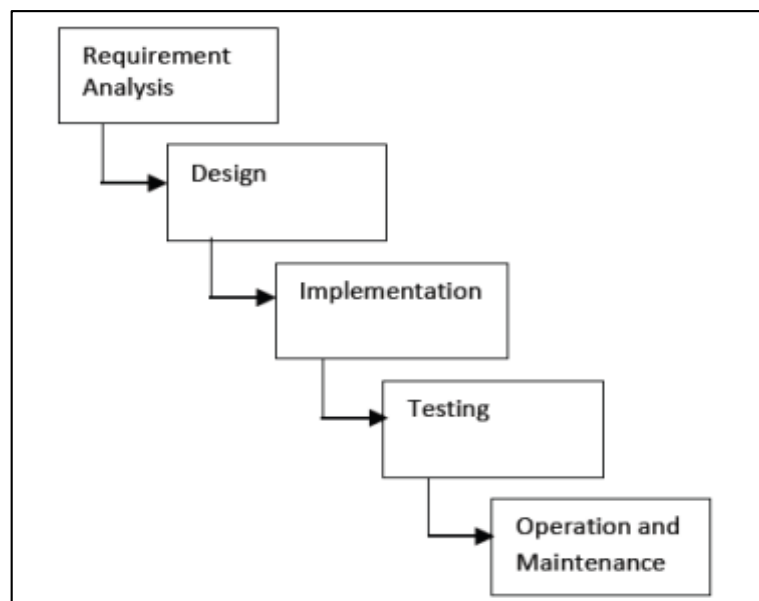
Por ende un modelo dentro del desarrollo de software es el que determina el orden en el que se van a llevar las actividades del proceso de tal manera que se entiende como un procedimiento que sigue

durante el proceso, a este modelo de desarrollo de software también es llamado un paradigma del proceso.

### 3.2.1.2 Modelos Cascada

Este es un modelo estático y se caracteriza por gestionar los desarrollos de sistemas de una manera lineal y secuencial, completando una actividad antes que la otra, de allí proviene su nombre. Se compone de las siguientes 5 fases: Análisis de requerimientos, diseño, implementación (codificación), pruebas y ejecución y mantenimiento [7].

**Ilustración 1:** Modelo cascada



**Fuente:** [7].

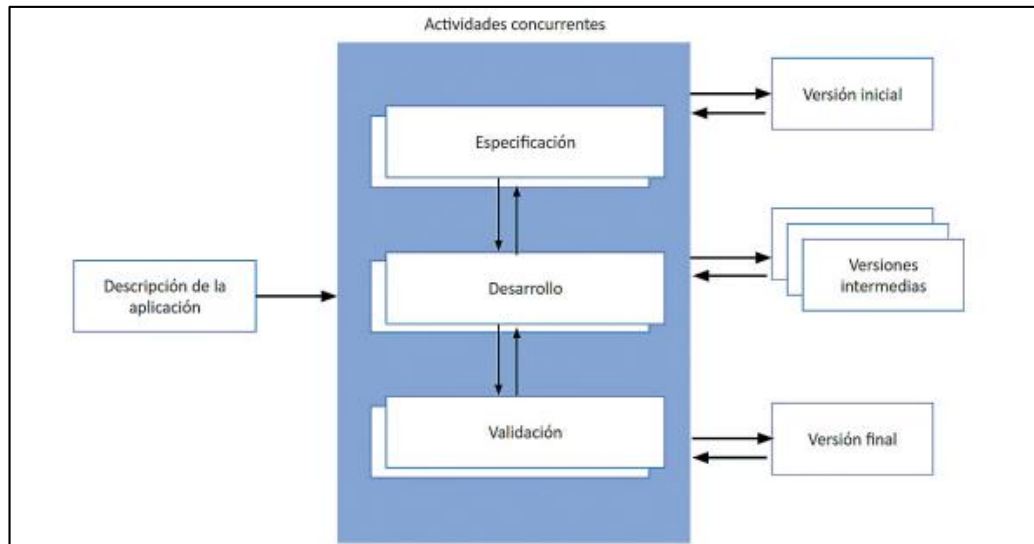
De tal manera que este modelo es muy útil en el desarrollo de un sistema estructurado en donde no su codificación ya no puede ser alterada, ya que los datos y los procesos están separados, motivo por el cual si se mueve la data también se tiene que modificar el código, de tal manera que se denomina como un software no reutilizable por lo que se le dificulta al momento que se quiera actualizar, provocando un proceso complejo y costoso.

### 3.2.1.3 Modelo Iterativo - Incremental

El modelo iterativo incremental combina elementos del modelo en cascada en una propuesta de iteraciones. Este modelo construye una implementación parcial de la aplicación; cada secuencia lineal o iteración produce un incremento funcional en el desarrollo del producto final [8].

De acuerdo a lo antes mencionado al modelo iterativo incremental se ha convertido en una base para arrancar con nuevas metodologías y prácticas de desarrollo de software, y se ha tomado como una evolución de nuevas propuestas en la que el cambio de las necesidades de la información es cada vez son mucho más frecuentes.

**Ilustración 2:** Modelo iterativa incremental



**Fuente:** [8].

### 3.2.2 Metodologías de desarrollo de software

Una metodología de desarrollo de software (SDM) es definido como un conjunto de documentos de políticas, procesos y procedimientos que forman parte de un marco de trabajo usado por los equipos de desarrollo para estructurar, planificar y controlar el proceso de desarrollo de software, optimizándolo mediante el aumento de la productividad del personal de tecnologías de información y una mejora de la solución o producto de software final [7].

De tal manera que al adoptar las metodologías se las puede considerar como un factor crítico en el desarrollo y evolución para empresas desarrolladoras aunque por otro lado, permite un

mejoramiento en la organización de los procesos de desarrollo para que tenga un aumento en la producción y un nivel de calidad en los productos desarrollados.

**Ilustración 3:** Comparación entre enfoque tradicional y ágil

Tópico	Enfoque tradicional	Enfoque ágil
Ciclo de vida de desarrollo	Lineal, modelo ciclo de vida (cascada, espiral o alguna variación)	Iterativo, modelo evolutivo
Estilo de desarrollo	Anticipado	Adaptativo
Requerimientos	Conocidos tempranamente, estables, claramente definidos y documentos.	Emergentes, de cambios rápidos, desconocidos (Se descubren durante el desarrollo del proyecto)

Fuente: [7].

### 3.2.2.1 Metodologías de desarrollo tradicionales

Las metodologías de desarrollo tradicionales o también llamadas metodologías pesadas, tales como cascada, modelo-V o RUP, son metodologías que se basan en una serie de iteraciones secuenciales, llamadas fases. Estas fases dependen de un conjunto de procesos y documentación predeterminada que son elaboradas durante el desarrollo del proyecto y sirven de guía para futuros desarrollos [7].

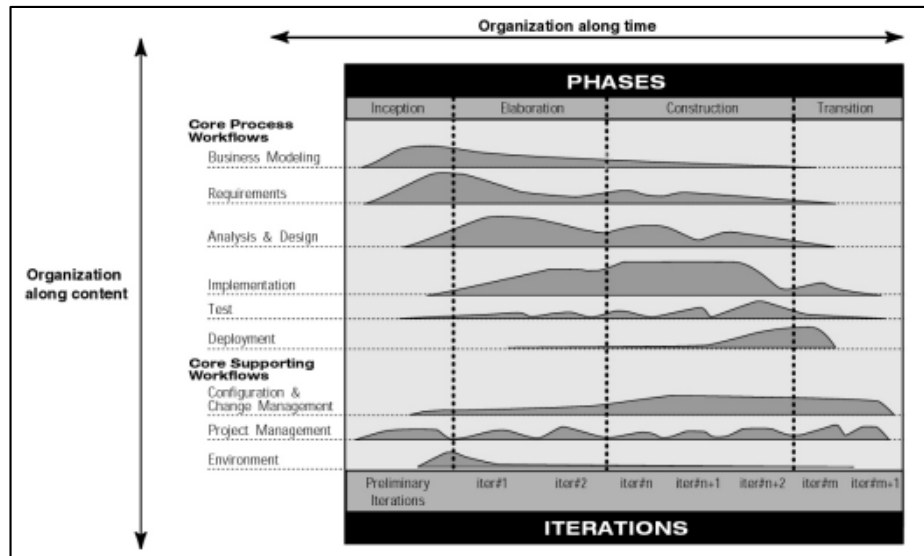
Por ende se puede acotar que dentro de las metodologías de desarrollo tradicional se basa en determinar y conocer los requerimientos para empezar con la fase del desarrollo, tomando en cuenta que puede existir cambios en dicha fase lo que provoca que se complique, por otro lado facilita una mejor determinación de los costos del proyecto, una organización del cronograma y asigna los recursos necesarios para cada actividad.

#### a. Rational Unified Process

RUP es un proceso de ingeniería de software que ofrece un enfoque estructurado y disciplinado para poder asignar las tareas y recursos dentro de la organización para mejorar la productividad ya que proporciona a cada miembro del equipo un acceso rápido a una base de conocimientos con guías, plantillas y herramientas para cualquier actividad crítica de desarrollo [7].

En este método ya no importa la fase en la que va a trabajar ya que todos los miembros del equipo comparten una visión, lenguaje y procesos del desarrollo del software.

**Ilustración 4:** Rational Unified Process



Fuente: [7].

RUP se divide en cuatro fases: inepción, elaboración, construcción y transición, estas fases están divididas en iteraciones en donde se demuestra una parte del software, dentro de la fase de inepción se determina los objetivos del proyecto, en la fase de elaboración se analiza el problema, se crea un plan y se detalla los procesos e infraestructura, además se identifica actores y casos de uso, en la fase de construcción es donde se integran todos los componentes y funcionalidades en el producto, y finalmente la fase de transición en donde el producto ya puede ser entregado al usuario.

### 3.2.2.2 Metodologías de desarrollo ágiles

Las metodologías ágiles proporcionan una serie de pautas y principios junto a técnicas pragmáticas que hacen que la entrega del proyecto sea menos complicada y más satisfactoria tanto para los clientes como para los equipos de trabajo, evitando de esta manera los caminos burocráticos de las metodologías tradicionales, generando poca documentación y no haciendo uso de métodos formales. [9]

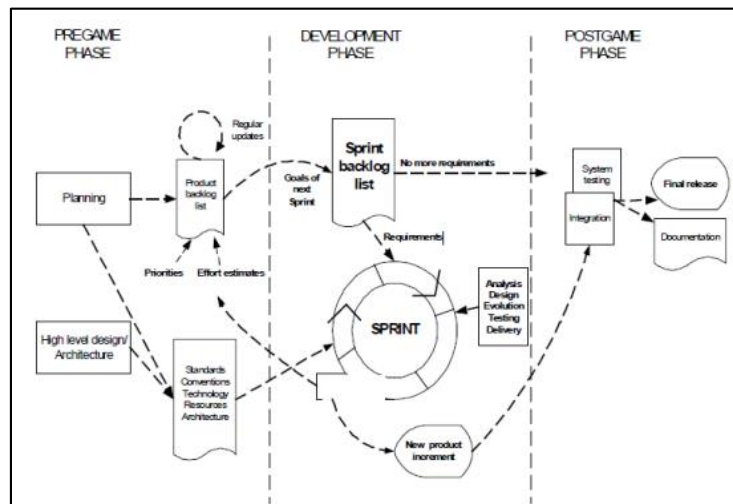
Por lo que las metodologías ágiles son caracterizadas por tener un proceso incremental, además que existe una relación de trabajo constante entre los clientes y desarrolladores, incluso para el equipo es fácil de aprender y de modificar, dentro de este método es posible realizar cambios de último momento lo que le hace una mejor opción para adoptar esta metodología ágil.

### a. Scrum

Scrum es un framework estructurado para soportar el desarrollo de productos de software complejos. Se compone de Scrum Teams y sus roles, eventos, artefactos y reglas asociadas. Cada componente dentro del framework tiene un propósito en específico y es esencial para su éxito y uso. [7]

De tal manera que la metodología Scrum permite que se involucren variables de entorno y técnicas que puede realizar cambios durante el desarrollo del sistema, sin embargo esta técnica no aplica para la fase de implementación, motivo por el que esta técnica se basa solamente en la funcionalidad de los miembros del equipo para que puedan producir un software flexible.

**Ilustración 5:** Modelo Scrum



**Fuente:** [7].

El proceso de Scrum consiste en tres fases, la fase de pre-game que se encarga de la planificación y la arquitectura, la fase de desarrollo puede existir cambios durante el proceso el cual son observadas y controladas por varias prácticas de Scrum durante los Sprints que son ciclos iterativos en donde la funcionalidad es desarrollada o mejorada para nuevos avances y finalmente la fase de post-game es el cierre del producto.

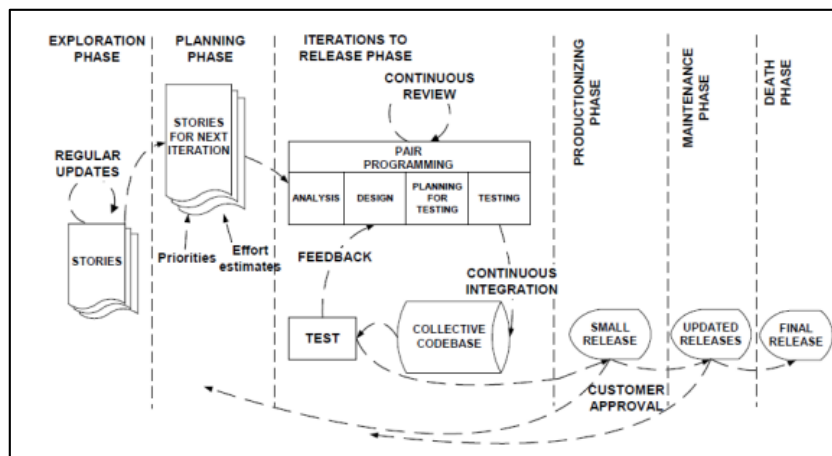
### b. Extreme Programming

Extreme Programming es uno de los procesos ágiles más populares junto con Scrum que han sido validados como casos de éxitos por varias compañías de diferentes tipos e industrias a través de todo el mundo. Debe su éxito a que se enfoca en la satisfacción del cliente, ya que ofrece el software

que se necesite en el momento que se necesite. Esto se debe a que capacita a sus desarrolladores para que puedan responder de la manera más eficiente a los cambios en los requerimientos del cliente, aun cuando estos se den en la última etapa del ciclo de vida [7].

Por ende se puede acotar que esta metodología se basa en la comunicación, la reutilización del código desarrollado y la realimentación, para una mayor calidad y mejor eficiencia del equipo de desarrollo ya que aplica prácticas de ingeniería apropiada para la creación de un software.

**Ilustración 6:** Extreme Programming



**Fuente:** [7].

Dentro de esta metodología se presentan cinco fases, la fase de exploración en donde los clientes escriben las funcionalidades que quieren que tenga el software, en la fase de planificación se toma como prioridad todas las historias de usuario para hacer un acuerdo de la primera versión, en la fase de iteraciones en el despliegue se descompone lo anterior mencionado para hacerlos por partes, en la fase de pases a producción se realiza pruebas para ver el rendimiento del producto, y finalmente la fase de la muerte en donde el cliente ya no tiene historias de usuario para implementar y es donde ya se escribe la documentación el proceso final del proyecto.

### 3.3 DevOps

#### 3.3.1 Definición de DevOps

DevOps se centra en los objetivos más que en los medios, por lo que DevOps se define como un conjunto de prácticas destinadas a reducir el tiempo entre la realización de un cambio en un sistema y la puesta en producción normal del cambio asegurando una alta calidad. [18]

Por ende se puede acotar que DevOps es una metodología en el desarrollo de software en donde se basa de la integración de codificadores y administradores de sistemas, de tal manera que los desarrolladores se prioricen del desarrollo como tal y no tanto del despliegue ya que el código se lo puede desplegar en segundos. De tal manera que DevOps es hacer Integración Continua (CI) y Despliegue continuo (DC) en el desarrollo de software.

### **3.3.1.1 Principios de DevOps**

Damon Edward y John Willis describieron a DevOps en 2010 con el acrónimo (CAMS) [19], que describen elementos importantes para la organización de DevOps:

#### **a. Cultura:**

Establece expectativas, prioridades y creencias fundamentales que guíen, determinando un vínculo entre la cultura y las prácticas de gestión, derivadas de principios de administración ágil y un enfoque para la resolución de problemas, es por ello que dicha colaboración debe estar presente dentro y entre los equipos.

#### **b. Automatización:**

Para lograr una automatización de procesos es necesario conocer las ineficiencias, de tal manera que la automatización permita potenciar las capacidades de los equipos liberando tiempo para un trabajo de alto nivel y abriendo oportunidades para innovaciones.

#### **c. Medición:**

Mide la producción y los comparte con los involucrados en el desarrollo y entrega de software, permitiendo mejorar los procesos, midiendo y reportando todo.

#### **d. Compartir (Sharing):**

Las organizaciones y el software requieren de equipos de personas con diferentes habilidades y conocimientos especializados para que de esta forma se mantenga juntos y compartan datos de progreso que ayudara a fomentar la cultura de DevOps, y que se exponga métricas de forma simple y clara a todas las personas dentro de la organización.

### **3.3.1.2 Ciclo de vida DevOps**

DevOps es una profunda integración entre los desarrolladores y operaciones que consta de un ciclo de vida [20]:

#### **a. Desarrollo:**

El desarrollo de software se lleva a cabo constantemente, por lo que todo el proceso se divide en pequeños ciclos de vida de desarrollo, beneficiando al equipo de DevOps para agilizar el desarrollo de software y el proceso de entrega.

#### **b. Prueba:**

El equipo de QA situado entre el cliente y el equipo de desarrollo usa herramientas con entorno de pruebas para identificar y corregir errores en una nueva parte del código.

#### **c. Integración:**

Dentro de esta nueva etapa, la nueva funcionalidad se integra con el código vigente y las pruebas se llevan a cabo, tomando en cuenta que el desarrollo continuo es posible gracias a la integración y pruebas continuas.

#### **d. Despliegue:**

En esta etapa el proceso de implementación se lleva a cabo de forma continua ya que al realizar un cambio en el código en cualquier momento, este no debe afectar el funcionamiento del sitio web.

#### **e. Monitoreo:**

Dentro de esta fase, el equipo de operación se encarga del comportamiento adecuado del sistema o de los errores que se puede encontrar en producción.

## **3.4 INTEGRACIÓN CONTINUA**

### **3.4.1 Definición**

la Integración Continua es una práctica de desarrollo de software en la que los miembros de un equipo integran su trabajo con frecuencia, por lo general, cada persona se integra al menos a diario, lo que lleva a múltiples integraciones por día. Cada integración se verifica mediante una compilación automatizada (incluida la prueba) para detectar errores de integración lo más rápido posible. Muchos equipos encuentran que este enfoque conduce a problemas de integración

significativamente reducidos y permite que un equipo desarrolle software cohesivo más rápidamente [15].

De tal manera que la Integración Continua es una práctica en donde se fusiona el código de trabajo en canales compartidos, reuniendo todas las modificaciones del código, preparándolo para su producción, y probarla para finalmente lanzar el código, ya que la Integración Continua indica una infraestructura de pruebas y desarrollo de software mejorando la calidad del proyecto con poco de esfuerzo y tiempo.

La Integración Continua y el despliegue continuo tienen que ir de la mano en un entorno ágil ya que permite incorporar, corregir errores y configurar código en un ambiente de producción automático, motivo por el cual la Integración Continua se determina como una fase inicial para la entrega y el despliegue continuo en donde combina partes del código principal sin interrumpir el código colectivo, llevándola a una ejecución de pruebas para que los errores no interfieran con el producto final y que esta no se dañe al incorporar nuevos códigos.

Por otro lado la entrega Continua hace referencia a la práctica de desarrollo de software en la cual se crea, prueban y preparan de modo automático los cambios y se entregan para la fase de producción. Esto significa que la Entrega Continua permite extender la Integración Continua al implementar todos los cambios en un entorno de pruebas y de producción después de la fase de creación, lográndose obtener un artefacto listo para su implementación, que ha sido sometido a un proceso de pruebas estandarizado [16].

Se puede acotar que la entrega continua es un método en el cual se puede liberar el cambio, ya que este procedimiento lo autoriza y garantiza llevándolo al último paso de implementación, tanto el despliegue como la entrega continua se produce cuando las modificaciones pasan por la fase de desarrollo y están listas para ser lanzadas.

### **3.4.2 Prácticas de integración continua**

Todas las prácticas de integración continua se muestran a continuación:

**a. Mantener siempre un repositorio de código fuente.**

El desarrollo de software implica a un conjunto de ficheros que necesitan ser organizados para elaborar un producto, así mismo hacer el debido seguimiento de ellos implica altos costos y más aún si se involucran varias personas para realizar la actividad. Como solución a este problema existe la herramienta de gestión de código “GIT” que además forma parte de las herramientas implementadas en el presente proyecto [17].

Para la práctica de integración continua es importante principalmente que el código fuente del proyecto se encuentre en el repositorio y sea actualizado constantemente. Sin embargo, se debe tomar en cuenta que esta es una práctica en donde puede llevarse a cabo en cualquier proceso de desarrollo, el cual podría intervenir la creación y uso de más elementos de trabajo. Pero en la actualidad existen muchas herramientas que permiten gestionar el repositorio con nuestro código fuente.

**b. Automatizar la construcción.**

La construcción de un software implica compilar el código, para ganar tiempo y evitar muchos errores, esta tarea puede ser automatizada por dos herramientas como son Maven o Ant [17].

Una construcción grande a menudo toma tiempo, no desea realizar todos estos pasos si solo ha hizo un pequeño cambio. En donde el objetivo principal de automatizar la construcción es simplificar y agilizar los procesos de construcción.

**c. Entregar todos los días el código actualizado a la línea principal.**

La integración ronda principalmente alrededor de la comunicación. La integración permite a los desarrolladores decir a otros desarrolladores acerca de los cambios que han hecho. La comunicación frecuente permite a la gente a conocer más rápidamente los cambios que se desarrollan. El pre-requisito para que un desarrollador realice un commit a la línea principal (Base line – Línea Base) es que pudo construir correctamente su código lo que incluye, por supuesto, superar las pruebas de compilación. [17]

La comunicación frecuente permite que los desarrolladores sepan rápidamente a medida que se realizan los cambios en el código fuente del proyecto. El único requisito previo para que un desarrollador se comprometa con la línea principal es que pueda construir correctamente su código.

El desarrollador primero actualiza su copia de trabajo para que coincida con la línea principal, resuelve cualquier conflicto con la línea principal. Las confirmaciones frecuentes alientan a los desarrolladores a dividir su trabajo en partes pequeñas de unas pocas horas cada uno ayudando a seguir el progreso y proporciona una sensación de progreso.

**d. Realizar pruebas en un ambiente clonado del entorno de producción.**

El riesgo de que ocurran errores en desarrollo y no en producción o viceversa es mayor si se realizan las pruebas en un ambiente con una versión distinta a la de producción. Para reducir estos riesgos es recomendable homologar en ambos entornos la misma versión de archivos de configuración, las librerías a usar, etc. [19]

El objetivo aquí es eliminar el riesgo asociado al entorno, ya que si los entornos son diferentes no estamos en condiciones de asegurar que lo que pasó en el entorno de pruebas será lo que sucederá (o no) en el entorno de producción. Sabemos que puede ser muy difícil e incluso hasta imposible tener ambiente clonados por cuestiones de costo. Debido a esto, una buena opción es utilizar máquinas virtuales para simular los ambientes.

**e. Automatizar el despliegue.**

La integración continua necesita de varios entornos para llevarse a cabo, como todos conocemos las áreas comunes son Desarrollo, aseguramiento de la calidad y producción quienes son las que interactúan diariamente y por lo tanto el proceso operativo debe ser más engorrosos para esto es mejor que los despliegues sean automatizados mediante scripts y hacer esta actividad de manera sencilla. [18]

Convertir las fuentes en un sistema en funcionamiento a menudo puede ser un proceso complicado. Implica compilación, movimiento de archivos, carga de esquemas en las bases de datos de manera rápida. Sin embargo, como la mayoría de las tareas en esta parte del desarrollo de software, puede ser automatizado y como resultado debe ser automatizado. Pedirle a la gente que escriba extraños comandos o hacer clic en los cuadros de diálogo es una pérdida de tiempo y muchos errores por eso es muy buena idea automatizar esos procesos o tareas para lograr un rápido despliegue en diferentes ambientes.

### 3.4.3 Beneficios de la integración continua

A continuación, podemos mencionar los beneficios principales de la integración continua:

#### **a. Mayor productividad para el desarrollo.**

Con la práctica de la IC un equipo puede obtener una retroalimentación temprana ya que en el caso de que el software no compile o no pase las pruebas automáticas el sistema alertará al equipo. Se puede obtener un “build” de forma automática cada vez que se suba el código al repositorio si así se lo desea. Esto implica una reducción del riesgo a la hora de integrar el producto y permite la detección temprana de errores, ya que estos se detectarán también en la subida de código al ejecutarse las pruebas de integración y aceptación de forma automática. [20]

Con los procesos automatizados, los desarrolladores dejan de realizar tareas manuales y complicadas en donde les quitan el tiempo y energía al momento de realizar un proyecto. De esta forma, el equipo de trabajo aumenta su productividad, creando soluciones que sean ágiles para el proceso de desarrollo de software.

#### **b. Detección rápida de las fallas.**

Debido a que integramos y ejecutamos las pruebas e inspecciones (de código) varias veces al día, hay una mayor probabilidad de que se descubran y reparen los defectos encontrados más rápidamente. Esto se debe a que es más acotado el espectro en el cual se debe buscar y corregir el mismo porque la cantidad de código modificado en cada integración es menor. [21]

Al realizar las pruebas es posible encontrar e investigar bugs antes de que los proyectos se desarrollen y así poder evitar que el trabajo realizado se pierda y tenga que iniciar a realizar el proyecto desde cero.

#### **c. Distribución rápida de las actualizaciones.**

La integración frecuente de código conduce a la reducción de los niveles de riesgo de cualquier proyecto. Los defectos de código se pueden detectar más pronto y se arreglan más rápido, por tanto, podemos tener métricas para medir el estado de la aplicación. Cuando un equipo de desarrollo integra su trabajo con frecuencia significa que hay poco recorrido entre el estado actual de la aplicación y lo que el desarrollador está implementando. Y así se reduce la posibilidad de supuestos. [21]

Los sistemas de integración continua ayudan a los equipos de trabajo distribuyan las actualizaciones sobre los usuarios o empresas de manera frecuente y rápida.

### **3.5 HERRAMIENTAS DE INTEGRACIÓN CONTINUA**

Existe una gran cantidad de herramientas en el mercado las cuales permite implementar Integración y Despliegue Continuo, dentro de las principales herramientas están Jenkins, Travis, Circle CI, TeamCity, Bamboo.

#### **3.5.1 Jenkins**

Es una aplicación que ayuda a los desarrolladores a realizar Integración Continua de proyectos de software y ayuda detectar y corregir errores de integración de forma continua. Fue creado del desarrollo original Hudson. Esta herramienta está implementada en lenguaje de programación Java por lo siguiente para poder utilizarlo es necesario una máquina virtual. [22]

Por ende Jenkins es una herramienta de Integración Continua y ayuda a localizar, corregir y detectar errores de una manera rápida y continua, además es muy útil para diferentes lenguajes de programación permitiendo facilitar el trabajo a los desarrolladores de una forma rápida y sobre todo disminuyendo el costo y el tiempo.

##### **a. Jenkins Pipeline**

Uno de los mayores beneficios de usar pipelines y Jenkins es poder hacer los jobs más configurables, fáciles de editar y más durables. También podemos utilizar entradas (input) humanos o esperar la aprobación de otro job para ejecutar el nuestro mediante pipelines [19].

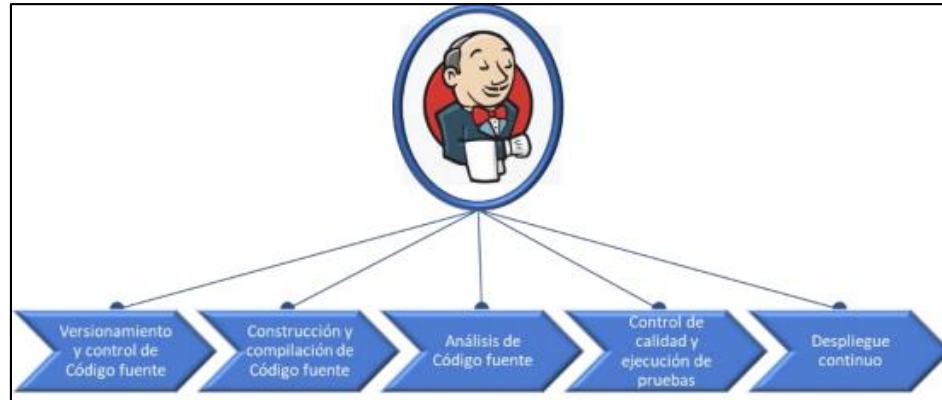
Por ende pipeline con Jenkins tiene una capacidad de suspender o reanudar de ejecución de trabajos, comprueba pipeline en el control de código fuente a través de un archivo, incluso cuenta con soporte para ampliar el idioma específico del dominio con pasos adicionales de la organización por medio de bibliotecas compartidas.

##### **3.5.1.1 Modelo de integración continua con Jenkins**

Como se vio anteriormente, Jenkins es utilizado como una herramienta para la automatización de la integración continua, por lo que este cuenta con una serie de plugins o complementos, que no

son más que conectores a otras aplicaciones para interactuar dinámicamente con ellas; es decir, Jenkins sería el encargado de las ejecuciones de cada aplicación en cada una de las fases de integración continua.

**Ilustración 7:** Modelo básico de Integración Continua con Jenkins



**Fuente:** [21].

Para Jenkins, el modelo es muy sencillo, ya que se basa en dichas conexiones y sus configuraciones de entrada para que cada aplicación ejecute las operaciones que cada una deba realizar en cada una de las fases del proceso de integración continua. Lo que lo hace complejo, es la configuración de las operaciones que cada herramienta de las que está conectada debe realizar independientemente.

Jenkins permite la ejecución de tareas manuales o programadas en cada fase; brindan como una respuesta, las estadísticas, reportes e informes del resultado de la operación, el cual puede servir como el disparador o trigger, para la siguiente operación dentro del flujo en el modelo previamente diseñado. Pero al final, la intención de Jenkins es tener todo el flujo automatizado para obtener un resultado final con la menor intervención humana posible en cada paso del flujo.

### **3.5.1.2 Plugins en Jenkins**

Los complementos (plugin) extienden su uso a proyectos escritos en lenguajes distintos de Java. Hay complementos disponibles para integrar Jenkins con la mayoría de los sistemas de control de versiones y bases de datos, los complementos también pueden cambiar la apariencia de Jenkins o agregar nuevas funciones.

### 3.5.2 Travis

Travis CI se basa en su proceso de desarrollo donde crea y prueba automáticamente los cambios de código, proporcionando información inmediata sobre el éxito del cambio. Travis CI también puede administrar otros procesos del desarrollo administrativo, como los despliegues de nuestros proyectos en algún entorno específico y también provee notificaciones, lo cual parece bien para mantenernos al tanto de lo ocurrido. [23]

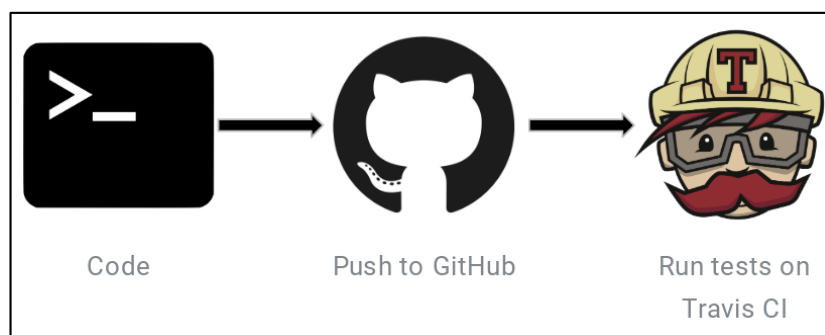
Por ende la herramienta Travis se centra en el rendimiento del proceso de construcción con pruebas automatizadas y con sistema de alerta, aquí los usuarios pueden probar el código rápidamente, ya que hace un seguimiento de los cambios dando como notificación si el cambio fue o no exitoso.

#### a. Archivo `.travis.yml`

Travis YML es un archivo de configuración ubicado en la carpeta raíz de su proyecto para definir cómo configurar la información relacionada con la compilación, esto permite que su configuración sea flexible y con control de versiones, YAML tiene su propia sintaxis amigable y es un lenguaje útil para agregar a su desarrollo, tiene para lenguajes más populares [24].

Por ende el archivo `.travis.yml` permite configurar el comportamiento de la compilación dentro de Travis CI, ya que es la raíz del repositorio del proyecto y si este archivo no está entonces la compilación no se llevará a cabo, este es un archivo de lenguaje humano quiere decir que el desarrollador configura de manera manual de acuerdo a su necesidad.

**Ilustración 8:** Integración Continua con Travis CI



**Fuente:** [24].

Travis CI nos permite automatizar el proceso de prueba, integración, entrega e implementación del código, lo que nos permite incorporar los principios de DevOps y Agile en nuestras prácticas de

código desde la implementación hasta la producción. Completamente alojado y listo para usar con solo unos pocos clics de un botón, Travis CI se vincula con su sistema de control de versiones existente y lo prepara para automatizar las pruebas de compilación y más con un mínimo esfuerzo inicial: se acabaron los días de tener que suministrar su propia prueba de infraestructura!

### **3.5.3 CircleCI**

CircleCI es una herramienta basada en la nube que automatiza el proceso de integración e implementación. También se enfoca en probar cada cambio de código antes de su implementación, utilizando métodos como pruebas unitarias, pruebas de integración y pruebas funcionales. Se integra perfectamente con el sistema de control de versiones actual. [25]

De tal manera se puede acotar que la herramienta CircleCI es una plataforma de integración y entrega continuas en donde ayuda a que los equipos de software liberaren código mucho más rápido y con confianza al automatizar el proceso de construcción, prueba e implementación, además tiene una plataforma de desarrollo de software moderna que permite a los equipos aumentar.

### **3.5.4 TeamCity**

La herramienta comprueba los cambios en el código antes de integrarlos a la línea principal. Únicamente cuando el código está libre de errores, pasa a formar parte del código base para todo el equipo. TeamCity lleva a cabo las pruebas automáticamente en un segundo plano, de modo que el desarrollador puede continuar trabajando. [26]

Es por ello que la herramienta TeamCity viene siendo un servidor de Integración Continua llegando a ser fácil de usarlo para aquellas personas desarrolladoras, además es ligero al momento de configurar incluso es gratis para equipos pequeños y ciertos proyectos de código abierto.

Para los desarrolladores que se van por el lado de TeamCity buscan una mejor capacidad de codificación, en donde integran herramientas de compilación y pruebas visualizando procesos de DevOps permitiendo un mejor rendimiento, así como la facilidad de configurar los procesos sin hacer uso de códigos, de tal manera que lleva un registro de las pruebas que identifica los fallos del proceso en tiempo real.

### **3.5.5 Bamboo**

Es el servidor de automatización importante utilizado para CI que crea, prueba e implementa automáticamente el código fuente y configura la aplicación para su implementación. Esta herramienta con licencia ofrece una visibilidad de alto nivel sobre todo el proceso de desarrollo, prueba e implementación, es compatible con numerosos idiomas y permite a los desarrolladores de software utilizar metodologías de CI / CD. [24]

Es por ello que la herramienta Bamboo es está orientado a la Integración Continua y despliegue continuo ya que permite resistencia, fiabilidad y escalabilidad de equipos de todo tipo de tamaño además se destaca en ser la más sencilla integración con diferencia de otras herramientas.

### **3.5.6 Características de las herramientas de Integración Continua**

Debido al rápido crecimiento y a la gran variedad de opciones en el mercado de la tecnología de información, ya que la selección de las herramientas adecuadas para un desarrollo de software se convierte en una tarea muy difícil.

Al principio del proceso de la elección de las herramientas para la implementación de la Integración Continua no son muy claros, de tal manera que es importante averiguar las características que se requiere para hacer una buena selección de la herramienta.

Según [27] afirma que la mejor herramienta de Integración Continua debe ahorrar el tiempo y costes de un proyecto, además que la elección de la herramienta no debe ser la única que use, más bien que el desarrollador aplique las herramientas que se adapte a sus necesidades, entre las características que se toma en cuenta para una buena Integración Continua son:

- La ejecución del build: característica básica de una herramienta de Integración Continua, según [27] manifiesta que el verdadero sistema de Integración Continua debe ser del tipo de ejecución de build impulsada por eventos. De tal manera la compilación se realiza después de cualquier cambio que se haga en el repositorio, es una opción sin embargo requiere tiempo adicional en la configuración del repositorio de control del versiones, mientras que un sistema impulsado por sondeo no es necesario.
- El mantenimiento del sistema de control de versiones en un desarrollo de software es muy importante ya que se debe prestar atención a factores como la forma en que la herramienta

de Integración Continua interactúa con los métodos que usa los repositorios, aunque el servidor de Integración Continua tiene que reconocer los archivos modificados y sus versiones.

- La retroalimentación es otro de los factores clave de la Integración Continua, ya que soporta diferentes tipos de retroalimentación, puesto que es crítica para optimizar los procesos de DevOps (creación, prueba, entrega, implementación y retroalimentación del código para iniciar de nuevo con el código), en donde las herramientas habilitan bucles de retroalimentación es ahí donde los equipos de DevOps deben reducir el ruido innecesario y medir continuamente el éxito.
- La interfaz de usuario debe tenerse en cuenta ya que su diseño puede facilitar el tiempo en los primeros periodos mientras trabaja con las herramientas de Integración Continua.
- Las características de seguridad están incorporadas en cada herramienta dentro su sistema, esto quiere decir que incluyen autenticación y autorización, que permite el acceso de control a la revisión de resultados y modificación de la configuración.
- La extensibilidad quiere decir que existe la posibilidad de ampliar la funcionalidad.
- La compatibilidad es otro de los factores importantes al momento de seleccionar la herramienta de Integración Continua.
- La fiabilidad es importante ya que la herramienta de Integración Continua con un largo periodo de existencia en el mercado y una buena reputación es una opción fiable a comparación de otras que recién se incorporen al mercado tecnológico.
- La longevidad es un factor que afecta al futuro, ya que la fiabilidad se centra en el pasado y presente de la vida útil de la herramienta.
- La usabilidad se refiere a la facilidad de configurar y usar la herramienta para el usuario.

Posterior al análisis de una revisión bibliografía de las diferentes herramientas que compara StackShare se determina que las herramientas de Integración Continua más efectivas y popularmente las más utilizadas son Jenkins y Travis, por ello el caso de estudio a realizarse será entorno a estas dos herramientas.

## **3.6 METODOLOGIAS**

La calidad de un instrumento depende de sus características psicométricas, las cuales se conocen mediante su fiabilidad y validez, estos permiten conocer la consistencia interna del instrumento, la pertinencia y claridad de los ítems, así como la estructura latente del cuestionario [28].

De tal manera que la medición es una parte importante en un proyecto de investigación ya que da validez a la investigación que se realiza, esto a través de escalas de medición de intervalo o un test para tener resultados estadísticos y ver su fiabilidad.

### **3.6.1 Metodologías de investigación**

La metodología de investigación es el conjunto de decisiones coherentes, generales y abstractas que el investigador toma sobre cómo obtener determinados tipos de datos de la realidad que estudia, pero los cuales quedarán objetivamente reflejados en los modos en que se acercara a la realidad y obtendrá datos de esta, con la utilización de métodos, técnicas y herramientas [11]. De tal manera que las metodologías conforma un plan de estudio el cual permite llegar a sustentar el tema que se va a investigar a través de varios métodos, técnicas e instrumentos que ayuda a la recolección de la información.

#### **3.6.1.1 Tipo de investigación**

##### **a. Investigación cualitativa**

La investigación cualitativa es aquella que se basa en la obtención de datos en principio no cuantificables, basados en la observación. Aunque ofrece mucha información, los datos obtenidos son subjetivos y poco controlables y no permiten una explicación clara de los fenómenos. Se centran en aspectos descriptivos [12].

De tal manera que la investigación cualitativa no es más que un análisis de los datos que no son numéricos para poder obtener una información de los fenómenos que se va a estudiar, de forma detallada, este tipo de investigación también es adaptable a cualquier tipo de estudio.

### **3.6.1.2 Nivel de investigación**

#### **a. Investigación descriptiva**

El objetivo de este tipo de investigación es exclusivamente describir; en otras palabras: indicar cuál es la situación en el momento de la investigación. Su informe debe de contener el ser, no el deber ser. Después de describir se puede interpretar, inferir y evaluar. Esta es la base y fundamento de otras investigaciones [13].

De tal manera que la investigación descriptiva como su nombre lo indica esta establece una descripción casi completa de un objeto de estudio, sin la búsqueda de causas ni consecuencias del fenómeno, esta puede medir características simplemente se busca tener una idea clara de la situación.

### **3.6.1.3 Diseño de la investigación**

#### **a. Diseño narrativo**

El diseño narrativo en diversas ocasiones es un esquema de investigación, pero también es una forma de intervención, ya que el contar una historia ayuda a procesar cuestiones que no estaban claras. Se usa frecuentemente cuando el objetivo es evaluar una sucesión de acontecimientos [14].

De tal manera el diseño de investigación narrativo consta de la recolección de datos que realiza el investigador sobre las experiencias determinadas personas para describirlas y analizarlas, los datos se obtiene a través de entrevistas, documentos, entre otros.

## **3.6.2 Metodologías de desarrollo**

### **3.6.2.1 Prácticas ágiles**

#### **a. Concepto**

Normalmente nos referimos a las prácticas Ágiles como a un conjunto de prácticas que son empleadas habitualmente en los métodos y marcos de trabajo Ágiles. El concepto también incluye algunas prácticas inventadas, o aplicadas exclusivamente en entornos de trabajo Ágil. Sin embargo, eso no quiere decir que sean exclusivamente Ágiles. De hecho muchas de estas prácticas se empelaban anteriormente en entornos tradicionales. [10]

De tal manera que las prácticas ágiles no solo se aplican en metodologías ágiles sino que también se las puede involucrar en las metodologías tradicionales siempre y cuando haya un beneficio para el sistema así se puede ajustar la práctica al sistema.

### **b. Ejemplos**

Las prácticas ágiles pueden ser beneficiosas tanto para las metodologías tradicionales como para las metodologías ágiles dependiendo si es que se requiere de la implementación de las prácticas que se muestran a continuación:

- La propiedad colectiva del código
- Programación por paros
- Reuniones diarias de pie (daily stand-ups)
- Radiadores de Información
- Refactorización continua
- Auto-organización

## **3.7 TÉCNICA DE VALIDACIÓN**

### **3.7.1 Validación por expertos**

Según [29] afirma que “La evaluación mediante el juicio de experto consiste, básicamente, en solicitar a una serie de personas la demanda de un juicio hacia un objeto, un instrumento, un material de enseñanza, o su opinión respecto a un aspecto concreto”.

La validación por experto es un método que permite saber si una investigación es fiable o no en donde intervienen personas expertas en el área y evalúan de acuerdo a una escala de medición que permite conocer los resultados de dichos expertos.

### **3.7.2 Alfa de Cronbach**

El coeficiente alfa de Cronbach es una fórmula general para estimar la fiabilidad de un instrumento en el que la respuesta a los ítems es dicotómica o tiene más de dos valores, expresa esta consistencia interna a partir de la covariación entre los ítems del cuestionario o test, de manera que cuanto mayor es la covariación, mayor puntuación alfa [30].

Por ende el test de Cronbach requiere una admiración de medición, aquí no es necesario que se dividan en dos mitades los ítems del instrumento de medición, únicamente se aplica la medición y se calcula el coeficiente, lo cual permite una calcular la fiabilidad de la investigación.

Para el cálculo de alfa de Cronbrach se emplean las siguientes variables [31]:

Mediante la varianza de los ítems:

$$\alpha = \frac{K}{K - 1} \left[ 1 - \frac{\sum Vi}{Vt} \right]$$

Donde:

$\alpha$  = Alfa de Cronbach

$K$  = Número de ítems

$Vi$  = Varianza de cada ítem

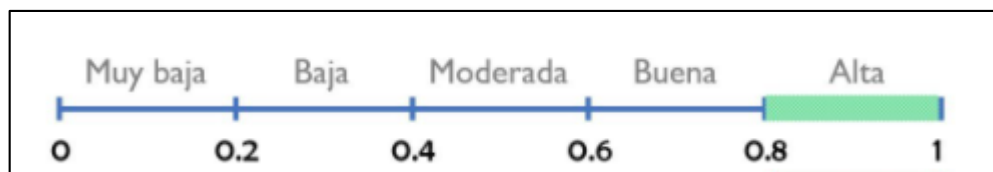
$Vt$  = Varianza del total

De tal manera que al emplear la formula este permite conocer el valor de la investigación, de acuerdo a los datos que se recolecte de los expertos que van a evaluar.

### **3.7.3 Interpretación del Coeficiente de Alfa de Cronbach**

El valor mínimo aceptable para el coeficiente alfa de Cronbach es 0,70; por debajo de ese valor la consistencia interna de la escala utilizada es baja. Por su parte, el valor máximo esperado es 0,90; por encima de este valor se considera que hay redundancia o duplicación. Varios ítems están midiendo exactamente el mismo elemento de un constructo; por lo tanto, los items redundantes deben eliminarse. Usualmente, se prefieren valores de alfa entre 0,80 y 0,90. Sin embargo, cuando no se cuenta con un mejor instrumento se pueden aceptar valores inferiores de alfa de Cronbach, teniendo siempre presente esa limitación.

**Ilustración 9:** Valoración del Coeficiente de Alfa de Cronbach



Fuente: [35].

Es necesario tener en cuenta que el valor de alfa es afectado directamente por el número de ítems que componen una escala. A medida que se incrementa el número de ítems, se aumenta la varianza sistemáticamente colocada en el numerador, de tal suerte que se obtiene un valor sobreestimado de la consistencia de la escala. De igual manera, se debe considerar que el valor del alfa de Cronbach se puede sobreestimar si no se considera el tamaño de la muestra: a mayor número de individuos que completen una escala, mayor es la varianza esperada.

## 4 MATERIALES Y MÉTODOS

### 4.1 METODOLOGÍA DE INVESTIGACIÓN

Para la ejecución del proyecto propuesto es importante recurrir al siguiente tipo de investigación el cual ha sido analizado por los investigadores, a continuación se presentará las razones o criterios de selección de cada una de ellas:

#### 4.1.1 Tipo de investigación cualitativa

En el proyecto de investigación se usará el tipo de investigación cualitativa porque se basa en una comparativa que interpreta conclusiones tentativas y revisiones que se comprueban ya que el estudio que se realiza no es lo último, más bien es tomado para el caso de estudio que estamos resolviendo, realizando estudios más estructurados con la aplicación de dos estrategias.

#### 4.1.2 Nivel de estudio descriptivo

La investigación cualitativa con nivel de estudio descriptivo con medición de variables independientes permitirá valorar el uso de las herramientas de Integración Continua, Jenkins y Travis CI, ya que en nuestro caso para realizar el proceso de Integración Continua en proyectos de software depende de las herramientas para llevar a cabo el proceso.

### **4.1.3 Diseño de investigación narrativo**

Esta investigación de diseño narrativo permitirá narrar lo que sucederá al probar las dos herramientas para la integración continua, aquí se detallara de principio a fin el uso de las herramientas Jenkins y Travis CI aplicando a un software web creado por las investigadoras.

## **4.2 TÉCNICAS E INSTRUMENTOS DE INVESTIGACIÓN**

### **4.2.1 Observación**

Esta técnica de investigación ayudó a seleccionar la herramienta que mejor alternativa tiene para los procesos de integración continua basándose en los resultados que se obtiene a través del análisis de las características.

### **4.2.2 Fichas bibliográficas**

Esta técnica permitió revisar el material bibliográfico sobre las herramientas de integración continua Jenkins vs Travis CI ya que se recolectó datos importantes que ayudó a comprobar la hipótesis del proyecto.

### **4.2.3 Juicio de expertos**

Esta técnica se utilizó para validar el contenido del proyecto de investigación, enfocándose en el caso de estudio de las herramientas Jenkins y Travis CI para el proceso de integración, permitiendo conocer la escala en la que se encuentra nuestro estudio.

### **4.2.4 Cuestionario.**

Este instrumento permitió formular preguntas que ayuda a obtener información para ver fiabilidad del proyecto de investigación a través del coeficiente Alfa de Cronbach y con los datos recolectados poder analizar que herramienta es la mejor alternativa en los procesos de integración continua.

## **4.6 METODOLOGÍA DEL DESARROLLO**

### **4.6.1 Prácticas ágiles**

Esta metodología ayudó con la ejecución de los casos prácticos, ya que se usó la propiedad colectiva del código y trabajo en equipo, al ser un trabajo en equipo se necesitó de un repositorio

para almacenar el código fuente del proyecto que este caso se utilizó GitHub. Para el proceso práctico de integración continua se aplicó en un sistema web realizado en Visual Studio Code.

#### **4.5.1.1 Herramienta Jenkins**

La herramienta Jenkins ayudará en la ejecución de los procesos de integración continua, a través del trabajo en equipo, para realizar las pruebas, se hará uso del repositorio GitHub para tener el código fuente compartido lo cual será realizado en Visual Studio Code con el lenguaje de programación Python y durante la práctica poder analizar la funcionalidad y las características que tiene.

#### **4.5.1.2 Herramientas Travis**

Esta herramienta nos ayudará en la demostración de los procesos de integración continua, para realizar la ejecución de las pruebas se utilizará GitHub como repositorio del código fuente el cual será realizado en Visual Studio Code en lenguaje Python, y durante la practica poder observar y analizar la funcionalidad y las características que tiene.

## 5 ANÁLISIS Y DISCUSIÓN DE LOS RESULTADOS

### Resultados de la comparación de las herramientas de Integración Continua caso de estudio Jenkins vs Travis.

#### 5.1 Resultado del análisis comparativo.

Se realiza un cuadro comparativo de las características de las herramientas de Integración Continua Jenkins y Travis CI tras realizar una práctica con cada una de ellas, con el fin de analizar y mostrar resultados realizado por los investigadores.

**Tabla 1:** Cuadro comparativo de las herramientas de Integración Continua: Jenkins y Travis CI

Base de comparación	Jenkins	Travis
<b>Licencia</b>	Bajo licencia de software libre accesible para todos los usuarios que deseen usar la herramienta.	Tiene un plan anual y mensual: El mensual tiene en un mes de prueba gratis y los siguientes meses son de paga según los créditos que se necesite, y el plan anual no tiene prueba gratis.
<b>Instalación</b>	Se ha realizado la instalación de 2 maneras: <ul style="list-style-type: none"><li>• <b>Ubuntu:</b> se necesita de máquina virtual (VirtualBox), SSH, Docker, e imagen de Jenkins, esta instalación se realiza a través de comandos.</li><li>• <b>Windows:</b> se necesita de JDK y el programa ejecutable</li></ul>	No se requiere de instalación debido a que es un servicio en la nube en donde se necesita registrarse con una cuenta de Github, Bitbucket, Gitlab, y Assembla, además se necesita de una tarjeta de crédito para empezar a trabajar en la herramienta.

	de Jenkins, en esta instalación no se usa comandos.	
<b>Configuración</b>	Se realiza configuraciones manuales al momento de haber creado una nueva tarea.	Solo se necesita configurar el archivo YML directamente en el código fuente del proyecto.
<b>Notificaciones</b>	Se puede recibir notificaciones a través de Slack, para esto se necesita de plugins y configuración en Jenkins.	Las notificaciones de los resultados llegan automáticamente al correo electrónico, esto debido a que la cuenta de Github está vinculada con correo electrónico.
<b>Ejecución del proyecto (Build)</b>	La construcción del proyecto se necesita de configuraciones extras, como la conexión de Jenkins con Github usando credenciales y el tiempo en el que se va a realizar el build.	La construcción del proyecto se realiza automáticamente, lo cual no se necesita de una configuración previa.
<b>Control de versiones</b>	Para el control de versiones se usa la herramienta Git ya que registra los cambios del código fuente.	En Travis CI se usa la herramienta Git para guardar el registro de las modificaciones que se realiza en el código.
<b>Cambios (commit)</b>	Los cambios que se realizan primero se deben subir al repositorio para que la herramienta de Jenkins realice la construcción.	Los cambios en Travis CI se hacen de manera automática.

<b>Seguridad</b>	Jenkins tiene una autenticación y autorización muy amplia ya que permite manejar roles de usuarios.	En Travis CI la autenticación es vulnerable debido a que está vinculado a un repositorio público, sin embargo puede tener seguridad en proyectos privados.
------------------	---	--

**Fuente:** Los investigadores

Tras el análisis realizado por los investigadores ante la práctica de las herramientas de Jenkins y Travis CI, se puede observar que la herramienta Jenkins es la mejor opción para el proceso de Integración Continua ya que es amplia en muchos aspectos tanto como la configuración y seguridad además existe más información para conocer más sus funcionalidades, por otro lado la herramienta de Travis CI resulta más fácil para los procesos de Integración Continua sin embargo no proporciona seguridad para proyectos públicos.

Se realiza una investigación de artículos tecnológicos para realizar un análisis bibliográfico de las herramientas de Integración Continua Jenkins y Travis CI. Además se consultó en la herramienta

StackShare en donde varias personas que participan son expertas en el manejo de tendencias tecnológicas [32].

**Tabla 2:** Análisis bibliográfico de las herramientas de Integración Continua Jenkins y TravisCI

<b>Herramienta</b>	<b>Artículos bibliográficos</b>
<b>Jenkins</b>	<p>En el artículo [33] intenta reducir el riesgo y fortalecer pruebas de aceptación con Jenkins CI, en donde pasantes del Laboratorio de Pregrado en Ciencias ayudan a investigar y configurar un prototipo de sistema de prueba automatizado, se utilizó Jenkins como servidor maestro para ejecutar pruebas de Puppet, el proceso de pruebas consistía en hacer un cambio al entorno de producción de Puppet y al instante se realiza una prueba automatizada en Jenkins en donde permite ver el proceso de su ejecución en donde detalla el estado de la prueba y saber que salió mal o si todo salió con éxito.</p>
	<p>En el artículo [35] el sistema informático ATLAS Nightly System sirvió como una herramienta importante en los esquemas de administración y organización de software colaborativo de ATLAS. La adopción de prácticas de Integración Continua y el reemplazo de elementos antiguos del sistema con herramientas modernas de código abierto como Jenkins y GitLab permitió racionalizar la asignación de recursos de hardware y las operaciones administrativas, proporcionando un flujo de trabajo de desarrollo de software, acelerando los ciclos de innovación y aumentando la confianza en las implementaciones de software nuevo. Permitiendo mejorar la capacidad de ATLAS de soportar muchos desarrolladores y pruebas en grupos.</p>
	<p>En el artículo [37] demostraron que Jenkins es de código abierto y reconocido para formar una plataforma ágil para la ciencia, procesamiento de imágenes, datos que tiene facilidad en la configuración, mantenimiento y tiene un soporte para varios sistemas operativos y una gran cantidad de complementos preconstruidos, Jenkins puede ser utilizado en ambientes científicos donde haya necesidad de integrar y automatizar cálculos científicos y gestión de datos.</p>

	<p>En el artículo [39] Jenkins ayudó a los desarrolladores de software HPC ya que puede crear pipelines completos la cual ayuda en el proceso de gestión y aumento el potencial para entregar productos confiables y de alta calidad.</p>
	<p>En el artículo [41] presentó la fase del funcionamiento de Jenkins a través de pruebas de la TICs del ejército ecuatoriano en donde la herramienta permitió la verificación con actualizaciones y modificaciones que realizan el departamento de desarrollo al departamento de operaciones los cuales se conecta a través del repositorio subversión.</p>
<p><b>Travis</b></p>	<p>En el artículo [34] manifiesta que para el proceso de implementación utiliza un mecanismo automático para detectar problemas de integración lo antes posible. El sistema compila y prueba automáticamente en servicios de back-end para acelerar la implementación proceso. Travis CI permitió hacer el proceso de prueba, pasando por 3 procesos, subir el código al repositorio de Github, y se aprovechó la Integración Continua para hacer un despliegue automático y continuo usando Travis CI que construye y prueba e implementa el servicio en el servidor y lo implementan en el servidor de Heroku.</p>
	<p>En el artículo [36] los modelos computacionales son herramientas poderosas para explorar las propiedades de los sistemas biológicos complejos, es por ello que ha desarrollado Open Source Brain una plataforma que comparte, analiza y simula modelos de diferentes regiones cerebrales y especies que vincula repositorios de código abierto, donde el desarrollo de modelos necesita pruebas sistemáticas del código base, por ello el modelo de validación se instala localmente y se ejecuta usando Integración Continua con Travis CI permitiendo garantizar cualquier problema de compatibilidad con el simulador.</p>
	<p>En el artículo [38] una plataforma de generación de malla de código abierto para el modelo biofísico utilizando geometrías celulares realistas, y para la disponibilidad del código y fomentar la colaboración de la comunidad se ha usado repositorio de Github para hacer Integración Continua con TravisCI y Appveyor que garantizó la compatibilidad del código en una amplia gama de</p>

	<p>sistemas operativos, compiladores y control de versiones basados en etiquetas git para poder rastrear la versión de software.</p>
	<p>En el artículo [40] se estudió dos formas de depresión sináptica producida por la neuromodulación diferencial de los canales de calcio presináptico se utilizó un modelo computacional en python, para obtener una lista completa de los parámetros del modelo, se sometió a pruebas de Integración Continua usando pytest con Travis CI lo que permitió garantizar la ejecución en una amplia gama de combinaciones de parámetros válidos y no válidos.</p>
	<p>En el artículo [42] presenta modelos regidos por enzimas y análisis ómicos de <i>Streptomyces coelicolor</i> revelan cambios metabólicos que mejoran la producción heteróloga, tras la reconstrucción del modelo se realizó un seguimiento del desarrollo usando Git para el control de versiones para el desarrollo de modelos transparentes y colaborativos en la cual la herramienta se incorpora en el repositorio a través de Travis CI para el seguimiento del desarrollo en cada cambio del modelo, garantizando un control de calidad continuo.</p>
	<p>En el artículo [43] muestra un tutorial practico para que expertos y principiantes aprovechen el potencial de los datos fenómicos en programas de mejoramiento genético Alled, en donde crearon un cuaderno ejecutable para ocho algoritmos que subieron al repositorio de Github permitiendo instruir a los investigadores a través de un proceso de ejecutar código para reproducir resultados o ejecutar algoritmos implementados en nuevo conjunto de datos utilizando Integración Continua Travis CI para que las pruebas unitarias sean implementadas usando pytest y pueda ejecutar automáticamente cuando introduzca actualizaciones al código. Lo que permitió salvaguardar contra las actualizaciones que pueden dañar la funcionalidad al generar informes.</p>

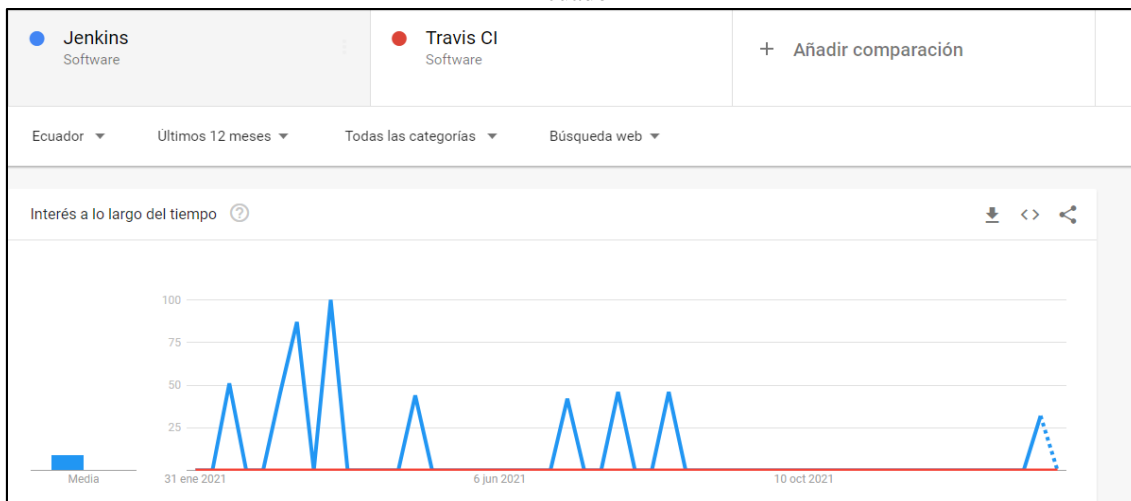
**Fuente:** Los investigadores

En la comparación bibliográfica realizada en los artículos [34], [36], [38], [40], [42] y [43] se determinó que Travis CI tenía licencia gratuita por ello era la herramienta más usada para procesos de Integración Continua, ya que permitía configuraciones menos complejas que la herramienta

Jenkins, por otro lado en los artículos [33], [35], [37], [39] y [41] se demuestra que actualmente Jenkins es la herramienta más usada para procesos de integración debido a que es Open Source.

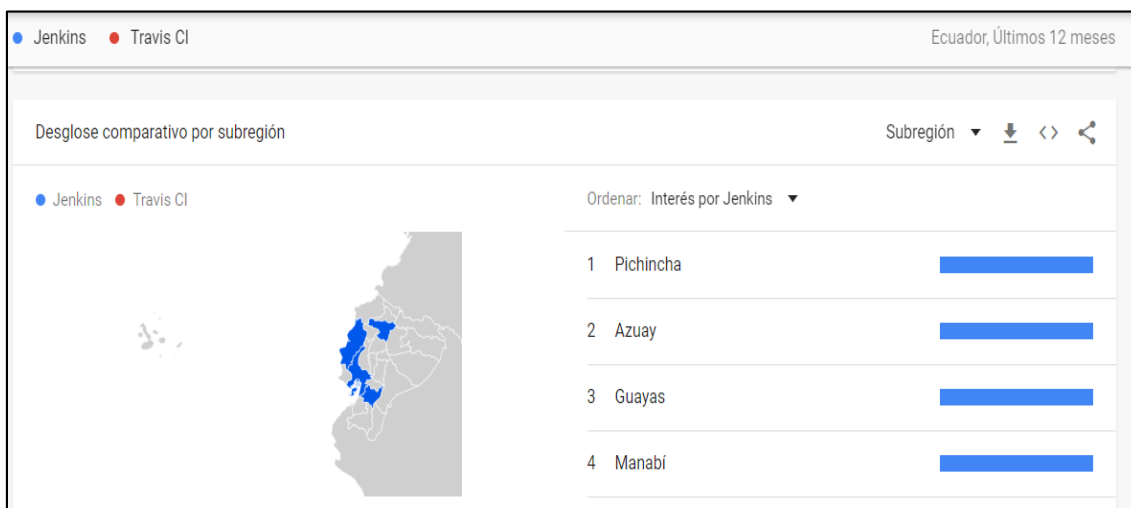
Además de las comparaciones realizadas se hace comparaciones (ilustración 7) que muestra la tendencia de información de las herramientas de Integración Continua: Jenkins y Travis CI en el Ecuador desde los últimos 12 meses hasta el día de hoy, según Google Trends, en la ilustración 8 muestra el desglose comparativo por subregión.

**Ilustración 10:** Comparación de tendencia de las herramientas de Integración Continua: Jenkins y Travis CI en el Ecuador



**Fuente:** Google Trends

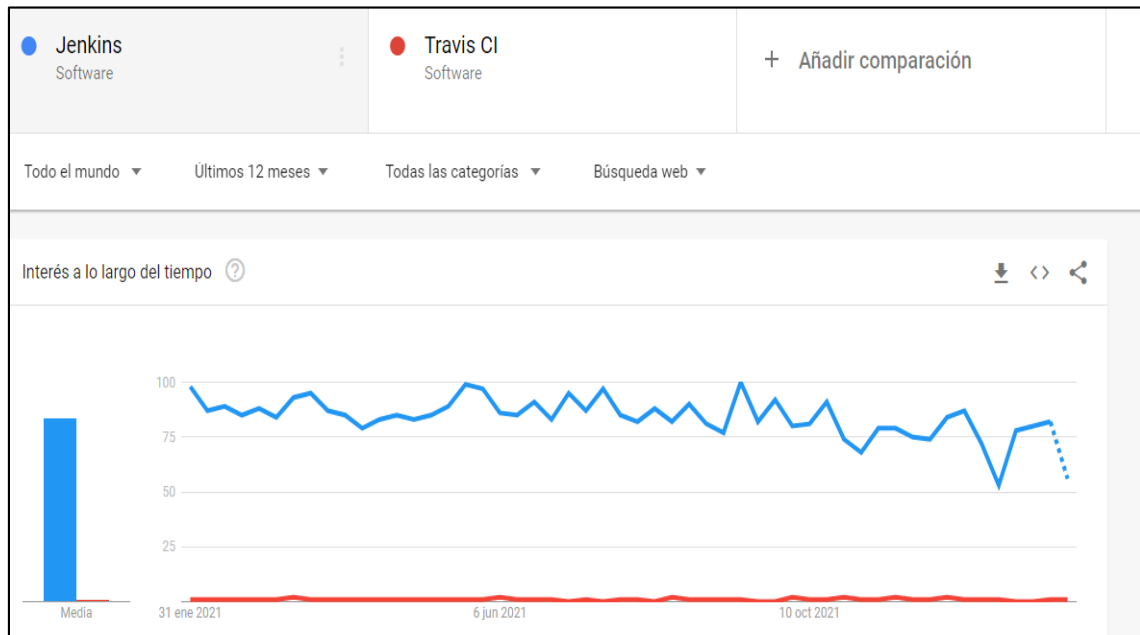
**Ilustración 11:** Desglose comparativo por subregión de las herramientas de Integración Continua: Jenkins y Travis CI



**Fuente:** Google Trends

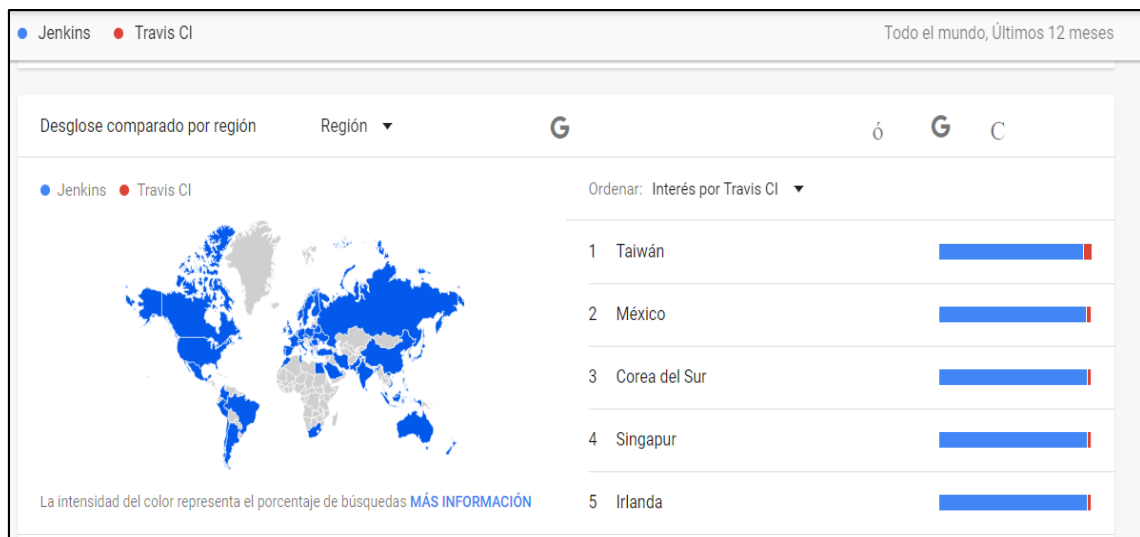
En la ilustración 9 muestra la tendencia de información de las herramientas de Integración Continua: Jenkins y Travis CI en todo el mundo desde los últimos 12 meses hasta el día de hoy, según Google Trends.

**Ilustración 12:** Comparación de tendencia de las herramientas de Integración Continua: Jenkins y Travis CI en el mundo



**Fuente:** Google Trends

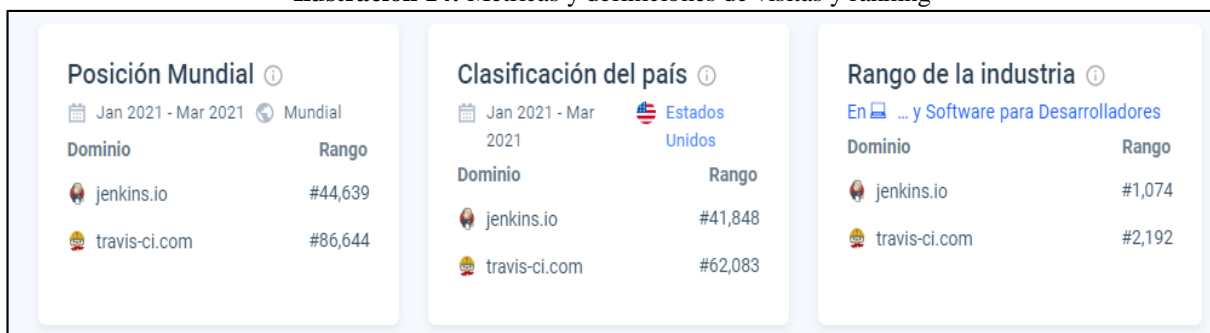
**Ilustración 13:** Desglose comparativo por región de las herramientas de Integración Continua: Jenkins y Travis CI



**Fuente:** Google Trends

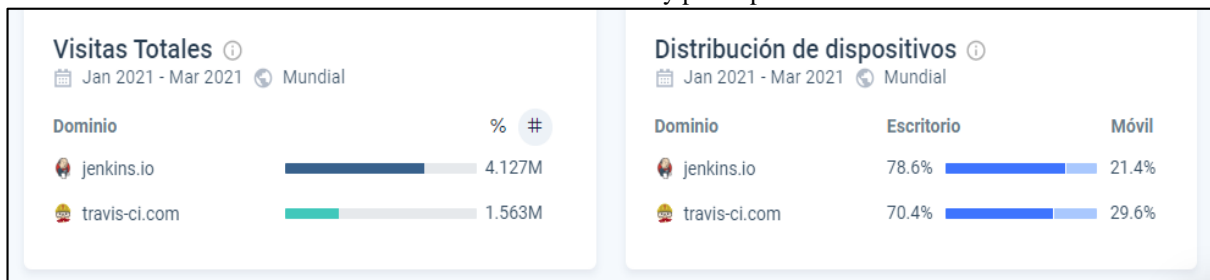
La página Rendimiento del sitio web proporciona una descripción general amplia de las estadísticas clave de tráfico para su sitio web, incluidos datos sobre el tráfico y la participación (Ilustración 11,12), las visitas a lo largo del tiempo (ilustración 13), el tráfico compartido por región geográfica (ilustración 14).

**Ilustración 14:** Métricas y definiciones de visitas y ranking



Fuente: SimilarWeb

**Ilustración 15:** Tráfico y participación



Fuente: SimilarWeb

**Ilustración 16:** Visitas a lo largo del tiempo



Fuente: SimilarWeb

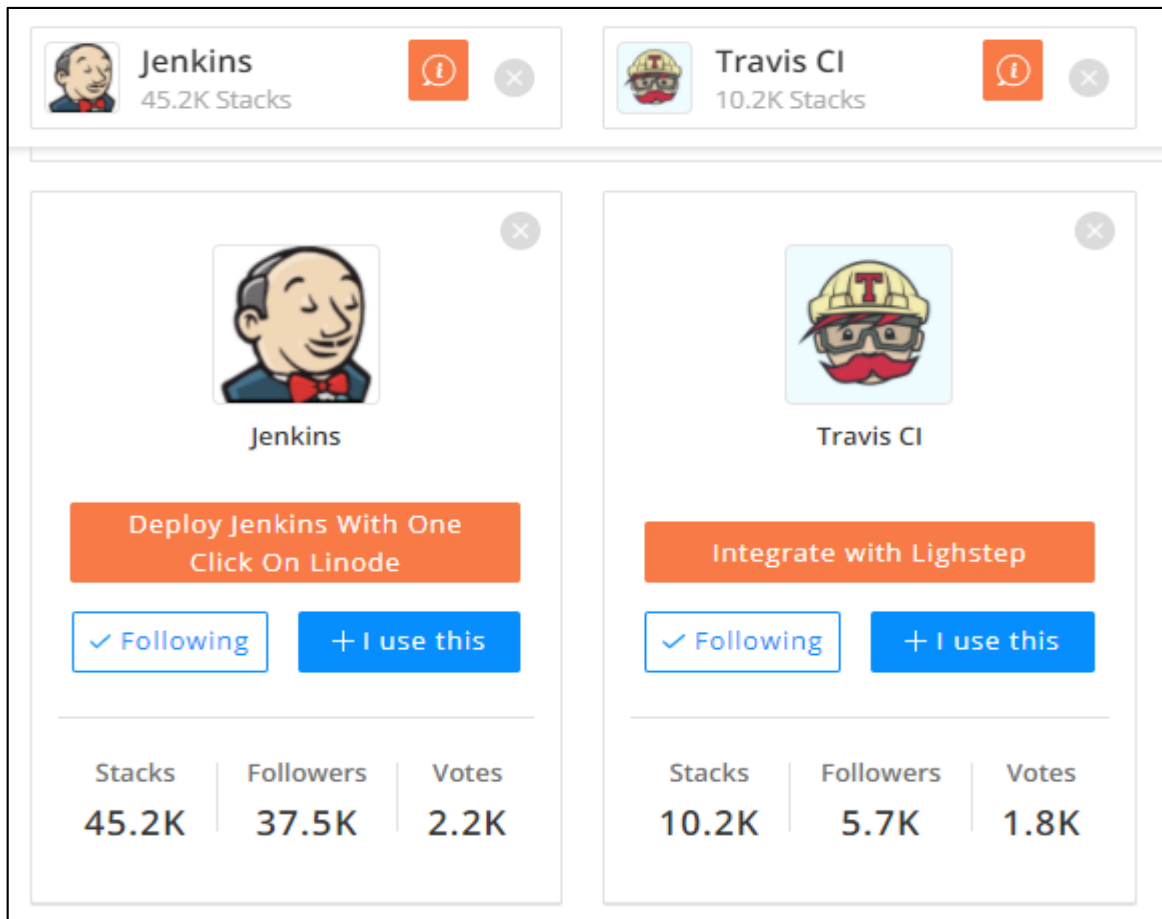
**Ilustración 17:** Tráfico compartido por región geográfica.



**Fuente:** SimilarWeb

SimilarWeb permite acceder a un análisis de cada herramienta en donde muestra una medida definitiva del mundo digital, lo cual permite conocer la lista ordenada de cada una de ellas.

**Ilustración 18:** Comparación de las herramientas de Integración Continua más usadas en el mundo.



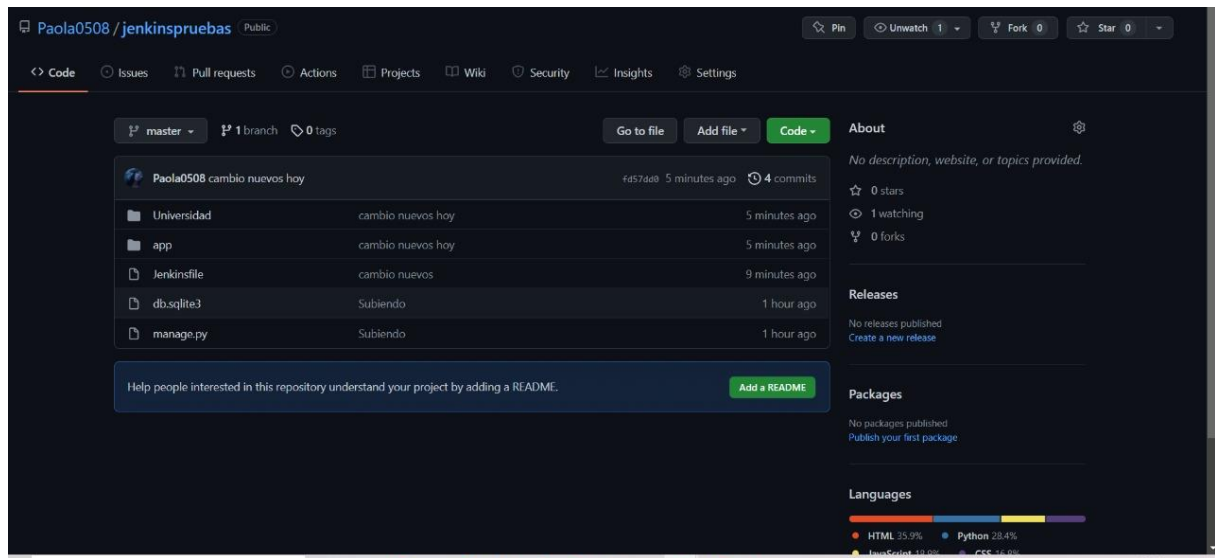
**Fuente:** SimilarWeb

## 5.2 Práctica con la herramienta Jenkins

A continuación, se presenta los resultados de la implementación CI en Jenkins.

Crear un proyecto y subir al repositorio de Github

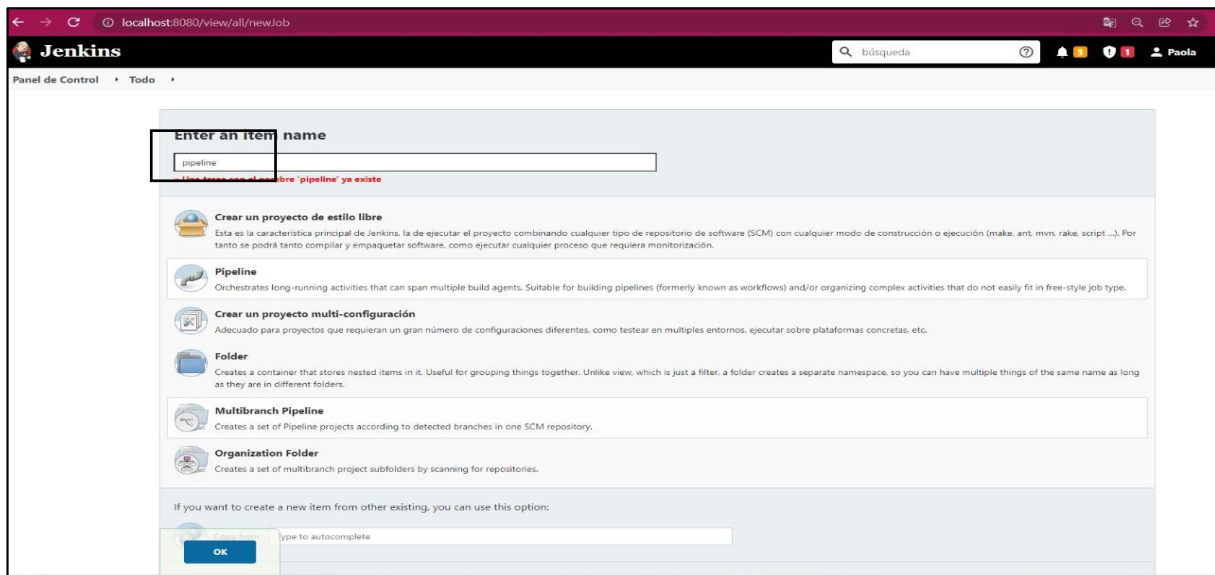
**Ilustración 19:** Proyecto en el repositorio



**Fuente:** Los investigadores

Crear una tarea en Jenkins para empezar con la Integración Continua.

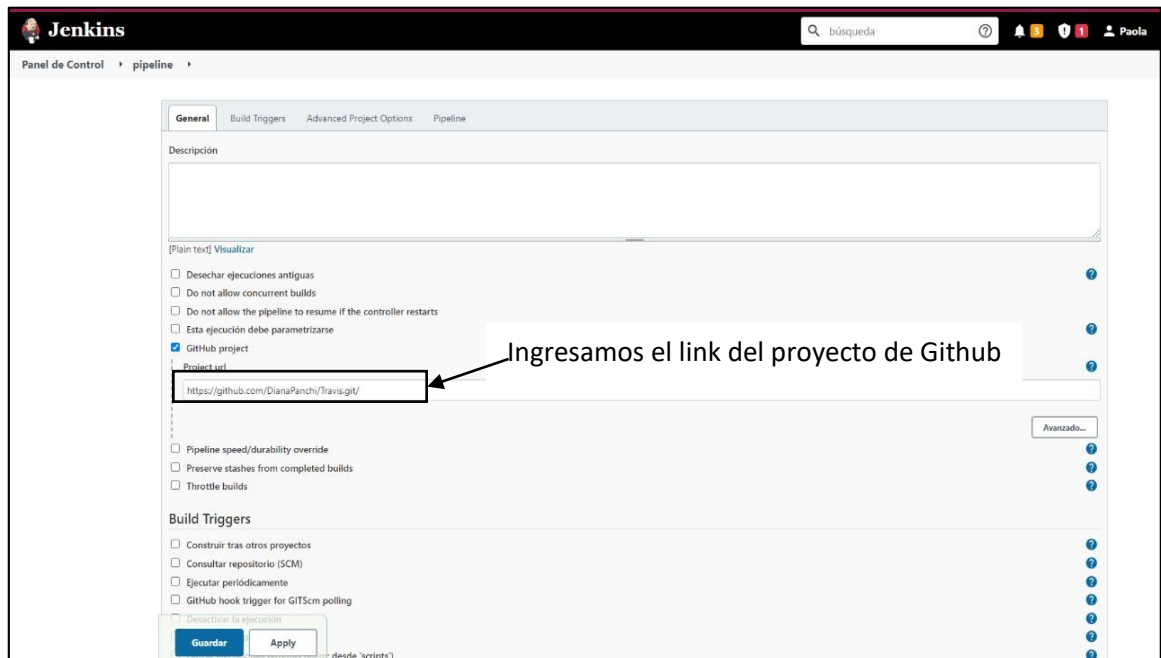
**Ilustración 20:** Crear tarea en Jenkins



**Fuente:** Los investigadores

En este apartado se realiza la configuración de Pipeline.

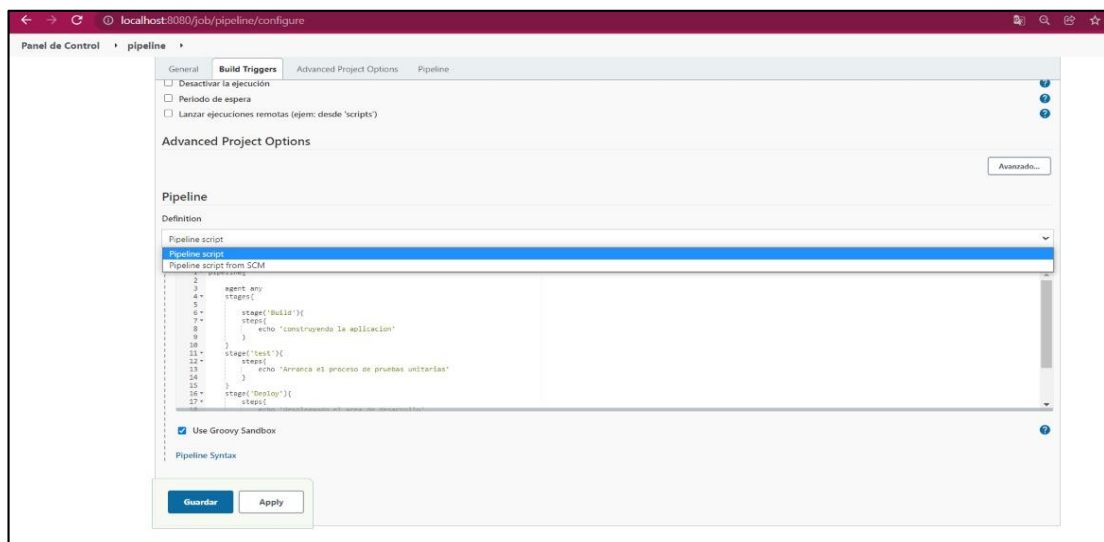
**Ilustración 21:** Configuración de Pipeline



**Fuente:** Los investigadores

Para realizar el script se lo puede realizar directamente en Jenkins o a su vez en la codificación del proyecto.

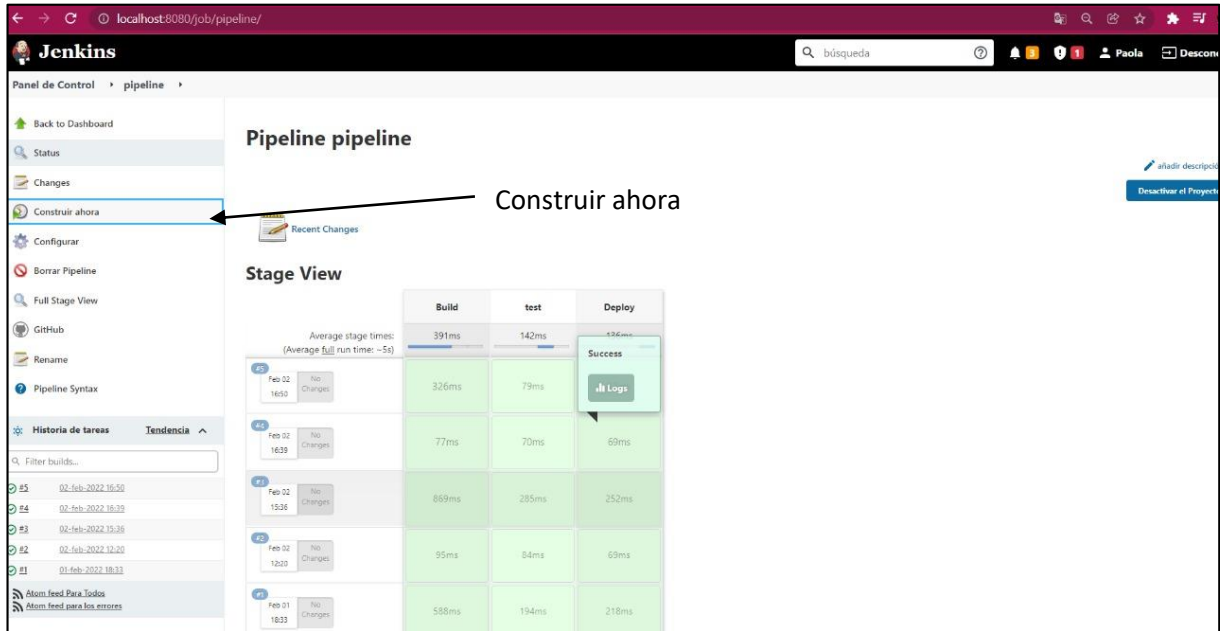
**Ilustración 22:** Codificación Pipeline Script



**Fuente:** Los investigadores

Ejecutamos el proyecto para ver los resultados (build, test y deploy)

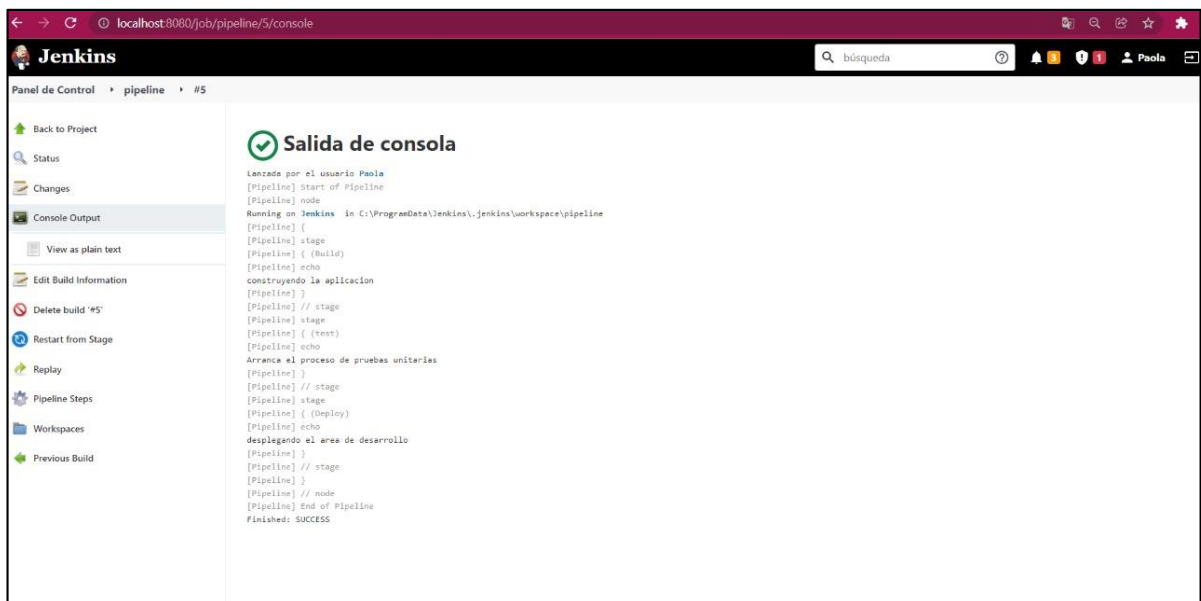
**Ilustración 23:** Resultados



**Fuente:** Los investigadores

En de Salida de consola se puede ver el proceso de la ejecución.

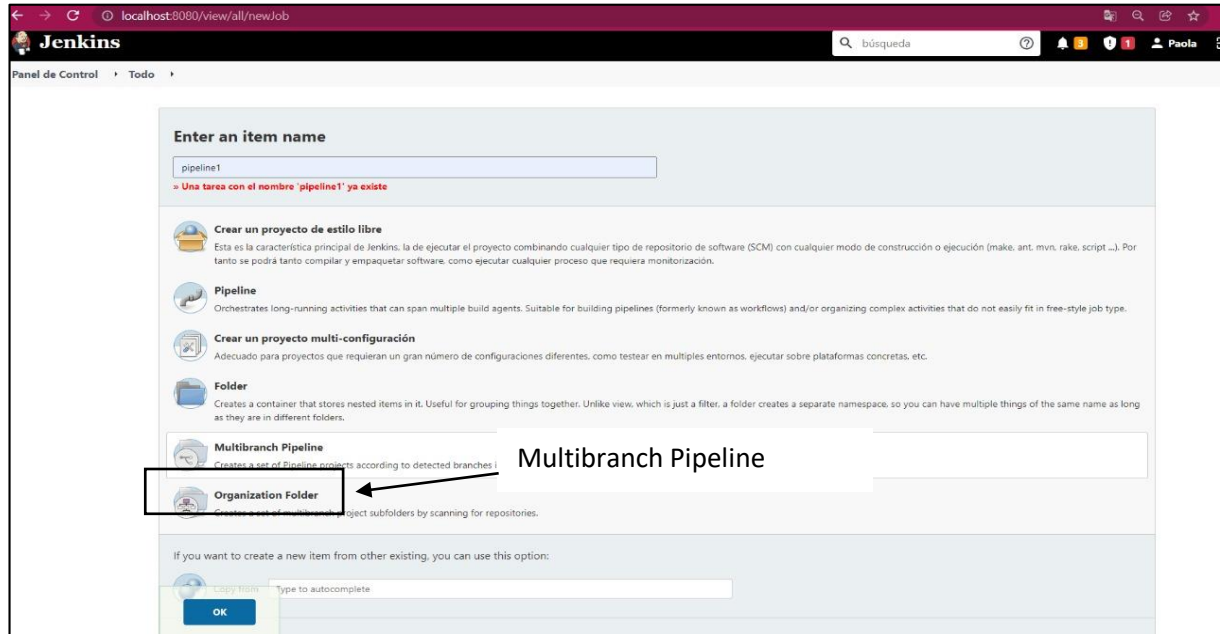
**Ilustración 24:** Resultados modo Consola



**Fuente:** Los investigadores

## Pipeline Automático (Branch)

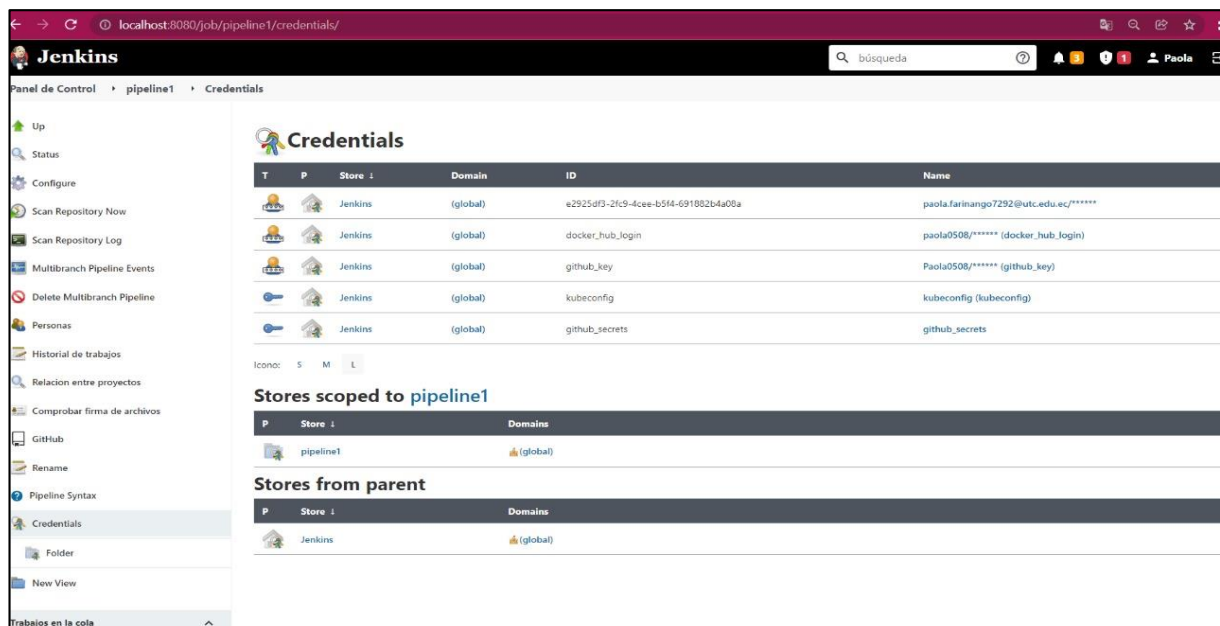
Ilustración 25: Multibranch Pipeline



Fuente: Los investigadores

## Creación de credenciales para la tarea en Multibranch Pipeline

Ilustración 26: Creación de credenciales

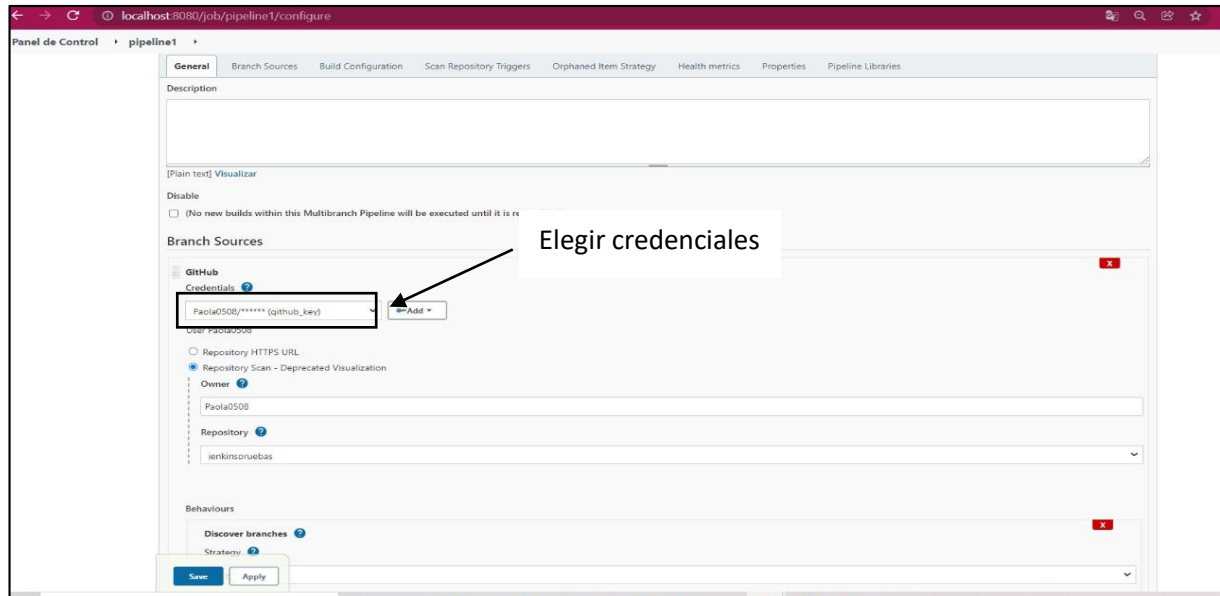


Fuente: Los investigadores

Para la configuración de Pipeline se ha realizado los siguientes pasos:

- Conexión al repositorio de Github

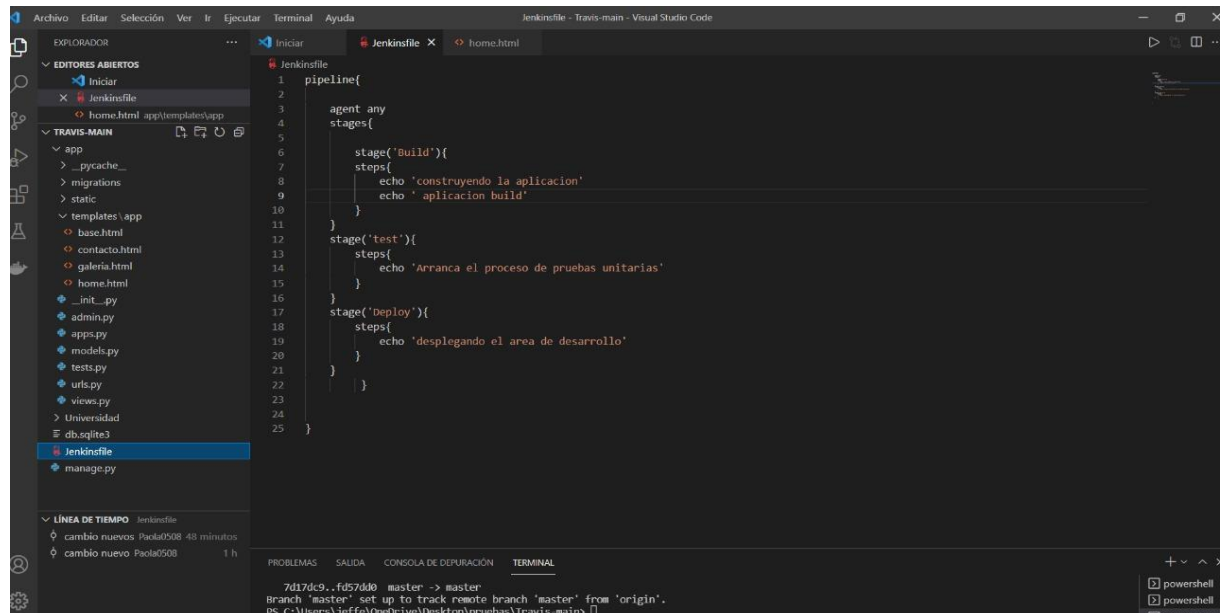
**Ilustración 27:** Conexión a Github



**Fuente:** Los investigadores

Script de Pipeline creado en el archivo Jenkinsfile

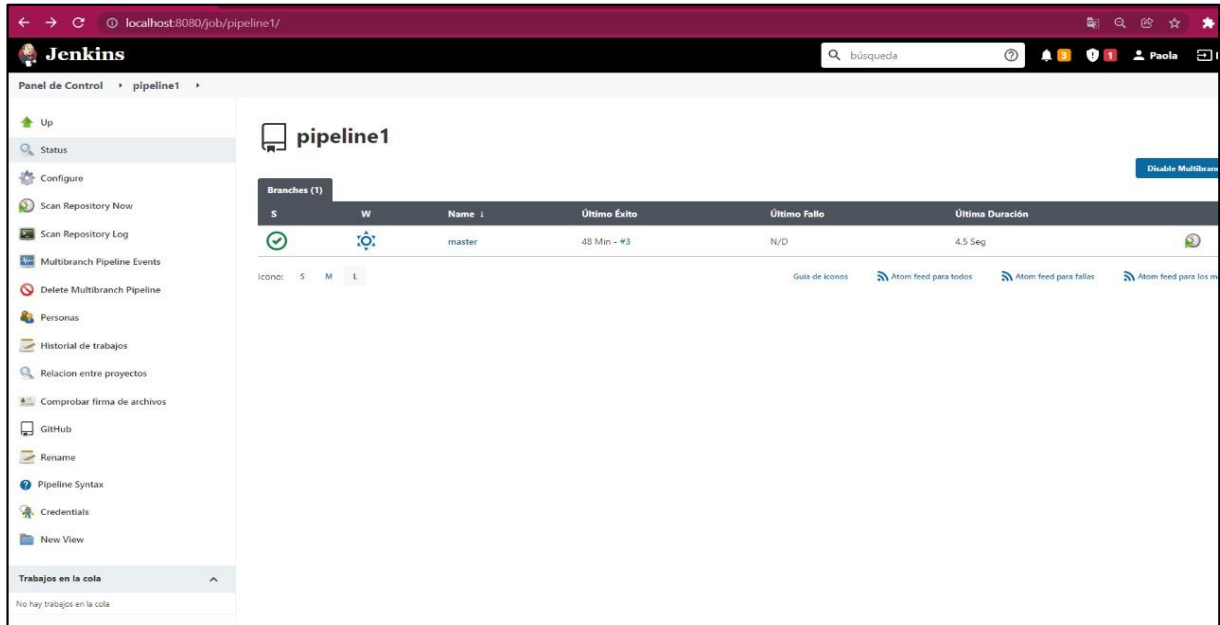
**Ilustración 28:** Crear archivo Jenkinsfile



**Fuente:** Los investigadores

Para la ejecución del proyecto seleccionamos (master).

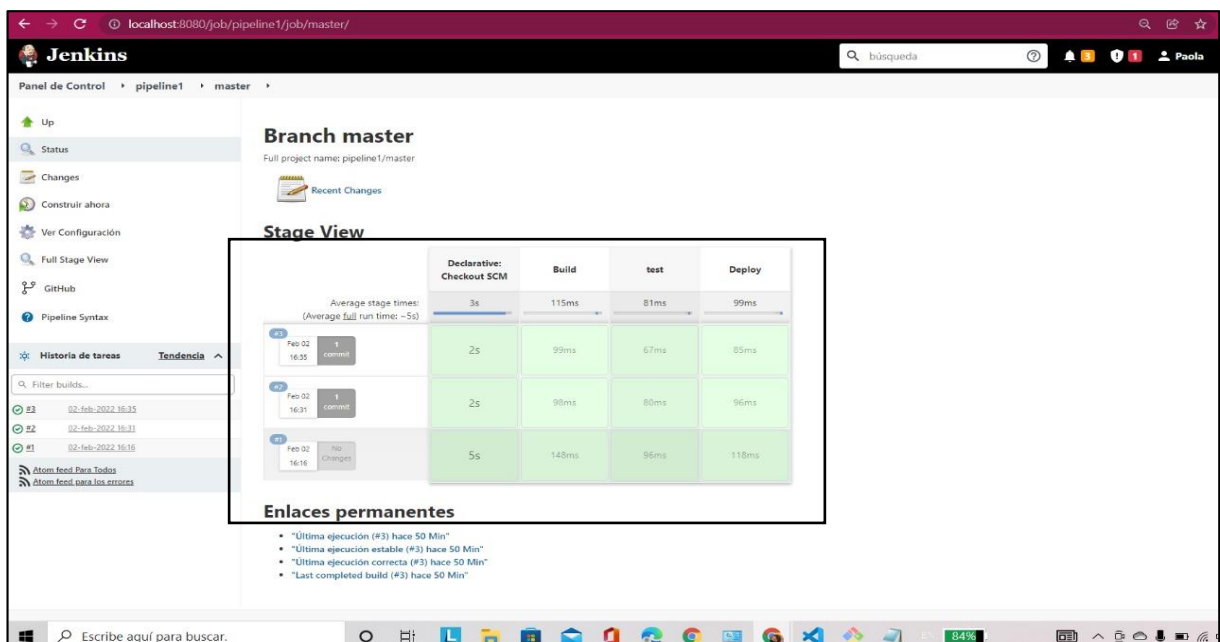
**Ilustración 29:** Selección para ejecución del proyecto



Fuente: Los investigadores

Construimos el proyecto y se muestra los resultados de Pipeline

**Ilustración 30:** Resultado de Pipeline



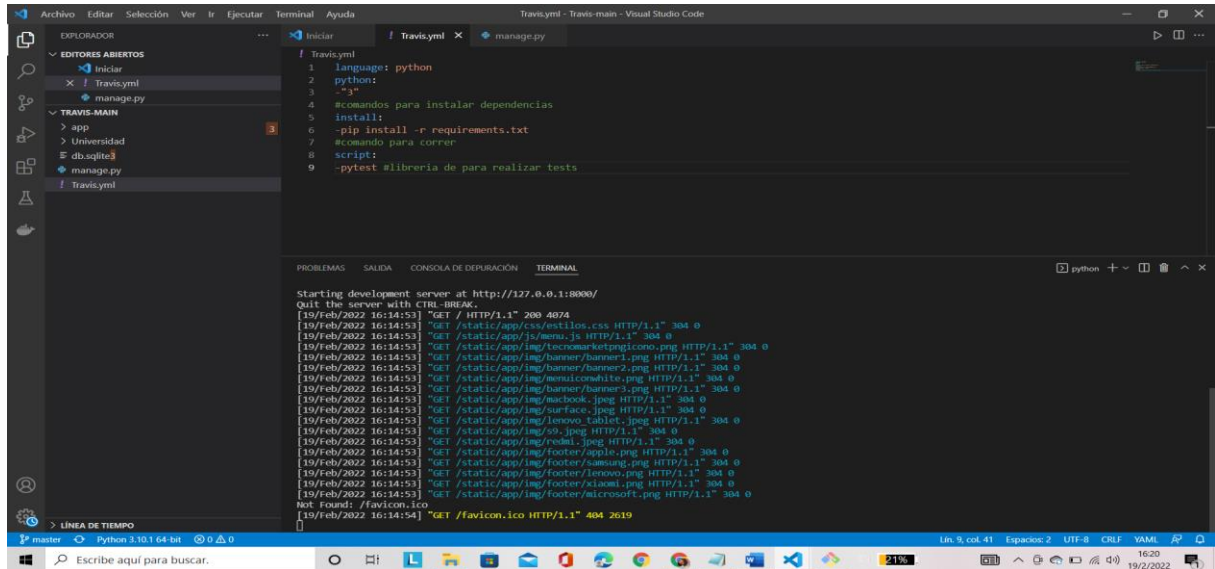
Fuente: Los investigadores

### 5.3 Práctica con la herramienta Travis CI

A continuación, se presenta los resultados de la implementación de la Integración Continua en Travis CI.

Se crea un archivo yml en el proyecto, se lo guarda y se lo sube al repositorio.

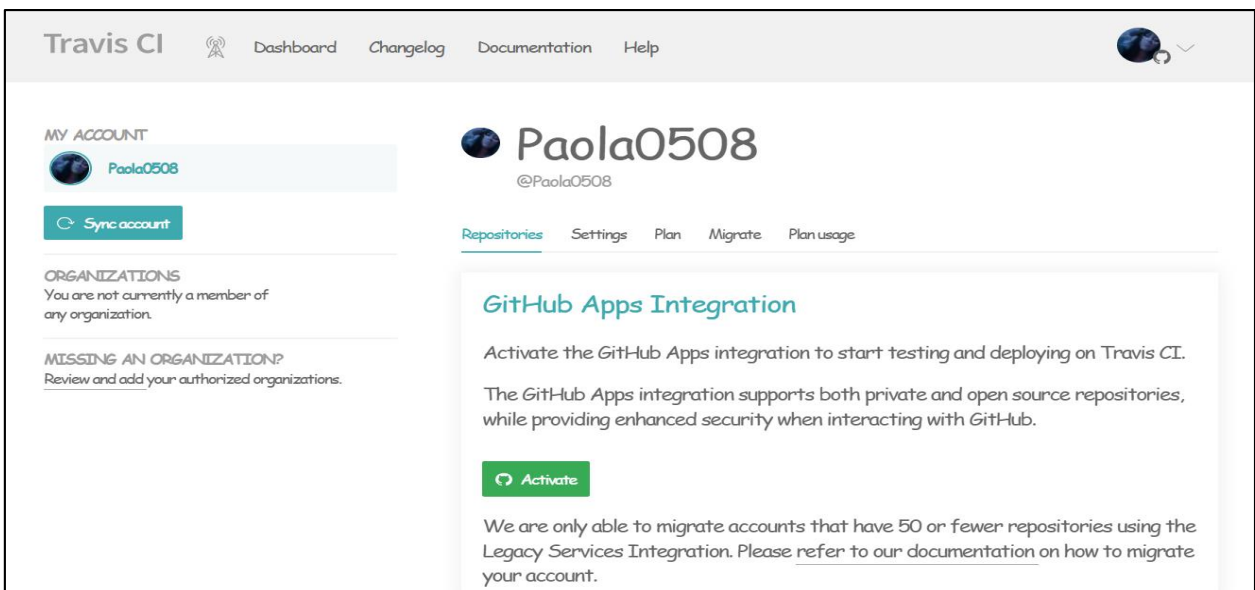
**Ilustración 31:** Crear archivo yml en el proyecto de Visual Studio Code



**Fuente:** Los investigadores

Se activa el repositorio que se va a conectar a Travis CI.

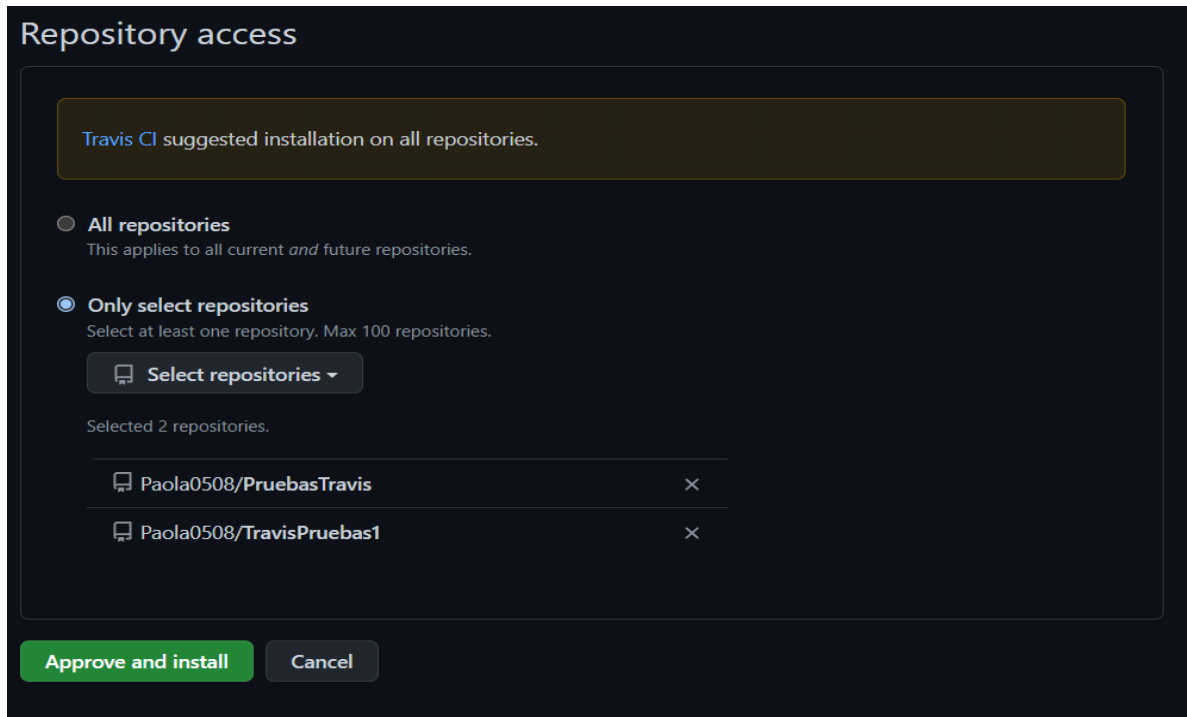
**Ilustración 32:** Activación del repositorio



**Fuente:** Los investigadores

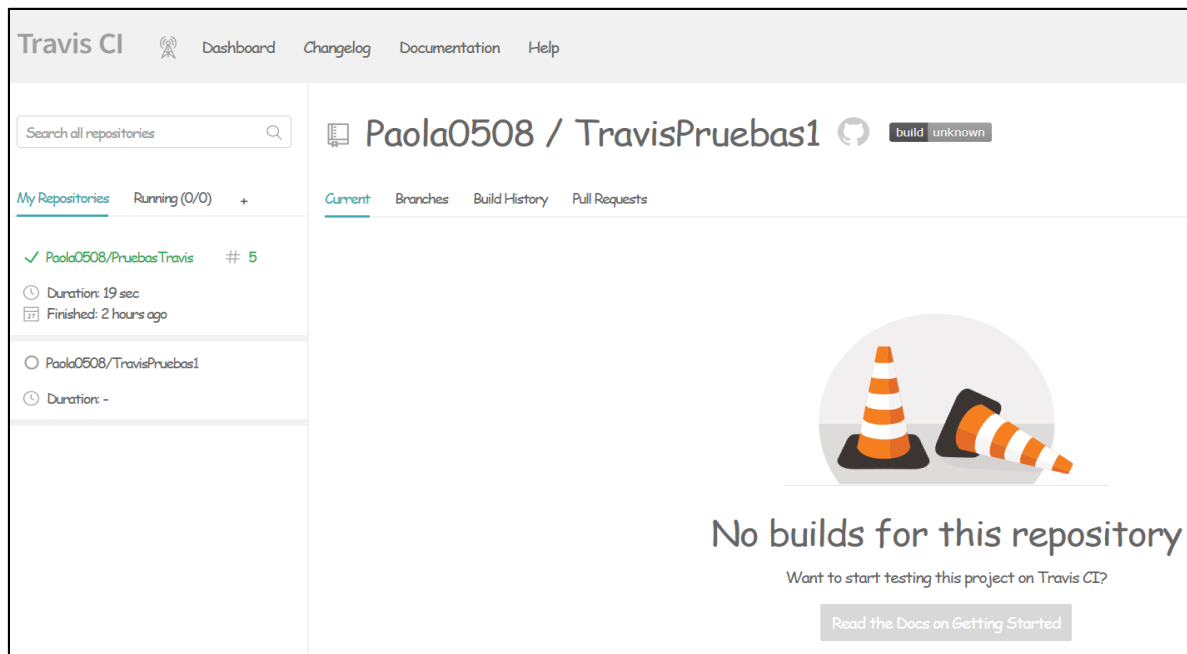
En este caso tiene dos opciones si quiere acceder a todos los repositorios o solo uno en específico.

**Ilustración 33:** Selección de repositorios



**Fuente:** Los investigadores

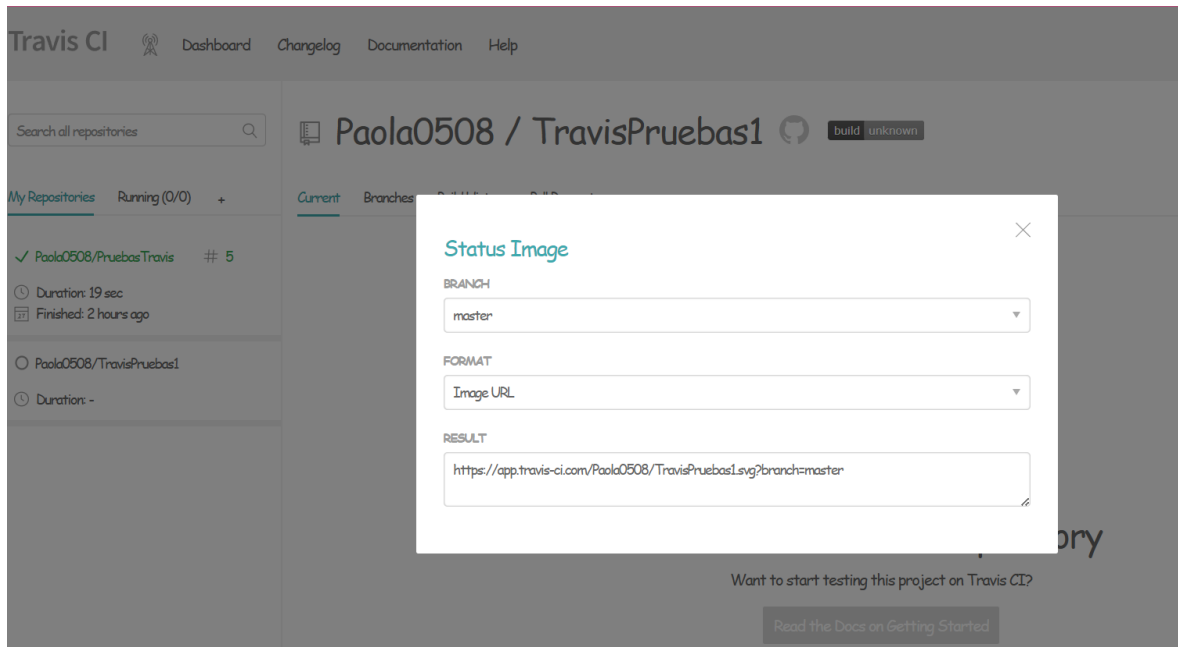
Una vez elegido los repositorios se puede observar en Travis CI los repositorios para realizar las pruebas.



**Fuente:** Los investigadores

No requiere de ninguna configuración en Travis CI debido a que esto se realiza automáticamente.

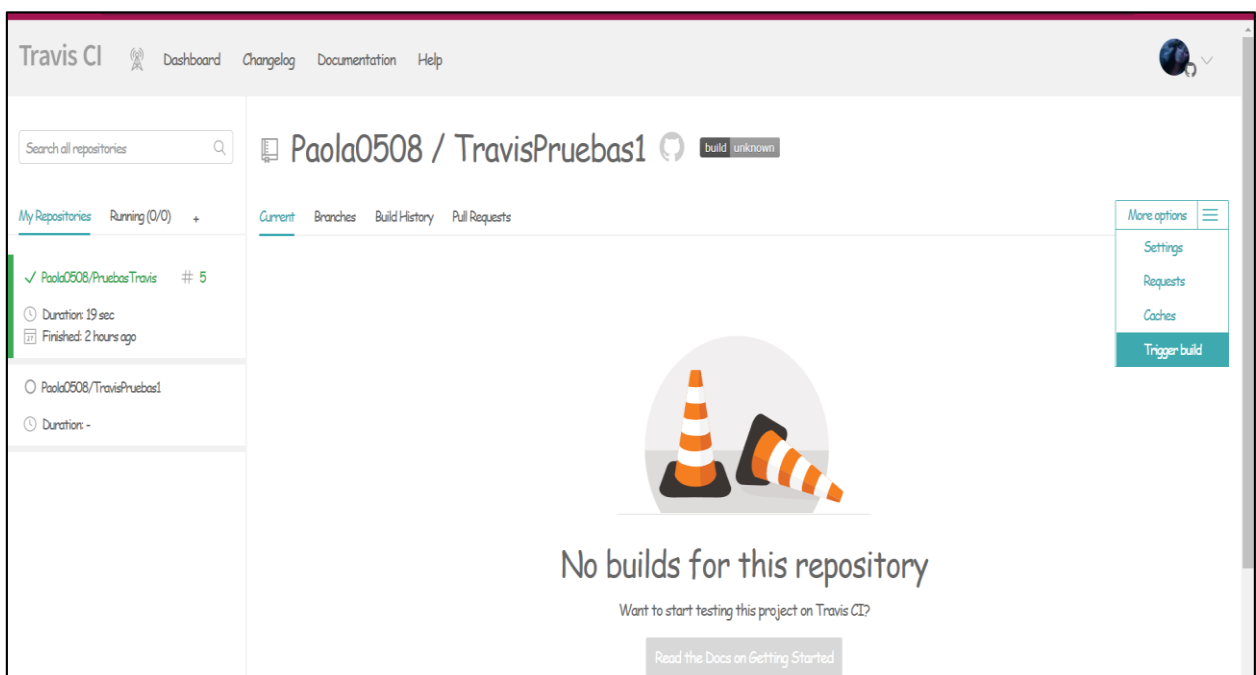
**Ilustración 34:** Configuración automática en Travis CI



**Fuente:** Los investigadores

Luego activamos el repositorio para que se realice la construcción (build).

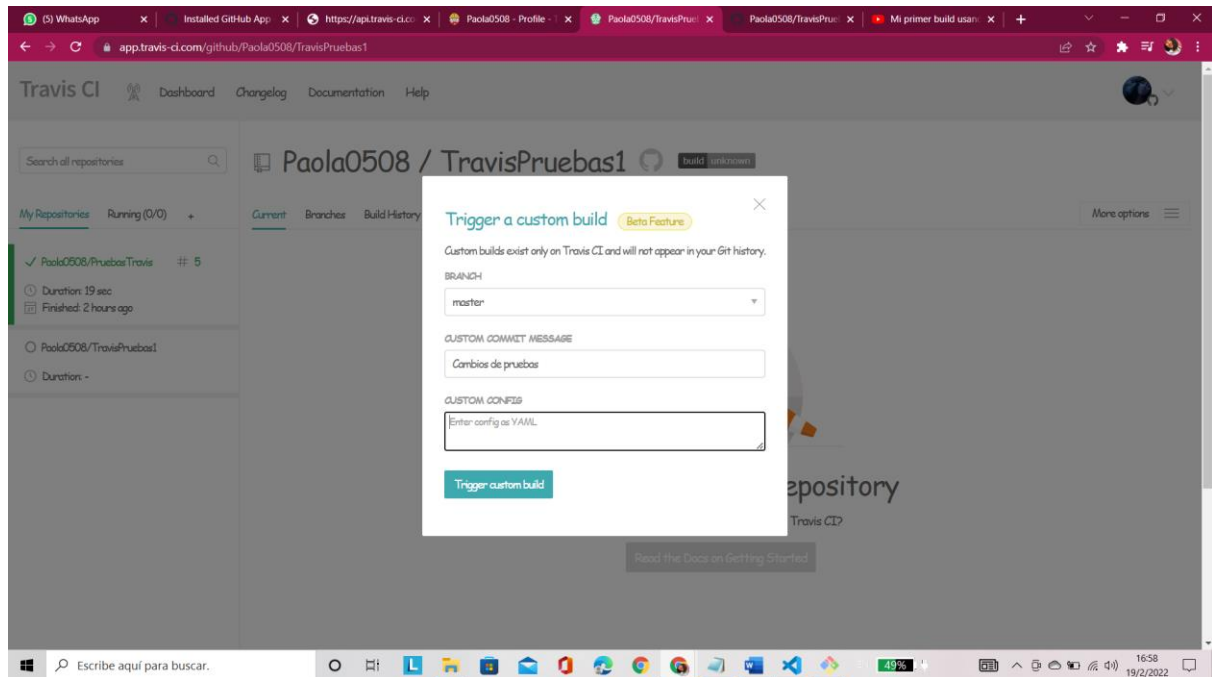
**Ilustración 35:** Activación del repositorio



**Fuente:** Los investigadores

Configuración para activar la compilación.

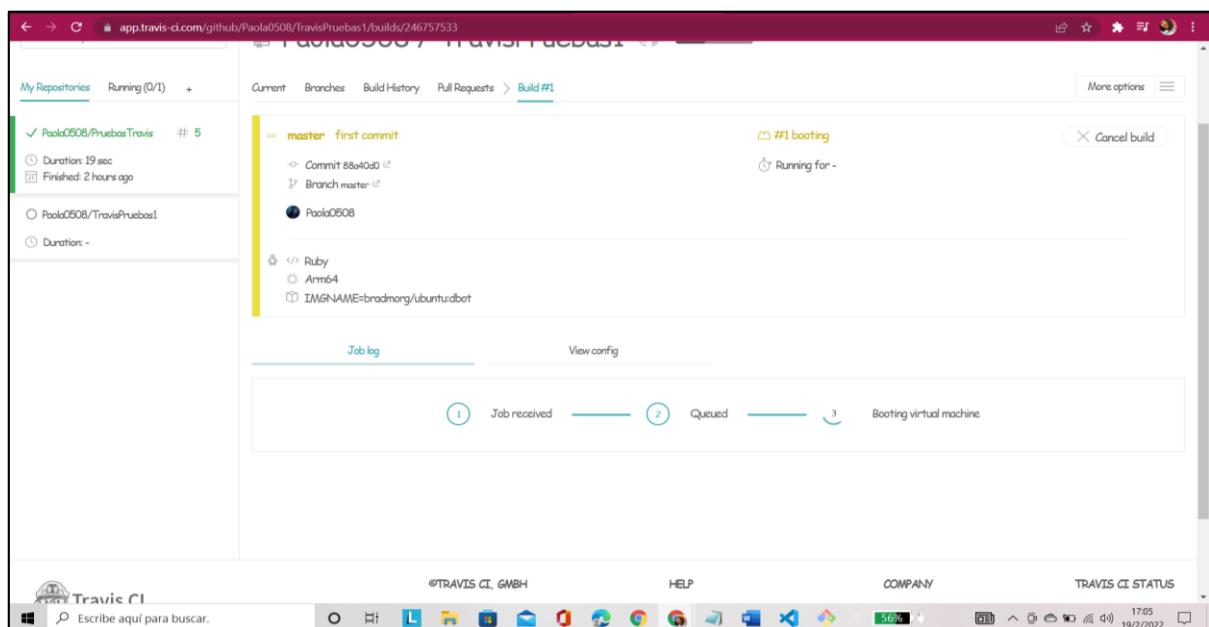
**Ilustración 36:** Configuración para activar la compilación



Fuente: Los investigadores

Se muestra el estado de la ejecución en Travis CI, en donde se ve que resultado se obtiene.

**Ilustración 37:** Resultado en Travis CI



Fuente: Los investigadores

### 5.3 Validación de expertos

A continuación se muestra un cuadro de indicadores el cual permite calificar cada uno de los ítems.

<b>CATEGORÍA</b>	<b>CALIFICACIÓN</b>	<b>Indicador</b>
Eficiencia	1.- Regular	La herramienta no es eficiente para el proceso de Integración Continua.
	2.- Bueno	La herramienta tiene poca eficiencia en el proceso de Integración Continua.
	3.- Muy bueno	La herramienta cumple con algunas funciones para el proceso de Integración Continua.
	4.- Sobresaliente	La herramienta cumple con todas funciones para el proceso de Integración Continua.
Características	1.- Regular	La herramienta no tiene las características necesarias para el proceso de Integración Continua.
	2.- Bueno	La herramienta cuenta con pocas características para el proceso de Integración Continua.
	3.- Muy bueno	La herramienta tiene características que ayudan el proceso de Integración Continua.
	4.- Sobresaliente	La herramienta cuenta con todas las características para realizar el proceso de Integración Continua.
Alternativa	1.- Regular	La herramienta no es una alternativa para el proceso de Integración Continua.
	2.- Bueno	La herramienta podría ser una alternativa para el proceso de Integración Continua.
	3.- Muy bueno	La herramienta es una alternativa para el proceso de Integración Continua.
	4.- Sobresaliente	La herramienta es una de las mejores alternativas para el proceso de Integración Continua.

## EXPERTO NÚMERO 1

Respetado juez: Usted ha sido seleccionado para evaluar las herramientas de Integración Continua que hace parte de la investigación Análisis Comparativo de herramientas de Integración Continua: caso de estudio Jenkins vs Travis CI. La evaluación de los instrumentos es de gran relevancia para lograr que sean válidos y que los resultados obtenidos a partir de estos sean utilizados eficientemente. Agradezco su valiosa colaboración.

<b>Nombres:</b>	Blanca Lucia
<b>Apellidos:</b>	Reinoso Chicaiza
<b>Cédula:</b>	1719470609
<b>Filiación</b> (ocupación, y lugar de trabajo):	Analista IT en empresa TATA Consultancy Service Área de business intelligence del Banco Pichincha. Manejo de Metodologías ágiles de programación. Programador en empresa Bypros S.A Desarrollo de aplicaciones informáticas Desarrollo de aplicaciones en PHP. Manejo de metodología XP.
<b>Grado académico.</b>	Universidad Técnica del Norte Facultad de Ingeniería en Ciencias Aplicadas (FICA). Carrera de Ingeniería en Sistemas Computacionales (CISIC).
<b>e-mail</b>	lucia1202utn@gmail.com
<b>Teléfono o celular</b>	09 59466575
<b>Fecha de la validación</b> (día, mes y año):	26-02-2022

ITEMS	EFICIENCIA	CARACTERÍSTICAS	ALTERNATIVA	OBSERVACIONES
¿Considera usted que la herramienta Jenkins es eficiente para realizar todos los procesos de Integración Continua?	4			
¿Considera usted que la herramienta Travis CI es eficiente para realizar todos los procesos de Integración Continua?	3			
¿Cree usted que características de Jenkins son más relevantes para el proceso de Integración Continua?		4		
¿Cree usted que características de Travis CI son más relevantes para el proceso de Integración Continua?		2		
¿Considera que Jenkins es una mejor alternativa para los procesos de Integración Continua?			4	
¿Considera que Travis CI es una mejor alternativa para los procesos de Integración Continua?			3	

El primer experto considera que la calificación de la herramienta Jenkins es sobresaliente debido a que la eficacia de la herramienta cumple con todas funciones para el proceso de Integración Continua, así como también cuenta con todas las características para realizar el proceso de Integración Continua y consideró que es una de las mejores alternativas para realizar la Integración Continua. En cuanto a la herramienta de Travis CI su calificación es muy buena en eficiencia ya que cuenta con algunas funciones para el proceso de Integración Continua y puede ser considerada como alternativa para dichos procesos, finalmente, en características su calificación es buena ya que cuenta con poca de ellas.

## EXPERTO NÚMERO 2

Respetado juez: Usted ha sido seleccionado para evaluar las herramientas de Integración Continua que hace parte de la investigación Análisis Comparativo de herramientas de Integración Continua: caso de estudio Jenkins vs Travis CI. La evaluación de los instrumentos es de gran relevancia para lograr que sean válidos y que los resultados obtenidos a partir de estos sean utilizados eficientemente. Agradezco su valiosa colaboración.

<b>Nombres:</b>	William Jair
<b>Apellidos:</b>	Dávila Vargas
<b>Cédula:</b>	0928363993
<b>Filiación</b> (ocupación, y lugar de trabajo):	Aplicaciones Móviles. Departamento de Ensamblaje y reparación de computadoras. Desarrollo-Programador Departamento de Proyectos Informáticos-Desarrollador Web Manejo de diferentes herramientas de programación. Experiencia en el desarrollo de sistemas informáticos tanto web y escritorio.
<b>Grado académico.</b>	Ingeniero en Sistemas Computacionales Milagro Superior: Universidad de Estatal de Milagro (UNEMI)
<b>e-mail</b>	williamjair94@hotmail.com
<b>Teléfono o celular</b>	0979014551-2977557
<b>Fecha de la validación</b> (día, mes y año):	27-02-2022

ITEMS	EFICIENCIA	CARACTERÍSTICAS	ALTERNATIVA	OBSERVACIONES
¿Considera usted que la herramienta Jenkins es eficiente para realizar todos los procesos de Integración Continua?	4			
¿Considera usted que la herramienta Travis CI es eficiente para realizar todos los procesos de Integración Continua?	4			
¿Cree usted que características de Jenkins son más relevantes para el proceso de Integración Continua?		4		
¿Cree usted que características de Travis CI son más relevantes para el proceso de Integración Continua?		3		
¿Considera que Jenkins es una mejor alternativa para los procesos de Integración Continua?			4	
¿Considera que Travis CI es una mejor alternativa para los procesos de Integración Continua?			2	

El

segundo experto considera que la calificación de la herramienta Jenkins es sobresaliente debido a que la eficacia de la herramienta cumple con todas funciones para el proceso de Integración Continua, así como también cuenta con todas las características para realizar el proceso de Integración Continua y consideró que es una de las mejores alternativas para realizar la Integración Continua. En cuanto a la herramienta de Travis CI su calificación es sobresaliente en eficiencia ya que cumple con todas funciones para el proceso de Integración Continua, en características la calificación es muy buena ya que si puede ayudar en el proceso, finalmente, el experto determina que podría ser una alternativa dependiendo del tipo de proyecto por lo que le da una calificación de bueno.

### EXPERTO NÚMERO 3

Respetado juez: Usted ha sido seleccionado para evaluar las herramientas de Integración Continua que hace parte de la investigación Análisis Comparativo de herramientas de Integración Continua: caso de estudio Jenkins vs Travis CI. La evaluación de los instrumentos es de gran relevancia para lograr que sean válidos y que los resultados obtenidos a partir de estos sean utilizados eficientemente. Agradezco su valiosa colaboración.

<b>Nombres:</b>	Omar Santiago
<b>Apellidos:</b>	Bedón Zambrano
<b>Cédula:</b>	0502630437
<b>Filiación</b> (ocupación, y lugar de trabajo):	Analista de información aplicada a las TICs. Programador de servicios y control de dispositivos. Analista de seguridad de información. Desarrollador web.
<b>Grado académico.</b>	Ingeniería Sistemas de información Universidad autónoma de los Andes UNIANDES
<b>e-mail</b>	omarbe2013@hotmail.com
<b>Teléfono o celular</b>	0969021016
<b>Fecha de la validación</b> (día, mes y año):	27-02-2022

ITEMS	EFICIENCIA	CARACTERÍSTICAS	ALTERNATIVA	OBSERVACIONES
¿Considera usted que la herramienta Jenkins es eficiente para realizar todos los procesos de Integración Continua?	3			
¿Considera usted que la herramienta Travis CI es eficiente para realizar todos los procesos de Integración Continua?	3			
¿Cree usted que características de Jenkins son más relevantes para el proceso de Integración Continua?		3		
¿Cree usted que características de Travis CI son más relevantes para el proceso de Integración Continua?		2		
¿Considera que Jenkins es una mejor alternativa para los procesos de Integración Continua?			4	
¿Considera que Travis CI es una mejor alternativa para los procesos de Integración Continua?			2	

El tercer experto considera que la calificación de la herramienta Jenkins es muy buena en cuanto a la eficiencia ya que cumple con algunas funciones para el proceso de Integración Continua, así como también cuenta características que podría ayudar a dicho proceso, sin embargo, el experto consideró que es una de las mejores alternativas para realizar la Integración Continua a pesar de la amplia configuración y licencia. En cuanto a la herramienta de Travis CI su calificación es muy buena en eficiencia ya también cuenta con algunas funciones para la Integración Continua, y en características su calificación es buena ya que considera que no es amplia para configurar es por ello que la calificación en alternativa es buena ya que podría ser considerada como una alternativa para el proceso de Integración Continua.

Expertos	Items					
	I1	I2	I3	I4	I5	I6
1	4	3	4	2	4	3
2	4	4	4	3	4	2
3	3	3	3	2	4	2

$$\alpha = \frac{K}{K-1} \left[ 1 - \frac{\sum Vi}{Vt} \right]$$

$$\alpha = \frac{6}{6-1} \left[ 1 - \frac{1,11}{2,88} \right]$$

$$\alpha = \frac{6}{5} [1 - 0,3854]$$

$$\alpha = 1,2 [0,6146]$$

$$\alpha = 0,738461$$

Para el criterio de validación de las herramientas Jenkins y Travis CI para el proceso de integración continua se determinó a través del Coeficiente de Alfa de Cronbach, el cual está conformado por 6 ítems, divididos en 3 ítems para Jenkins (I1, I3, I5) y 3 ítems para Travis CI (I2, I4, I6), evaluando la eficiencia, características y alternativa correspondientemente, cuyas escalas tienen como respuestas cuatro alternativas, para el cálculo de la validación se realizó con el apoyo del programa Excel, obteniendo como resultado un valor de 0,73, en donde dicho valor representa una calificación de Bueno, lo que indica una aceptación favorable en el uso de las herramientas ya que ayudan a los procesos de Integración Continua.

Por otro lado los resultados obtenidos tras aplicar el método de juicio de expertos se determinó que los tres expertos mencionan que para el proceso de Integración Continua la mejor alternativa es la herramienta Jenkins debido a su amplia configuración y sobre todo al acceso de la licencia gratuita.

## 5.6 PRESUPUESTO

En el presupuesto se detallan cada uno de los costos directos y costos indirectos que intervinieron para el desarrollo del proyecto de la investigación.

### 5.6.1 Costos directos

En los costos directos se encuentran los valores económicos se emplearon para realizar la práctica de las herramientas. A continuación, se muestra una tabla dónde se detalla cada una de los materiales con sus precios.

**Tabla 3:** Costos directos

Detalle	Cantidad	Valor unitario	Valor total
Curso de Jenkins	1	18,00	18,00
Servidor Travis CI	1	10,00	10,00
Desarrollador Master	1		300
Desarrollador	1		150
<b>Total</b>			478,00

### 5.6.2 Costos indirectos

Estos costos están ligados indirectamente a los gastos del desarrollo del proyecto de investigación, como comida, transporte entre otros.

**Tabla 4:** Costos indirectos

Detalle	Cantidad	Valor unitario	Valor total
Transporte	General		114,00
Comida	General		60,00
Internet			10,00
Impresiones	100	0,8	8,00
Anillados	3	3,00	9,00
Copias		0,3	3,00
<b>Total</b>			204,00

### 5.6.3 Costos totales

Se presentan los costos totales del proyecto de investigación.

**Tabla 5:** Costos totales

Detalle	Valor total
Costos directos	478,00
Costos indirectos	204,00
<b>Total</b>	682,00

## **6 CONCLUSIONES Y RECOMENDACIONES**

### **6.1 CONCLUSIONES**

- La revisión bibliográfica permitió crear el marco teórico del presente proyecto de investigación, lo cual ayudó a comprender las diferentes definiciones sobre la Integración Continua en el desarrollo de software, además permitió conocer los beneficios que tiene esta práctica para la automatización de los procesos de pruebas de proyectos de software.
- La aplicación de las herramientas de Integración Continua, Jenkins y Travis CI, permitió determinar las características técnicas de cada herramienta a través de las cuales, se pudo concluir que la mejor herramienta es Jenkins, ya que se adapta mejor a las necesidades del desarrollador y sobre todo porque es Open Source.
- Por medio de la técnica de juicios de expertos aplicando el Coeficiente Alfa de Cronbach se pudo determinar que hay una aceptación favorable del análisis de las dos herramientas de Integración Continua, Jenkins y Travis CI, con un valor de 0,73, lo cual demuestra que la investigación tiene una valoración cualitativa de Bueno.

### **6.2 RECOMENDACIONES**

- El tema de Integración Continua al no ser tan conocido en este medio, se recomienda realizar nuevas investigaciones relacionadas para promover el uso de estas nuevas prácticas de desarrollo de software.
- Se debería realizar un siguiente estudio para complementar el que se presenta que se enfoque en la práctica de Despliegue Continuo para el cual se podría utilizar las mismas herramientas.
- Para calcular la validez del proyecto de investigación se puede aplicar otras técnicas para conocer la fiabilidad que tiene la investigación.

## 7 BIBLIOGRAFÍA

- [1] M. Rojas Rodríguez, Y. Moreira Gamboa y S. Torres López, «Gestión de Proyectos,» *Serie Científica de la Universidad de las Ciencias Informáticas*, vol. 12, n° 10, pp. 102-116, 2019.
- [2] «repositorio.espe,» 08 09 2020. [En línea]. Available: <http://repositorio.espe.edu.ec/bitstream/21000/24954/1/M-ESPEL-sit-0099.pdf>. [Último acceso: 06 12 2021].
- [3] «Desarrollo de Software,» Aurens Global SA, 2013.
- [4] F. Joly, «La cartografía,» Oikos-Tau, Barcelona, 1998.
- [5] S. Ríos, «Modelización,» AU 822. Alianza Ediciones, Madrid, 1995.
- [6] S. Ian, Ingeniería del Software, 7ma Ed, Madrid: Pearson Educación.S.A, 2005.
- [7] D. A. Godoy Álvarez y R. S. Taype Calderón, «Modelos de aceptación de metodologías de desarrollo de software,» Lima-Perú, 2015.
- [8] A. Alshamrani y A. Bahattab, «A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model,» *International Journal of Computer Science Issues (IJCSI)*, vol. 12, p. 106, 2015.
- [9] E. G. Maida y J. Pacienza, «Metodologías de desarrollo de software,» Argentina, 2015.
- [10] J. L. Vila Grau, «Proagilist,» 11 Noviembre 2015. [En línea]. Available: <https://proagilist.es/blog/agilidad-y-gestion-agil/agile-scrum/la-diferencia-entre-metodos-y-practicas-agiles/>. [Último acceso: 12 Diciembre 2021].
- [11] «QuestionPro,» [En línea]. Available: <https://www.questionpro.com/blog/es/metodos-de-investigacion/>. [Último acceso: 20 Febrero 2020].
- [12] O. Castellero Mimenza, «Psicología y Mente,» 17 Octubre 2017. [En línea]. Available: <https://psicologiaymente.com/miscelanea/tipos-de-investigacion>. [Último acceso: 25 Febrero 2022].
- [13] G. Fernández Guerrero, «Metodología de la Investigación Licenciatura en Diseño Gráfico CONOCERSE ACEPTARSE AMARSE CUIDARSE SUPERARSE TRA...,» 2000.
- [14] J. Creswell, Educational research: Planning, conducting, and evaluating quantitative and qualitative research. Upper Saddle River: Pearson Education., Saddle River, NJ: Prentice Hall , 2005.
- [15] M. Fowler, «Continuous Integration,» 01 Mayo 2006. [En línea]. Available: <https://www.martinfowler.com/articles/continuousIntegration.html>.

- [16] R. L. Avila, «Implementación de una plataforma de Integración Continua y Entrega Continua en un marco de trabajo ágil,» Lima, Perú, 2017.
- [17] D. Villahermosa, «Desde el bastión del software,» 27 Noviembre 2019. [En línea]. Available: <https://danielvillahermosa.wordpress.com/2019/11/27/la-evolucion-de-devops/>. [Último acceso: 12 Diciembre 2021].
- [18] L. Bass, I. Weber y L. Zhu, DevOps: A Software Architect's Perspective, Estados Unidos: Pearson Education, Inc, 2015.
- [19] A. Earnshaw, «DEVOPS LATAM,» 17 Julio 2019. [En línea]. Available: <https://devopslatam.com/principios-devops-gerentes-ti/>. [Último acceso: 1 Febrero 2022].
- [20] «Tecnologías Información,» [En línea]. Available: <https://www.tecnologias-informacion.com/devops.html>. [Último acceso: 2022 Enero 31].
- [21] T. Hall, Atlassian, [En línea]. Available: <https://www.atlassian.com/devops/devops-tools/devops-pipeline>. [Último acceso: 30 Enero 2022].
- [22] A. M. A, C. A. D y L. P, « Jenkins Integración Continua.,» Universidad Nacional de Quilmes, 2013.
- [23] H. Bustamante Cruz, «Herramientas Open Source para la Integración Continua, en Proyectos de Desarrollo de Software,» Cochabamba, Bolivia, 2018.
- [24] R. Jain, «Jenkins contra Travis contra Bamboo contra TeamCity: Choque de titanes,» DZone, [En línea]. Available: <https://dzone.com/articles/jenkins-vs-travis-vs-bamboo-vs-teamcity-clash-of-t?fromrel=true>.
- [25] «Stackshare,» 29 Abril 2020. [En línea]. Available: <https://stackshare.io/stackups/circleci-vs-jenkins-vs-travis-ci#stats>. [Último acceso: 1 Diciembre 2021].
- [26] «Jet Brains,» [En línea]. Available: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/herramientas-de-integracion-continua/>. [Último acceso: 30 Noviembre 2021].
- [27] P. M. Duvall, S. Matyas y A. Glover, Continuous Integration: Improving Software Quality and Reducing Risk, Person Education, Inc, 2007.
- [28] R. López Fernández, J. Lalangui Ramírez, D. E. Palmero Urquiza y J. Lalangui Ramírez, «VALIDACIÓN DE UN INSTRUMENTO. SOBRE LOS DESTINOS TURÍSTICOS PARA DETERMINAR LAS POTENCIALIDADES TURÍSTICAS EN LA PROVINCIA DE EL ORO, ECUADOR,» *Revista Científica de la Universidad de Cienfuegos*, vol. 11, n° 2, 2019.
- [29] J. Cabero Almenara y M. d. C. Llorente Cejudo, «La aplicación del juicio de experto como técnica de evaluación de las tecnologías de la información y comunicación

(TIC),» *Revista de Tecnología de Información y Comunicación en Educación*, vol. 7, nº 2, 2013.

- [30] J. Rodríguez-Rodríguez y M. Reguant-Álvarez, «Calcular la fiabilidad de un cuestionario o escala mediante el SPSS: el coeficiente alfa de Cronbach,» *REIRE Revista d'Innovació i Recerca en Educació*, pp. 1-13, 2020.
- [31] L. Cronbach, «Coeficiente alfa y la estructura interna de las pruebas,» *Psychometrik*, vol. 16, p. 297–334, 1951.
- [32] [En línea]. Available: <https://stackshare.io/feed>.
- [33] J. A. Smith, G. Richman, J. DeStefano, J. Pryor, T. Rao, W. StreckerKellogg y T. Wong, «The Careful Puppet Master: Reducing risk and fortifying acceptance testing with Jenkins CI,» *IOPscience*, pp. 1-9, 2015.
- [34] R. I F, «Implementation of microservices architecture on certification information system (case study: LSP P1 State Polytechnic of Malang),» *IOPscience*, pp. 1-7, 2020.
- [35] J. Elmsheuser, A. Krasznahorkay, E. Obreshkov y A. Undrus, «A Roadmap to Continuous Integration for ATLAS Software Development,» *IOPscience*, pp. 1-6, 2017.
- [36] P. Gleeson, M. Cantarelli, B. Marin, M. Earnshaw y R. A. Silver, «Open Source Brain: A Collaborative Resource for Visualizing, Analyzing, Simulating, and Developing Standardized Models of Neurons and Circuits,» *Cell Press*, vol. 103, nº 3, pp. 1-23, 2019.
- [37] I. Moutsatsos, I. Hossain, C. Agarinis, H. Fred y A. Yann, «Jenkins-CI, an Open-Source Continuous Integration System, as a Scientific Data and Image-Processing Platform,» *SLAS Discovery*, vol. 22(3), nº 10, pp. 238-249, 2018.
- [38] C. T. Lee, J. G. Laughlin, J. B. Moody, R. E. Amaro, J. A. McCammon, M. Holst y P. Rangamani, «An Open-Source Mesh Generation Platform for Biophysical Modeling Using Realistic Cellular Geometries,» *Cell Press*, pp. 1-6, 2019.
- [39] A. Holt, T. Hauser y Z. Sampedro, «Continuous Integration and Delivery for HPC,» *PEARC*, pp. 26-28, 2018.
- [40] K. J. Burke, C. M. Keeshen y K. J. Bender, «Two Forms of Synaptic Depression Produced by Differential Neuromodulation of Presynaptic Calcium Channels,» *Cell Press*, pp. 1-24, 2018.
- [41] S. Darío, «Automatización del proceso de desarrollo de software mediante una herramienta de Integración Continua (Jenkins) para el control de versionamiento en la Dirección de Tecnologías de la Información y Comunicación del Ejército.,» *Repositorio*, vol. I, p. 77, 2020.
- [42] S. Sulheim, T. Kumelj, D. van Dissel, A. Salehzadeh-Yazdi, C. Du, G. P. van Wezel, K. Nieselt, E. Almaas, A. Wentzel y E. J. Kerkhoven, «Enzyme-Constrained Models and

Omics Analysis of *Streptomyces coelicolor* Reveal Metabolic Changes that Enhance Heterologous Production,» *Cell Press*, pp. 1-67, 2020.

- [43] F. Nakhle y A. L. Harfouche, «Ready, Steady, Go AI: A practical tutorial on fundamentals of artificial intelligence and its applications in phenomics image analysis,» *Cell Press*, pp. 1-21, 2021.
- [44] I. Alfonzo, *Técnicas de Investigación bibliográfica*, Caracas: Contexto Editores, 1994.
- [45] A. Alonso Serrano , L. García Sanz , I. León Rodrigo, E. García Gordo , B. Gil Álvaro y L. Ríos Brea, «MÉTODOS DE INVESTIGACIÓN DE ENFOQUE EXPERIMENTAL,» [En línea]. Available: <https://www.postgradoune.edu.pe/pdf/documentos-academicos/ciencias-de-la-educacion/10.pdf>. [Último acceso: 7 Diciembre 2021].
- [46] C. Martínez, «Metodologías de investigación educativa (descriptivas, experimentales, participativas, y de investigación-acción),» *reciMundo*, pp. 163-173, 24 Enero 2018.
- [47] J. J. Plaza Guzmán y P. A. Uriguen Aguirre, «Metodología de la Investigación,» *Revista ARJÉ*, vol. 11, nº 21, pp. 3-5, 2017.
- [48] S. McConnell, *Code Complete, Second Edition (2 ed.)*, Redmond, Washington : Microsoft, 2004.
- [49] «Apiumhub,» 9 Enero 2018. [En línea]. Available: <https://apiumhub.com/es/tech-blog-barcelona/integracion-continua/>. [Último acceso: 9 Junio 2021].
- [50] R. S. Pressman, *Ingeniería del Software Un enfoque práctico* Septima edición, México, D. F: McGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V., 2010.
- [51] «Jenkins,» [En línea]. Available: <https://www.jenkins.io/doc/>. [Último acceso: 9 Junio 2021].
- [52] «getapp,» [En línea]. Available: <https://www.getapp.es/software/125307/jenkins>. [Último acceso: 2021 06 04].
- [53] S. Perez, «Apiumhub,» 23 Octubre 2017. [En línea]. Available: <https://apiumhub.com/es/tech-blog-barcelona/ventajas-de-jenkins/>. [Último acceso: 2021 06 06].
- [54] R. Pac, «ciberninjas,» *Herramienta de Integración Continua*, pp. 10-15, 22 mayo 2020.
- [55] I. S. Ginés, «enmilocalfunciona,» 06 mayo 2016. [En línea]. Available: <https://enmilocalfunciona.io/integracion-continua-de-una-app-ios-con-travis-ci/>. [Último acceso: 06 junio 2021].
- [56] C. F. San Nicolás Martínez, «Herramientas de integración y desarrollo continuo para proyectos de software colaborativo,» pp. 25-47, 17 julio 2017.

- [57] C. F. S. N. Martínez, «repositorio.upct,» 17 julio 2017. [En línea]. Available: <https://repositorio.upct.es/bitstream/handle/10317/6565/tfg-sanher.pdf?sequence=1&isAllowed=y>. [Último acceso: 06 06 2021].
- [58] J. I. Piovani, «Estudios Comparativos,» 04 enero 2017.
- [59] J. Pacheco, «webyempresas,» 15 octubre 2019. [En línea]. Available: <https://www.webyempresas.com/metodo-comparativo/>. [Último acceso: 09 06 2021].
- [60] A. M. GARCÍA OROZCO, *LA INTEGRACIÓN CONTINUA Y SU APORTE AL ASEGURAMIENTO DE LA CALIDAD EN EL CICLO DE VIDA DEL DESARROLLO DE SOFTWARE*, Bogotá, 2015.
- [61] «surveymonkey,» [En línea]. Available: <https://es.surveymonkey.com/mp/sample-size-calculator/>. [Último acceso: 13 Julio 2021].

## 8 ANEXOS

### Hoja de vida investigador 1

#### HOJA DE VIDA



#### DATOS PERSONALES

<b>NOMBRES Y APELLIDOS:</b>	Paola Elizabeth Farinango Alcásiga
<b>LUGAR Y FECHA DE NACIMIENTO:</b>	05 de agosto del 1997
<b>CÉDULA DE CIUDADANÍA:</b>	1753097292
<b>SEXO:</b>	Femenino
<b>ESTADO CIVIL:</b>	Soltera
<b>DIRECCIÓN:</b>	Fco. Orellana- Coca
<b>TELÉFONO:</b>	0988962234
<b>E-MAIL:</b>	paola.farinango7292@utc.edu.ec

#### FORMACIÓN ACADÉMICA

<b>EDUCACIÓN BÁSICA:</b>	Unidad Educativa José Acosta Vallejo
<b>EDUCACIÓN SECUNDARIA:</b>	Unidad Educativa Dr. Byron Efrein Reyes
<b>TÍTULO:</b>	Bachiller General Unificado
<b>EDUCACIÓN SUPERIOR:</b>	Universidad Técnica de Cotopaxi
<b>CARRERA:</b>	Ingeniería en Informática y Sistemas Computacionales

## Hoja de vida investigador 2

### HOJA DE VIDA



#### DATOS PERSONALES

<b>NOMBRES Y APELLIDOS:</b>	Diana Pamela Panchi Zapata
<b>LUGAR Y FECHA DE NACIMIENTO:</b>	01 de agosto del 1998
<b>CÉDULA DE CIUDADANÍA:</b>	0550239636
<b>SEXO:</b>	Femenino
<b>ESTADO CIVIL:</b>	Soltera
<b>DIRECCIÓN:</b>	Thomas de Berlanga y Archipiélago
<b>TELÉFONO:</b>	0983705331
<b>E-MAIL:</b>	diana.panchi9636@utc.edu.ec

#### FORMACIÓN ACADÉMICA

<b>EDUCACIÓN BÁSICA:</b>	Escuela "Elvira Ortega"
<b>EDUCACIÓN SECUNDARIA:</b>	Unidad Educativa "Victoria Vásquez Cuví"
<b>TÍTULO:</b>	Bachiller General Unificado
<b>EDUCACIÓN SUPERIOR:</b>	Universidad Técnica de Cotopaxi
<b>CARRERA:</b>	Ingeniería en Informática y Sistemas Computacionales

## Hoja de vida del experto 1

**BLANCA LUCÍA REINOSO CHICAIZA**

### **DATOS PERSONALES**

**Cédula de ciudadanía:** 1719470609

**Dirección:** Ciudad Bicentenario (Pomasqui)

**Celular:** 09 59466575

**Nacionalidad:** Ecuatoriana

**Estado civil:** Soltera

**Lugar y fecha de nacimiento:** Cangahua, 03 de marzo de 1988

**Correo electrónico:** [lucia1202utn@gmail.com](mailto:lucia1202utn@gmail.com)



### **IDIOMAS**

INGLÉS: Intermedio

### **EDUCACIÓN**

#### **PRIMARIA**

Escuela José Acosta Vallejo

1995 – 2001

Cayambe-Pichincha

#### **SECUNDARIA**

Colegio Técnico “Cayambe”

2001 – 2007

Cayambe-Pichincha

#### **SUPERIOR**

Universidad Técnica del Norte

Facultad de Ingeniería en Ciencias Aplicadas (FICA).

Carrera de Ingeniería en Sistemas Computacionales (CISIC).

2008-2014

Ibarra - Imbabura.

### **EXPERIENCIA LABORAL**

**Analista IT en empresa TATA Consultancy Service**

Área de business intelligence del Banco Pichincha.

- Desarrollo de aplicaciones en SAS
- Desarrollo de ETLs en herramienta SAS
- Desarrollo de ETLs(DTSX) en Visual Studio 2015-2016.
- Conocimiento de manejo de DTS en SQL 2000 - 2005.
- Conocimiento de configuraciones centralizada de información.
- Conocimiento de Procesos de Versionamiento en tres ambientes DESARROLLO- TEST – PRODUCCION.
- Conocimiento de desarrollo de cubos de reportería utilizados para análisis de información financiera.
- Conocimiento de Herramienta Control M.
- Conocimientos de Herramienta Oracle
- Conocimiento de Tablas Clúster, cuando manejan grandes volúmenes de datos.
- Conocimiento de Procedimientos almacenados genéricos.
- Conocimiento de proceso de armar tablas analíticas utilizadas para reportería.
- **Manejo de Metodologías ágiles de programación.**
- Manejo de herramienta Visual Studio.
- Manejo de herramienta de reportería Power BI
- Manejo de Todos los complementos de herramienta SAS
- Manejo de Base de datos SQL Server 2008- 2015.
- Conocimiento de herramienta de reportería Visual Analytics
- Conocimiento de funcionalidades de reportería Financiera.
- Conocimiento de Lógicas financieras.
- Conocimiento de herramienta de Reportería de Oracle Business.
- Conocimientos del modelo de trabajo financiero:
  - Participantes.
  - Cartera
  - Localizaciones
  - Canales
  - Tarjetas de Crédito

Área de mantenimiento de Aplicaciones de Banco Pichincha.

- Generación de reportería para la súper intendencia de bancos.
- Generación de reportería para el área de BI.
- Conocimientos de lógicas aplicadas en la generación de reportes para el área de Riesgo.
- Análisis y solución de problemas reportados de aplicaciones de Banco Pichincha.
- Soporte para aplicaciones de DWH.
- Soporte para aplicaciones de BDWH (BI).
- Conocimientos básicos del manejo de bus de datos de Banco Pichincha.
- Conocimientos básicos de compra de cartera.

- Conocimiento medio de sincronización de cartera y validación de errores.

2015 – Actualidad

**Analista Programador en empresa Bypros S.A**

Desarrollo de aplicaciones informáticas:

- Desarrollo de aplicaciones en PHP.
- Manejo de metodología Metrica 3.
- Manejo de herramienta ScriptCase.
- Manejo de Ajax.
- Manejo de Base de datos Postgresql.
- Manejo de Base de datos Mysql.
- Manejo de Base de datos SQL SERVER 2008.

2014-2015

**Prácticas pre –profesionales:**

Gobierno Autónomo descentralizado Municipal San Miguel de Ibarra, departamento de Informática.

- Desarrollo del proyecto Mercado Amazonas Ciudad Comercial.
- Manejo de herramienta Symfony complemento de java.
- Manejo de librería jquery.
- Desarrollo java.

2013 – 2014

**Desarrollo de Tesis**

Desarrollo de la Aplicación Informática

- Manejo de tecnología JSF con IDE eclipse (JAVA).
- Manejo de librerías primefaces, jstl, itext, fileUpload, entre otras.
- Manejo de ireport.
- Manejo de metodología XP.

2013-2014

### **CURSOS REALIZADOS**

#### **Cisco CCNA1: “Aspectos básicos de redes”**

Universidad Técnica del Norte –Academia CISCO

2012-2013

#### **Cisco CCNA2 “Routing and Switching Essentials”**

Universidad Técnica del Norte –Academia CISCO

2013-2014

#### **CISCO IT ESSENTIAL “Mantenimiento de hardware y software de computadoras”**

Universidad Técnica del Norte –Academia CISCO

Marzo – julio 2013

#### **Programación básica en C#**

Universidad Técnica del Norte –Centro de Capacitación Continua

2008

#### **Armar y Dar Mantenimiento a un Computador**

Universidad Técnica del Norte –Centro de Capacitación Continua.

2008

### **APTITUDES Y CONOCIMIENTOS**

#### **Programación**

- Programación Java con framework JSF
- Conocimiento en librerías primefaces, jstl, itext, fileUpload, entre otras.
- Programación php con base de datos.
- Conocimientos de Symfony(JAVA).
- Manejo de librería JQuery.
- Manejo de herramientas para reportes lreport
- Conocimientos de Html5 y Css3.
- Conocimiento de ScriptCase y sus librerías.
- Programación SAS y todos sus paquetes.
- Desarrollo de ETLs.

- Desarrollo de DTS.
- Manejo de herramienta Power BI.
- Mantenimiento de aplicaciones en general.
- Análisis y generación de reportería financiera y entes de control.
- Manejo de mitologías de desarrollo:
  - Métrica 3.
  - XP.
  - Scrum
  - Cascada.

#### **Bases de datos**

- Manejo de las siguientes suites de Bases de datos:
  - Microsoft SQL Server 2005,2008 -2015
  - MySQL
  - PostgreSQL
  - Oracle.

#### **Otros**

- Conocimientos básicos de:
  - Desarrollo de aplicaciones en .NET.
  - Sistemas de Información Geográfica (GIS).
  - SOA (Arquitectura Orientada a Servicios) en java IDE eclipse.
  - Arquitectura de software.
- Ayuda a resolución de conflictos informáticos.
- Prioridad de trabajo en equipo.
- Habilidad para aprender e investigar herramientas adaptables a cualquier situación.
- Habilidad de Análisis de problemas para solucionar en un tiempo de Prioridad 1.

#### **REFERENCIAS PERSONALES**

Ing. Mónica Gómez Viteri (PL)	Celular:0998004461
Mg. Edwin Catagña(PL)	Celular:0984559757
Ing. Rosa Sagñay	Celular: 0996809868
Ing.Katy Arias	Celular: 0990893501
Ing. Miguel Angel López	Celular: 0998212836
Ing. Johana Villacres	Celular: 0958781385
Ing.Wendy Mera	Celular: 0999274014

## Hoja de vida del experto 2

<b>Nombres :</b> William Jair <b>Apellidos :</b> Dávila Vargas <b>Fecha de nacimiento:</b> 19 de Octubre de 1994 <b>Domicilio:</b> Cdla. Dager, Calle Rio Zamora entre Av. Tumbes y Av. Tarqui <b>Teléfono de domicilio:</b> 2977557 <b>E-mail 1:</b> davilawilliam94@gmail.com	<b>C.I.:</b> 0928363993 <b>Tipo de sangre :</b> O+ <b>Edad:</b> 25 años <b>Ciudad:</b> Milagro, Guayas  <b>Celular:</b> 0979014551 <b>E-mail 2:</b> williamjair94@hotmail.com	
--	---	---

### FORMACIÓN ACADÉMICA

<b>2017</b> <i>Milagro</i>	<b>Ingeniero en Sistemas Computacionales</b> Superior: Universidad de Estatal de Milagro (UNEMI)
<b>2012</b> <i>Milagro</i>	<b>Aplicaciones Informáticas</b> Secundaria: Colegio Particular Mixto "Madre de Dios"
<b>2006</b> <i>Milagro</i>	<b>Educación Básica</b> Primaria: Escuela Particular "Victoria Concha de Valdez"

### PERFIL PROFESIONAL

Especializado en dar soluciones informáticas a medida en múltiples plataformas de programación, responsable, dinámico, innovador, siempre con ideas nuevas, y con la capacidad de adaptarme a cualquier proyecto así sea en condiciones de alta presión.

### CONOCIMIENTO ADQUIRIDO

N°	Descripción	Programas
1	Base de datos	▪ SQLServer, PostgreSQL, MySQL, SQLITE y Access
2	Lenguajes de programación de escritorio	▪ Visual.Net, C#, Java y Python
3	Aplicaciones Móviles	▪ Android
4	Lenguaje de programación para la web	▪ PHP, Django y ASP.NET
5	Sistemas operativos, programas utilitarios, entre otros	▪ Windows 7,8.1,10, Linux Ubuntu, Microsoft Word, Excel, PowerPoint, Publisher, Project, Visio, Prezi, Lucidchart
6	Conocimiento adicional para el desarrollo de aplicaciones	▪ POO, XHTML, HTML5, CSS, Javascript, JQuery, Ajax, XML, JSON, Bootstrap y GIT.

### EXPERIENCIA LABORAL

N°	Empresa	Área	Posición	De mes-año	A mes-año
1	UNEMI	Departamento de Ensamblaje y reparación de computadoras	Técnico	Abril, 2015	Junio, 2015
2	UNEMI	Facultad Ciencias de la Ingeniería	Ayudante de Catedra	Febrero, 2013	Octubre, 2016
3	Cavnet S.A.	Helpdesk	Analista	Abril, 2016	Junio, 2016
4	Pc & Servicios	Desarrollo	Programador	Enero, 2016	Octubre, 2016
5	IN.PLANET S.A.	Departamento de Proyectos Informáticos	Desarrollador Web	Abril, 2018	Mayo, 2019

6	IN.PLANET S.A.	Departamento de Proyectos Informáticos	Ingeniero de Proyectos	Junio, 2019	Actualidad
---	----------------	--	------------------------	-------------	------------

### CURSOS, JORNADAS, TALLERES Y SEMINARIOS

N°	Nombre	Institución	Horas	Año
1	Curso de operador, programador y diseñador gráfico	ICD Computer Center	140	2012
2	V Jornadas Tecnológicas e Industriales	UNEMI	20	2012
3	Jornadas Tecnológicas Calidad en la Web	UNEMI	5	2014
4	Curso Online "Analítica Web"	Gobierno de España- Ministerio de industria, energía y turismo-Escuela de organización industrial	40	2016
5	Curso Online "Productividad Personal en la Era Digital"	Fundación Santa María la Real del Patrimonio Histórico – España	40	2016
6	Curso de Desarrollo de Aplicaciones Web usando Tecnología Open Source	Academia Cisco Espol	50	2018
7	Curso Online "Internet of Everything"	Cisco Networking Academy	30	2018
8	Security Day 2018	Escuela Superior Politécnica del Litoral	8	2018
9	Aplicaciones Web Responsivas con ASP.NET, WCF y Entity Framework 6	SIPECOM	24	2019

### MERITOS Y GALARDONES

N°	Nombre	Institución	Año
1	Mención honorífica por haber sido el segundo escolta del pabellón de Milagro	Escuela Particular "Victoria Concha de Valdez"	2005 - 2006
2	Medalla de reconocimiento por haber sido el segundo escolta del pabellón de Milagro	Escuela Particular "Victoria Concha de Valdez"	2005 - 2006
3	Reconocimiento por excelencia académica de buen desempeño como estudiante y abanderado del pabellón nacional	Club de Leones de Milagro y su Comité de Damas	2011
4	Mención honorífica por haber sido el abanderado del pabellón nacional del Ecuador	Colegio Particular Mixto "Madre de Dios"	2012
5	Placa de reconocimiento por la elaboración del mejor proyecto de informática	Colegio Particular Mixto "Madre de Dios"	2012
6	Medalla de oro del mejor bachiller	Colegio Particular Mixto "Madre de Dios"	2012
7	Reconocimiento por haber realizado y concluido con éxito las ayudantías de cátedra de "ESTRUCTURA DE DATOS" Y "REALIDAD NACIONAL"	Universidad Estatal de Milagro	2014
8	Beca de Alto promedio y distinción académica	Universidad Estatal de Milagro	2013 - 2016
9	Participación en el concurso de Galardones Nacionales 2016 con el tema "Diseño e implementación de una aplicación"	Secretaría de Educación Superior, Ciencias, Tecnología	2016

tecnológica que permita detectar el maltrato y acoso en niños, niñas y adolescentes en las instituciones educativas"	(SENESCYT)-Investigaciones Científicas	
--	--	--

## HABILIDADES PROFESIONALES

- Manejo de diferentes herramientas de programación y administración de bases de datos.
- Experiencia en el desarrollo de sistemas informáticos tanto web y escritorio.
- Facilidad y eficacia en la resolución de problemas así sea bajo presión.
- Creatividad e innovación en cada trabajo realizado.
- Colaboración, argumentación y toma de decisiones.
- Capacidad de adaptación al grupo de trabajo.
- Comunicación eficaz y liderazgo.

## LOGROS PERSONALES

- Creador y fundador de AlgoriSoft, sitio web dedicado a la comercialización de fragmentos de sistemas informáticos que he ido desarrollando a lo largo de mi carrera profesional como programador para aquellas personas que quieran incursionar en el mundo de la programación y quieran tener una base ideal para realizar cualquier tipo de proyecto tecnológico a través de un gestor de contenido que desarrolle para la subida dinámica de la información. Enlace del sitio web: <https://algorisoft.com/>
- Instructor en la plataforma UDEMY. Curso denominado "Implementa tu sitio web Django en un VPS Ubuntu Español 2019". Enlace del curso: <https://www.udemy.com/implementa-tu-sitio-web-django-en-un-vps-ubuntu-espanol-2019/>
- Creador del canal de YouTube AlgoriSoft para dictar cursos de programación, Curso de Python con Django de 0 a Máster | Español. Enlace del canal <https://www.youtube.com/c/AlgoriSoft>

## REFERENCIAS PERSONALES

<b>MSC. William Bazán Vera</b> Universidad Agraria del Ecuador Docente de la Facultad de Ciencias de la Informática william_bazan@hotmail.com 0996852974	<b>MSc. Oscar Berneo Almeida</b> Universidad Estatal de Milagro Docente de la Facultad de Ciencias de la Ingeniería oscar_majic@hotmail.com 0960458705
<b>Ing. Marcos Naranjo Pino</b> Papelera Nacional S.A. Auditor de tecnologías de la información marcos_naranjo_pino@hotmail.com 0991324716	<b>MgTI. Rufina Narcisca Bravo</b> Universidad Estatal de Milagro Docente de la Facultad de Ciencias de la Educación rbravo@unemi.edu.ec 0996460826

## REFERENCIAS LABORALES

<b>Ing. Jordy Malo</b> IN.PLANET S.A. Desarrollador de aplicaciones móviles jmalo@in-planet.net 0982467563	<b>Ing. Freddy Valenzuela Ortega</b> Cavnet S.A. Gerente Técnico del departamento Helpdesk fvalenzu@hotmail.com 0994657465
--	--

## Hoja de vida del experto 3

[CURRICULUM VITAE] Omar Bedón

### OMAR SANTIAGO BEDÓN ZAMBRANO

Número de cédula: 0502630437

Fecha de nacimiento: 31/Enero/1987

Edad: 33 años

Teléfono móvil: 0969021016

Dirección: Miraflores Alto entre Thomas de Berlanga

Ciudad: Latacunga

Estado civil: Casado

E-mail: omarbe2013@hotmail.es



### PERFIL PROFESIONAL

Soy profesional de Ingeniería en Sistemas, experto en brindar soluciones informáticas, técnicas de seguridad SGSI, Controles de seguridad según la ISO 27001, 27002, seguridad perimetral, Ethical hacking, estoy actualizado en las innovaciones tecnológicas para la aplicación de métodos vigentes, analista, innovador, investigativo, realice mi especialización en temas de sistemas, gestión, estructura y análisis de la información aplicada al TICs, me encanta enfrentar retos para mi carrera, me adapto fácilmente a un nuevo empleo, no soy reacio a los cambios, me gusta trabajar en equipo y sometido a grandes tensiones, tengo habilidad para solucionar los problemas y alcanzar las metas propuestas por la empresa y el equipo de trabajo.

### FORMACIÓN ACADÉMICA

#### *FORMACIÓN PRIMARIA*

Titulación:	Certificación de culminación estudiantil
Año graduación:	1999
Institución:	Escuela Isidro Ayora

***FORMACION SECUNDARIA***

Titulación: Bachiller  
Especialidad: Químico Biólogo  
Institución: Instituto tecnológico Superior  
"Vicente León"

***FORMACION SUPERIOR- UNIVERSITARIA – Tercer Nivel Técnico***

Titulación: Tecnólogo  
Especialidad: Informática mención Análisis de  
Sistemas  
Institución: Instituto tecnológico Superior  
"Vicente León"

***FORMACION SUPERIOR- UNIVERSITARIA – Tercer Nivel***

Titulación: Ingeniería  
Especialidad: Sistemas  
Institución: Universidad Tecnológica  
Indoamericana