



**ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ
MANUEL FÉLIX LÓPEZ**

CARRERA DE COMPUTACIÓN

**INFORME DE TRABAJO DE INTEGRACIÓN CURRICULAR
PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
CIENCIAS DE LA COMPUTACIÓN**

**MECANISMO: SISTEMATIZACIÓN DE EXPERIENCIAS PRÁCTICAS
DE INVESTIGACIÓN Y/O INTERVENCIÓN**

TEMA:

**CARACTERIZACIÓN DE LAS PRÁCTICAS DE DEVOPS EN LAS
ORGANIZACIONES**

AUTOR:

ÁNGEL YOEL MOREIRA ÁLAVA

TUTOR:

MGTR. ALFONSO TOMÁS LOOR VERA

CALCETA, FEBRERO DE 2023

DECLARACIÓN DE AUTORÍA

Yo ÁNGEL YOEL MOREIRA ÁLAVA, con cédula de ciudadanía 1316331212, declaro bajo juramento que el Trabajo de Integración Curricular titulado: **CARACTERIZACIÓN DE LAS PRÁCTICAS DE DEVOPS EN LAS ORGANIZACIONES** es de mi autoría, que no ha sido previamente presentado para ningún grado o calificación profesional, y que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración, concedo a favor de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos, conservando a mi favor todos los derechos patrimoniales de autor sobre la obra, en conformidad con el Artículo 114 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación.



ÁNGEL YOEL MOREIRA ÁLAVA
CC: 1316331212

AUTORIZACIÓN DE PUBLICACIÓN

ÁNGEL YOEL MOREIRA ÁLAVA, con cédula de ciudadanía 1316331212, autorizo a la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López, la publicación en la biblioteca de la institución del Trabajo de Integración Curricular titulado: **CARACTERIZACIÓN DE LAS PRÁCTICAS DE DEVOPS EN LAS ORGANIZACIONES**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.



ÁNGEL YOEL MOREIRA ÁLAVA
CC: 1316331212

CERTIFICACIÓN DEL TUTOR

ALFONSO TOMÁS LOOR VERA, certifica haber tutelado el Trabajo de Integración Curricular titulado: **CARACTERIZACIÓN DE LAS PRÁCTICAS DE DEVOPS EN LAS ORGANIZACIONES**, que ha sido desarrollado por **ÁNGEL YOEL MOREIRA ÁLAVA**, previo a la obtención del título de **INGENIERO EN CIENCIAS DE LA COMPUTACIÓN**, de acuerdo al **REGLAMENTO DE LA UNIDAD DE INTEGRACIÓN CURRICULAR DE CARRERAS DE GRADO** de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López.

MGTR. ALFONSO TOMAS LOOR VERA
CC: 1311655938
TUTOR

APROBACIÓN DEL TRIBUNAL

Los suscritos integrantes del Tribunal correspondiente, declaramos que hemos **APROBADO** el Trabajo de Integración Curricular titulado: **CARACTERIZACIÓN DE LAS PRÁCTICAS DEVOPS EN LAS ORGANIZACIONES**, que ha sido desarrollado por **ÁNGEL YOEL MOREIRA ÁLAVA**, previo a la obtención del título de **INGENIERO EN CIENCIAS DE LA COMPUTACIÓN**, de acuerdo al **REGLAMENTO DE LA UNIDAD DE INTEGRACIÓN CURRICULAR DE CARRERAS DE GRADO** de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López.

MGTR. DANIEL A. MERA MARTINEZ
CC: 1301932156
PRESIDENTE DEL TRIBUNAL

MGTR. RICARDO A. VÉLEZ
VALAREZO
CC:1306391614
MIEMBRO DEL TRIBUNAL



Firmado
digitalmente por
FERNANDO
RODRIGO
MOREIRA MOREIRA

MGTR. FERNANDO R. MOREIRA
MOREIRA
CC:1311726689
MIEMBRO DEL TRIBUNAL

AGRADECIMIENTO

A la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López que me dio la oportunidad de crecer como ser humano a través de una educación superior de calidad y en la cual he forjado mis conocimientos profesionales día a día;

A Dios, por ser el guía y conductor de mi vida, por darme la fuerza necesaria para seguir adelante día a día y enfrentar cualquier obstáculo que se presente.

A mi tutor el Ing. Alfonso Tomás Loor Vera, por haberme guiado con sus conocimientos, brindarme su tiempo y por darme la oportunidad de participar en sus proyectos y desarrollar en la UDC mi trabajo de integración curricular.

Al Ing. Edwin Moreira Santos, por brindarme su apoyo en el diseño y ejecución de mi trabajo de integración curricular.

A los docentes y amigos que me brindaron su apoyo durante la formación profesional y personal, gracias por estar presentes en todo momento.

ÁNGEL YOEL MOREIRA ÁLAVA

DEDICATORIA

A Dios, por darme la vida, la salud y la sabiduría necesaria en mis estudios, guiándome por el camino del bien.

A mis padres, quienes me han enseñado con sus sabios consejos a creer en mis capacidades, afrontar las adversidades, y nunca desmayar, ellos son el testimonio fiel de la existencia de Dios, son mi fuente de motivación y mi apoyo incondicional.

A mis hermanos que aportaron con su apoyo, confianza y oraciones para que todo bajo la voluntad de Dios culmine con éxito.

A mi pareja, por su amor, su paciencia, su comprensión, su cariño, por acompañarme, apoyarme y emprender conmigo nuevos retos para un mejor futuro

Y a todas aquellas personas que siempre han estado a mi lado de alguna u otra forma pendiente de mi progreso profesional.

Mil gracias, ¡sin ustedes no habría logrado este éxito!

ÁNGEL YOEL MOREIRA ÁLAVA

CONTENIDO GENERAL

CARÁTULA	I
DECLARACIÓN DE AUTORÍA.....	II
AUTORIZACIÓN DE PUBLICACIÓN	III
CERTIFICACIÓN DEL TUTOR.....	IV
APROBACIÓN DEL TRIBUNAL	V
AGRADECIMIENTO	VI
DEDICATORIA	VII
CONTENIDO GENERAL	VIII
CONTENIDO DE TABLAS	IX
CONTENIDO DE FIGURAS.....	X
CONTENIDO DE GRÁFICOS.....	X
RESUMEN	XI
PALABRAS CLAVE.....	XI
ABSTRACT	XII
KEY WORDS.....	XII
CAPÍTULO I. ANTECEDENTES.....	1
1.1. DESCRIPCIÓN DE LA INSTITUCIÓN	1
1.2. DESCRIPCIÓN DE LA INTERVENCIÓN	3
1.3. OBJETIVOS	5
1.3.1. <i>OBJETIVO GENERAL</i>	5
1.3.2. <i>OBJETIVOS ESPECÍFICOS</i>	5
CAPÍTULO II. DESARROLLO METODOLÓGICO DE LA	6
INTERVENCIÓN	6
2.1. EFECTUAR UNA REVISIÓN SISTEMÁTICA DE LA LITERATURA SOBRE LAS CARACTERIZACIONES DEVOPS.....	6
2.1.1. <i>REVISIÓN SISTEMÁTICA DE LA LITERATURA</i>	6
2.1.2. <i>USO DE LA REVISIÓN SISTEMÁTICA DE LA LITERATURA</i>	6
2.1.3. <i>PASOS DE LA REVISIÓN SISTEMÁTICA DE LA LITERATURA</i>	7
2.2. EFECTUAR UN ANÁLISIS COMPARATIVO DE LAS HERRAMIENTAS DE DEVOPS.	9
2.3. ELABORAR UNA GUÍA DE DEVOPS REFERENTE AL CICLO DE VIDA DE DESARROLLO DE SOFTWARE.....	9

2.4. SOCIALIZAR LOS RESULTADOS CON LA UDIV DE DESARROLLO COMPUTACIONAL.	10
CAPÍTULO III. DESCRIPCIÓN DE LA EXPERIENCIA	11
3.1. REVISIÓN SISTEMÁTICA DE LA LITERATURA	11
3.1.1. PREGUNTAS DE INVESTIGACIÓN	11
3.1.2. PROCESO DE BÚSQUEDA	11
3.1.3. CRITERIOS DE INCLUSIÓN Y EXCLUSIÓN	12
3.1.4. EVALUACIÓN DE CALIDAD	2
3.1.5. EXTRACCIÓN Y ANÁLISIS DE DATOS	3
3.2. EFECTUAR UN ANÁLISIS COMPARATIVO DE LAS HERRAMIENTAS DE DEVOPS	7
3.2.1. PLANEACIÓN	8
3.2.2. CODIFICACIÓN	8
3.2.3. LANZAMIENTO	9
3.2.4. DESPLIEGUE	10
3.2.5. OPERACIÓN	10
3.2.6. MONITOREO	11
3.2.7. PRUEBAS	12
3.2.8. CONSTRUCCIÓN	13
3.3. ELABORAR UNA GUÍA DE DEVOPS REFERENTE AL CICLO DE VIDA DE DESARROLLO DE SOFTWARE.	15
3.4. SOCIALIZAR LOS RESULTADOS CON LA UDIV DE DESARROLLO COMPUTACIONAL.	16
CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES	1
4.1. CONCLUSIONES	1
4.2. RECOMENDACIONES	2
BIBLIOGRAFÍA	3
ANEXOS	6
ANEXO 1. CREACIÓN DE LA GUÍA DEVOPS	7
ANEXO 2. ENTREGA DE GUÍA DE DEVOPS A LOS MIEMBROS DE LA UNIDAD DE DESARROLLO COMPUTACIONAL	51
ANEXO 3. CERTIFICADO DE ENTREGA DE GUÍA A LA UDIV	52

CONTENIDO DE TABLAS

Tabla 1. Resultados de la búsqueda en las bases de datos.	12
Tabla 2. Criterios de inclusión y exclusión.	13

Tabla 3. Resultados de base de datos, aplicando los criterios de inclusión y exclusión.	13
Tabla 4. Filtro por título.....	13
Tabla 5. Filtro por resumen.....	14
Tabla 6. Resultados obtenidos por medio de una lectura rápida.....	14
Tabla 7. Artículos encontrados en las bases de dato	1
Tabla 8. Evaluación de la calidad de las preguntas de investigación.	2
Tabla 9. Resultado de la P1.....	3
Tabla 10. Resultado de la P2.	3
Tabla 11. Respuesta P3.	4
Tabla 12. Respuesta a P4.....	5
Tabla 13. Respuesta a P5.....	6

CONTENIDO DE FIGURAS

Figura 1. Ecosistema DevOps.....	7
Figura 2. Resultado del análisis.....	15

CONTENIDO DE GRÁFICOS

Gráfico 1. Resultados de las bases de datos en porcentaje.	1
Gráfico 2. Retos al adoptar DevOps por continentes.....	5

RESUMEN

El presente trabajo de titulación tuvo como propósito desarrollar una investigación sobre la caracterización de las prácticas de DevOps en las organizaciones con el fin de elaborar un documento base sobre el proceso de actualización tecnológica en la UDIV de Desarrollo Computacional. De igual manera, se realizó una revisión sistemática de la literatura la cual sirvió para conocer las metodologías, las prácticas, las herramientas, los beneficios y los retos de DevOps en las organizaciones, asimismo se aplicó el método cualitativo y analítico con el fin de obtener información por medio del análisis de datos, de igual manera se procedió a diseñar un ecosistema de DevOps con las herramientas más adecuadas en cada fase del ciclo de vida DevOps. Seguidamente, con los resultados de cada uno de los objetivos se creó una guía de DevOps con el fin de socializarla con los miembros de la UDIV de Desarrollo Computacional para que pueda servir de insumo para futuros proyectos.

PALABRAS CLAVE

DevOps, revisión sistemática, UDIV de Desarrollo Computacional.

ABSTRACT

The purpose of this degree work was to develop an investigation on the characterization of DevOps practices in organizations in order to prepare a base document on the process of technological updating in the UDIV of Computational Development. In the same way, a systematic review of the literature was carried out, which served to know the methodologies, practices, tools, benefits and challenges of DevOps in organizations, also the qualitative and analytical method was applied in order to obtain information through data analysis, in the same way we proceeded to design a DevOps ecosystem with the most appropriate tools in each phase of the DevOps life cycle. Next, with the results of each of the objectives, a DevOps guide was created in order to share it with the members of the UDIV of Computational Development so that it can serve as input for future projects.

KEY WORDS

Devops, systematic review, Computational Development Unit

CAPÍTULO I. ANTECEDENTES

1.1. DESCRIPCIÓN DE LA INSTITUCIÓN

De acuerdo al artículo 350 de la Carta Magna, señala que: el Sistema de Educación Superior tiene como finalidad la formación académica y profesional con visión científica y humanista; la investigación científica y tecnológica; la innovación, promoción, desarrollo y difusión de los saberes y las culturas; la construcción de soluciones para los problemas del país (Consejo de Educación Superior, 2016). La Educación Superior cumple un papel fundamental en la sociedad, ya que son los impulsores de la evolución, se vincula con su entorno, guiando y aportando de conocimiento integral a los estudiantes por el camino del progreso, formándolos con valores morales y éticos, ya que son el futuro del país.

Asimismo, el rol que desempeñan las universidades en la actualidad, de acuerdo a Padilla (2018) como se citó en la Universidad de Guadalajara (2018) es crear un ambiente propicio, en donde se vea reflejado lo intelectual, al irradiar conocimientos científicos. De igual manera, establecer disciplinas para el individuo, formando humanos inteligentes, por su capacidad y su acción benéfica: fundar pensamiento crítico, porque la producción del conocimiento no es un hecho simplemente abstraído, es la manifestación de complejas interrelaciones sociales restringidos por los individuos en los métodos de innovación social, con el fin de lograr una prosperidad común.

Según la Universidad del Ecuador (2022) la Escuela Superior Politécnica Agropecuaria de Manabí "Manuel Félix López", ESPAM MFL, es una universidad pública ecuatoriana ubicada en la ciudad de Calceta, Cantón Bolívar, es la principal universidad de la zona norte de la provincia de Manabí. De igual manera, ESPAM MFL (2016) hace referencia que esta universidad se convierte en una institución de educación pública del Ecuador y nace como: "Persona jurídica de derecho público, autónoma, que se rige por la Constitución Política del Estado, Ley de Educación Superior, su Estatuto Orgánico y Reglamentos, para preparar a la juventud ecuatoriana y convertirla en profesionales".

La ESPAM MFL, a inicios de sus labores surge con cuatro carreras acordes a las necesidades de la población como es Agroindustria, Medio Ambiente, Agrícola y Pecuaria, pero de acuerdo a muchos estudios se llegó a la conclusión de implementan la Carrera de Informática, tiempo después para satisfacer el contexto empresarial se creó dos Carreras más con es la Administración Pública y de Empresas, todo esto con el único fin de formar profesionales de alta calidad que elaboren proyectos sustentables y sean fundadores de fuentes de trabajo.

Según la ESPAM MFL (2021) la Carrera de Computación tiene como objetivo “formar profesionales que aporten innovaciones computacionales para la solución de problemas sociales, regionales y nacionales, vinculados al modelo constructivista y desarrollador productivo, dentro de equipos multidisciplinares interdisciplinares, con énfasis en el sector agropecuario y agroindustrial, que actúen con responsabilidad económica, ambiental, ética y social, en sintonía con los planes y políticas públicas”.

La Carrera Computación cuenta con tres Unidades de Docencia Investigación y Vinculación – UDIV, entre una de ellas se encuentra la Unidad de Desarrollo Computacional (UDC), de acuerdo a la ESPAM MFL (2018) en la resolución 012-2018 de sesión ordinaria del Honorable Consejo Politécnico de la ESPAM MFL de fecha viernes 20 de septiembre de 2018: N° 9 se procedió aprobar la propuesta de la Carrera de Computación respecto a la reestructuración de: "Unidad de Producción de software a: "Unidad de Desarrollo Computacional", acogiendo recomendación de la Comisión Académica.

La UDIV de Desarrollo Computacional es una unidad dedicada a capacitar, asesorar y realizar el acompañamiento a estudiantes de la Carrera en el desarrollo y gestión de proyectos de desarrollo de software que contribuyan a solucionar las necesidades tecnológicas internas y externas. Tiene como misión el apoyar en los procesos de docencia, investigación y vinculación de la Carrera de Computación mediante la gestión de proyectos de desarrollo de software.

1.2. DESCRIPCIÓN DE LA INTERVENCIÓN

En la actualidad, la tecnología está avanzando constantemente en muy poco tiempo, es por ello que últimamente en las organizaciones se está escuchando hablar de DevOps, este término hace referencia a estrategias de transformación digital y a entregas continuas o en otras palabras desarrollo ágil. El objetivo de DevOps en las organizaciones es ayudarles a generar productos y servicios software de una calidad extraordinaria y con un costo menor, pero de una manera rápida.

En una investigación realizada por la Universidad Politécnica de Madrid, encuestaron a once empresas aleatorias de desarrollo de software, el cual manifestaron que tenían como problema la falta de comunicación entre el desarrollo y las operaciones y, por lo tanto, obtener un tiempo de respuesta rápida a las demandas comerciales era bastante difícil. Es por ello que decidieron implementar DevOps, dando como solución, tiempos de respuesta más cortos e implementaciones más frecuentes, mayor productividad, mejores comentarios del cliente y menores costos de TI. (Díaz et al., 2018)

DevOps maneja un conjunto de principios enfocado en la entrega e integración continua de software, esto envuelve una nueva cultura para crear y extender software en argumentos soberanamente colaborativos y ágiles, este paradigma implica a todos los representantes como creadores, profesionales de las Tecnologías de la Información con sus siglas (TI), usuarios y stakeholders (Dittrich et al., 2018, como se citó en Mamani et al., 2020).

Según Senapathi et al. (2018); Ruiz, (2018); Claranet, s.f.; Mike, (2021) y Creative Commons, (2020) afirman que DevOps promueve una comunicación más fluida entre los equipos de desarrollo de aplicaciones y sus semejantes en operaciones tecnológicas. Gupta et al., (2017) aluce que empresas como Netflix y Fotopedia descubrieron que al utilizar DevOps unieron el conocimiento, las habilidades y las capacidades tanto para el equipo de desarrollo de software como para el de operaciones, aumentando la colaboración, la automatización y la medición de su organización.

Asimismo, Centro de Recursos It User (2020) concuerda, que DevOps es responsable de combinar dos equipos, del desarrollo y los semblantes operativos del proyecto, y cerciorarse de que haya bastante comunicación entre ellos para conseguir resultados efectivos. DevOps sigue manteniendo la idea ágil de que las personas están por encima de los procesos y herramientas. Lo que se quiere ahora es fomentar una cultura de equipo, una cultura de empresa a distintos niveles donde haya un poco más de transparencia y que por ejemplo gente de desarrollo sepa lo que hace sistemas y viceversa, negocio tenga más visibilidad sobre ciertos temas de desarrollo. (De la Torre et al., 2017)

En la Carrera de Computación los estudiantes que realicen sus prácticas pre profesionales, trabajo de Integración Curricular o que se encuentren involucrados en la construcción de algún proyecto de software, necesitan desarrollar sistemas confiables y cumplir con las metas establecidas con los clientes en un tiempo acordado. Por este motivo la UDIV de Desarrollo Computacional se encuentra actualizando su stack de desarrollo acorde a las tecnologías y herramientas que se utilizan actualmente en el mercado e incorporar el tema de DevOps en su proceso de desarrollo de software, con la finalidad de estar a la vanguardia de las tecnologías y así los estudiantes de la Carrera puedan concluir sus proyectos en menor tiempo. Con estos antecedentes, el presente trabajo tuvo como propósito realizar una búsqueda bibliográfica en diferentes fuentes de base de datos y así poder conocer cómo involucrar el enfoque de trabajo DevOps, sus buenas prácticas, las herramientas de automatización que se utilizan en cada una de las fases del desarrollo de software, si existe alguna metodología o modelo de aplicación y otros aspectos de interés que surjan durante esta investigación, todo el resultado obtenido se plasma en un documento que sirve como insumo para la UDIV de Desarrollo Computacional en el proceso de actualización tecnológica.

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Desarrollar una investigación sobre caracterización de las prácticas de DevOps en organizaciones con el fin de elaborar un documento base para el proceso de actualización tecnológica en la UDIV de Desarrollo Computacional de la Carrera de Computación.

1.3.2. OBJETIVOS ESPECÍFICOS

- Efectuar una revisión sistemática de la literatura sobre las caracterizaciones DevOps.
- Efectuar un análisis comparativo de las herramientas de DevOps.
- Elaborar una guía de DevOps referente al ciclo de vida de desarrollo de Software.
- Socializar los resultados con la UDIV de Desarrollo Computacional.

CAPÍTULO II. DESARROLLO METODOLÓGICO DE LA INTERVENCIÓN

El presente trabajo de investigación curricular, tiene como propósito desarrollar una investigación sobre caracterización de las prácticas de DevOps en organizaciones con el fin de elaborar un documento base para el proceso de actualización tecnológica en la UDIV de Desarrollo Computacional, el cual consta de cinco objetivos específicos; se aplicó el método cualitativo para recopilar datos de la misma, asimismo se utilizó el método analítico con el fin de descubrir información relevante sobre un tema.

2.1. EFECTUAR UNA REVISIÓN SISTEMÁTICA DE LA LITERATURA SOBRE LAS CARACTERIZACIONES DEVOPS.

2.1.1. REVISIÓN SISTEMÁTICA DE LA LITERATURA

Se dice que la revisión sistemática es un medio de investigación que recolecta y facilita un resumen sobre algún tema en específico, por otro lado, se refiere a una pregunta estructurada o búsqueda, basada en criterios explícitos aplicados uniformemente a todos los artículos, de igual manera, es un resumen cualitativo y/o cuantitativo y es basado en evidencias.

De igual manera, se dice que la revisión sistemática posee características concretas como: que es un instrumento de síntesis de información, investigación cualitativa, es una investigación secundaria (investigación sobre lo investigado), que ofrece una síntesis de la información científica disponible, que aumenta la eficacia de las conclusiones de estudios propios y finalmente que identifica aquellas áreas en las que existe incertidumbre y que requieren una investigación.

2.1.2. USO DE LA REVISIÓN SISTEMÁTICA DE LA LITERATURA

La revisión sistemática es muy útil para reducir la realidad sobre un procedimiento o conjunto de técnicas, para abreviar la certidumbre empírica de

las protecciones y restricciones de un método rápido delimitado, identificar una brecha en la exploración presente para proponer espacios para una mayor investigación, y proporcionar antecedentes para posicionar debidamente nuevas actividades de investigación. Sin embargo, también se pueden realizar revisiones sistemáticas para inspeccionar hasta qué punto la evidencia empírica apoya/contradice las hipótesis teóricas, o incluso para ayudar a crear nuevas hipótesis.

Según Arteaga (2020), el uso de este tipo de revisión conlleva una serie de ventajas inherentes. En primer lugar, la realización de la revisión sistemática de la literatura con sus siglas (SLR) permite al investigador consultar un número muy elevado de publicaciones para dar respuesta a una serie de preguntas muy concretas planteadas de antemano. Esta sistematización evita, por un lado, que la persona que lleva a cabo la investigación se desvíe de su objetivo inicial de manera innecesaria y, por otro lado, que se introduzcan sesgos en la revisión, aunque hay que tener en cuenta el efecto del sesgo de publicación que se abordará más adelante.

De igual manera, el autor afirma que, como segundo lugar, al tratarse de una metodología claramente establecida con un protocolo muy desarrollado, su ejecución resulta muy accesible, es relativamente fácil encontrar materiales de calidad que nos permiten su aprendizaje e incluso existen herramientas específicas diseñadas para llevarlas a cabo, lo que facilita a la persona que realiza la investigación tener una idea clara de la calidad metodológica de su trabajo.

2.1.3. PASOS DE LA REVISIÓN SISTEMÁTICA DE LA LITERATURA

Existen muchas teorías acerca de los pasos correctos en una revisión sistemática de la literatura, de acuerdo a Pardo (2020) son: formular la pregunta, definición de las fuentes y del método de búsqueda, definición y aplicación de criterios de selección de los estudios, análisis crítico de los estudios, extracción y síntesis de los resultados de los estudios, conclusiones e inferencias, evaluar

el riesgo de sesgo de los estudios, analizar y presentar los resultados, interpretar los resultados y obtener conclusiones; y mejorar y actualizar la revisión (párr. 8).

En cambio, Lerardi et al. (2017) manifiesta que una vez definido el tema de investigación y una vez localizada la necesidad de iniciar una eficiente RSL es muy recomendable seguir las diferentes etapas que la constituyen. Si bien es cierto, por la influencia que tiene la fase de planificación es un poco más significativa y está compuesta por una serie de preguntas de investigación de forma clara y concisa, asimismo, por designar un proceso de estudio o base de datos y establecer algunos de los criterios de inclusión y exclusión.

2.1.4.1. PREGUNTAS DE INVESTIGACIÓN

Una manera fundamental de la revisión sistemática es delimitar el problema y expresar con claridad las preguntas de investigación a la hora de la RSL, debido a que lo expuesto permite conocer el estado actual y sintetizar la literatura existente. De acuerdo con Codina (2020) las preguntas de investigación se consideran necesarias, incluso imprescindibles, en todo tipo de investigaciones, tanto las que, además utilizan hipótesis, como aquellas que no las necesitan.

2.1.4.2. PROCESO DE BÚSQUEDA

“La estrategia de búsqueda comprende determinar los términos de búsqueda, los recursos literarios y el proceso de búsqueda (Wen et al., 2012, como se citó en Navarro et al., 2018).

2.1.4.3. CRITERIOS DE INCLUSIÓN Y EXCLUSIÓN

De acuerdo con la Universidad de Navarra (2022) los criterios de inclusión y exclusión establecen los límites de la revisión sistemática (RS). Se determinan después de establecer la pregunta de investigación y antes de realizar la búsqueda. Se pueden utilizar muchos factores diferentes como criterios de inclusión o exclusión. La información sobre los criterios de inclusión y exclusión generalmente se pone como un párrafo o una tabla, dentro de la sección de métodos de la RS (párr. 1)

2.1.4.4. EVALUACIÓN DE CALIDAD

Es deber de cada revisor aplicar de manera exhaustiva, cuidadosa y responsable la lista de chequeo para establecer la calidad metodológica de la revisión y evitar el sesgo de clasificación al subestimar o sobreestimar (Pizarro et al., 2019). Es decir, una evaluación de calidad significa realizar a cabalidad la investigación, encontrando lo bueno y lo malo de la investigación.

2.1.4.5. EXTRACCIÓN Y ANÁLISIS DE DATOS

La extracción de datos es el "proceso mediante el cual, a partir de los estudios primarios, los revisores obtienen la información necesaria para contestar a la pregunta de investigación planteada" (Universidad de Navarra, 2022). El proceso de selección de estudios debe ser explícito para que disminuya el riesgo de sesgos y errores. Se puede utilizar algún software de gestión de referencias como EndNote o Mendeley para recopilar, recolectar y organizar las referencias.

2.2. EFECTUAR UN ANÁLISIS COMPARATIVO DE LAS HERRAMIENTAS DE DEVOPS.

Se procedió a realizar un análisis con todas las herramientas utilizadas por fases, tomando en cuenta varios parámetros de las herramientas, se determinó que herramienta puede ser de mayor beneficio para cada fase de DevOps, optimizando recursos, dando como resultado un ecosistema más comprensivo.

2.3. ELABORAR UNA GUÍA DE DEVOPS REFERENTE AL CICLO DE VIDA DE DESARROLLO DE SOFTWARE.

Este objetivo comprende la etapa preliminar del proyecto, que como toda investigación busca documentar los resultados obtenidos, aquí se elaboró una guía de DevOps referente al ciclo de vida de desarrollo de software, que contiene información sobre las herramientas en cada fase de DevOps, que se obtuvieron en el análisis comparativo.

2.4. SOCIALIZAR LOS RESULTADOS CON LA UDIV DE DESARROLLO COMPUTACIONAL.

La socialización de resultados en los trabajos investigativos, es el apto de dar a conocer a las partes involucradas todo lo concerniente a la creación de algún tipo de documento u otro medio que haya sido estudiado. Como último objetivo se socializaron los resultados con los miembros de la UDIV de Desarrollo Computacional, el cual se realizó mediante la interacción de cada una de las partes, con el propósito de mostrar la guía que fue creada por el autor del trabajo de investigación.

CAPÍTULO III. DESCRIPCIÓN DE LA EXPERIENCIA

3.1. REVISIÓN SISTEMÁTICA DE LA LITERATURA

3.1.1. PREGUNTAS DE INVESTIGACIÓN

Las preguntas de investigación que se realizaron para llevar a cabo este estudio son las siguientes:

- P1. ¿Qué metodologías se utilizan para aplicar DevOps?
- P2. ¿Cuáles prácticas de DevOps son utilizadas en las organizaciones que desarrollan software?
- P3. ¿Cuáles han sido los beneficios al adoptar DevOps en las organizaciones?
- P4. ¿Cuáles han sido los retos al adoptar DevOps en las organizaciones por continente?
- P5. ¿Qué herramientas se emplea en las diversas etapas del ciclo de DevOps?

De acuerdo a la P1, se observaron las metodologías acordes para la aplicación de DevOps; P2 se analizaron las diferentes prácticas que son utilizadas en las organizaciones; P3 se redactaron los diferentes beneficios que tiene DevOps al adoptarlo como una cultura en las organizaciones; seguido en la P4 se establecieron los retos que enfrenta DevOps al adoptarlo en las organizaciones de cada continente; y finalmente en la P5 se definieron las herramientas necesarias para cada etapa del ciclo de DevOps empleada en las organizaciones.

3.1.2. PROCESO DE BUSQUEDA

Se realizó el proceso de búsqueda con base de datos científicos como ACM, IEEE Xplore, ScienceDirect y Springer, haciendo uso de las cadenas de string que hacen parte de la investigación, como por ejemplo DevOps, devops and

Integración continua or Despliegue continuo, DevOps and Methology, practicas DevOps”

Sucesivamente, se ejecutó una investigación avanzada de motores de búsqueda, el cual contenía filtros de los últimos cinco años de creación, es decir desde el 2017 hasta la actualidad. En la siguiente tabla el autor presenta un resumen de la investigación que realizó en concordancia a las cadenas de textos que se mencionaron con anterioridad:

Tabla 1. Resultados de la búsqueda en las bases de datos.

Base de Datos	Cantidad
ACM	953
IEEE	196
ScienceDirect	109
Springer	557
Total	1,815

Fuente: Elaboración propia.

Con base en la tabla 1 se puede connotar las bases de datos como es ACM con una cantidad de 953 artículos, seguido de IEEE Xplore con una cantidad de 196, sucesivamente ScienceDirect con una cantidad de 109 y finalmente Springer con una cantidad de 557, que da como resultado un total de 1,815 artículos científicos.

3.1.3. CRITERIOS DE INCLUSIÓN Y EXCLUSIÓN

En concordancia a ello, el autor de este trabajo investigativo inició este proceso, suprimiendo aquellos documentos que estaban duplicados en diferentes bases de datos, seguidamente se aplicaron criterios de inclusión y exclusión a los títulos y resúmenes, posteriormente para seguir refinando los artículos, el autor realizó una lectura rápida de aquellos para descartar artículos que no poseían información relevante al tema de investigación; y después de este proceso de selección se obtuvieron un total de 119 artículos.

Tabla 2. Criterios de inclusión y exclusión.

Criterios de inclusión	Criterios de exclusión
Artículos de los últimos 5 años.	Artículos que contengan información similar a la ya consultada.
Títulos que contengan palabras como DevOps, prácticas DevOps, DevOps and Methology, DevOps and Integración continua.	Información de tesis, blogs y repositorios.
Resúmenes que contengan características, conceptos, retos, herramientas y métodos de DevOps.	
Idioma Inglés y español	
Información de DevOps en las organizaciones	

Fuente: Elaboración propia.

Para facilitar la obtención de resultados, el autor optó por aplicar los criterios de tabla 2 usando la herramienta Excel, en el cual, se logró eliminar una gran cantidad de artículos que serían irrelevantes para la investigación, como primer paso se descargaron los resultados de las diferentes bases de datos en archivos CSV, posteriormente se convirtieron los archivos a XLSX.

Tabla 3. Resultados de base de datos, aplicando los criterios de inclusión y exclusión.

BASE DE DATOS	CANTIDAD
ACM	76
IEEE XPLORE	10
SPRINGER	23
SCIENCE DIRECT	10
TOTAL	119

Fuente: Elaboración propia.

En la tabla 3 se procedió a la aplicación de los criterios antes expuestos, dando como resultado en ACM: 76, IEEE Xplore: 10, Springer: 23 y Science Direct: 10 un total de 119 artículos científicos.

Tabla 4. Filtro por título.

BASE DE DATOS	CANTIDAD
ACM	7
IEEE XPLORE	5
SPRINGER	11
SCIENCE DIRECT	4
TOTAL	27

Fuente: Elaboración propia.

Seguidamente, en la tabla 4 se aplicó un filtro por título de los resultados de la base de datos en donde se aplicaron los criterios de inclusión y exclusión, dando como resultado que en ACM se encontraron 7, seguido en IEEE Xplore 5, de igual manera en Springer 11 y finalmente en Sciencedirect 4, llegando a un total de 27 artículos científicos.

Tabla 5. Filtro por resumen.

BASE DE DATOS	CANTIDAD
ACM	8
IEEE XPLORE	6
SPRINGER	12
SCIENCE DIRECT	8
TOTAL	34

Fuente: Elaboración propia.

De igual manera, en la tabla 5 se aplicó un filtro por resúmenes de los resultados de la base de datos en donde se aplicaron los criterios de inclusión y exclusión para obtener información más detallada, dando como resultado que la base de datos ACM obtuvo 8 artículos científicos, la IEEE Xplore 6, el Springer 12 y Science Direct 8, con un total de 34 artículos filtrados por resumen.

Tabla 6. Resultados obtenidos por medio de una lectura rápida.

BASE DE DATOS	CANTIDAD
ACM	6
SCIENCEDIRECT	4
IEEE XPLORE	6
SPRINGER	7
TOTAL	23

Fuente: Elaboración propia.

Como aún se contaba con una gran cantidad de artículos se realizó una lectura rápida, considerando las preguntas de investigación propuestas, dejando así artículos con mayor relevancia. Los resultados obtenidos fueron ACM: 6, Sciencedirect: 4, IEEE Xplore: 6, y Springer con 7 artículos, con un total de 23 artículos como se detalla en la tabla 6.

A continuación, como se expresa en el gráfico 1, se muestran los porcentajes de las bases de datos filtradas:

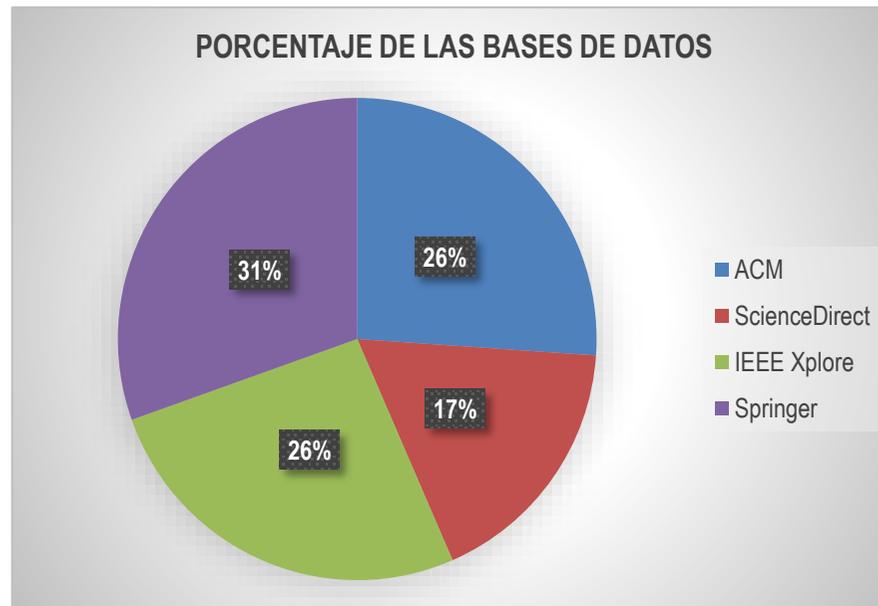


Gráfico 1. Resultados de las bases de datos en porcentaje.
Fuente: Elaboración propia.

A continuación, se detallan los artículos seleccionados ordenados por año.

Tabla 7. Artículos encontrados en las bases de dato

REF	TITULO	AUTORES	AÑO
1	DevOps: IT-Entwicklung im Industrie 4.0-Zeitalter	(Thomas et al)	2017
2	Fünf kritische Erfolgsfaktoren für eine erfolgreiche DevOps Transformation	(Lichtenberger)	2017
3	Modeling and measuring attributes influencing DevOps implementation in an enterprise using structural equation modeling	(Gupta et al)	2017
4	Praxisbasierte Capability-Modelle für DevOps-Einsätze in Unternehmen	(Juner & Benlian)	2017
5	The emergence of a new EAM function for a BizDevOps-based fast IT	(Drews et al)	2017
6	Building a collaborative culture: A grounded theory of well succeeded devops adoption in practice	(Luz et al)	2018
7	DevOps capabilities, practices, and challenges: Insights from a case study	(Senapathi et al)	2018
8	DevOps in practice - An exploratory case study	(Díaz et al)	2018
9	DevOps metrics	(Forsgren et al)	2018
10	Adopting DevOps in the real world: A theory, a model, and a case study	(Luz et al)	2019
11	Adoption Issues in DevOps from the Perspective of Continuous Delivery Pipeline	(Zulfahmi et al)	2019
12	The Practice: The DevOps Phenomenon	(Wiedemann et al)	2019
13	DevOps in practice: A multiple case study of five companies	(Lwakatare et al)	2019
14	DevOps in practice for education management information system at ECNU	(Yang et al)	2020
15	From Agile to DevOps: Smart Skills and Collaborations	(Hemon et al)	2020
16	Identification and prioritization of DevOps success factors using fuzzy-AHP approach	(Akbar et al)	2020
17	Prioritization Based Taxonomy of DevOps Challenges Using Fuzzy AHP Analysis	(Akbar et al)	2020
18	Anomaly Detection in DevOps Toolchain	(Antonio et al)	2020
19	Quality-Aware DevOps Research: Where Do We Stand?	(Alnafessah et al)	2021
20	Why are many businesses instilling a DevOps culture into their organization?	(Díaz et al)	2021
21	Toolchain Development for Midcore Scale Game Products through DevOps and CI/CD Approach	(Thobari et al)	2021
22	Critical Challenges to Adopt DevOps Culture in Software Organizations: A Systematic Review	(Khan et al)	2022
23	Toward Successful DevOps : A Decision-Making Framework	(Akbar et al)	2022

Fuente. Autoría propia

3.1.4. EVALUACIÓN DE CALIDAD

Este proceso se realizó leyendo de forma detallada los artículos que se encuentran en la tabla 8 para así obtener los datos más relevantes. Esta evaluación se realizó conjuntamente con los criterios de las preguntas de investigación que fueron:

- P1. ¿Qué metodologías se utilizan para aplicar DevOps?
- P2. ¿Cuáles prácticas de DevOps son utilizadas en las organizaciones que desarrollan software?
- P3. ¿Cuáles han sido los beneficios al adoptar DevOps en las organizaciones?
- P4. ¿Cuáles han sido los retos al adoptar DevOps en las organizaciones?
- P5. ¿Qué herramientas se emplea en las diversas etapas del ciclo de DevOps?

Tabla 8. Evaluación de la calidad de las preguntas de investigación.

Ref	TITULO	P1	P2	P3	P4	P5
1	DevOps IT-Entwicklung im Industrie 4.0-Zeitaler	X				
2	Fünf kritische Erfolgsfaktoren für eine erfolgreiche DevOps Transformation	X	X			
3	Modeling and measuring attributes influencing DevOps implementation in an enterprise using structural equation modeling	X	X			
4	Praxisbasierte Capability-Modelle für DevOps-Einsätze in Unternehmen		X			
5	The emergence of a new EAM function for a BizDevOps-based fast IT					X
6	Building a collaborative culture: A grounded theory of well succeeded devops adoption in practice	X				
7	DevOps capabilities, practices, and challenges: insights from a case study			X	X	
8	DevOps in practice - An exploratory case study			X		
9	DevOps metrics			X		
10	Adopting DevOps in the real world: A theory, a model, and a case study		X			
11	Adoption Issues in DevOps from the Perspective of Continuous Delivery Pipeline	X		X	X	
12	The Practice: The DevOps Phenomenon		X		X	
13	DevOps in practice: A multiple case study of five companies	X	X			X
14	DevOps in practice for education management information system at ECNU					X
15	From Agile to DevOps: Smart Skills and Collaborations	X				
16	Identification and prioritization of DevOps success factors using fuzzy-AHP approach			X		
17	Prioritization Based Taxonomy of DevOps Challenges Using Fuzzy AHP Analysis				X	
18	Anomaly Detection in DevOps Toolchain					X
19	Quality-Aware DevOps Research: Where Do We Stand?					X
20	Why are many businesses instilling a DevOps culture into their organization?			X		
21	Toolchain Development for Midsize Scale Game Products through DevOps and CI/CD Approach					X
22	Critical Challenges to Adopt DevOps Culture in Software Organizations: A Systematic Review				X	
23	Toward Successful DevOps: A Decision-Making Framework		X			

Fuente. Autoría propia.

Con el análisis de cada uno de los artículos y haciendo uso del software Atlas.ti, se fueron seleccionando los datos que daban respuesta a las preguntas de investigación. El resultado del análisis se puede apreciar en la tabla 8.

3.1.5. EXTRACCIÓN Y ANÁLISIS DE DATOS

El análisis de datos se realizó con las lecturas detalladas de los artículos seleccionados, haciendo uso de Atlas.ti. A continuación, se detallan los resultados de las interrogantes con base la tabla 10.

- P1. ¿Qué metodologías se utilizan para aplicar DevOps?

Tabla 9. Resultado de la P1.

METODOLOGÍAS	REFERENCIAS	TOTAL
Lean	REF[13], [3], [6], [11], [2]	5
Scrum	REF[13], [1], [15], [2]	4
Kanban	REF[13]	1

Fuente. Autoría propia.

Como respuesta a la P1 se puede concluir que las metodologías ágiles que más se utilizan en las organizaciones son Lean, Scrum y Kanban, siendo la metodología Lean la más mencionada, debido a que esta metodología busca la mejora continua y optimizar a las personas, los recursos, el esfuerzo y la energía de una organización.

- P2. ¿Cuáles prácticas de DevOps son utilizadas en las organizaciones que desarrollan software?

Tabla 10. Resultado de la P2.

PRÁCTICAS	REFERENCIAS	TOTAL
Desarrollo continuo	REF[10], [12]	2
Pruebas continuas	REF[10], [12], [2], [23], [3], [4]	6
Integración continua CI	REF[10], [12], [13], [2], [23], [3], [4]	7
Entrega Continua	REF[10], [12], [2], [3]	4
Puesta en marcha continua CD	REF[10], [3], [4],	3
Supervisión continua	REF[10], [12], [13], [23], [3]	5

Infraestructura como código	REF[10], [13], [2], [4]	4
Gestión de la configuración	REF[12], [13]	2
Inicio de Sesión	REF[12]	1

Fuente. Autoría propia.

Con base en los artículos encontrados para el análisis de las prácticas más usadas en las organizaciones, el autor determinó las prácticas más usadas, de acuerdo a la tabla 10 son la realización de pruebas continuas e integración continúa CI.

- P3. ¿Cuáles han sido los beneficios al adoptar DevOps en las organizaciones?

Tabla 11. Respuesta P3.

BENEFICIOS	REFERENCIAS	TOTAL
Rápida velocidad en los lanzamientos	Ref [8], [7], [9], [11], [20]	5
Un rápido tiempo de respuesta a demandas de clientes	Ref [8], [7], [11], [20]	4
La no interrupción y respuesta rápida de la producción	Ref [8], [11], [16]	3
Mayor autonomía	Ref [7], [16], [20]	3
Aprender nuevas tecnologías	Ref [7]	1
Mejora la calidad de las aplicaciones	Ref [7], [11], [20]	3
Los equipos están más felices y comprometidos	Ref [7], [20]	2
Visibilidad continua	Ref [9]	1
Enfoque amigable para los negocios	Ref [11], [20]	2
Ahorro en el costo de desarrollo	Ref [11], [20]	2
Equipos capacitados establecidos	Ref [16]	1

Fuente. Autoría propia.

Los beneficios más relevantes como expresa la tabla 12 en las organizaciones son la rápida velocidad en los lanzamientos y un rápido tiempo de respuestas a demandas de clientes.

- P4. ¿Cuáles han sido los retos al adoptar DevOps en las organizaciones por continente?

Tabla 12. Respuesta a P4

RETOS	REFERENCIAS	TOTAL
Cultura de equipo	Ref [11], [22], [17]	3
Uso de herramientas	Ref [11], [17]	2
No hay un proceso y una guía clara	Ref [11], [7]	2
Control de recursos	Ref [11]	1
Las organizaciones y los productos de software varían en madurez e implementación	Ref [12], [7], [17]	3
L contratación de nuevo personal con la capacitación y las habilidades técnicas adecuadas	Ref [7], [22], [17]	3
Estructura organizativa y distribución geográfica	Ref [7], [22]	2
Falta de comunicación y colaboración	Ref [22], [17]	2
Problemas de confianza y seguridad	Ref [22], [17]	2
Des interminación de roles dentro del equipo	Ref [17]	1
Falta de retroalimentación y priorización de errores	Ref [17]	1

Fuente. Autoría propia

En concordancia a la tabla 13, los retos que se presentan al adoptar DevOps, han sido las organizaciones y los productos de software que varían en madurez e implementación y de igual manera, la contratación de nuevo personal con la capacitación y las habilidades técnicas adecuadas.

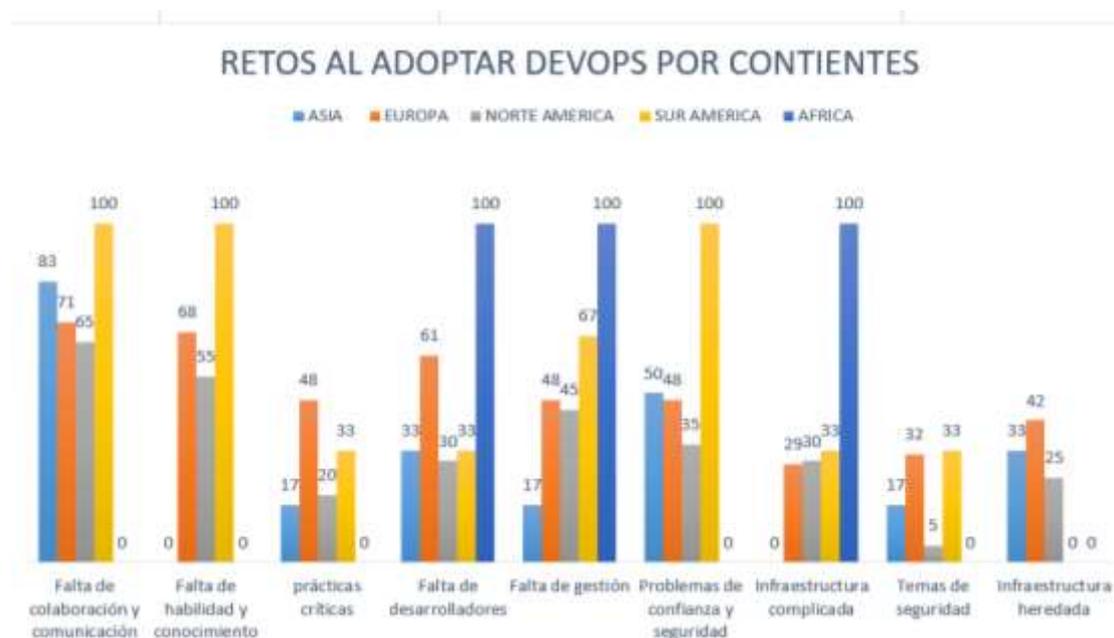


Gráfico 2. Retos al adoptar DevOps por continentes.

Fuente: Elaboración propia.

De acuerdo al gráfico 2, se muestran los retos que han sufrido las organizaciones para adoptar DevOps, dando como resultado que la falta de colaboración y comunicación ha desencadenado una trágica tarea al momento de adaptarse a la cultura de DevOps.

- P5. ¿Qué herramientas se emplea en las diversas etapas del ciclo de DevOps?

Tabla 13. Respuesta a P5

Herramientas	Referencia	Utilización
Git	Ref [5], [13], [14]	Planeación, código
GitLab	Ref [21], [14]	Código
Docker	Ref [18], [21], [5], [13], [14]	Despliegue, Construcción
Kubernetes	Ref [14]	Operación
Jenkins	Ref [18], [5], [13], [14]	Lanzamiento
GitHub	Ref [19], [18]	Código
Grafana	Ref [19]	Monitoreo
Prometheus	Ref [19]	Prueba
InfluxDB	Ref [19]	Monitoreo
CODE CLIMATE	Ref [18]	Prueba
Jira	Ref [18]	Planeación
Node Package Manager (NPM)	Ref [18]	Construcción
WebLOAD	Ref [5]	Prueba
Selenium	Ref [5]	Prueba
Ansible	Ref [5]	Operación
Swagger	Ref [5]	Construcción
Bitbucker	Ref [13]	Código
Puppet	Ref [13]	Operación
SonarQube	Ref [13], [14]	Prueba

Fuente. Autoría propia.

De acuerdo con la tabla 14, existen múltiples herramientas utilizadas en las etapas del ciclo de DevOps, las que se usan para la **planeación** es Git y Jira; para **codificación**: Git, GitLab, GitHub y Bitbucker; en **operación** se usa kubernetes, ansible y puppet; en **construcción** es Docker y, npm; en **lanzamiento** sólo se aplica Jenkins; en **monitoreo** se emplea Grafana y

InfluxDB; en **pruebas** se utiliza Prometheus, CODE CLIMATE, WebLOAD, Selenium y SonarQube; y finalmente en **despliegue** se practica docker.

3.2. EFECTUAR UN ANÁLISIS COMPARATIVO DE LAS HERRAMIENTAS DE DEVOPS

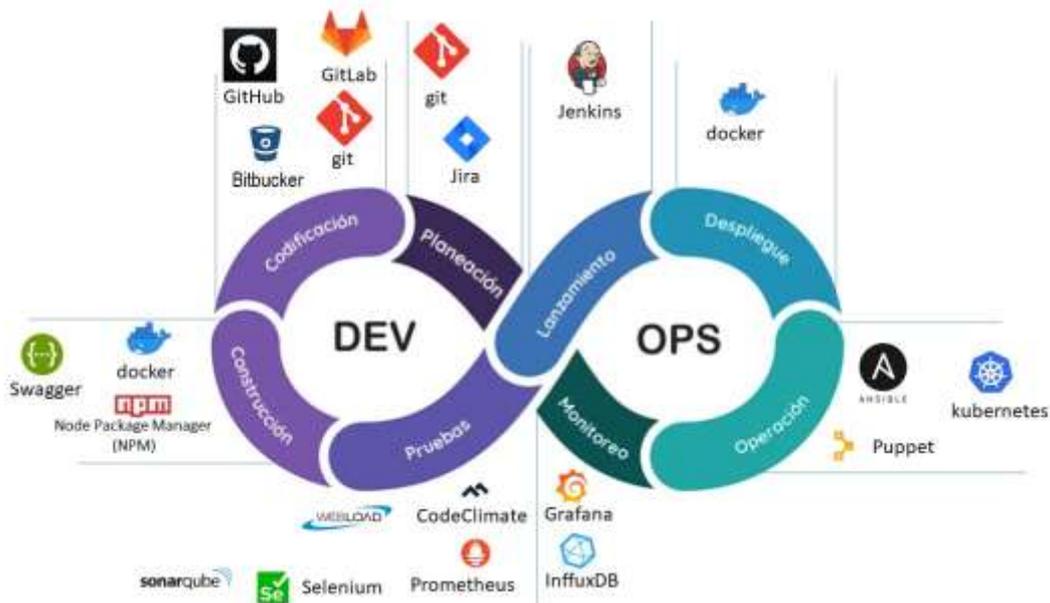


Figura 1. Ecosistema DevOps.
Fuente. Autoría propia.

Tomando como base el apartado de la pregunta 5 que indica cuales son las todas herramientas que se emplean en DevOps, se las categorizó entorno al ecosistema DevOps y quedan de la siguiente forma repartidas como se visualiza en la figura 1, para lo cual se va a realizar un análisis de cada una de estas herramientas con el propósito de evaluar cuál de todas ellas tienen mejor rendimiento en cada fase del ecosistema.

En este objetivo se describen las características principales de las herramientas del ciclo de vida DevOps con el fin de realizar un análisis de estas, el cual despliega sus respectivas fases como son: planeación, codificación, lanzamiento, despliegue, operación, monitoreo, pruebas y construcción.

3.2.1. PLANEACIÓN

En esta fase se define las exigencias y valores empresariales, se realiza un rastreo de los problemas conocidos y se lleva a cabo una gestión en los proyectos, para esta fase la herramienta que más se destacó dentro de la RLS es Git y Jira. Es de fácil acceso, su tamaño de descarga es de 47.1 MB lo cual hace que su descarga sea rápida, también se encuentra disponible para los sistemas operativos Windows, Linux y macOS.

Jira es una herramienta para la gestión de proyectos, procesos y acciones de cualquier tipo que puedan tener respectivos flujos de trabajos, permite definir filtros, se puede definir diferentes tipos de tareas, se puede definir el flujo de trabajo, función a las necesidades, se pueden definir esquemas de seguridad y de permisos.

3.2.2. CODIFICACIÓN

En la codificación se involucra el diseño de software y la creación del código, en esta fase como se muestra en la RLS las herramientas que se utilizan son Git, GitHub, GitLab y Bitbucket. Asimismo, es un método de control de traducciones segmentado, que ayuda a los programadores a descargar un software y subir la versión que han cambiado, tras realizar una modificación.

Es de fácil acceso, su tamaño de descarga es de 47.1 MB lo cual hace que su descarga sea rápida, también se encuentra disponible para los sistemas operativos Windows, Linux y macOS.

GitHub es el sitio web más usado para almacenar repositorios Git, alberga más de 50 millones de proyectos de código abierto, entre sus beneficios están el seguimiento de errores, la búsqueda rápida, el contar con una comunidad, se puede compartir y descargar un código fuente. Entre sus desventajas se pudo encontrar que el servicio no es completamente gratuito para acceder a todas las funciones de GitHub y tiene limitaciones en el tamaño de los archivos.

GitLab es un servicio que está desarrollado en la base de control de versiones de Git, entre sus beneficios está que es gratis, opera bajo una licencia de código abierto, seguimiento de errores y edición de código basado en la web, tiene como desventajas una interfaz considerablemente lenta y tiene problemas con los repositorios.

Y finalmente, Bitbucket también ofrece un servicio muy similar a GitHub, tiene beneficios como los repositorios privados, tiene una búsqueda consistente de código, también viene con Trello para realizar una búsqueda de problemas de manera gratuita, como desventaja es que no es de código abierto, pero admite proyectos de este tipo.

Una vez que se realizó el análisis de los repositorios, al autor recomienda utilizar GitHub o Bitbucket, ya que ambos son similares en cuanto a características, el autor tomará GitHub como referencia para la fase de codificación ya que es una herramienta con una interfaz cómoda y también la más usada para almacenar repositorios de Git.

3.2.3. LANZAMIENTO

En esta fase, se efectúan los ensayos. El crecimiento continuo sólo es viable debido a la composición y a los ensayos duraderos. En esta fase se encontró a Jenkins, la cual es una herramienta de automatización de tareas en procesos de integración continua, tiene como características que cuenta con más de 1400 plugins, puede orquestar cualquier tipo de proceso, es utilizada para ejecutar tareas manuales, automáticas o periódicas, fácil de usar, dashboard para visualizar el estado de tareas, monitoreo de procesos y alertas ante problemas en algún proceso

3.2.4. DESPLIEGUE

En esta etapa, se gestiona el software durante su producción. En cualquier momento que se realice un cambio, no debe afectar el funcionamiento del sitio web de alto tráfico. Para esta etapa se puede utilizar Docker el cual es un proyecto de código abierto con el que se pueden crear fácilmente contenedores, entre sus características se puede encontrar autogestión de los contenedores, fiabilidad, capacidad para desplegar multitud de contenedores en un mismo equipo físico, puesta en marcha de los servicios de manera rápida y puede ejecutar una amplia gama de aplicaciones.

3.2.5. OPERACIÓN

La fase de operación de DevOps se encarga del comportamiento impropio del sistema o de los errores que se encuentran en la producción. En este apartado se pudieron encontrar tres herramientas esenciales para el desenlace de esta fase tales como, Ansible, Kubernetes y Puppet las cuales se detallan a continuación:

Ansible es una herramienta open source de DevOps, en la cual se pueden automatizar todas aquellas tareas de administración de servidores que un sistema debe ejecutar día a día, esto trae como beneficio el ahorro de tiempo, efectividad y simplificación de tareas complejas, cuenta con herramientas de depuración y validación, con poco conocimiento de programación se puede operar.

Se dice que Kubernetes, es una plataforma de código abierto que administra cargas, entre sus características se tiene que organizar los contenedores en diversos host o archivos, realiza un buen uso del hardware, automatiza y controla las actualizaciones, realiza comprobaciones de estado y se encarga de reiniciar los contenedores que fallan, así también de eliminar los contenedores que no responden.

La herramienta Puppet, es un instrumento de gestión esencial y utilizado en la configuración de código abierto, entre sus características se tiene que impulsar tareas de automatización, prueba código, tiene un lenguaje declarativo, una librería y adopta enfoques basado en modelos de corrección de desviaciones escalable.

Finalmente, después de revisar las tres herramientas, el autor tomó de referencia Ansible como instrumento a utilizar para la fase de operación, dado que cuenta con las características necesarias para cumplir con el objetivo de esta fase de operaciones al ser una plataforma libre para administrar ordenadores y es declarado herramienta de orquestación.

3.2.6. MONITOREO

En la fase del monitoreo se recolecta la información de los problemas encontrados en la versión de software que se localiza en producción; es decir en esta fase el equipo de operación monitorea y administra el funcionamiento de las aplicaciones y lo que conlleva su infraestructura. En esta se pudieron determinar tres herramientas como son: Grafana, InfluxDB y Slack.

Grafana es un herramienta que sirve para efectuar análisis de datos, monitorean aplicaciones y recursos, entre sus características encontramos que se conecta a todas las fuentes de datos posibles, ayuda a rastrear el comportamiento del usuario y de la aplicación, existen paneles donde tiene una gran variedad de opciones para la visualización de mapas y cuadros para poder estudiar datos, además se pueden configurar alertas.

Influxdb, es un software de código abierto diseñado para el almacenamiento rápido y de alta disponibilidad y la recuperación de datos de series temporales, también evalúa datos de sensores o protocolos. InfluxDB también reconoce que

las preferencias del esquema pueden cambiar con el tiempo, asimismo no es necesario definir esquemas por adelantado.

Como conclusión en la fase de monitoreo, el autor una vez que leyó las características de ambas aplicaciones determinó que la herramienta que mejor puede definirse en la fase de monitoreo es Grafana, por lo cual se la utilizó como guía en esta fase.

3.2.7. PRUEBAS

Esta fase incluye la realización de pruebas continuas (manuales o automatizadas) para garantizar la calidad de la programación. En esta etapa se encontró varias herramientas como, codeClimate, AWS kinesis, WEBLOAD, Prometheus, Selenium y sonarQube.

WebLOAD es una herramienta de prueba de rendimiento que ayuda a las empresas a plantear y administrar entornos de pruebas personalizados para varias aplicaciones, sitios web, entre otros, también permite configurar tableros en tiempo real utilizando gráficos y métricas para obtener información sobre los resultados de las pruebas, comparar múltiples sesiones e identificar problemas a través del análisis. Cuenta con una versión gratuita, disponible solo para Estados Unidos y su único idioma es el inglés.

SonarQube, es una aplicación que sirve para la inspección de la calidad del código a través de los instrumentales de estudios estáticos de código fuente, admite los lenguajes de programación más populares, realiza revisiones automáticas, facilita informes, se integra con toda la cadena de herramientas de DevOps e igual que muchas otras herramientas cuenta con una versión gratuita.

Selenium, es una herramienta para realizar pruebas de sistema, a nivel completo, se puede utilizar en todos los buscadores, sirve para los principales sistemas operativos, de igual manera es de código abierto y portátil, cuenta con

combinación de herramientas, reduce el tiempo de ejecución en las pruebas, usa menos recursos, admite múltiples sistemas operativos y se puede usar con varios navegadores.

CodeClimate es una empresa privada de software que ofrece un conjunto de herramientas de análisis de programas estáticos para ayudar a los desarrolladores de software a mejorar la calidad de su código fuente. CodeClimate es una herramienta que examina el código investigando duplicidad y problemas comunes. De esta forma te puede marcar aquellos puntos que se pueden mejorar.

Prometheus es considerado un framework de recolección de datos de estructura no definida. Esto permite definir sintaxis para su evaluación y de esa forma almacenar datos solo en caso de eventos de cambio, no cuenta con algún tipo de informe, por lo que se tiene que programar usando su API para extraer datos, Prometheus en su página web recomienda usar otras herramientas para los desarrolladores que desean monitorizar aplicaciones o generar gráficos.

El autor basándose en el análisis de la fase de prueba pudo deducir que la herramienta con mejores características para esta fase es Selenium, ya que puede realizar pruebas de sistema a nivel completo.

3.2.8. CONSTRUCCIÓN

En esta fase es donde se construye, ya sea diseñando infraestructura, automatizando procesos, escribiendo código, definiendo pruebas o estableciendo la seguridad. En esta fase se encontraron tres herramientas, como Swagger, NPM o Node Package manager y Docker, en las que se detalla a continuación:

La herramienta de Swagger es un framework que está conformado por un conjunto de herramientas de código abierto, que sirven para generar documentación de APIs, diseñar, construir, documentar, también incluye

documentación automatizada, generación de código y generación de casos de prueba.

NPM o Node Package Manager, es un gestor de paquetes para los proyectos de Node.js, NPM permite utilizar fácilmente el código escrito por otros sin necesidad de escribirlo durante el desarrollo, entre sus ventajas se tiene que, permite instalar librerías, acceder a proyectos Node.js seguros para el desarrollo, acelerar la fase de desarrollo y también puede utilizar una gran variedad de herramientas de manera gratuita.

Docker es un proyecto de código abierto con el que se pueden crear fácilmente contenedores, entre sus características se puede encontrar autogestión de los contenedores, fiabilidad, capacidad para desplegar multitud de contenedores en un mismo equipo físico, puesta en marcha de los servicios de manera rápida y se puede ejecutar una amplia gama de aplicaciones.

Una vez analizadas las herramientas de la fase de construcción se pudo notar que la herramienta Docker es la indicada, ya que también se usa en la fase de despliegue, y por ende al usarlo en esta fase se puede ahorrar recursos y tiempo, ya que no se usaría otra herramienta para esta fase sino una que se aplica en otra.

Una vez realizado el análisis de las herramientas que arrojó como resultado la pregunta 5 de la RLS se pudo obtener la figura 2, la cual es un ecosistema con las herramientas que son las representantes de cada fase del ciclo de vida DevOps. Como GitHub y Git para Codificación, Jira para planeación, Jenkins para lanzamiento, Docker para despliegue, Ansible para operación, Grafana para

monitoreo, Selenium para pruebas y Docker para construcción.

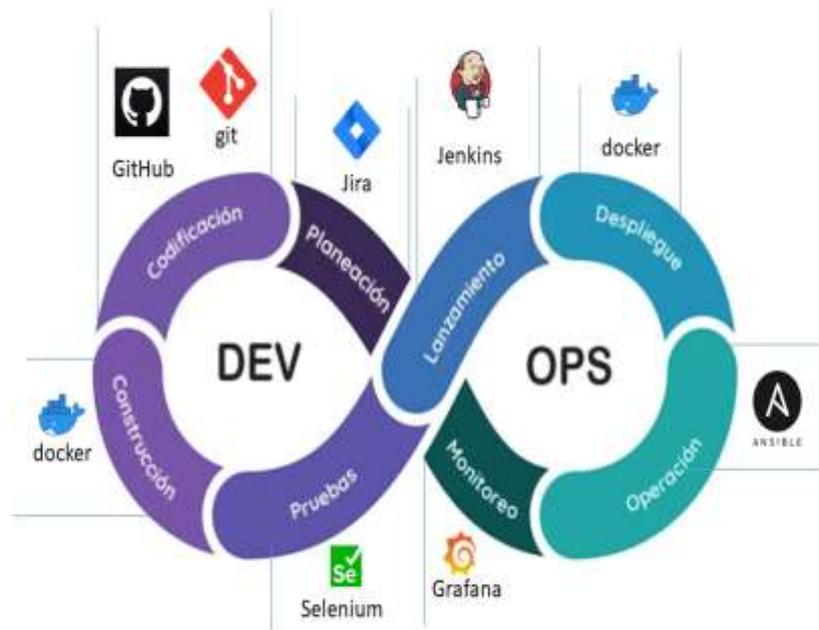


Figura 2. Resultado del análisis.
Fuente. Autoría propia.

3.3. ELABORAR UNA GUÍA DE DEVOPS REFERENTE AL CICLO DE VIDA DE DESARROLLO DE SOFTWARE.

La elaboración de la guía se creó con el aporte de los resultados de cada uno de los objetivos que tributan para el diseño de la guía de DevOps. La guía comprende paso a paso cada una de las fases del ciclo de vida DevOps ubicado en el ecosistema antes expuesto (ver Anexo 3). No obstante, esta guía servirá de insumo para que los estudiantes de la Carrera de Computación puedan levantar este ecosistema y desarrollen sus proyectos de titulación o proyectos de cualquier índole, asimismo la UDIV como organización puede usar la guía para no cometer los errores que las otras organizaciones cometieron al momento de implementar DevOps.

3.4. SOCIALIZAR LOS RESULTADOS CON LA UDIV DE DESARROLLO COMPUTACIONAL.

Una vez desarrollada la guía se procedió a concretar una cita con el encargado de la Unidad de Desarrollo Computacional (ver Anexo 2) para ejecutar el cuarto objetivo, el cual tuvo como propósito presentar la guía a los miembros de la UDIV de Desarrollo Computacional, con el fin de que se puedan acoger al ecosistema junto a las herramientas que se encuentran en las fases del ciclo de vida DevOps.

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- La revisión sistemática de la literatura permitió obtener resultados significativos que ayudaron a dar respuestas a las preguntas de investigación, mismas que permitieron determinar las herramientas utilizadas en las diversas etapas del ciclo de vida de DevOps.
- El análisis comparativo permitió al autor presentar un ecosistema de trabajo y así seleccionar las herramientas más idóneas según sus propiedades, ventajas y desventajas, y así de esta manera saber cuál es la que más se destaca de cada grupo de herramientas en las diferentes etapas del ciclo de vida DevOps y poder crear un ecosistema adecuado a las tecnologías de desarrollo actuales.
- La socialización de la guía elaborada permitió dar a conocer a los integrantes de la UDIV de Desarrollo Computacional, paso a paso el proceso de instalación de las herramientas más significativas de DevOps producto de esta investigación, con el propósito de que sirva de insumo para el desarrollo profesional de los estudiantes de la Carrera.

4.2. RECOMENDACIONES

- Al momento de realizar una investigación cualitativa, se recomienda el uso de una revisión sistemática de la literatura para obtener resultados relevantes que ayuden a dar respuesta a las diferentes preguntas de investigación.
- Al momento de realizar un análisis comparativo se recomienda tener una lista previa a la investigación con los aspectos más importantes de cada herramienta, de esta manera se agiliza la búsqueda de información y se puede comparar los resultados de una forma más sencilla.
- Se recomienda a los miembros de la UDIV de Desarrollo Computacional y a los estudiantes revisar la guía de DevOps para conocer paso a paso el proceso de instalación de las herramientas más destacadas y no cometan los mismos errores que las otras organizaciones.

BIBLIOGRAFÍA

- Akbar, M; Mahmood, M y Shafiq, E. (2020). Identification and prioritization of DevOps success factors using fuzzy-AHP approach. Soft Computing. <https://sci-hub.se/https://doi.org/10.1007/s00500-020-05150-w>
- Centro de Recursos It User. 2020). La importancia de DevOps en el lugar de trabajo. <https://discoverthenew.ituser.es/devops/2020/08/la-importancia-de-devops-en-el-lugar-de-trabajo>
- Claranet, A. (s.f.). DevOps: qué es y cómo lo aplicamos. <https://www.claranet.es/devops-que-es-y-como-lo-aplicamos>
- Codina, L. (2020). Preguntas de investigación en tesis doctorales y trabajos académicos. <https://www.lluiscodina.com/preguntas-de-investigacion-tesis-doctorales/>
- Consejo de Educación Superior. 2016). República del Ecuador Consejo de Educación Superior.
- Creative Commons, (2020). DevOps Now and Then. Raju Singh. Articleinfo
- De la Torre, C; De la Torre, M y De la Torre, A. (2017). DevOps (Development & Operation). <chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.scprogress.com/NOTICIAS/e-commerce5-20170428.pdf>
- Díaz, J; Almaraz, R y Pérez, J. (2018). DevOps in Practice – An Exploratory Case Study. In Proceedings of ACM XP '18 Companion. ACM, New York.
- ESPAM MFL (2018). Resolución 012-2018.
- ESPAM MFL (Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López). (2016). Plan de vinculación con la colectividad.
- ESPAM MFL (Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López). (2021). Carrera de Computación. <http://www.espam.edu.ec/web/oferta/grado/computacion.aspx>

- Guerrero, N. (2021). La importancia de la opción de DevOps en los equipos.
<https://www.programaenlinea.net/la-importancia-de-la-adopcion-de-devops-en-los-equipos/>
- Gupta, V; Kapur, P y Kumar, D. (2017). Modeling and measuring attributes influencing DevOps implementation in an enterprise using structural equation modeling. DOI: 10.1016/j.infsof.2017.07.010
- Lwakatare, L; Kilamo, T y Karvonen, E. (2019). DevOps in practice: A multiple case study of five companies
- Mamani, Z; Del Pino, L y Gonzales, J. (2020). Arquitectura basada en Microservicios y DevOps para una ingeniería de software continúa. Industrial Data, 23(2) 141-149. ISSN: 1560- 9146.
<https://www.redalyc.org/articulo.oa?id=81665362014>
- Microsoft. (2021). ¿Qué es DevOps? <https://azure.microsoft.com/es-es/overview/what-is-devops/#faq>
- Mike, C. (2021). DevOps: ¿qué es? Definición, ventajas, cursos.
<https://datascientest.com/es/devops-que-es-definicion-ventajas-cursos>
- Pizarro, A; Carvajal, S y Buitrago, A. (2019). ¿Cómo evaluar la calidad metodológica de las revisiones sistemáticas a través de la herramienta AMSTAR? [ARCHIVO PDF] Chrome-extension://efaidnbmnnnibpcajpcgclefindmkaj/http://www.scielo.org.co/pdf/rca/v49n1/es_2256-2087-rca-49-01-e501.pdf
- Red Hat. (2018). El concepto de DevOps.
<https://www.redhat.com/es/topics/devops#:~:text=DevOps%20es%20un%20modo%20de,constante%20de%20servicios%20de%20TI.>
- Ruiz, J. (2018). ¡Necesito un DevOps! https://www.linkedin.com/pulse/necesito-un-devops-juan-carlos-ruiz?trk=read_related_article-card_title
- Senapathi, M; Buchan, J y Osman, H. (2018). DevOps Capabilities, Practices, and Challenges: Insights from a Case Study

Universidad de Guadalajara. 2018). El papel de la Universidad en la sociedad actual. <http://www.cucsh.udg.mx/noticia/el-papel-de-la-universidad-en-la-sociedad-actual>

Universidad del Ecuador. 2022). Escuela Superior Politécnica Agropecuaria de Manabí <https://www.universidades.com.ec/escuela-superior-politecnica-agropecuaria-de-manabi>

Universidad de Navarra. (2022). Revisiones sistemáticas: Extracción de datos de los estudios seleccionados. https://biblioguias.unav.edu/revisionessistematicas/extraccion_de_datos_de_los_estudios_seleccionados#:~:text=La%20extracci%C3%B3n%20de%20datos%20es,la%20pregunta%20de%20investigaci%C3%B3n%20planteada%22

ANEXOS

ANEXO 1. CREACIÓN DE LA GUÍA DEVOPS



GUÍA DE DEVOPS

HERRAMIENTAS DE DEVOPS



SEPTIEMBRE 2022
MOREIRA ÁLAVA ÁNGEL YOEL

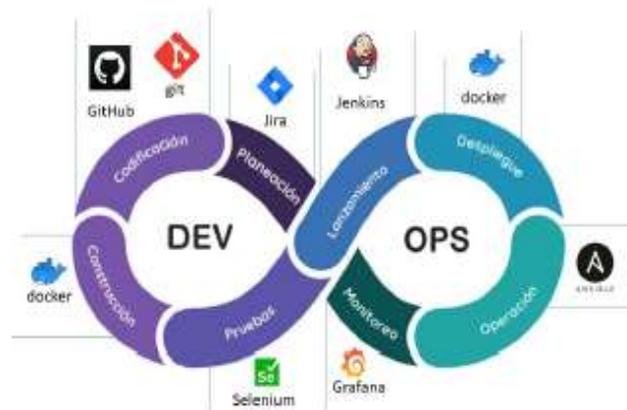
¿Qué es DevOps?

DevOps es un marco de trabajo y una cultura que evoluciona constantemente, realiza una mejor gestión con los proyectos de software en menos tiempo y revisa las funciones de los productos para los clientes de manera rápida.

DevOps genera una comunicación fluida entre los equipos de desarrollo de aplicaciones (Dev) y sus equivalentes en operaciones tecnológicas (Ops). Esta relación se extiende a las fases del ciclo de vida DevOps, como son: Planificación, codificación, lanzamientos, despliegue, operación, monitoreo, pruebas y construcción.

Ecosistema Devops

En DevOps se incorporan un conjunto de herramientas que se adaptan a los métodos. El objetivo de estas herramientas es tratar de optimizar, acortar y automatizar las diversas etapas del flujo de trabajo de la creación de aplicaciones. A continuación se muestra el ecosistema DevOps, el cual cuenta con las herramientas más usadas en cada fase de DevOps.



PLANEACIÓN

Todo proyecto necesita un enfoque que indique a los desarrolladores el motivo y el último trabajo a realizar, limitando un conjunto de funcionalidades que permitan aportar valor en cada iteración. Para esta fase se mostrara la herramienta Jira.

¿QUÉ ES JIRA Y PARA QUÉ SIRVE?

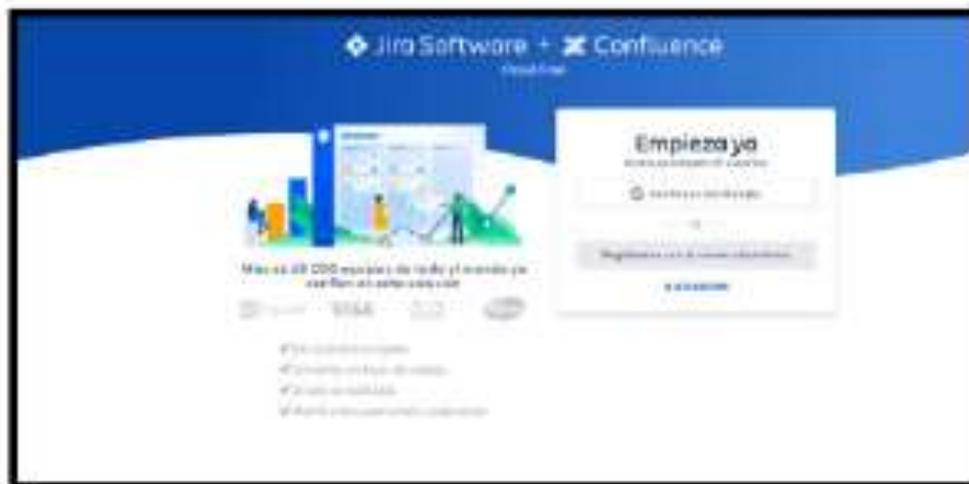
Es un software que proporciona herramientas de planificación y hojas de itinerarios para que los equipos puedan gestionar a los interesados, los presupuestos y los requisitos desde el primer día.

Pasos para la instalación de Jira:

1. Registrarse en la página oficial

Se puede encontrar esta página en siguiente link:

<https://www.atlassian.com/es/try/cloud/signup?bundle=jira-software-confluence&edition=free>



Seguido del registro se nos mostrará ciertas ventanas de diálogo las cuales servirán para configurar nuestro Jira según las herramientas que usemos y a qué nos dedicamos para tener una mejora experiencia.

Te damos la bienvenida a Jira Software. * *
Angel.

Si respondes a estas rápidas preguntas, nos ayudará a personalizar tu experiencia. Mantendremos tu información segura, de acuerdo con lo establecido en nuestra política de privacidad.

¿En qué tipo de equipo trabajas?

Recursos humanos

Soporte de TI

Marketing

Ventas

Operaciones

Frontend

Servicio de atención al cliente

Legal

Desarrollo de software

Otro

Guardar progreso

Te damos la bienvenida a Jira Software. * *
Angel.

Si respondes a estas rápidas preguntas, nos ayudará a personalizar tu experiencia. Mantendremos tu información segura, de acuerdo con lo establecido en nuestra política de privacidad.

¿Cuál de estas profesiones describe mejor lo que haces? Esto aparecerá en tu perfil.

Asesoría de negocio

Diseño de tecnología

Proyectos de producto

Estadística

Gestión de programas

Proceder

Experto en Scrum

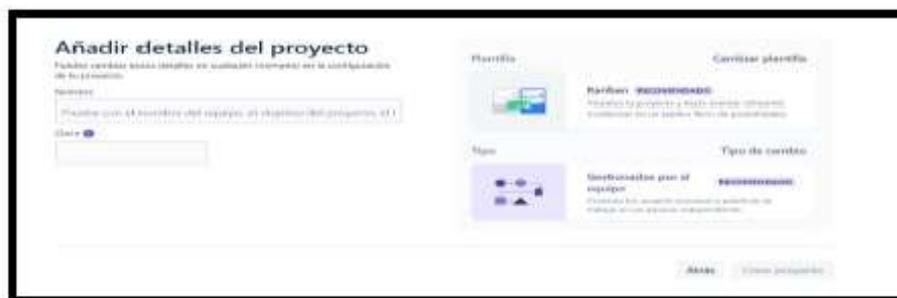
Otro

Guardar progreso



2. Crear un proyecto

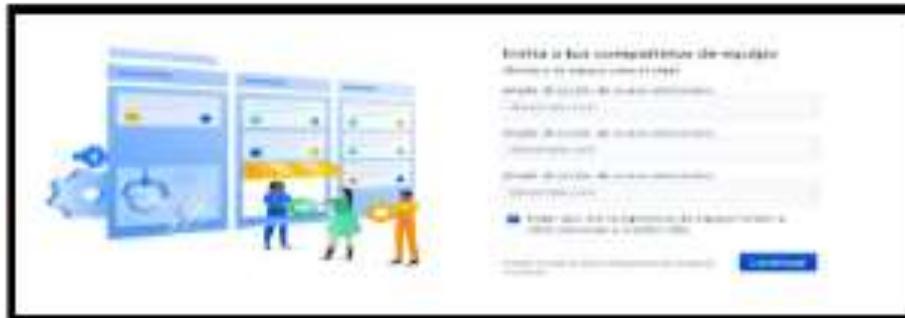
Una vez finalizado el registro y de haber configurado Jira, automáticamente, crearemos nuestro primer proyecto.



Seleccionamos las herramientas que vamos a utilizar en nuestro proyecto para que Jira tenga facilidad de conexión con dichas herramientas.



Podemos invitar a nuestros compañeros hacer parte de nuestros proyectos.



3. Configuración de proyecto

Una vez concluida la creación del proyecto, se muestra un cuadro de diálogo con la bienvenida a Jira.



Seguidamente podemos notar a simple vista que Jira tiene una interfaz amigable con el usuario. Cabe recalcar que el proceso es de manera fluida.



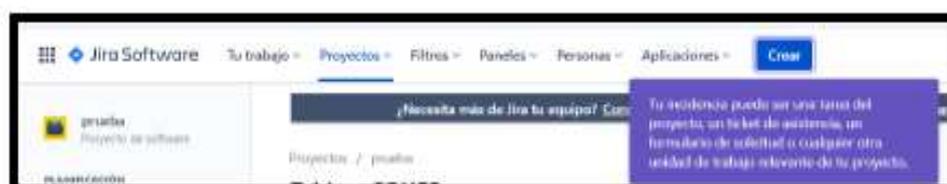
A continuación una pequeña descripción de lo que podemos hacer con Jira; empezando con la creación de incidencias.



Una pequeña descripción de lo que se puede hacer con algunas opciones del tablero.



Cuenta con un botón "Crear" el cual nos permite implantar una incidencia, subir un archivo y otras opciones como se puede mostrar en las imágenes.

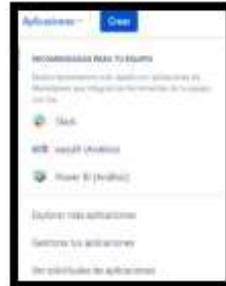


Posee la opción de agregar a alguien en el apartado de tablero, para ayudar a planificar o supervisar el proyecto.

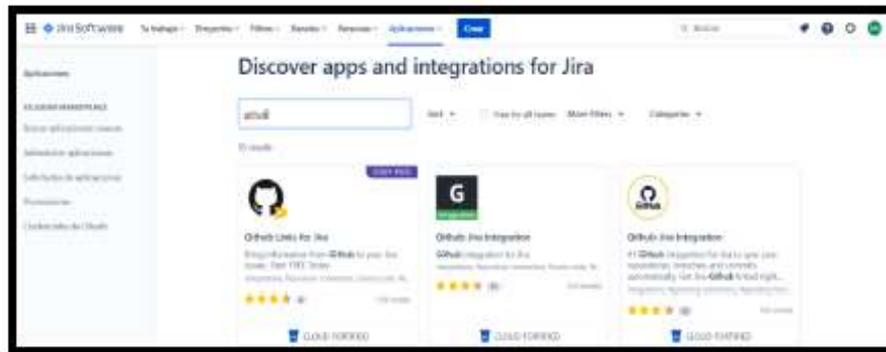


Tiene un menú desplegable de aplicaciones, donde puede escoger según las que necesite para el proyecto

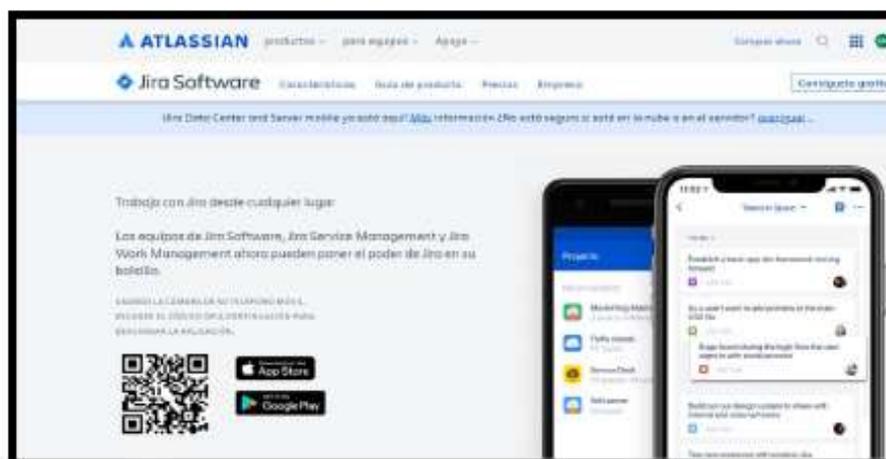




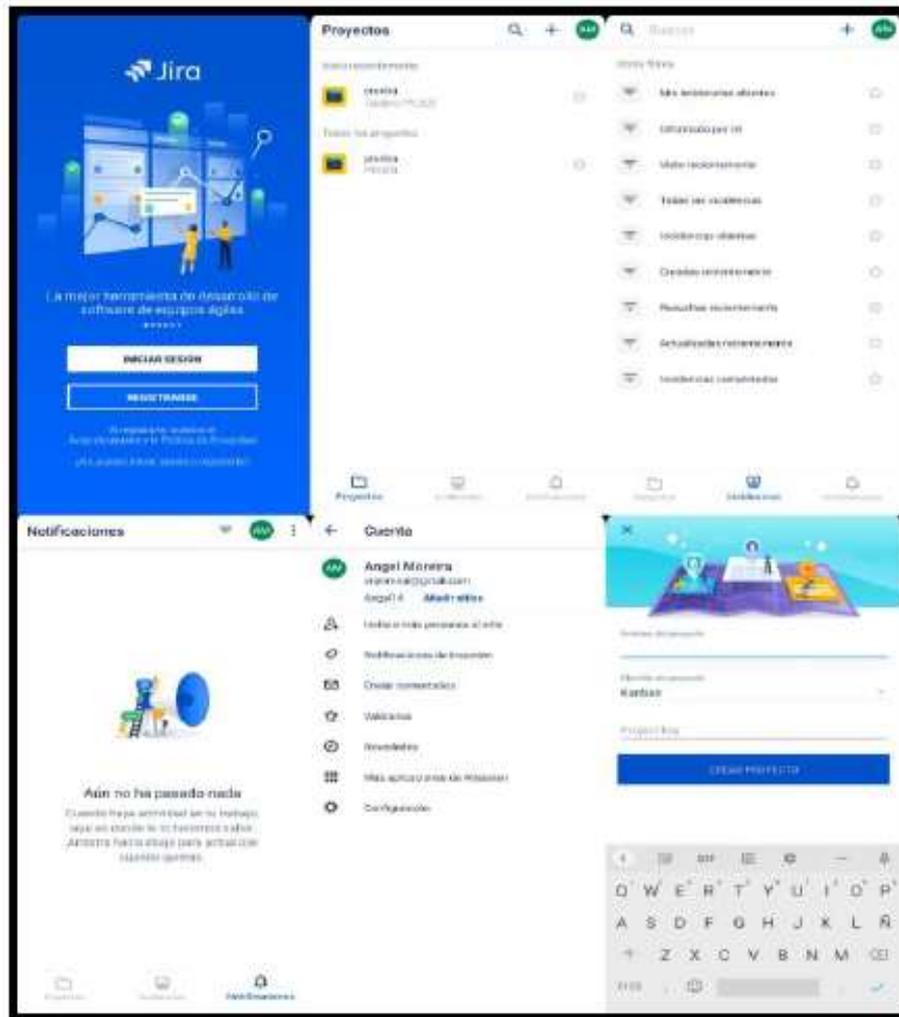
Cuenta con una amplia gama de aplicaciones donde puede escoger la que desea y también puede buscar las herramientas usando filtros.



Tiene su propia aplicación móvil donde puede monitorear, crear y recibir notificaciones de las incidencias.



Aquí podemos ver un resumen con las opciones que cuenta la aplicación móvil.



CODIFICACIÓN

En la codificación se involucra el diseño de software y la creación del código, para esta fase utilizaremos las herramientas Git y GitHub

¿QUÉ ES GITHUB Y PARA QUÉ SIRVE?

Es el sitio web más usado para almacenar repositorios git, alberga más de 50 millones de proyectos de código abierto, entre sus beneficios están el seguimiento de errores, la búsqueda rápida, el contar con una comunidad, se puede compartir y descargar un código fuente.

Pasos para usar GitHub

1. Crear una cuenta

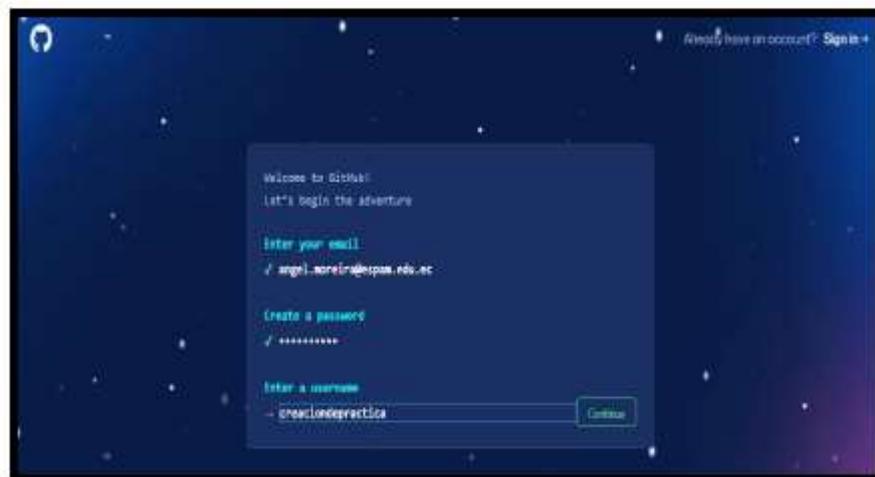
Puede crear una cuenta en siguiente link:

<https://github.com/>

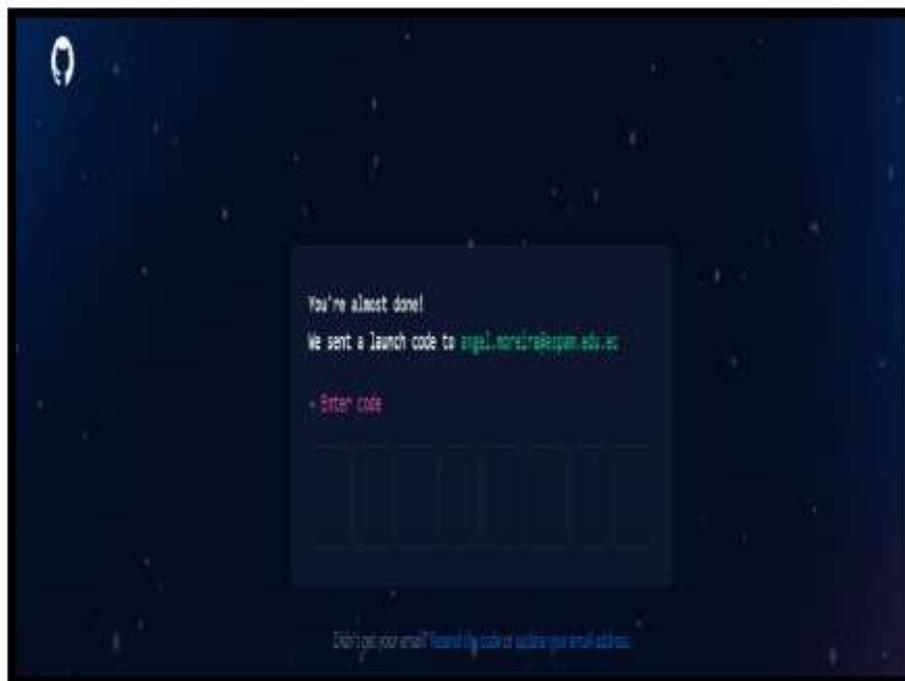
Se abrirá la siguiente ventana donde daremos clic en los botones ubicados en la parte superior derecha; Sing in es para iniciar sesión, Sign up para crear una cuenta.



Ahora seguiremos los pasos para crear una cuenta. Primero ubicar el correo, luego ubicar la contraseña, con símbolos, mayúsculas y números, luego ubicar el nombre de usuario.



Luego nos pedirá un código que nos llegara al correo que escribimos al inicio.



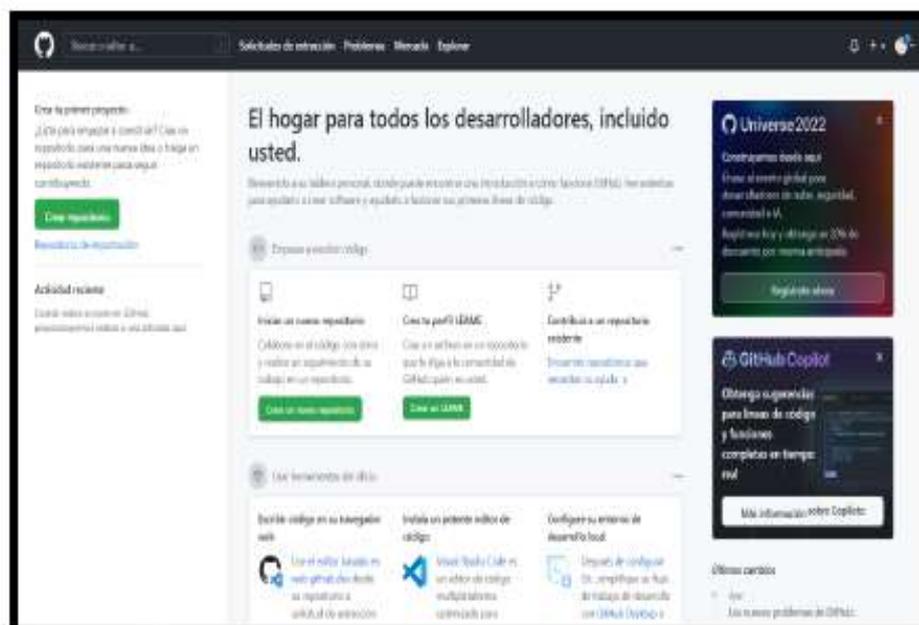
Seguido tenemos que llenar las siguientes preguntas para configurar nuestro GitHub



Seleccionamos las herramientas que vamos a utilizar, no es necesario responder o seleccionar las herramientas, se puede saltar ya que esto es solo para personalizar nuestro GitHub.



A continuación se mostrará la ventana de inicio de GitHub. En la cual podemos ver a simple vista sus opciones: crear repositorio, crear un perfil para que una comunidad te reconozca y usar editores de código.



Para crear nuestro primer repositorio seleccionamos crear repositorio y nos mostrará la siguiente ventana donde deberemos llenar los datos que nos solicita.

Crear un nuevo repositorio

Un repositorio contiene todos los archivos del proyecto, incluido el historial de revisiones. ¿Ya tienes un repositorio de proyectos en otro lugar? [Importar un repositorio](#).

Dueño: **creaciondepractica** / nombre del repositorio:

Los grandes nombres de repositorios son cortos y fáciles de recordar. ¿Necesitas inspiración? ¿Qué tal guacamole refactorizado?

Descripción (opcional):

Público
Cualquier persona en Internet puede ver este repositorio. Tú eliges quién puede comprometerse.

Privado
Tú eliges quién puede ver y comprometerse con este repositorio.

Inicialice este repositorio con:
Omita este paso si está importando un repositorio existente.

Agregar un archivo README
Aquí es donde puede escribir una descripción larga para su proyecto. [Aprenda más](#).

Agregar .gitignore:
Ejija qué archivos no rastree de una lista de patrones. [Aprenda más](#).

Plantilla: `.gitignore` Ninguna -

En este caso solo he colocado el nombre del repositorio, el cual tiene que tener el mismo nombre del proyecto que vamos alojar y para este ejemplo lo dejaremos en público, luego dar clic en crear repositorio. Seguido se mostrará la siguiente ventana.

Search GitHub · **creaciondepractica / Interfaces** · Home · Growth · Help · Settings

Quick setup — if you've done this kind of thing before

`git clone https://github.com/creaciondepractica/Interfaces.git`

Initiated by creating a new local subdirectory using the recommended repository template README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
cd <your-repo> && git init
git add .
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/creaciondepractica/Interfaces.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/creaciondepractica/Interfaces.git
git branch -M main
git push -u origin main
```

Aquí nos recomienda, subir el código mediante líneas de comando o por medio de un push. Entonces usaremos el paso a paso para subir nuestro código.

Para este punto debemos instalar Git para subir nuestro código de manera más fácil y rápida.

¿QUÉ ES GIT Y PARA QUÉ SIRVE?

Git es un sistema de control de versiones distribuido gratuito y de código abierto diseñado para manejar desde proyectos pequeños hasta proyectos grandes de manera rápida y eficaz.

Pasos para usar Git

1. Entrar en el siguiente enlace <https://git-scm.com/>
2. Dar clic en descargar para Windows.



3. Seleccionar **64-bit Git For Windows Setup**.



4. Abrir el archivo exe.



5. Dar clic en ejecutar.



6. Dar clic en next.



7. Seleccionamos los siguientes,



8. Seleccionamos la primera opción.



9. Seleccionamos la primera opción.



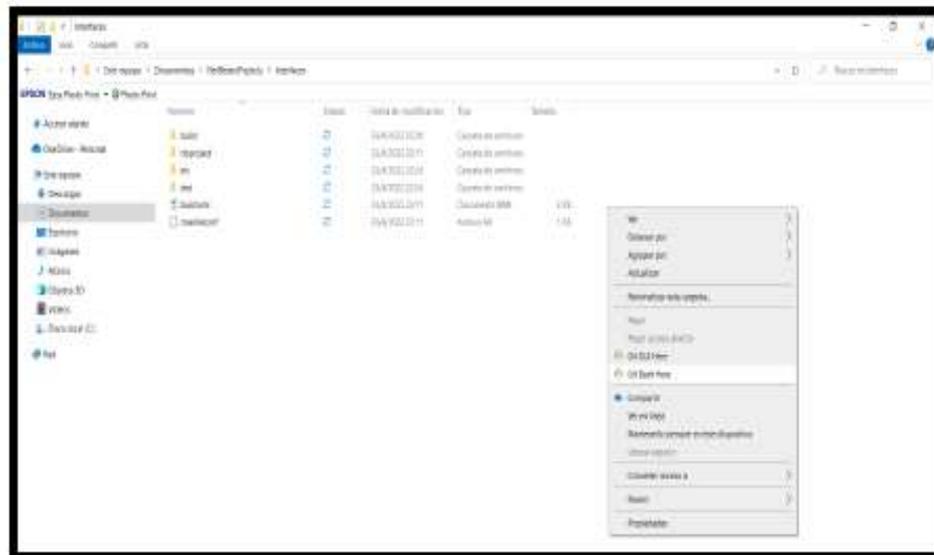
10. Comienza la instalación.



11. Pantalla de Git CMD.



Una vez instalado Git nos dirigimos al archivo el cual contiene nuestro proyecto y daremos clic derecho, nos mostrara una ventana donde daremos buscaremos la opción Git Bash Here



Nos abrirá Git con la ruta que queremos subir.



Seguiremos paso a paso la línea de comandos para subir nuestro proyecto.



Comenzamos inicializando nuestro proyecto con git init.

```

C:\Users\ANDRITA\OneDrive\Documents\NetDevOps\Projecto\ > git init
Initialized empty Git repository in C:\Users\ANDRITA\OneDrive\Documents\NetDevOps\Projecto\
C:\Users\ANDRITA\OneDrive\Documents\NetDevOps\Projecto\ > git status

```

Continuamos con un git status para ver los archivos que se van a subir y luego escribimos "Git add." para cargar todo nuestro proyecto.

```

C:\Users\ANDRITA\OneDrive\Documents\NetDevOps\Projecto\ > git status
On branch main
No commits yet
Untracked files:
  (use "git add <file>..." to include in what will be staged)
        .gitignore
        main.py
        main.py~
        README.md
nothing added to commit but untracked files present (use "git add" to track)
C:\Users\ANDRITA\OneDrive\Documents\NetDevOps\Projecto\ > git add
C:\Users\ANDRITA\OneDrive\Documents\NetDevOps\Projecto\ > git add

```

Seguimos con el primer commit -m "primer commit" seguido del nombre del proyecto entre comillas dobles el cual es para agregar los cambios del proyecto.

```

C:\Users\ANDRITA\OneDrive\Documents\NetDevOps\Projecto\ > git commit -m "primer commit"
[master (root-commit) 12345678] primer commit
 2 files changed, 10 insertions(+), 0 deletions(-)
 create mode 100644 .gitignore
 create mode 100644 main.py

```

Usamos git branch -M main para indicar que vamos a subir este proyecto a nuestra rama principal.

```

C:\Users\ANDRITA\OneDrive\Documents\NetDevOps\Projecto\ > git branch -M main
Switched to branch 'main'
C:\Users\ANDRITA\OneDrive\Documents\NetDevOps\Projecto\ > git status
On branch main
nothing to commit, working tree clean

```

Ahora vamos a subir este proyecto a nuestro repositorio usando el siguiente comando git remote add origin <https://github.com/Angelm321/Interfaces.git> el enlace debe ser el que les aparece en su repositorio.

```

MINGW64~/c:/Users/ANGELIDA/OneDrive/Documents/NetBeansProjects/Interfa...
create mode 100644 build/classes/netbeans_update_testarccas
create mode 100644 build/classes/accessdatos/IAccesoBatos.class
create mode 100644 build/classes/accessdatos/ImplementacionMySQL.class
create mode 100644 build/classes/test/TestInterfaces.class
create mode 100644 main/test.af
create mode 100644 nbproject/build-impl.xml
create mode 100644 nbproject/genfiles.properties
create mode 100644 nbproject/private/private.properties
create mode 100644 nbproject/project.properties
create mode 100644 nbproject/project.xml
create mode 100644 src/accessdatos/IAccesoBatos.java
create mode 100644 src/accessdatos/ImplementacionMySQL.java
create mode 100644 src/test/TestInterfaces.java

MINGW64~/c:/Users/ANGELIDA/OneDrive/Documents/NetBeansProjects/Interfa...
$ git branch -M main
MINGW64~/c:/Users/ANGELIDA/OneDrive/Documents/NetBeansProjects/Interfa...
$ git remote add origin https://github.com/Angelm321/Interfaces.git
MINGW64~/c:/Users/ANGELIDA/OneDrive/Documents/NetBeansProjects/Interfa...
$

```

Seguido usamos el ultimo comando git push -u origin main para hacer efectiva la sincronización del repositorio local con el remoto.

```

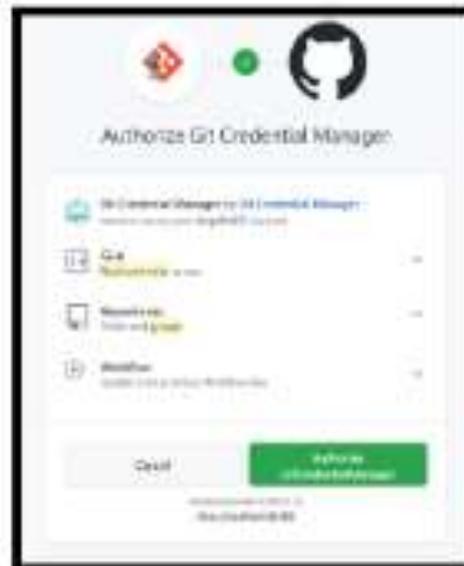
MINGW64~/c:/Users/ANGELIDA/OneDrive/Documents/NetBeansProjects/Interfa...
bash: create: command not found
bash: syntax error near unexpected token `;'
bash: $: command not found
bash: syntax error near unexpected token `;'
bash: $: command not found
bash: $: command not found
MINGW64~/c:/Users/ANGELIDA/OneDrive/Documents/NetBeansProjects/Interfa...
$
MINGW64~/c:/Users/ANGELIDA/OneDrive/Documents/NetBeansProjects/Interfa...
$ AC
MINGW64~/c:/Users/ANGELIDA/OneDrive/Documents/NetBeansProjects/Interfa...
$ git push -u origin main

```

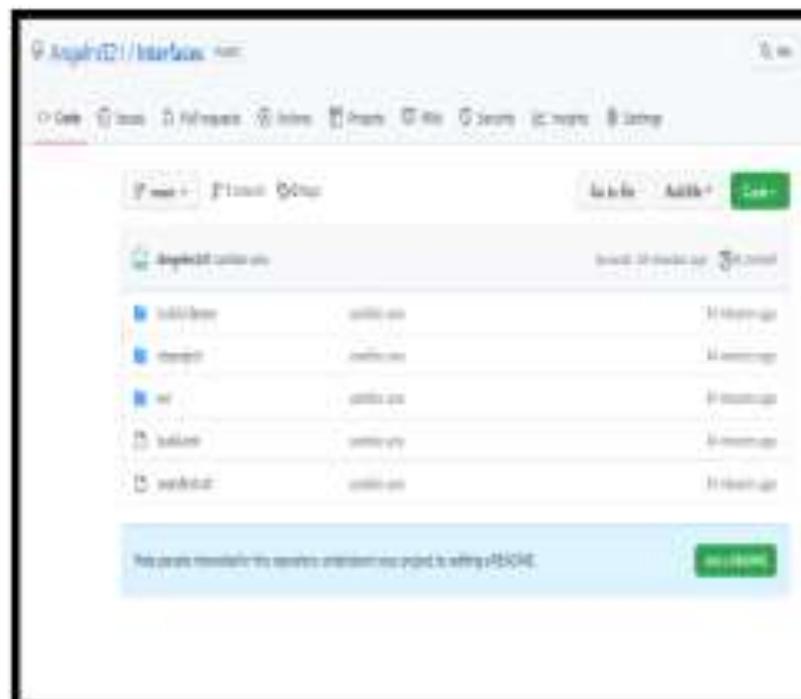
Aquí nos pide que iniciemos sesión en GitHub.



Autorizamos.



En este punto vamos a nuestro repositorio remoto y reiniciamos la página, entonces nos aparecerá nuestro proyecto.



LANZAMIENTO

En esta fase, se llevan a cabo las pruebas. El desarrollo continuo sólo es posible debido a la integración y las pruebas continuas.

¿QUÉ ES JENKINS Y PARA QUÉ SIRVE?

Es una herramienta de automatización de tareas en procesos de integración continua, tiene como características que cuenta con más de 1400 plugins, puede orquestar cualquier tipo de proceso, es utilizada para ejecutar tareas manuales, automáticas o periódicas, fácil de usar, dashboard para visualizar el estado de tareas, monitoreo de procesos y alertas ante problemas en algún proceso.

Pasos de instalación:

1. Primero debemos instalar JDK java lo podemos encontrar este link.

<https://www.oracle.com/technologies/javase/jdk11-archive-downloads.html>

2. Buscamos la versión para Windows, le damos clic al texto azul y descargamos.

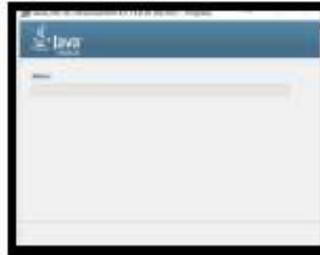


3. Ejecutamos el archivo exe.



4. Continuamos con la instalación.





5. Nos dirigimos a la página de descarga de Jenkins mediante el link.

<https://www.jenkins.io/download/>

6. Seleccionamos para Windows LTS.



7. Abrimos el ejecutable.



8. Damos clic en siguiente.







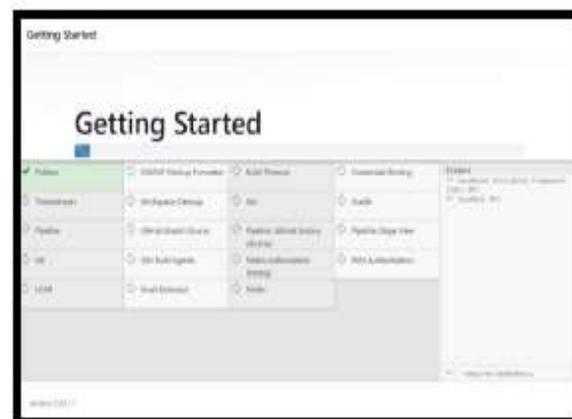
9. Nos abrió el siguiente localhost del puerto 8080, en cual nos pide una contraseña la cual está ubicada en la ruta que nos aparece en rojo, una vez que llegamos a esa ruta abrimos el archivo que nos pide initialAdminPassword en un block de nota y lo pegamos en la barra, damos clic en continuar.



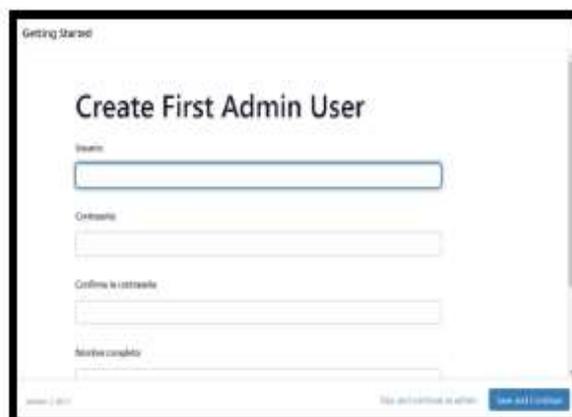
10. Instalamos los plugin de la izquierdo.



11. Esperamos.



12. Configuramos los datos, para iniciar sesion en Jenkins.



Getting Started

Name

URL

Confirm URL

Master computer

Email address

[Save and continue](#)

13. Configuramos la instancia y damos clic en guardar.

Getting Started

Instance Configuration

Jenkin (1) <http://192.168.1.100:8080>

[Save](#)

14. Damos clic en star y listo podemos usar jenkins.

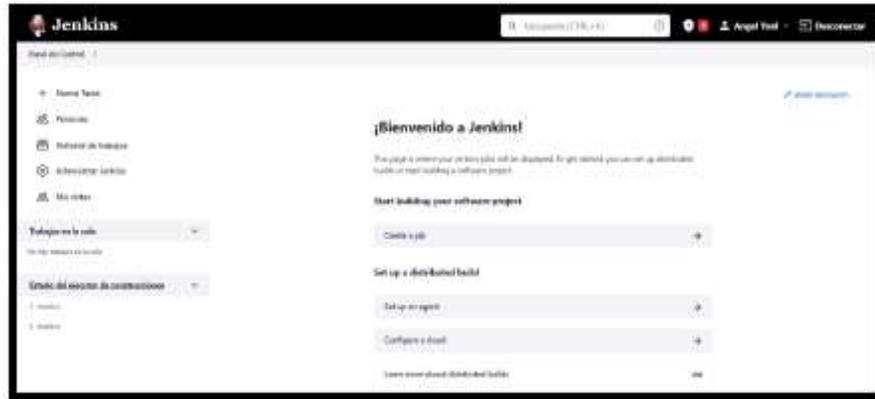
Getting Started

Jenkins is ready!

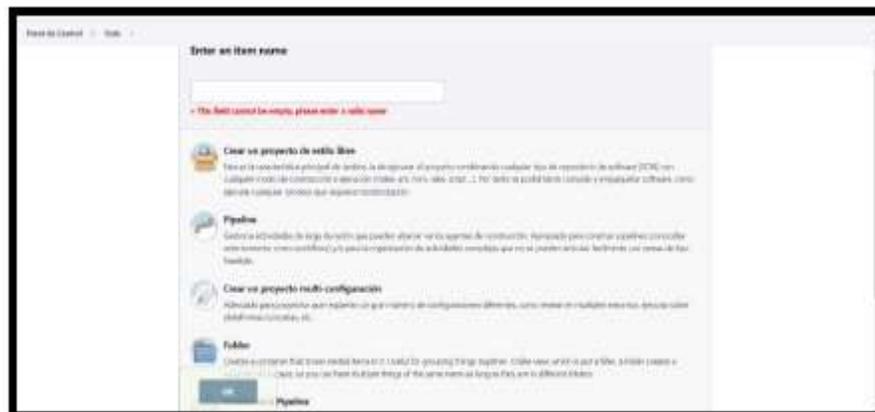
Your Jenkins setup is complete.

[Start Jenkins](#)

15. Interfaz de Jenkins.



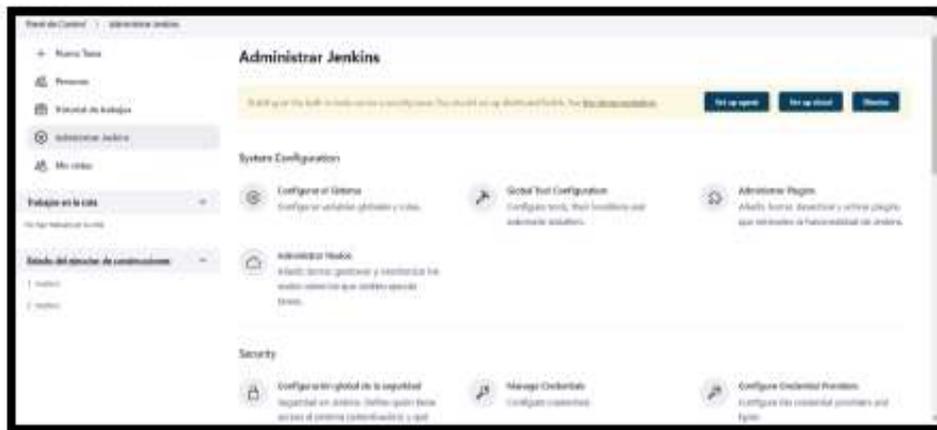
16. Interfaz para crear una nueva tarea nueva tarea.



17. Historial de tareas.



18. Configurar Jenkins.



DESPLIEGUE Y CONSTRUCCIÓN.

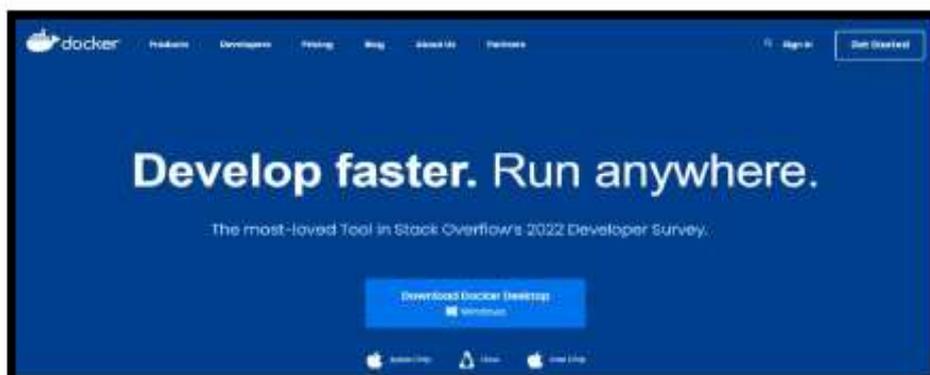
En la fase de construcción fase es donde se edifica, ya sea diseñando infraestructura, automatizando procesos, escribiendo código, definiendo pruebas o estableciendo la seguridad.

En despliegue, se gestiona el software durante su producción. En cualquier momento que se realice un cambio, no debe afectar el funcionamiento del sitio web de alto tráfico.

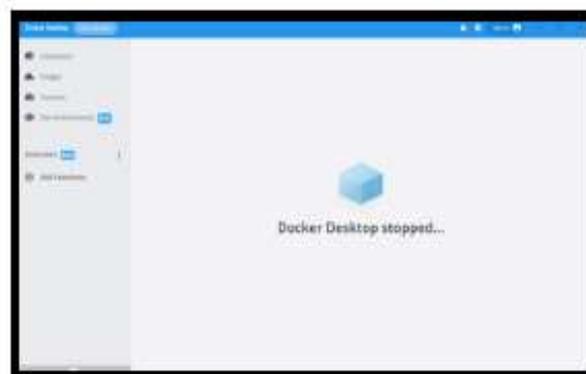
Para estas fases se optó por docker el cual es un proyecto de código abierto con el que se pueden crear fácilmente contenedores, entre sus características se puede encontrar autogestión de los contenedores, fiabilidad, capacidad para desplegar multitud de contenedores en un mismo equipo físico, puesta en marcha de los servicios de manera rápida y puede ejecutar una amplia gama de aplicaciones.

Instalación de docker:

1. Nos dirigimos al siguiente enlace <https://www.docker.com/>



2. Instalamos el archivo docker-desktop.exe.



3. Nos dirigimos a la documentación al siguiente enlace <https://docs.microsoft.com/en-us/windows/wsl/install-manual>, buscaremos la opción Step 4 y daremos clic en el texto [WSL2 Linux kernel update package for x64 machines](#)

Restart your machine to complete the WSL install and update to WSL 2.

Step 4 - Download the Linux kernel update package

1. Download the latest package.

- [WSL2 Linux kernel update package for x64 machines?](#)

Note

If you're using an ARM64 machine, please download the ARM64 package if instead. If you're not sure what kind of machine you have, open Command Prompt or PowerShell and enter: `systemtype`. Find "System Type". **Caution:** On non-English Windows versions, you might have to modify the search text, translating the "System Type" string. You may also need to escape the quotes for the find command. For example, in German: `systemtype` find: `"Systemtyp"`.

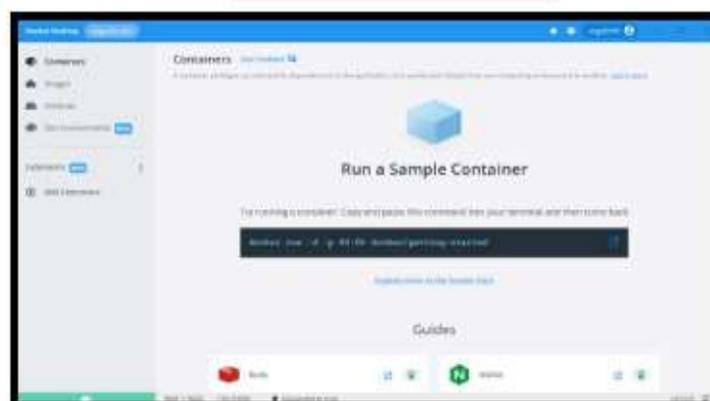
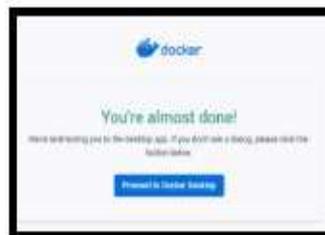
4. Instalamos el archivo descargado.



5. Escribimos el siguiente comando en PowerShell "wsl --set-default-version 2".



6. Una vez instalado docker y wsl procedemos a iniciar sesión en docker, donde podemos encontrar una interfaz amigable y varias opciones en la parte izquierda de la pantalla como son contenedores, imágenes, volumen y dev environments.



7. Podemos ir al cmd y con el comando `docker run hello-world` podemos verificar la creación y descarga de una imagen.

```

C:\Users\YARELI>docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (WARNING: This step may have caused a billing event.)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently seeing.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

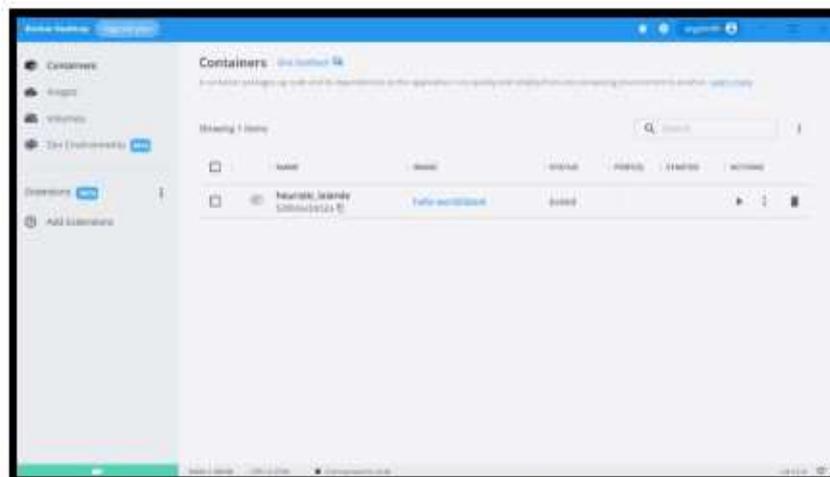
To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu:latest bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com

For more examples and ideas, visit:
https://docs.docker.com/get-started/

C:\Users\YARELI>

```



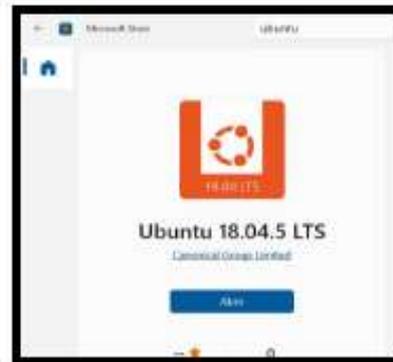
OPERACIÓN

Esta fase se encarga del comportamiento impropio del sistema o de los errores que se encuentran en la producción.

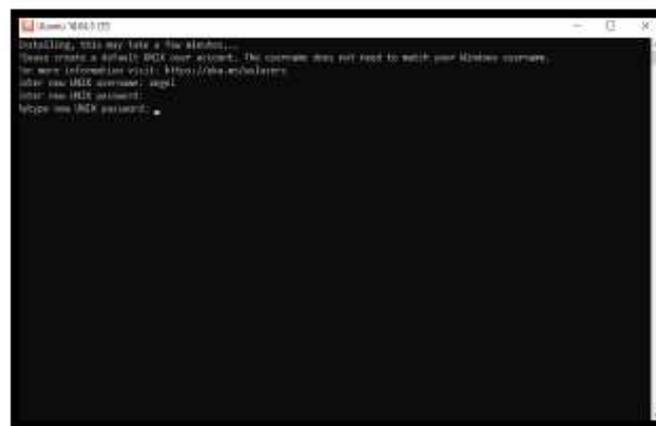
Ansible es una herramienta open source de DevOps, en la cual se pueden automatizar todas aquellas tareas de administración de servidores que un sistema debe ejecutar día a día, esto trae como beneficio el ahorro de tiempo, efectividad y simplificación de tareas complejas, cuenta con herramientas de depuración y validación, con poco conocimiento de programación se puede operar.

Pasos para instalación:

1. Instalamos Ubuntu el cual lo podemos encontrar en Microsoft Store.



2. Nos pedirá ingresar datos como usuario y contraseña, seguido podremos usar la consola. Con el comando `-i ../host` (archivo .yml).



3. Usamos los siguientes comandos en la terminal de Ubuntu para la instalación de ansible.

```
sudo apt-get update
sudo apt-get install software-properties-common
sudo apt-add-repository ppa:ansible/ansible
sudo apt-get update
sudo apt-get install ansible
```

4. Ya puede usar ansible en Windows.

```

angel@CCATDF-4C6376 ~$ git clone https://github.com/ansible/ansible
Cloning into 'ansible'...
remote: Enumerating objects: 45, done.
remote: Counting objects: 385 (45/45), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 45 (delta 8), reused 22 (delta 4), pack-reused 0
Receiving objects: 100% (45/45), done.
angel@CCATDF-4C6376 ~$

```

```

angel@CCATDF-4C6376 ~$ cd ansible
angel@CCATDF-4C6376 ~$ cd docs
angel@CCATDF-4C6376 ~$ cd 11
total 28
-rwxr-xr-x 4 angel angel 4096 Sep 18 10:41 /
-rwxr-xr-x 7 angel angel 4096 Sep 18 10:42 /
-rwxr-xr-x 8 angel angel 4096 Sep 18 10:42 /lib
-rwxr-xr-x 1 angel angel 1525 Sep 18 10:42 README.md
-rwxr-xr-x 1 angel angel 391 Sep 18 10:42 begin.txt
-rwxr-xr-x 1 angel angel 289 Sep 18 10:42 hosts
-rwxr-xr-x 3 angel angel 4096 Sep 18 10:42 playbooks
angel@CCATDF-4C6376 ~$ cd playbooks
angel@CCATDF-4C6376 ~$ cd ansible-test

```

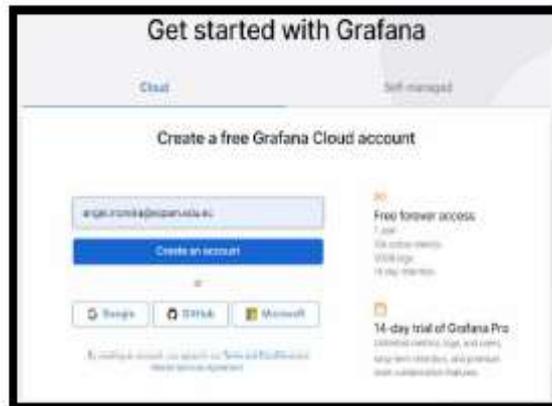
MONITOREO

En esta fase se identifica y recopila información sobre problemas que surgen en una versión de software específica que se encuentra en producción. Aquí el equipo de operación se encargará de definir las medidas para monitorizar y controlar el estado de salud de las aplicaciones y su infraestructura.

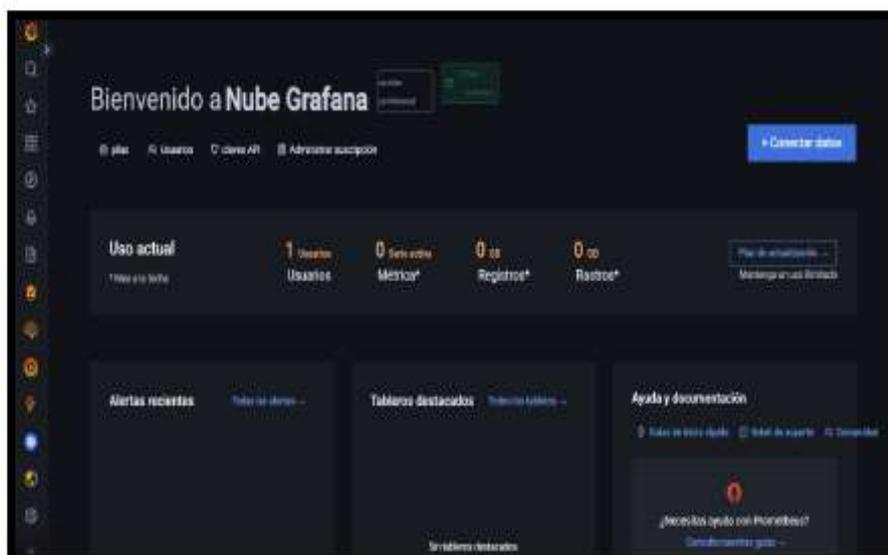
Grafana es un herramienta que sirve para efectuar análisis de datos, monitorear aplicaciones y recursos, entre sus características encontramos, se conecta a todas las fuentes de datos posibles, ayuda a rastrear el comportamiento del usuario y de la aplicación, existen paneles donde tiene un gran variedad de opciones para la visualización de mapas y cuadros para poder estudiar datos, se pueden configurar alertas.

Pasos

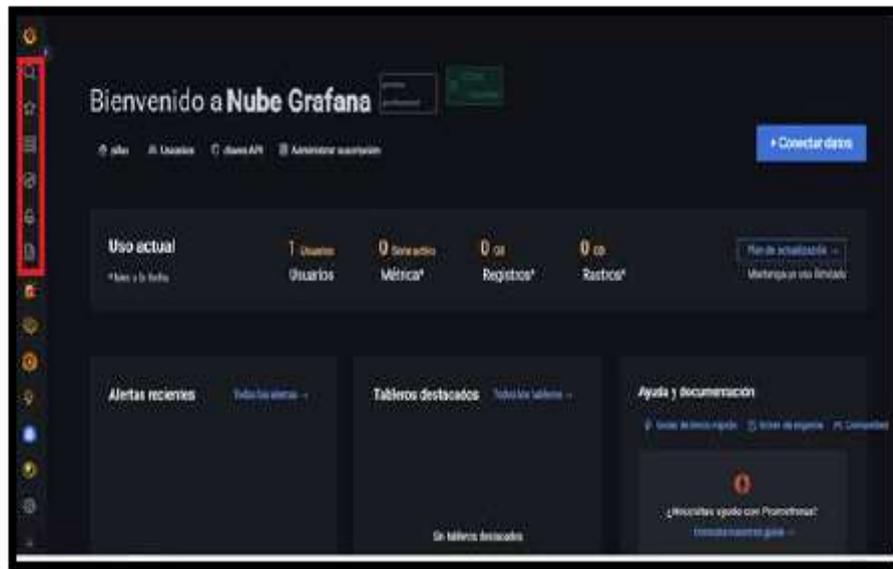
1. Creamos una cuenta en grafana.



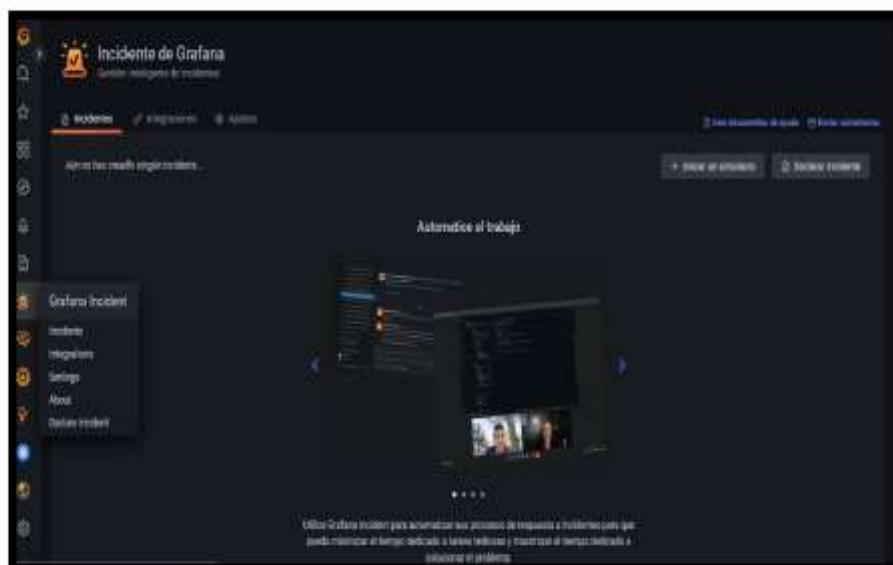
2. Seguimos los pasos que nos aparecen después del registro y por ultimo iniciamos sesión.

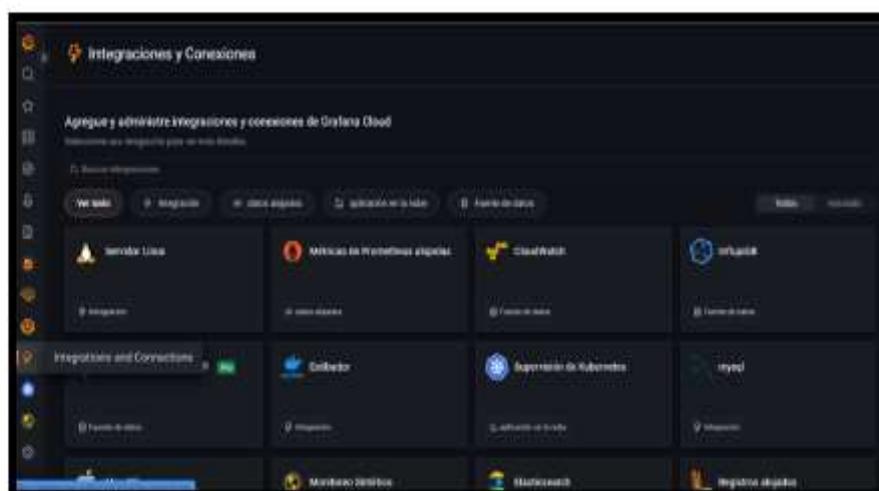
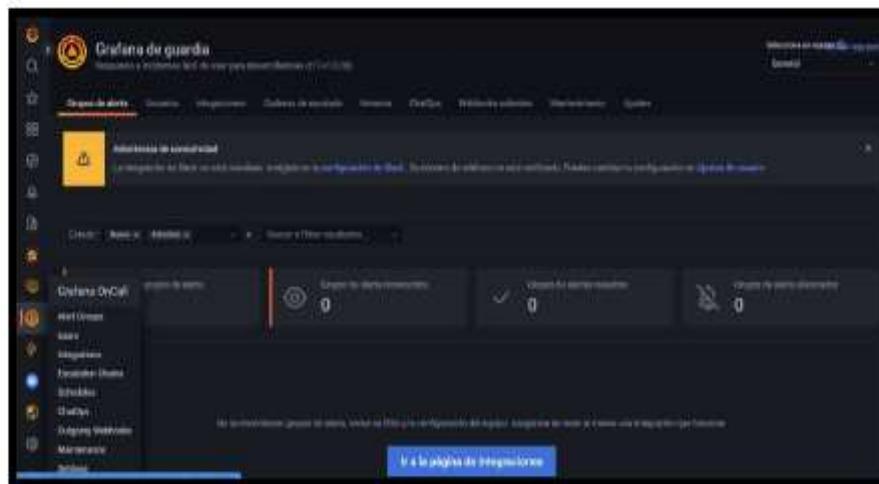
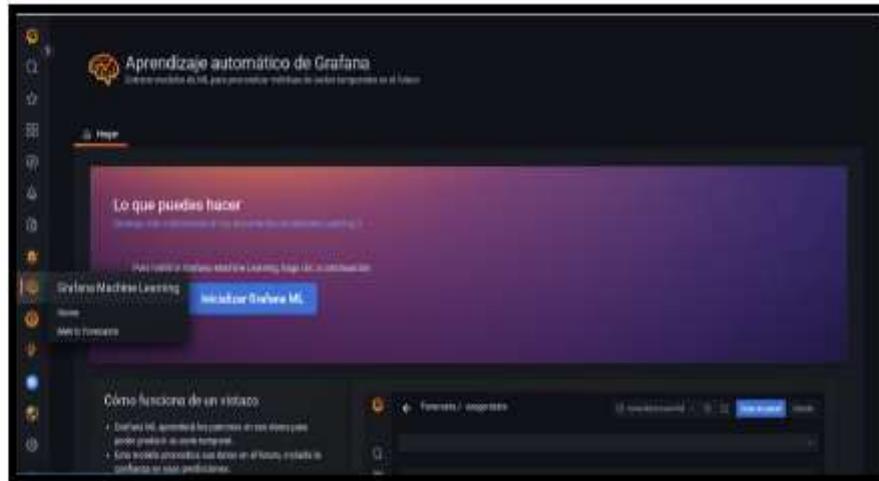


3. La ventana principal de grafana nos muestra un reporte de lo que estemos trabajando y con quienes, en el menú izquierdo nos muestra una variedad de opciones, entre ellas están, alertas, reportes y una librería con paneles.



4. y lo mejor de grafana lo podemos encontrar de color amarillo después de sus herramientas básicas. Estas son para el monitoreo de incidentes, aprendizaje automático, grafana de guardia, supervisión de kubernetes y la monitorización sistemática en la nube







PRUEBAS

Esta fase incluye la realización de pruebas continuas (manuales o automatizadas) para garantizar la calidad de la programación. Para esta fase utilizaremos Selenium, ya que entre sus características principales tenemos su gratuidad y otros servicios que puede ofrecer a los desarrolladores.

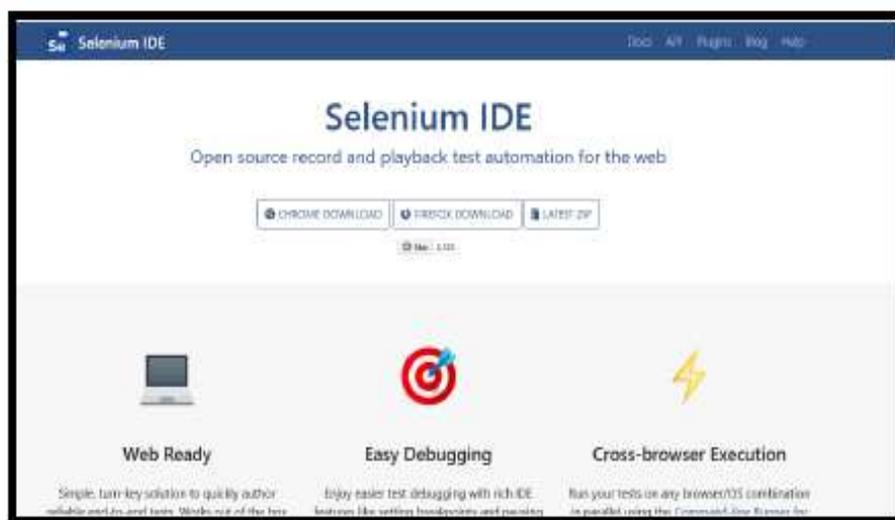
¿QUÉ ES SELENIUM Y PARA QUÉ SIRVE?

Selenium IDE es un plugin de Firefox, Chrome y diferentes navegadores, este pertenece al conjunto de herramientas de SeleniumHQ, permitiendo aplicar pruebas sobre múltiples aplicaciones web. Su objetivo fundamental es crear pruebas funcionales o tareas que requieren una cierta cantidad de procesamiento, además permite ahorrarse mucho tiempo a un equipo de desarrollo.

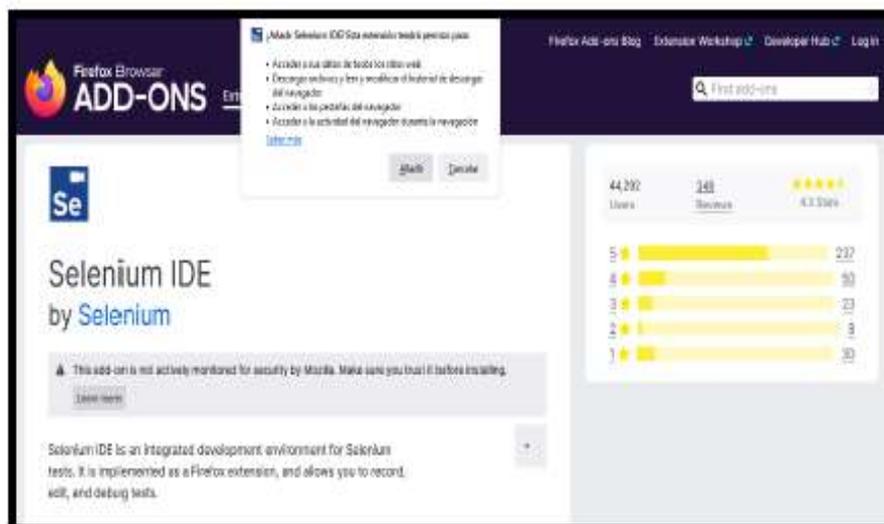
Este realiza una grabación del módulo o parte del funcionamiento que se desea testear permitiendo adaptarse a los diferentes casos logrando aplicarla las veces que sean necesarias.

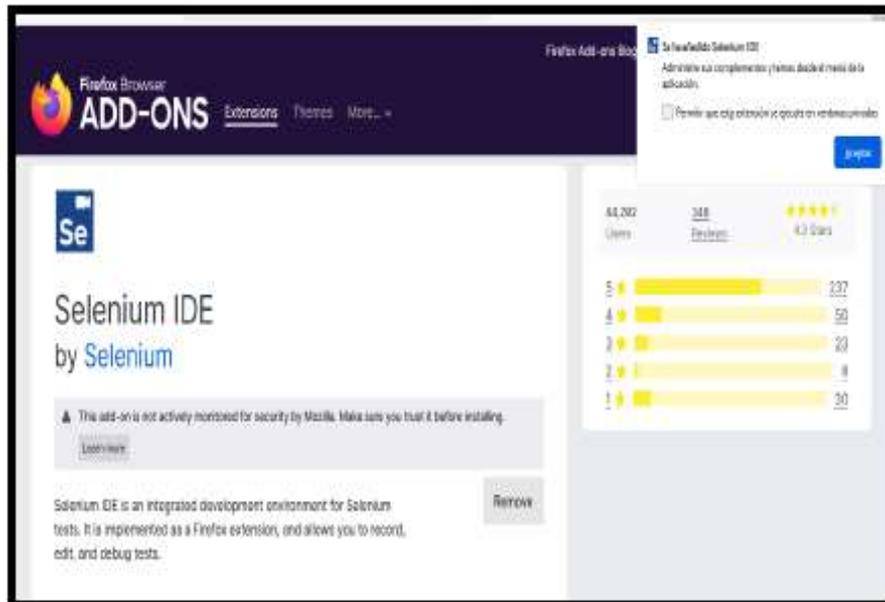
Pasos para instalación:

1. Nos dirigimos a la pagina <https://www.selenium.dev/selenium-ide/>

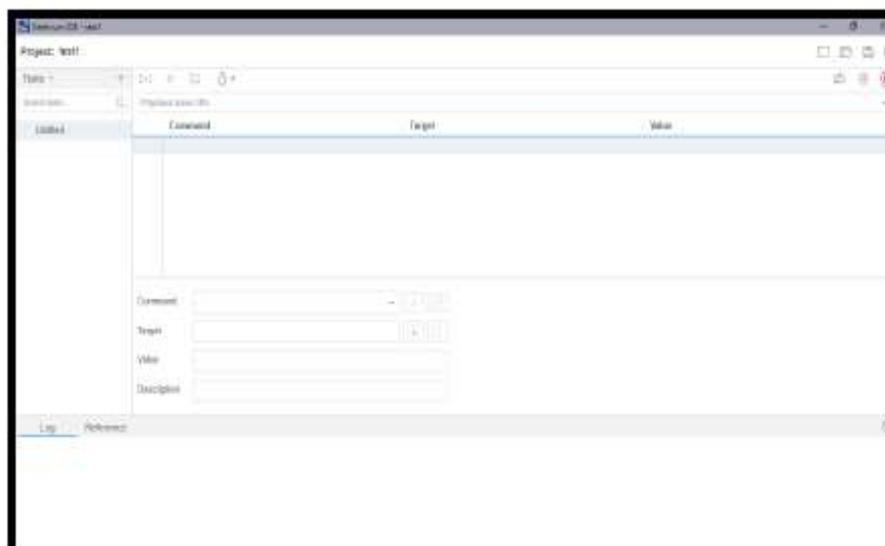


2. Descargamos el que necesitamos si es para Chrome o Firefox.
3. Luego se desplegará una ventana para agregarlo como extensión.





4. Interfaz de Selenium.

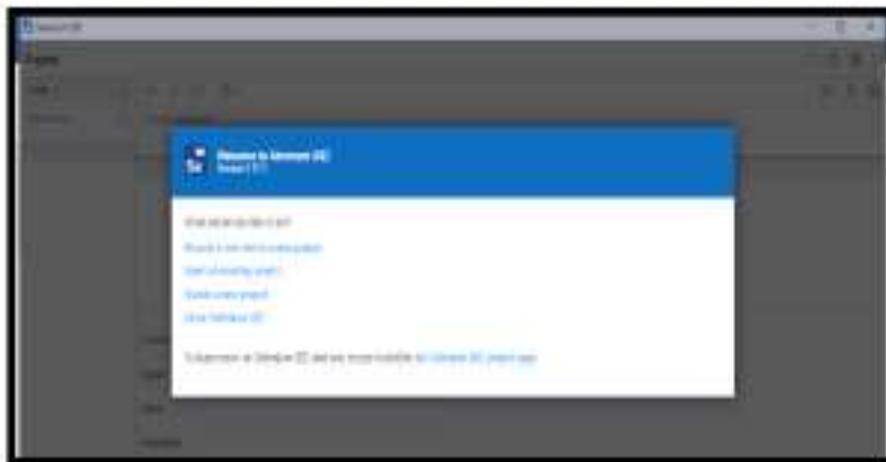


5. Ejemplo de navegación

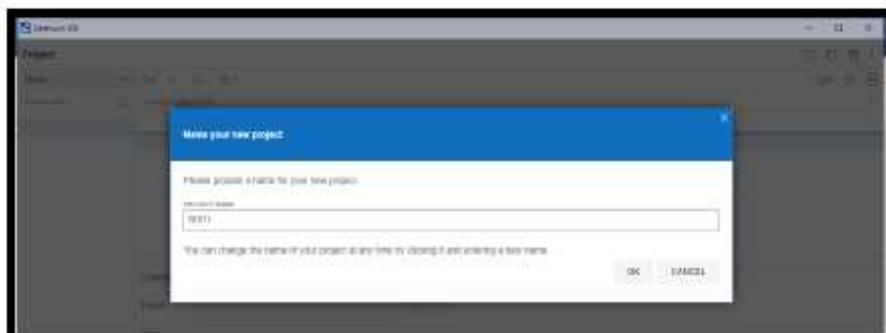
- Abrimos un navegador para iniciar el plugins de Selenium IDE , una vez descargado aparecerá en la parte superior derecha de nuestra ventana.



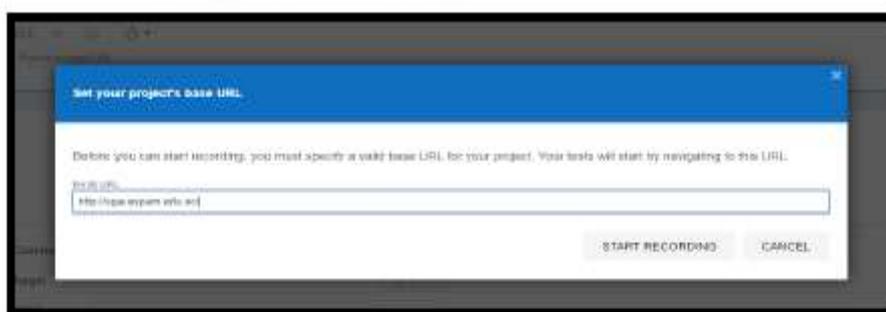
- Una vez seleccionado, se despliega una ventana con las opciones para iniciar un proyecto para este caso, iniciamos un nuevo test dando clic en la opción de "Record a new test in a new Project".



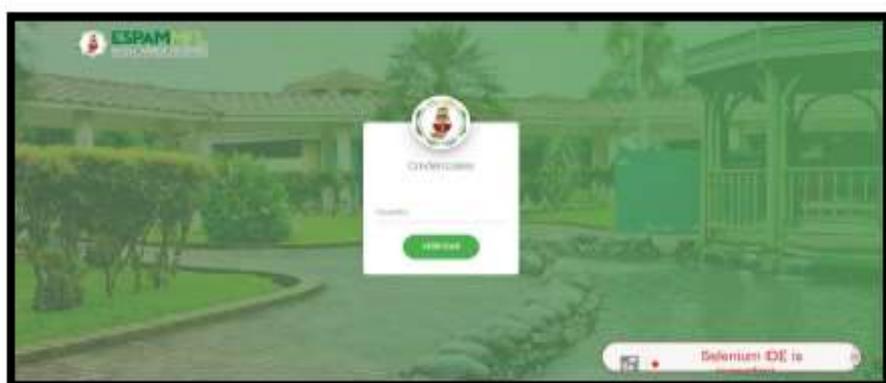
- Luego se despliega una nueva ventana para asignar un título al proyecto, ingresamos el que deseamos y pulsamos el botón ok.

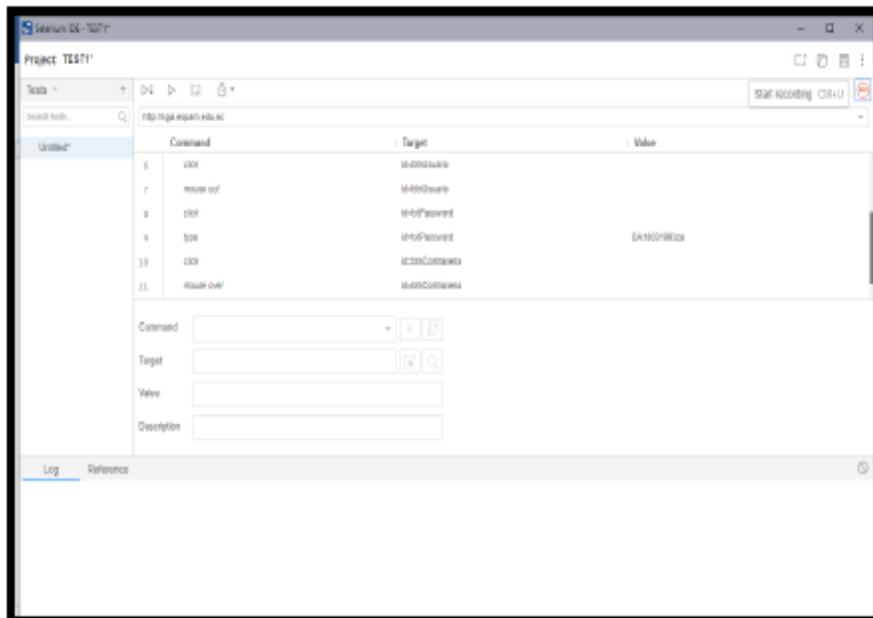


- Una vez ingresado el nombre, Selenium IDE nos despliega una ventana para ingresar la URL de nuestro proyecto a testear, la ventaja que este puede estar de manera local como global, para este caso vamos a analizar la página del control académico de la Escuela Superior Politécnica Agropecuaria de Manabí MFL, una vez colada la URL damos clic en el botón start recording.

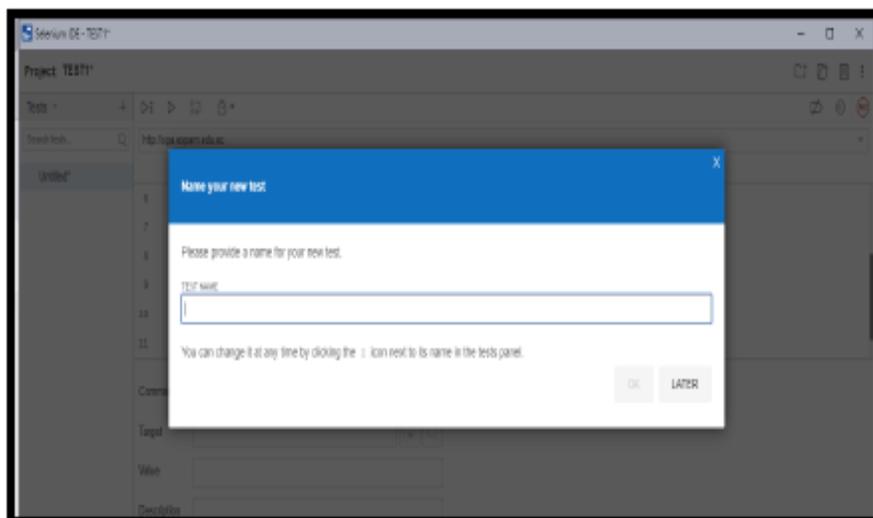


- Se despliega una vista para comenzar a testear la aplicación, ingresamos los datos necesarios para iniciar sesión, para que se quede grabado todo mediante la aplicación, una vez terminado el proceso de grabación nos dirigimos a selenium.

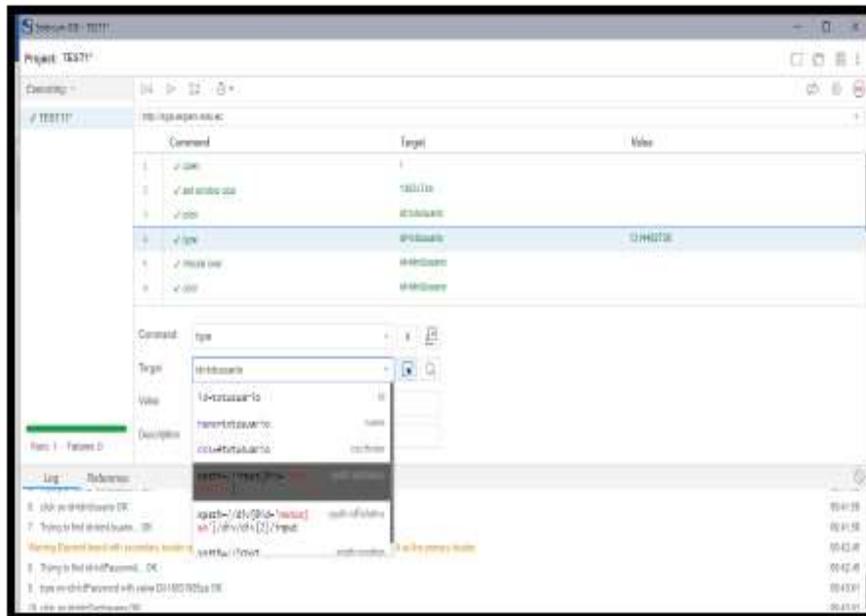




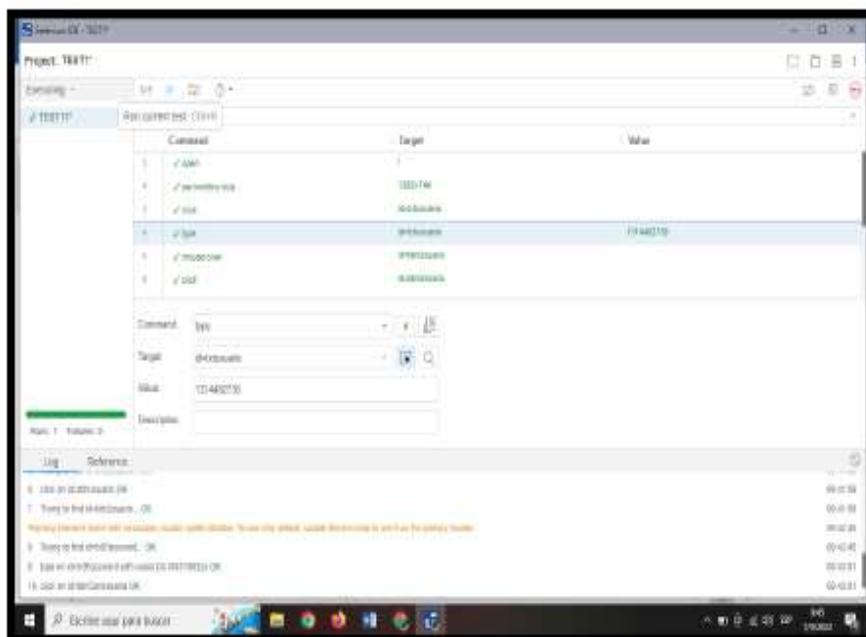
- Seleccionamos el botón de "REC" y se despliega una ventana, donde colocamos el nombre o un nuevo test.



- Una vez finalizado el test, en la vista de selenium nos muestra todas las acciones realizadas como la selecciona de la caja de texto como también si utilizamos el mouse si lo colocamos sobre alguna parte funcional de la página.
En la vista de selenium si damos clic sobre uno de los campos utilizado observamos que se despliega un target para observar los campos de esa caja o botón.



- Para realizar el test nuevamente si es que se encuentra un error seleccionamos el botón de play (Run current test) en la parte superior y automáticamente se abre una ventana en la cual ingresamos los datos y después SeleniumIDE coloca un check para ver si el proceso se corrigió o sigue con problemas.



ANEXO 2. ENTREGA DE GUÍA DE DEVOPS A LOS MIEMBROS DE LA UNIDAD DE DESARROLLO COMPUTACIONAL



ANEXO 3. CERTIFICADO DE ENTREGA DE GUÍA A LA UDIV

República del Ecuador



**ESCUELA SUPERIOR POLITÉCNICA
AGROPECUARIA DE MANABÍ MANUEL FÉLIX**
Carrera de Computación
Unidades de Docencia Investigación y Vinculación



Calceta, 11 de enero de 2023

CERTIFICACIÓN

Certifico que el estudiante Ángel Yoel Moreira Álava con C.I. N.º 1316331212 de la carrera de Computación, hizo la entrega del manual titulado "Guía de DevsOps" de forma digital e impresa a esta dependencia Universitaria.

El portador de este documento puede hacer uso del mismo como lo considere necesario.

Atentamente



Firma digitalizada por:
**YIMMY
SALVADOR**

Mgtr. Jimmy Salvador Loor Vera
DOCENTE RESPONSABLE
UDIVs CARRERA DE COMPUTACIÓN - ESPAM MFL