

UNIVERSIDAD PRIVADA DE TACNA



INGENIERIA DE SISTEMAS

TITULO:

**TRABAJO ENCARGADO N° 01: HERRAMIENTAS DE GESTION
DE PRUEBAS**

CURSO:

CALIDAD Y PRUEBAS DE SOFTWARE

DOCENTE:

Patrick Cuadros Quiroga

Integrantes:

Anahua Huayhua, Jenny Karen	(2018062150)
Coloma Colquehuanca, Kiara Estefani	(2018062218)
Cuadros Napa, Raul Marcelo	(2017057851)
Limache Durand, Rodrigo Jeral	(2017059278)

Tacna - Peru

2021

Herramientas de gestión de pruebas

Jenny Anahua, Kiara Coloma, Raul Cuadros y Rodrigo Limache

December 1, 2021

Resumen

En la presente artículo nos dedicamos a indagar sobre las peculiaridades de plataformas de Cloud Computing. Es el caso de Azure Devops, AWS Codepipeline, Google Code Build, Github Actions, etc , haciendo uso de versiones privadas de prueba en entornos educativos. Teniendo en cuenta sus características principales, se procede a realizar una comparación de las plataformas mencionadas, con la finalidad de elaborar una Tabla comparativa de los frameworks mencionados para presentar una serie de recomendaciones acerca de las características de las plataformas más seguras y adecuadas a la hora de decidir por una u otra. Con el fin de alcanzar el objetivo de este trabajo que resulta ser el de indagar y adquirir conocimientos teóricos sobre marcos de trabajo de plataformas de Cloud Computing

Abstract

In this article we are dedicated to investigating the peculiarities of Cloud Computing platforms. This is the case of Azure Devops, AWS Codepipeline, Google Code Build, Github Actions, etc., making use of private trial versions in educational environments. Taking into account their main characteristics, a comparison of the aforementioned platforms is carried out, in order to prepare a comparative table of the aforementioned frameworks to present a series of recommendations about the characteristics of the safest and most appropriate platforms at the time to decide for one or the other. In order to achieve the objective of this work, which turns out to be to investigate and acquire theoretical knowledge about Cloud Computing platform frameworks.

Keywords— Test, Pruebas, Software, Calidad, Azure Devops, Google Code Build

I. INTRODUCCION

Las herramientas de gestión de pruebas son aquellas que se utilizan para gestionar la información relativa a los «casos de prueba», normalmente los funcionales, para planificar actividades de testing, para gestionar los informes resultantes después de pasar dichos test, etc. Es fundamental para cualquier proyecto, salvo que sea muy pequeño, contar con alguna herramienta de gestión de pruebas. Hay herramientas que van por separado y otras que integran con herramientas complementarias, por ejemplo, con las de «bug tracking».

II. OBJETIVOS

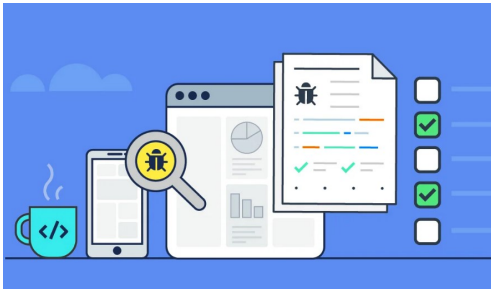
- Conocer sobre Herramientas de pruebas.

III. DESARROLLO

i. HERRAMIENTAS DE GESTIÓN DE PRUEBAS

Las herramientas de gestión de pruebas ayudan a gestionar todo el ciclo de pruebas de un producto. Una herramienta de gestión de pruebas útil debería poder integrarse con otros marcos de automatización y Integración CI.

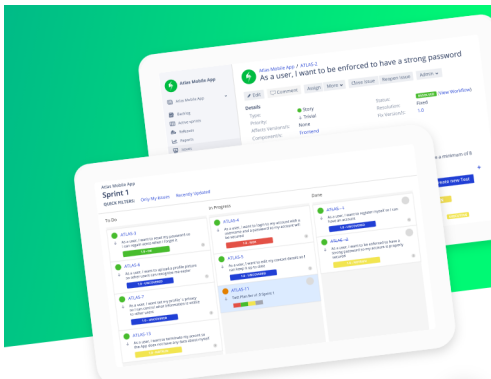
Las tareas de automatización de pruebas están cada vez más presentes en el mercado para automatizar las tareas de prueba. Existen una serie de herramientas de automatización, pero es poco probable que una sola herramienta pueda automatizar todas las tareas de prueba. La mayoría de las herramientas se centran en una tarea o grupo de tareas específicas, aunque algunas sólo tratan un aspecto de la tarea.



ii. Algunas Herramientas de gestion de Pruebas

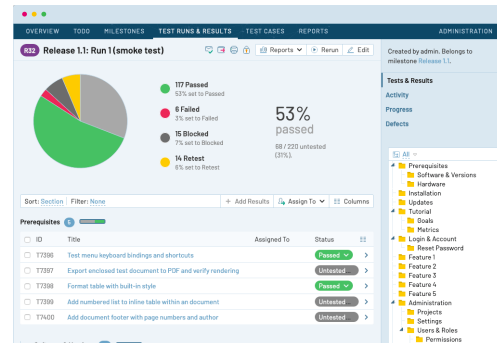
- XRAY

Radiografía es una de las herramientas de gestión de pruebas preferidas para pruebas manuales y automatizadas. Proporciona la estructura adecuada para organizar y clasificar conjuntos de pruebas y proporciona resultados de prueba eficientes en menos tiempo



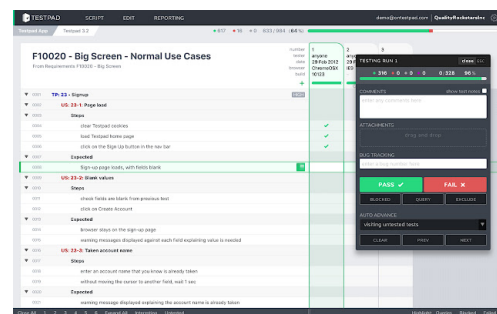
- TESTRAIL

TestRail es una herramienta de gestión de casos de prueba basada en la web que se puede configurar y utilizar fácilmente con la nube o la configuración local. Es altamente escalable y personalizable. Puede ver información en tiempo real sobre el progreso de las pruebas a través de paneles interactivos, métricas, informes de actividad, etc. Los casos de prueba automáticos y manuales se pueden administrar y documentar fácilmente mediante capturas de pantalla, comparación de resultados esperados versus reales.



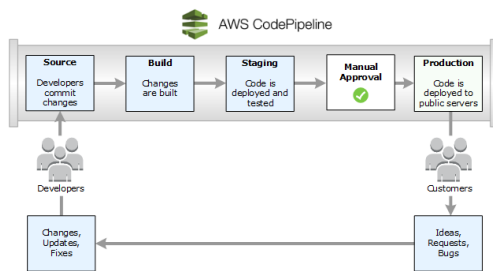
- TESTPAD

Testpad utiliza planes de prueba inspirados en listas de verificación para pruebas ágiles, pruebas exploratorias, gestión de casos de prueba tradicional, BDD resaltado por sintaxis y mucho más. Es una herramienta liviana con un editor controlado por teclado y tiene una interfaz de usuario altamente receptiva impulsada por JavaScript.

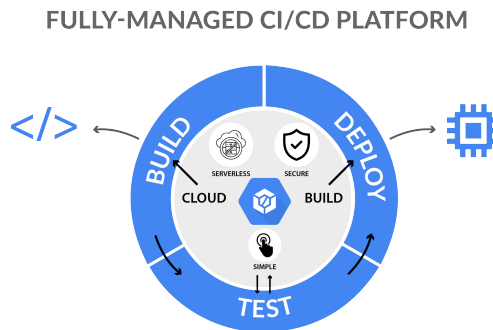


- AWS CodePipeline

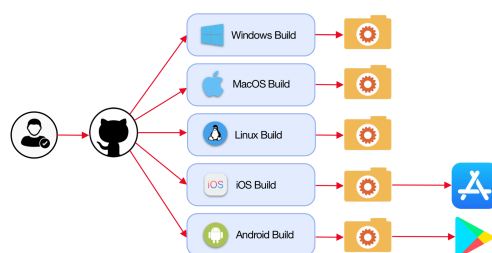
Servicio de entrega continua para actualizaciones de aplicaciones rápidas y confiables. CodePipeline compila, prueba e implementa su código cada vez que hay un cambio de código, según los modelos de proceso de lanzamiento que defina. ofrece modelado de flujo de trabajo, integraciones de AWS y complementos prediseñados.



- **Google Cloud Build**
Cloud Build permite crear software rápidamente. control completo sobre la definición de flujos de trabajo personalizados para compilar, probar e implementar en múltiples entornos, como VM, sin servidor, Kubernetes o Firebase.



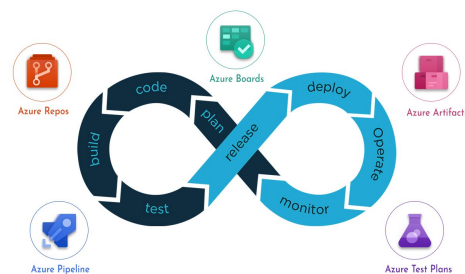
- **Github Actions**
GitHub Actions es una herramienta que permite reducir la cadena de acciones necesaria para la ejecución de código, mediante la creación de un flujo de trabajo encargado del Pipeline. Siendo configurable para que GitHub reaccione a ciertos eventos de forma automática según nuestras preferencias.



- **Azure DevOps**
Azure DevOps proporciona alojamiento Git privado ilimitado, compilación en la nube para la integración continua, planificación ágil y administración de versiones para la entrega continua en la nube y en las instalaciones. Incluye amplio soporte IDE. Azure DevOps es una herramienta en la categoría de herramientas de entorno de desarrollo integrado de una pila tecnológica. Usado por empresas como Microsoft, QRPoint, BigClarity iOLAP.

Características:

- Herramientas ágiles: tableros kanban, trabajos pendientes, tableros scrum
- Informes: paneles, widgets, Power BI
- Git: repositorios privados gratuitos, solicitudes de extracción
- Integración continua: construcciones y diagnósticos automatizados
- Agentes de compilación en la nube: agentes multiplataforma para Windows, Mac y Linux
- Herramientas de prueba: pruebas unitarias, pruebas de carga, pruebas manuales, exploratorias y de aceptación del usuario
- Gestión de versiones: automatic implementaciones, flujos de trabajo de aprobación cerrados, pistas de auditoría
- Administración de paquetes: paquetes npm y NuGet de host
- Integración: enlace de código y versiones a elementos de trabajo, compilaciones y resultados de pruebas.



IV. COMPARATIVA

	AWS CODEPIPELINE	GOOGLE CLOUD BUILD
Pros	Facil de configurar Servicio gestionado Integración con github Ejecución paralela Despliegue automatico	Basado en contenedor Facil integración con GCP. Implementa en varios entornos, como VM, Kubernetes, Firebase o sin servidores.
Contras	Sin tableros de proyectos Sin integración con las herramientas "Power" 365.	Dependencia de un proveedor
Empresas que utilizan	CRED, HENNGE, Educación Quero	Caredoc, Imroz, Nitra, Bux
Herramientas que se integran con	Github, Jenkins, Amazon S3, EC2	Google Cloud Run, WhiteSource, Doppler

V. CONCLUSIONES

Las pipelines han convertido en dos de los grandes proveedores de cloud pública y ofrecen tantísimos servicios que podemos encontrar sin dificultad soluciones que se adapten a nuestras necesidades, sean cuáles sea. nos permiten disponer en la nube de bases de datos, redes, herramientas de desarrollo y gestión, analytics, machine learning y mucho más. CodePipeline es otra característica revolucionaria de AWS que simplifica la forma de administrar su conjunto de herramientas CI / CD. Se integra con herramientas como GitHub, Jenkins y CodeDeploy que le permiten controlar visualmente el flujo de actualizaciones de aplicaciones desde la compilación hasta la producción. Se divide todo el ciclo de vida de la aplicación en una serie de etapas con cada etapa que consiste en una acción que se iniciará en su código. Al empujar una compilación al repositorio de Git, CodePipeline inicia la secuencia de acciones y pasa su código a cada etapa consecutiva automáticamente.

VI. RECOMENDACIONES

Utilice el cifrado y la autenticación para los repositorios de origen que se conectan a sus canalizaciones. Utilice las prácticas recomendadas que se proporcionan en esta sección para las canalizaciones con un proveedor de acciones de Jenkins. Puede crear canalizaciones que se integran con otras AWS Servicios de . Puede ser AWS Servicios de, como Amazon S3 o productos de terceros, como, por ejemplo, GitHub.

REFERENCES

- [1] Practicas recomendadas (2021)
[https://docs.aws.amazon.com/es_es/codepipeline/latest/userguide/best – practices.html/](https://docs.aws.amazon.com/es_es/codepipeline/latest/userguide/best-practices.html/)
- [2] Alvarado Abarca, Miguel(2017)Metodología de implementación de aplicaciones de software
<https://repositorio.una.ac.cr/handle/11056/14176>
- [3] Sierra Martin,Miguel (2021)Prácticas DevOps para la automatización de la integración
<https://oa.upm.es/68239/>
- [4] Mercado-Ramos, V. H., Zapata, J., Ceballos, Y. F. (2015). Herramientas y buenas prácticas para el aseguramiento de calidad de software con metodologías ágiles. Revista de Investigación, Desarrollo e Innovación
https://revistas.uptc.edu.co/index.php/investigacion_uitama/article/view/3277
- [5] Escobar-Rivera, Dalilis; Moreno-Pino, Mayra Rosario; Cuevas-Rodríguez, Luis (2016).La calidad de la auditoría en Sistemas de Gestión. Software
<http://efaidnbmnnnibpcajpcglclefindmkaj/viewer.html?pdfurl=https>
- [6] Tayché Capote-García (2015), Perspectivas del cuadro de mando integral personalizadas para laboratorios de pruebas de software.
<https://rii.cujae.edu.cu/index.php/revistaind/article/view/697>
- [7] Mera Paz, Julián Andrés (2016) , Análisis del proceso de pruebas de calidad de software.
<https://repository.ucc.edu.co/handle/20.500.12494/962>
- [8] Herramientas para la gestión de calidad de software (2015)
<http://www.pmoinformatica.com/2015/04/herramientas-gestion-calidad-software.html>