[INTEGRACIÓN CONTINUA]

[ENTREGA 7 - ESCENARIO 8]

Yesid David Martinez Sarmiento

Fabian Arley Méndez Farfán

Juan Pablo Monsalve Montes.

Jorge David Roa Espitia

Astrid Yurani Prieto Gomez

JUNIO DE 2021

POLITÉCNICO GRAN COLOMBIANO

INGENIERÍA DE SOFTWARE

Contenido

[1. HERRAMIENTAS: 3](#_Toc75856620)

[2. LINK DEL PROYECTO: 3](#_Toc75856621)

[3. CONCLUSIONES SOBRE LAS HERRAMIENTAS: 3](#_Toc75856622)

## HERRAMIENTAS:

* **Github**: Repositorio código fuente
* **Docker**: Empaquetar la aplicación
* **Github Projects**: Administrar las tareas del proyecto - reemplaza a ZenHub con la ventaja de que está integrado en Github, es gratis y no requiere crear una cuenta en otra página
* **Netlify**: Para desplegar automáticamente la aplicación cuando se realice un cambio
* **Playwright**: Para realizar prueba de integración de la aplicación
* **Github Actions**: Para compilar y probar automáticamente la aplicación cuando se realice un cambio

## LINK DEL PROYECTO:

<https://github.com/famendez5/integracion-continua>

## CONCLUSIONES SOBRE LAS HERRAMIENTAS:

* 1. Se evidencia que, al trabajar en un repositorio como GitHub, facilita el trabajo en grupo, ya que permite la modificación de un mismo proyecto por varias personas, almacenar esta información y llevar un registro de los cambios que se realizan y así mismo revertir si es necesario. Ya que se trabaja en ramas distintas las cuales posteriormente se van a poder integrar mediante un pull request que otro integrante del proyecto debe validar para asegurar la calidad del código.
  2. El uso de docker nos permite configurar el ambiente de cada componente del proyecto de manera que cuente exactamente con las dependencias que requiere para ejecutarse, lo cual nos da la posibilidad de replicar dichos ambientes sin necesidad de configurarlos cada vez.
  3. Al implementar Docker en un entorno de desarrollo, reducimos el tiempo a segundos ya que no arranca en un sistema operativo completo, lo que lo hace mucho más manejable para eliminarlo y volverlo a crear si arriesgar todos los datos y trabajando en conjunto con un repositorio GitHub es más sencillo ya que permite descargar como archivo el código fuente y hacer un seguimiento de errores mucho más detallado ya que interactúan varios integrantes del grupo
  4. Al realizar un trabajo compartido no solo se integran las personas sino también el conocimiento que tiene cada una aportando en los puntos del trabajo y se logra hacer de forma satisfactoria en relación al objetivo principal que era crear dos temas y darle su respectiva conexión.
  5. Github hace el trabajo en equipo más ágil y sencillo, ayuda a la detección de fallos, a disminuir errores humanos, al seguimiento por etapas del proyecto, al mantenimiento de diferentes entornos, etc. Al alojar proyectos como repositorios en Github, es una decisión correcta, ya que para repositorios públicos te beneficias de los conocimientos de otros programadores y en el caso de los privados, dispondrás de un sistema robusto de gestión de proyectos que hará que el trabajo en equipo sea mucho más rápido.
  6. Usando Docker evitamos tener que descargar y almacenar gran número de librerías. Puesto que podremos desarrollar cada aplicación en contenedores independientes además nos permitirá ahorrar tiempo a la hora de hacer los testeos. Abrir un contenedor es tarea rápida, por lo tanto, el tiempo que invertimos es prácticamente todo al testeo, sin largas esperas.
  7. Netlify es más que un hosting gratuito para alojar webs estáticas. Constituye una potente herramienta que permite completar el ciclo de despliegue continuo en nuestros proyectos web de manera rápida y segura.
  8. Playwright espera inteligentemente a que se completen las actividades del navegador antes de realizar acciones, por lo que el desarrollador no tiene que preocuparse de realizar una programación adicional para este paso