[INTEGRACIÓN CONTINUA]

[ENTREGA 1 - ESCENARIO 3]

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

[**Introducción** 3](#_Toc75811025)

[1. Crear repositorio en GitHub en donde creen un txt y cada uno escriba su nombre: 4](#_Toc75811026)

[2. Crear dos contenedores y subir los contenedores a GitHub: 4](#_Toc75811027)

[3. Comunicación entre los dos contenedores: 5](#_Toc75811028)

[4. Paso a Paso para la creación de los contenedores: 5](#_Toc75811029)

[**Contenedor LibreOfficeConvert** 5](#_Toc75811030)

[**Contenedor AppWeb** 5](#_Toc75811031)

[5. Conclusiones sobre las herramientas: 11](#_Toc75811032)

# **Introducción**

En este trabajo se trataron temas que no son muy comunes en el diario vivir y que algunos desconocemos los programas que se utilizaron para ello, como lo son Github y Docker, dos herramientas informáticas para realizar documentos y en este cada integrante desde la comodidad de la casa puede hacer sus aportes.

También se va a dar a conocer a LibreOfficeConver quien es la aplicación encargada de convertir un documento a formato PDF y una AppWeb que es la que nos va a permitir interactuar con la aplicación mencionada anteriormente, también se verá cómo se pueden complementar y configurar la comunicación que deben de tener entre ellas.

## Crear repositorio en GitHub en donde creen un txt y cada uno escriba su nombre:

El repositorio se encuentra en la URL <https://github.com/famendez5/integracion-continua> y el nombre de los integrantes se encuentra en el archivo [INTEGRANTES.txt](https://github.com/famendez5/integracion-continua/blob/main/INTEGRANTES.txt).

## Crear dos contenedores y subir los contenedores a GitHub:

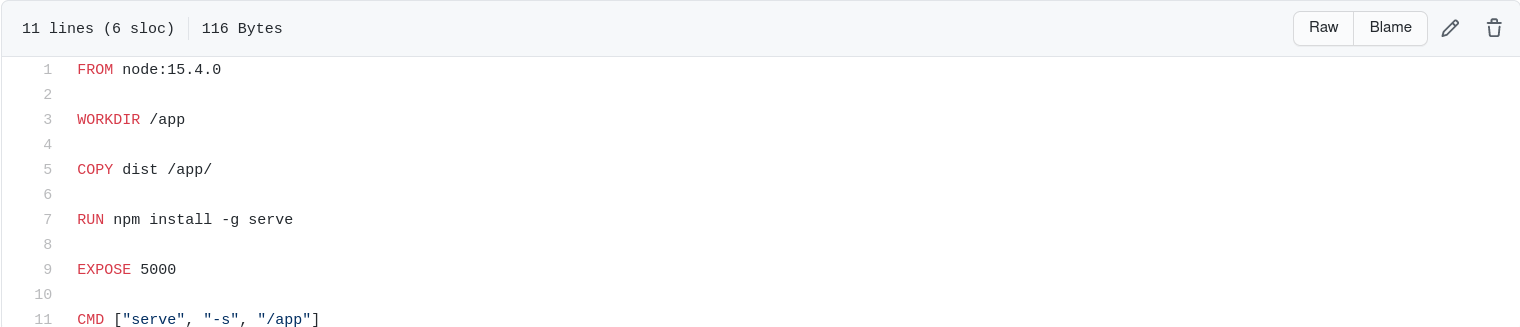
**Contenedor LibreOfficeConvert**

Esta imagen se encargará de convertir el documento a PDF. Esta imagen está publicada en Docker Hub: [fabianmendez/libreofficeconvert:0.1.0](https://hub.docker.com/repository/docker/fabianmendez/libreofficeconvert)

**Contenedor AppWeb**

Esta imagen es el Front que verá el usuario final. Esta imagen debe construirse porque no está publicada en Docker Hub ni ningún otro repositorio.

Esta imagen se encuentra en el archivo [integracion-continua/app/Dockerfile](https://github.com/famendez5/integracion-continua/blob/main/app/Dockerfile) en el repositorio:

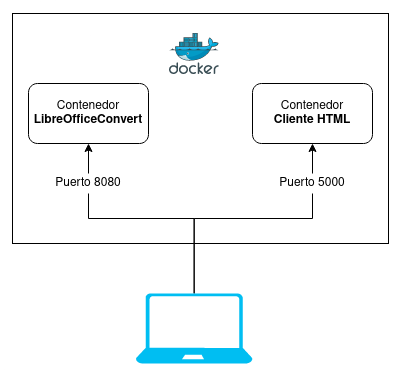


Se usa como base la imagen node:15.4.0, se instala el paquete serve de nodejs, se copia la aplicación web compilada y se define que al ejecutar el contenedor se ejecute el paquete serve en modo SPA con la aplicación guardada en la ruta /app.

## Comunicación entre los dos contenedores:

La comunicación entre los contenedores se realiza mediante una petición http de tipo POST, se envía desde el Front (App Web) al backend (LibreOfficeConvert) el cual convierte el documento a PDF y lo retorna al front.

La arquitectura de la solución se evidencia en el siguiente diagrama:



## Paso a Paso para la creación de los contenedores:

### **Contenedor LibreOfficeConvert**

Esta imagen está publicada en dockerhub y por lo tanto basta con ejecutar el siguiente comando para correr el servicio:

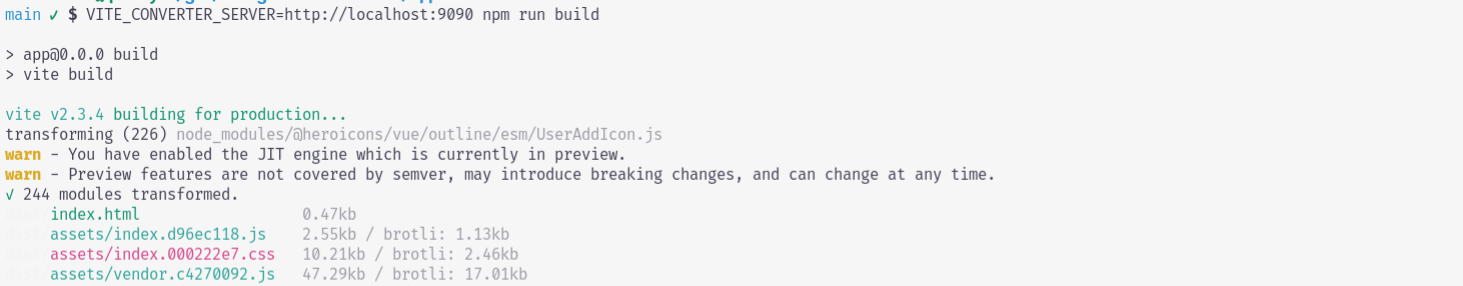
*docker run --rm -p 9090:8080 fabianmendez/libreofficeconvert:0.1.0*

En este ejemplo, se mapea el puerto 9090 de la máquina host al puerto 8080 del contenedor, en el cual el servicio está a la escucha.

### **Contenedor AppWeb**

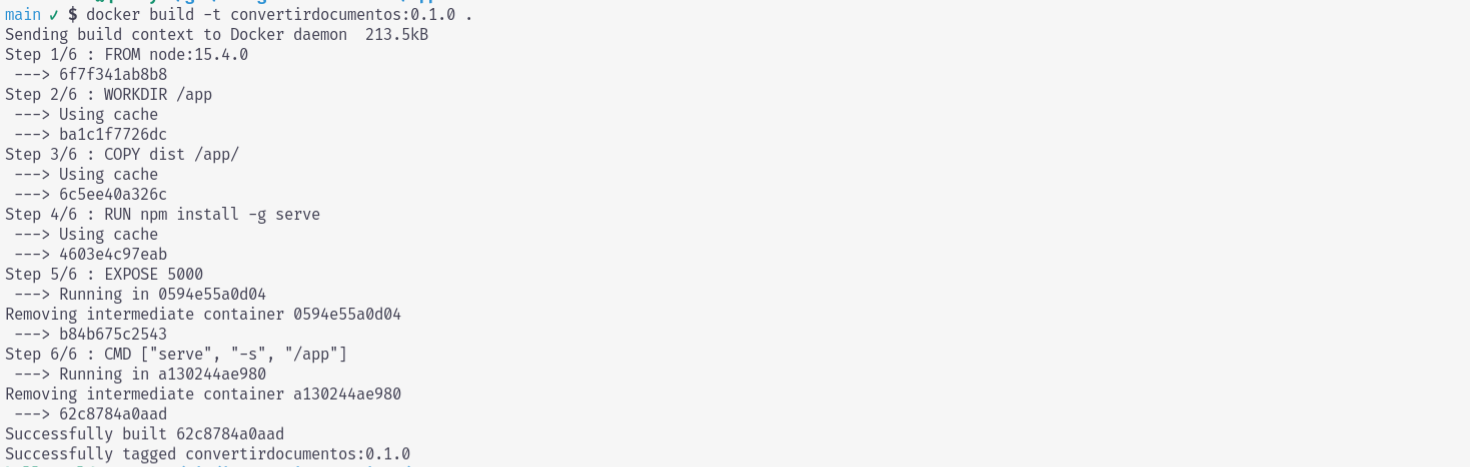
Primero se debe compilar la aplicación, ejecutando el siguiente comando en el directorio app:

*VITE\_CONVERTER\_SERVER=http://localhost:9090 npm run build*



La aplicación compilada queda en el directorio dist. Ahora se ejecuta el siguiente comando para generar la imagen docker:

*docker build -t convertirdocumentos:0.1.0 .*



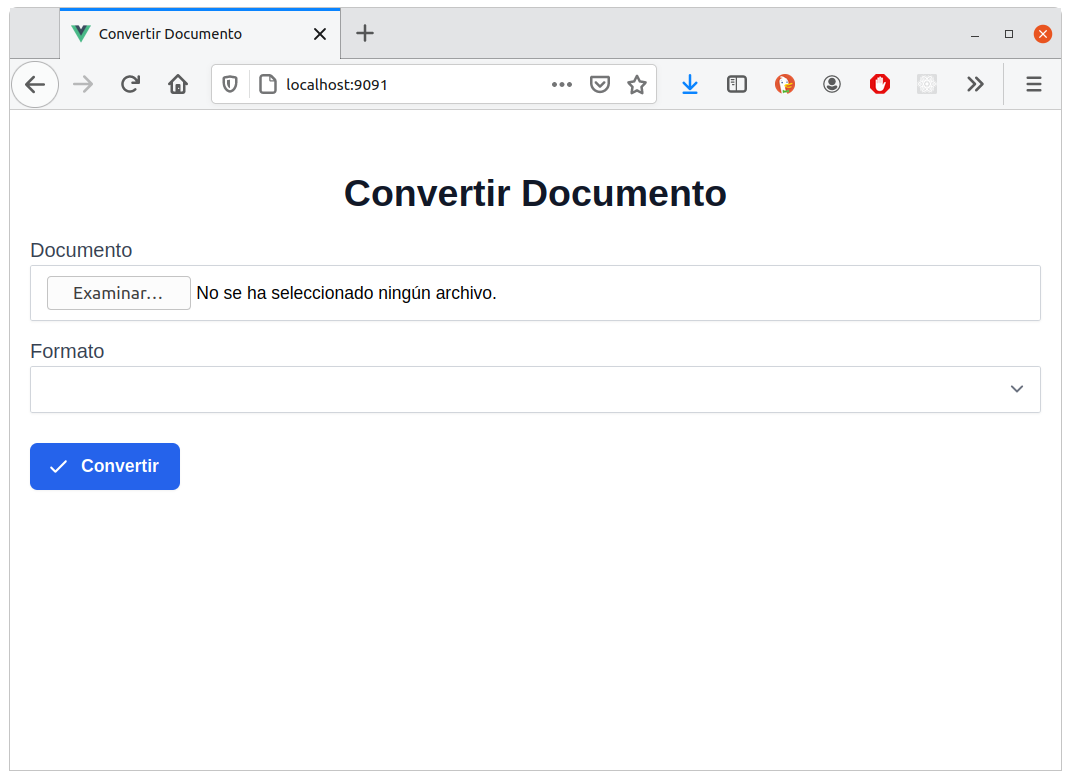
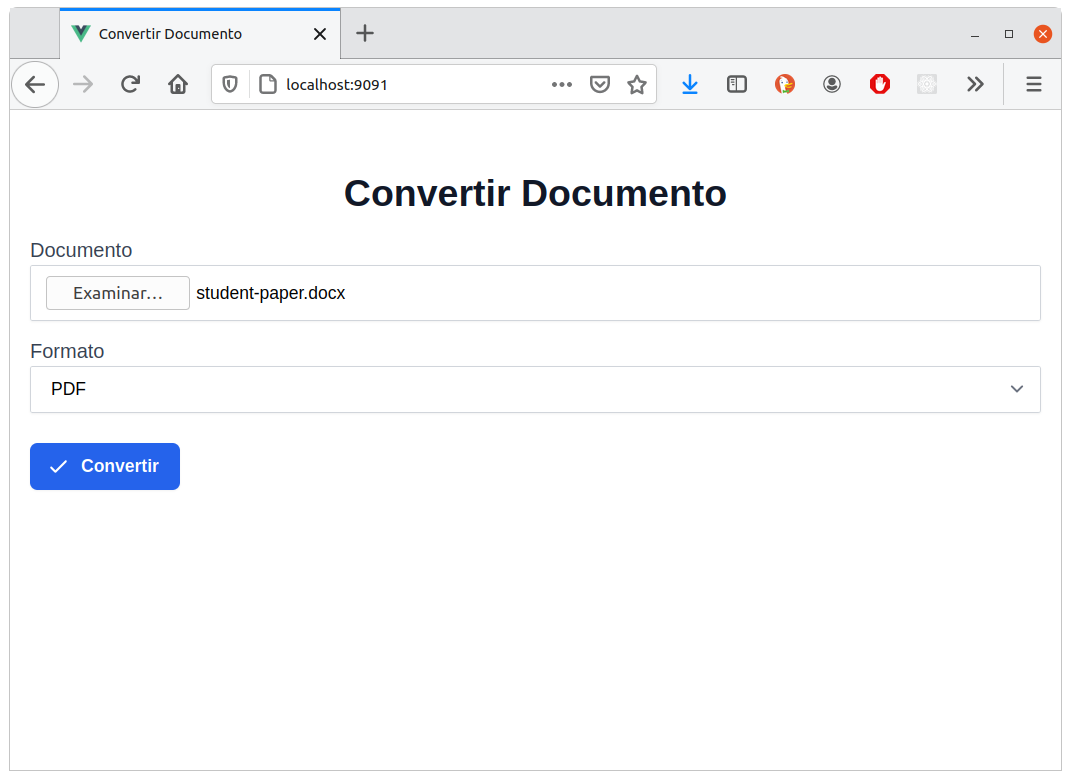
Finalmente, se inicia el contenedor con el siguiente comando:

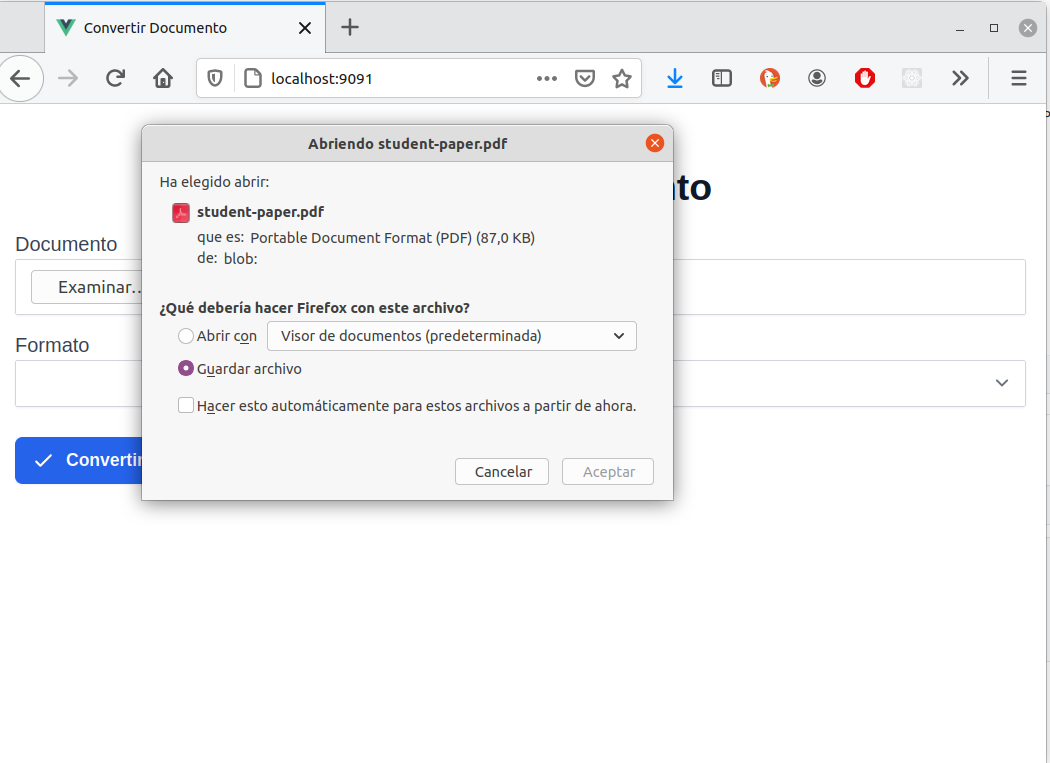
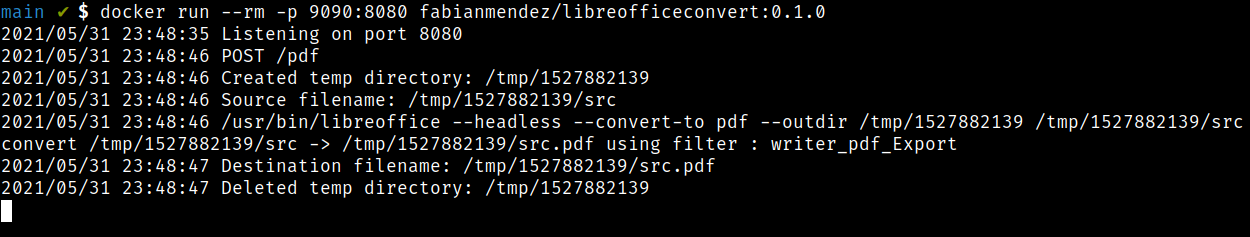
*docker run --rm -p 9091:5000 convertirdocumentos:0.1.0*



En este ejemplo, se mapea el puerto “9091” de la máquina host al puerto “5000” del contenedor, en el cual el servicio HTTP está a la escucha.

Por último se ingresa a la url http://localhost:9091

 Seleccionamos el documento y el formato:

Al hacer clic en el botón aceptar, después de unos segundos se obtiene el documento convertido:Adicionalmente, en los logs del servicio libreofficeconvert se evidencia que se ha procesado el documento

## 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