

---

## Tarea 1

### Servicios en contenedores

---

|                      |                      |
|----------------------|----------------------|
| Fecha de asignación: | 9 de Setiembre, 2020 |
| Grupos:              | Individual           |

|                   |                            |
|-------------------|----------------------------|
| Fecha de entrega: | 23 de Setiembre, 2020      |
| Profesores:       | Jason Leitón, Luis Barboza |

---

## 1. Objetivo

Implementar un servidor en un contenedor, con el fin de crear un ambiente controlado y aislar la ejecución del SO.

## 2. Atributos a evaluar

- Aprendizaje continuo. Se requiere que el estudiante valore las estrategias y el conocimiento adquirido para alcanzar el objetivo.
- Herramientas de Ingeniería. Se requiere que el estudiante sea capaz de adaptar técnicas, recursos y herramientas modernas para la solución de problemas.

## 3. Motivación

La virtualización de los recursos es una técnica que ha estado en pleno auge, debido a que cada vez más se requiere aplicaciones independientes de la arquitectura, además de existen equipos más especializados con los cuales se puede ejecutar mejor las tareas computacionales. Esta técnica está estrechamente relacionadas con los SO porque implica administrar los elementos del hardware, por lo que es de suma importancia que los Ingenieros en Computadores se familiaricen con los entornos de virtualización tanto a nivel de hardware como a nivel de empaquetamiento como lo son los contenedores.

## 4. Descripción

Cada estudiante creará una imagen de Docker la cual permitirá recibir imágenes en orden secuencial, una por una, utilizando sockets. Para que una conexión con el socket del contenedor sea aceptada, el programa que envía el dato debe pertenecer a una lista pre aprobada de IPs, o de no seguras, las cuales no se procesa nada. Cuando ha recibido los datos, debe procesar la imagen y decidir cuál es su color principal, si rojo, azul o verde. Y colocar la imagen en la carpeta destinada a ese color dentro de los volúmenes montados (SO host).

### 4.1. Algoritmo de color

El algoritmo debe recibir una imagen y colocarla en una de tres carpetas: R, G, B. Dependiendo de cuál sea su color predominante. El algoritmo que calcula el color, puede ser importado de cualquier biblioteca de uso público que el estudiante prefiera.

### 4.2. Archivo de configuración

Será el encargado de tener una lista con todas las IPs de los hosts que pueden enviar las imágenes, y de aquellos de los cuales se recibe la conexión, se acepta la imagen, pero no se procesa. Dicha lista se encontrará en un archivo ubicado directamente en el host (es decir, no en la imagen del contenedor) por lo que cambiará dependiendo del host que lo corra. El volumen que contiene el archivo no debe montarse desde el dockerfile, si no, ser cargado al momento de correr el contenedor. El archivo debe llamarse `configuracion.config`.

### 4.3. Comunicación

En el archivo `configuracion.config` se listan los “trusted hosts”, es decir, aquellos de quienes el contenedor sabe que es seguro recibir la información. Por lo que si recibe un mensaje proveniente de cualquier otro IP, el contenedor no debe aceptar la conexión. Estos trusted hosts también pueden ser del tipo de aceptados pero de los cuales no procesarán la imagen. Esta solamente se ubicará en una carpeta llamada: “Not trusted”

### 4.4. Las carpetas

El volumen a ser montado por el contenedor debe ser el mismo siempre, es decir, si corren varios contenedores de la misma imagen en un determinado host, todos deben montar el mismo volumen. Por lo que el contenedor deberá crear una carpeta dentro de este volumen para el uso exclusivo de las imágenes que procese únicamente él. Dentro de esta carpeta se encontrarán CUATRO más: R,G,B y Not trusted.

### 4.5. Programa que envía las imágenes (cliente)

Deberá recibir como único parámetro para ejecutarse, la ip donde debe enviar las imágenes. Una vez corriendo, debe preguntar por la ruta de la siguiente imagen a enviar. Una vez enviada la imagen, volverá a preguntar por otra ruta y así sucesivamente. Si el usuario desea finalizar el programa, simplemente en la ruta debe escribir fin y el programa será finalizado. Este programa NO debe de ejecutarse dentro de un contenedor, si no que en la máquina local.

#### 4.6. Requerimientos técnicos

- El contenedor se debe crear a partir de una imagen oficial, se recomienda Centos.
- Este proyecto se debe realizar en el lenguaje de programación C. Pueden utilizar las bibliotecas que sean necesarias.
- Debe ser implementado en Linux y se debe proporcionar todos los comandos en el makefile. No se permite realizar configuración alguna una vez ejecutado el contenedor.

### 5. Documentación- Estilo IEEE-Trans (máximo 5 páginas)

- Introducción: Teoría necesaria, breve descripción del proyecto y qué es lo que se espera en el escrito.
- Ambiente de desarrollo: Todos los detalles de implementación y herramienta durante el desarrollo del proyecto.
- Atributos: Esta sección deben de describirse cuales atributos fueron reforzados durante el desarrollo del proyecto (documento aparte que explique cómo se reforzó).
- Detalles del diseño del programa desarrollo, tanto del software como del hardware (en caso de que aplique): Diagramas UML, imágenes, descripciones entre otros, todo lo que sea necesario para entender de una mejor manera el diseño y funcionamiento del proyecto.
- Instrucciones de cómo se utiliza el proyecto.
- Tabla de actividades por cada estudiante: bitácora con el total de horas trabajadas.
- Conclusiones
- Sugerencias y recomendaciones.
- Referencias

### 6. Entregables

- Código fuente con documentación interna.
- Documentación.
- Archivos necesarios para ejecutar el programa.

## 7. Evaluación

- Contenedor 35 %
- Integración 10 %
- Servidor 15 %
- Cliente 15 %
- Archivos de configuración y makefiles 5 %
- Documentación 20 %

## 8. Fecha de entrega

- 23 de setiembre. 23:59 por tecdigital

## 9. Otros aspectos administrativos

- Para la revisión del proyecto se debe de entregar tanto la documentación como la implementación del software.
- No se reciben trabajos después de la hora indicada.
- En la revisión del proyecto pueden estar presentes el coordinador y asistente.
- Es responsabilidad del estudiante proveer los medios para poder revisar la funcionalidad del software, por ejemplo, si no se realiza la interfaz, se debe de proporcionar otro medio para la verificación, de lo contrario la nota será cero en los rubros correspondientes a la funcionalidad faltante.