

Trabajo Práctico

Final - Enunciado

Entorno de Programación

Introducción

El objetivo del trabajo práctico es diseñar y escribir un programa para procesar un lote de imágenes.

Este programa consta de tres partes principales:

- Generación y descarga de imágenes a través de un servicio web.
- Aplicación de una transformación solo a las imágenes de personas.
- Generación de un archivo comprimido con las imágenes procesadas.

Todo el trabajo debe ser realizado bajo control de versiones, con participación de todos los integrantes y debe ejecutarse dentro de un contenedor.

Pautas generales

- El trabajo práctico deberá ser realizado en grupos de dos o tres estudiantes.
- Deberá informarse al docente de la comisión cómo está conformado su equipo de trabajo vía mail.
- Todos los integrantes deben conocer todos los aspectos del trabajo entregado.
- Se admite una única entrega final, por lo que se recomienda revisar detenidamente las funcionalidades antes de la entrega formal, que debe comunicarse vía mail al docente de la comisión.

Aspectos evaluativos

- Programación bash.
- Manejo de control de versiones.

- Uso de contenedores.
- Generación de documentación.

Enunciado

El trabajo consiste en generar un contenedor que, al ejecutarse, presente un menú de opciones para procesar un lote de imágenes. Deberán editar un Dockerfile y construir una imagen.

Las imágenes a analizar deben ser descargadas o generadas desde internet y almacenadas dentro del contenedor para su posterior análisis. Luego del análisis, deberá generarse un archivo disponible fuera del contenedor.

Tanto los scripts como el Dockerfile deben estar en el repositorio GitLab o GitHub que creen para realizar el desarrollo en equipo.

Deberán trabajar en el repositorio manteniendo la prolijidad y las buenas prácticas de Git. Se recomienda crear una rama por cada ejercicio e ir integrando a la rama principal a medida que los scripts que resuelven cada ejercicio estén listos. Todos los integrantes deben realizar commits en el repositorio.

En ese mismo repositorio debe estar la documentación suficiente para comprender cómo desplegar el contenedor y ejecutar la aplicación, con algunas capturas de ejecución o cualquier otro elemento que consideren necesario para explicar el funcionamiento y visualizar los resultados. Esta documentación debe estar presente en un archivo README.md al acceder al repositorio en GitLab o GitHub.

Scripts

Se deberán programar los siguientes scripts:

generar.sh

Genera imágenes utilizando algún servicio web. Se recomienda tener en cuenta que al descargar desde una página conviene usar un sleep entre descarga y descarga para no saturar el servicio y

evitar problemas. Se debe poder indicar por argumento cuántas imágenes generar, y se deben asignar nombres de archivo al azar de una lista de nombres de personas. Luego, se deben comprimir las imágenes y generar un archivo con su suma de verificación.

descomprimir.sh

Se debe poder indicar por argumento dos archivos: uno con las imágenes comprimidas y otro con una suma de verificación. Si ocurre algún error, se debe informar al usuario; de lo contrario, se procede a descomprimir las imágenes.

procesar.sh

Se deberán recortar las imágenes a una resolución de 512x512 utilizando alguna utilidad como ImageMagick. Solamente deben procesarse aquellas imágenes que



Necesito un README.md para describir el programa creado

ChatGPT

Programa de Procesamiento de Imágenes

Este programa es una aplicación diseñada para procesar un lote de imágenes, aplicando transformaciones específicas a las imágenes de personas. Proporciona opciones para generar, descargar, procesar y comprimir imágenes, así como generar archivos con información relevante.

Características principales

- Generación y descarga de imágenes desde un servicio web.
- Recorte de imágenes a una resolución de 512x512.
- Procesamiento exclusivo de imágenes de personas válidas.
- Compresión de imágenes y generación de un archivo comprimido.
- Generación de diversos archivos con información estadística.

Requisitos del sistema

- Sistema operativo: Linux, macOS o Windows (con Git Bash u otro emulador de terminal).
- Docker instalado para ejecutar el programa dentro de un contenedor.

Instalación y configuración

Clona el repositorio del programa desde GitHub:

```
bash
```

Copy code

```
clone
```

Accede al directorio del repositorio:

```
bash
```

Copy code

```
cd
```

Construye la imagen de Docker utilizando el Dockerfile proporcionado:

```
bash
```

Copy code

Ejecuta el contenedor utilizando la imagen creada:

```
bash
```

Copy code

Uso del programa

El programa ofrece un menú con las siguientes opciones:

Generar y descargar imágenes: Permite generar y descargar imágenes utilizando un servicio web. Debes indicar la cantidad de imágenes a generar.

Procesar imágenes: Realiza el procesamiento de las imágenes generadas. Recorta las imágenes a una resolución de 512x512 y aplica transformaciones solo a las imágenes de personas válidas.

Comprimir imágenes: Genera un archivo comprimido que incluye las imágenes procesadas y archivos con información estadística.

Salir: Finaliza la ejecución del programa.

Para seleccionar una opción, ingresa el número correspondiente y presiona Enter.

Estructura del repositorio

- README.md: Documentación del programa, proporcionando instrucciones de instalación, uso y configuración.
- Dockerfile: Archivo para construir la imagen de Docker que contiene el entorno de ejecución del programa.
- generar.sh: Script para generar y descargar imágenes utilizando un servicio web.
- procesar.sh: Script para procesar las imágenes, recortándolas y aplicando transformaciones específicas.
- comprimir.sh: Script para comprimir las imágenes y generar archivos con información estadística.
- menu.sh: Script que muestra el menú de opciones y maneja la interacción con el usuario.

Contribución

- Si deseas contribuir a este proyecto, puedes seguir los pasos a continuación:
 - Realiza un fork del repositorio.
 - Crea una rama con la nueva funcionalidad o corrección: `git checkout -b nueva-funcionalidad`.

Realiza los cambios necesarios y haz commits: `git commit -m "Agrega nueva funcionalidad"`.

Envía tus cambios al repositorio remoto: `git push origin nueva-funcionalidad`.

Crea una Pull Request en GitHub para que los cambios sean revisados y fusionados.

Autores

- Palandri David Jose
- Dito Daniela
- Pinazo Silvana