Entrega 5 – Sistema de Conversión Cloud

Daniel Andrés Jiménez Riveros​

Andrés Martín Ochoa Toro​

Esteban Emmanuel Ortiz Morales​

Manuel Felipe Porras Tascón

Desarrollo de Soluciones Cloud

Universidad de los Andes, Bogotá, Colombia

Enlace del Repositorio:

<https://github.com/AmOchoat/Cloud-Conversion-Tool>

Enlace Documentación Postman:

<https://documenter.getpostman.com/view/5689272/2s93CPrYBi>

**Contexto del problema**:

Una nueva compañía de cloud denominada *Cloud Conversion Tool* desea crear una aplicación web que será ofrecida gratuitamente a usuarios de internet para que estos puedan subir abiertamente diferentes formatos multimedia de archivos y cambiarles su formato o realizar procesos de compresión.

El modelo general de funcionamiento de la aplicación se basa en crear una cuenta en el portal web y acceder al administrador de archivos. Una vez la cuenta ha sido creada, el usuario puede subir archivos y solicitar el cambio de formato de estos para descargarlos. La aplicación web le permite a un usuario convertir archivos multimedia en línea de un formato a otro, seleccionando únicamente el formato destino.

En esta primera versión, la herramienta sólo soporta la compresión de archivos en con tres tipos diferentes de algoritmos y utilidades, estos algoritmos son ZIP*,* TAR*.*GZ y TAR.BZ2.

**Implementación de la solución**:

Para brindar una solución a la compañía *Cloud Conversion Tool,* se realizó la implementación del API REST que permite la creación de una cuenta (*sign up)*, el inicio de sesión(*login*), la creación de una nueva tarea de compresión, el listado de todas las tareas de compresión de un usuario, borrar una tarea de compresión de un usuario y la obtención del archivo original o procesado.

Así mismo se realiza la implementación de una interfaz de usuario que permita consumir el API REST mencionada previamente.

**Arquitectura realizada:**

A partir de las necesidades manifestadas por el negocio, se definió el siguiente conjunto de diagramas conformado por el diagrama de clases, diagrama de componentes y diagrama de despliegue.

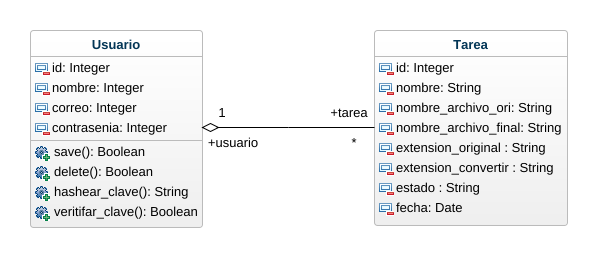


Imagen 1. Diagrama de clases entrega 3 *Cloud Conversion Tool*

En este diagrama se define el usuario y la tarea de conversión. Un usuario tiene muchas tareas de conversión, donde cada tarea tiene un nombre y nombre de los archivos generados, así como la extensión de inicio y final. Lo más importante en este modelo es el estado de una tarea, que puede ser *uploaded* y *processed,* y los nombres de los archivos. Para identificar un archivo en caso de ser necesario, se toma el nombre del archivo asociado a la tarea como valor de texto y se busca este nombre en un directorio de archivos internos.

Diagrama

Descripción generada automáticamente

Imagen 2. Diagrama de Componentes entrega 3 *Cloud Conversion Tool*

Nuestros componentes principales son una VPC que envuelve los otros componentes. Dentro de la VPC está el Front-end de la página web construido en Reactjs el cual está conectado al Back-end construido en Flask. Dentro de la VPC también se encuentra el API REST que tiene como objetivo conectar el backend con el frontend, además de tener a un componente Cloud storage para manejar los temas de almacenamiento de los archivos convertidos junto con los componentes PUB/SUB y Compression Worker que son los encargados de controlar todo el flujo de conversión de archivos de la aplicación.

**Diagrama

Descripción generada automáticamente**

Imagen 3. Diagrama de Despliegue entrega 5 *Cloud Conversion Tool*

En el caso de los diagramas de componentes y diagrama de despliegue se observa la relación de los elementos implementados y los servicios usados en su despliegue.

Se crea un ambiente de ejecución para la interfaz de usuario y se conecta mediante protocolo HTTP con un Application Server que tiene en ejecución el Front-End de la aplicación. Este Front-End mediante un protocolo HTTPS se conecta al Back-End que se encuentra desplegado en un Application Server que está conectado con el servicio desplegado de PUB/SUB que a su vez se conecta directamente mediante un protocolo HTTPS con el Compression Worker, el cual está desplegado con Cloud Run. Además, tanto el compression worker como el back-end mediante una conexión con un protocolo TCP con el Cloud storage y también se encuentran conectados con un VPC que esta exclusivamente desplegado para las fuentes de información que en este caso son para el Cloud Storage y para el Postgresql DB.