

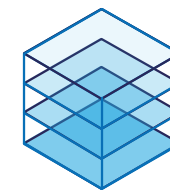
MISO

Maestría en Ingeniería de Software

Arquitectura

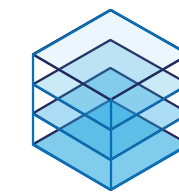
Cloud Conversion Tool

Camilo Ramírez Restrepo
Laura Daniela Molina Villar
Leidy Viviana Osorio Jiménez
Tim Ulf Pambor
Shadith Perez Rivera



Contenido

- Objetivo
- Vista de Contexto y dominio
- Vista Información
- Vista Funcional
- Vista de Despliegue



Objetivo

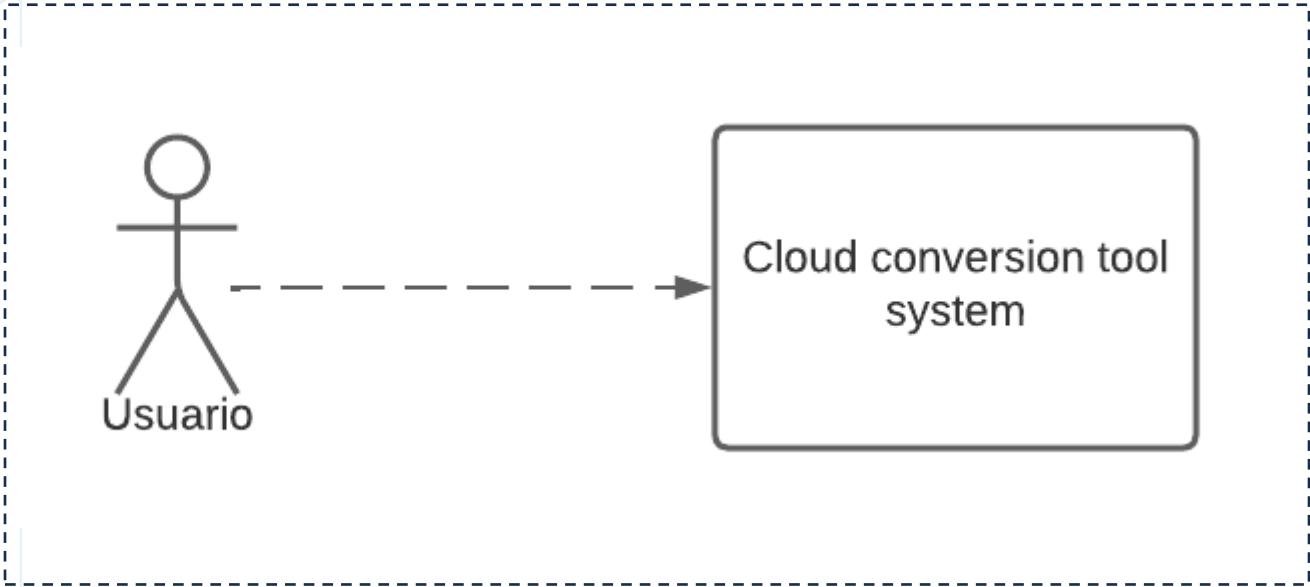
Proporcionar una plataforma eficiente y segura para que los usuarios conviertan archivos, puedan tener la administración de los archivos y tener una sección de registro y login, al mismo tiempo que se asegura la autenticación y control de acceso de los usuarios.



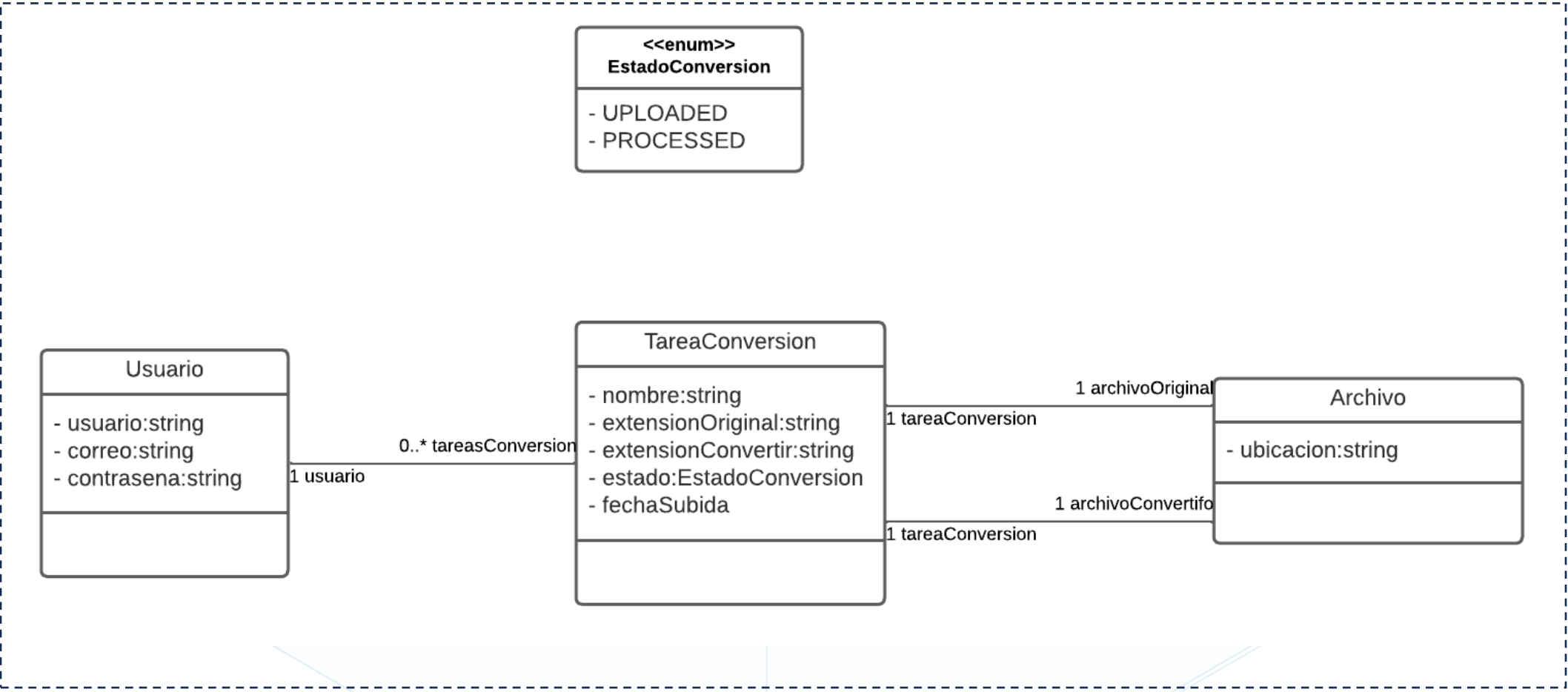
Vista de Contexto y Dominio

Proyecto	Cloud Conversion Tool	ID	VC-001	Elaboración	Equipo 1	Versión	1.0
Vista	Contexto y Dominio		Modelo	Contexto y Informacion			

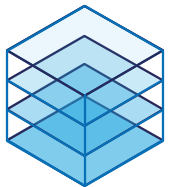
Convención
UML



Vista de contexto



Vista de dominio



Vista de Contexto y Dominio

Proyecto	Cloud Conversion Tool	ID	VC-001	Elaboración	Equipo 1	Versión	1.0
Vista	Vista de Contexto y Dominio		Modelo	Contexto e Información			

Razonamiento sobre el modelo

En el modelo de arquitectura que se desarrolló para la aplicación de Cloud Conversion Tool, algunas de las principales decisiones de arquitectura tomadas son las siguientes:

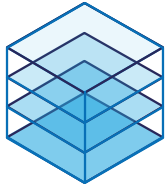
Arquitectura de Cliente-Servidor: La arquitectura de la aplicación se basa en un modelo cliente-servidor. El cliente (representado por el Usuario) se comunica con el servidor (Cloud Conversion Tool System) a través de servicios REST.

Escalabilidad y Mantenimiento: Al utilizar servicios REST, se crea una interfaz clara y estandarizada para que los clientes se comuniquen con el sistema. Esto facilita la escalabilidad, ya que se pueden agregar más servicios o servidores según sea necesario en el futuro. Además, la separación de la lógica del servidor y la interfaz de usuario facilita las actualizaciones y el mantenimiento.

Modelo de Datos: El modelo de datos se basa en tres clases principales: Usuario, TareaConversion, y Archivo. Cada entidad con sus pertinentes atributos y relaciones. La relación entre Usuario y TareaConversion es uno a muchos, lo que significa que un usuario puede tener múltiples tareas de conversión. Además, las relaciones 1 a 1 entre la clase TareaConversion y la clase Archivo (una para el archivo original y otra para el archivo convertido) es eficiente para gestionar y acceder a los archivos y el sistema almacena la ubicación de los archivos, lo que facilita la recuperación y descarga de los mismos.

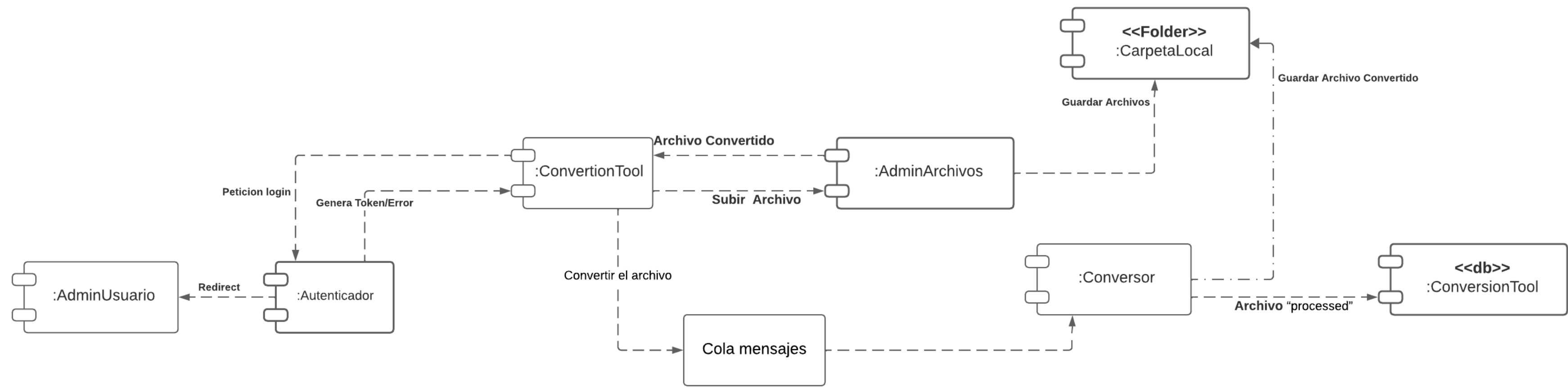
Autenticación de Usuarios: La autenticación de usuarios se gestiona a través de un token de sesión que se genera cuando un usuario inicia sesión correctamente. Los usuarios deben proporcionar un correo electrónico/usuario y contraseña para acceder a los recursos de la aplicación. Esto para garantizar la seguridad de la aplicación y controlar el acceso a los recursos.

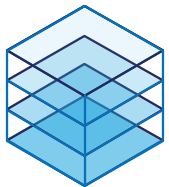
Vista Información



Proyecto	Sistema ABC Jobs	ID	VC-001	Elaboración	Equipo 1	Versión	1.0
Vista	Información		Modelo	Información			

Convención
UML





Vista de Información

Proyecto	Cloud Conversion Tool	ID	VC-004	Elaboración	Equipo 1	Versión	1.0
Vista	Información		Modelo	Información			

Razonamiento sobre las principales decisiones de arquitectura tomadas en este modelo

En esta vista de información, se observa una estructura que se centra en la autenticación, la gestión de archivos, la conversión de archivos y el almacenamiento de archivos. La inclusión de una cola de mensajes para la conversión de archivos es para para manejar de forma más eficiente múltiples solicitudes de conversión de manera ordenada. El sistema está diseñado para gestionar de manera práctica las operaciones relacionadas con la conversión y el almacenamiento de archivos.

ConversionTool:

Tenemos una relación con Autenticador para manejar la autenticación de los usuarios y se da una comunicación para la petición de inicio de sesión y la generación de tokens o errores de autenticación. La relación con Cola de mensajes para gestionar y procesar las solicitudes de conversión de manera eficiente. Y la relación con AdminArchivos para permitir a los usuarios subir archivos y acceder a los archivos convertidos.

Autenticador:

Relación con AdminUsuario para gestionar las operaciones de administración de usuarios, lo que incluye la creación de cuentas y la gestión de perfiles de usuario.

Conversor:

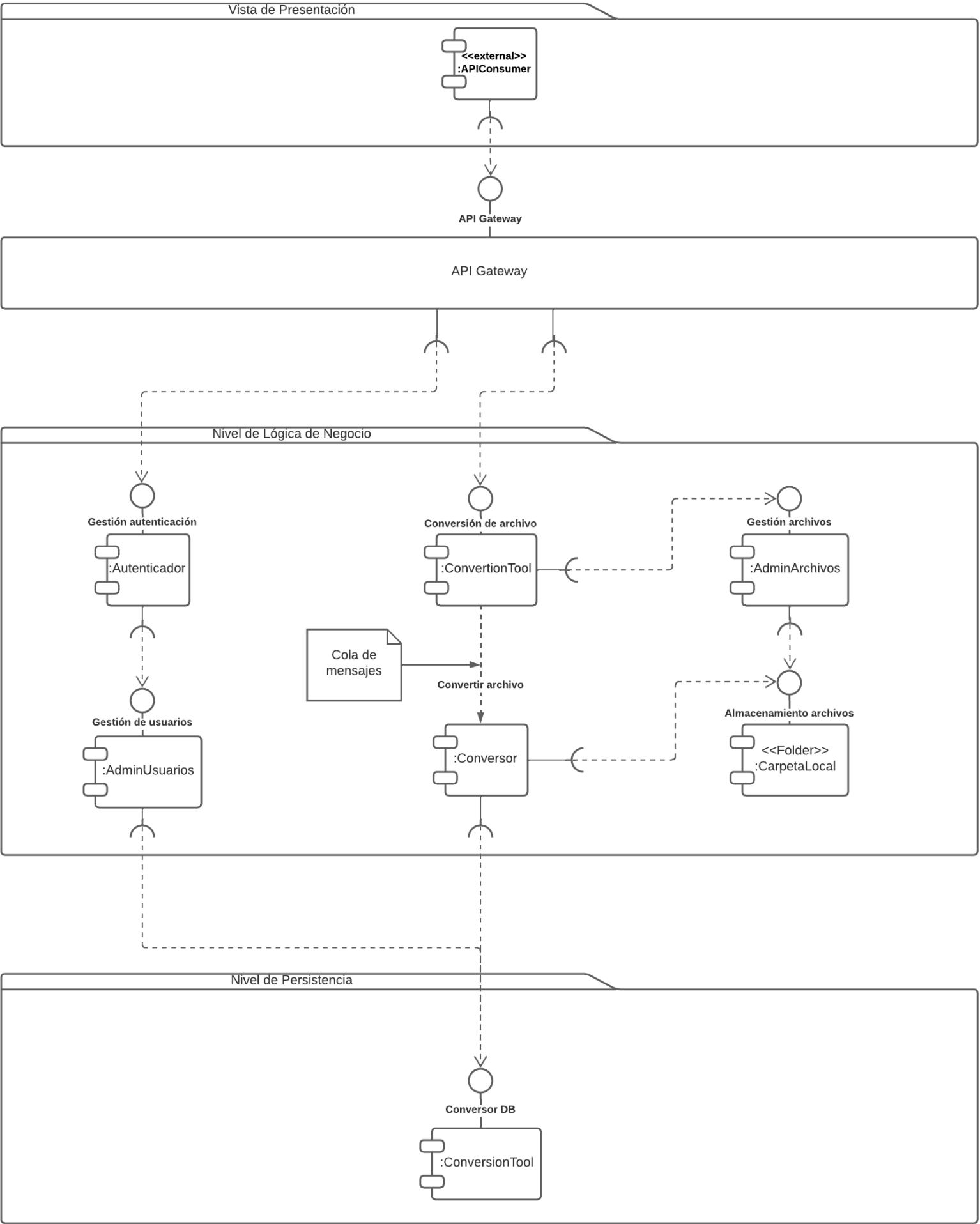
Su relación con la Cola de mensajes la usamos para procesar las tareas de conversión de archivos en el orden en que se reciben. Con Conversion Tool para acceder a algún componente relacionado con la conversión de archivos. Y la relación de Guardar Archivo convertido con CarpetaLocal es el resultado de la conversión y se almacena en una carpeta local.

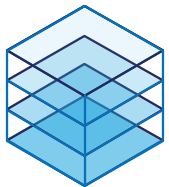
AdminArchivos:

La relación de "Subir archivo" con ConversionTool es para permitir a los usuarios cargar archivos en la aplicación y "Archivo Convertido" para acceder a los archivos convertidos. Finalmente, la relación "Guardar archivos" con CarpetaLocal implica que los archivos subidos y los archivos convertidos se almacenan en una carpeta local.

Vista Funcional

Proyecto	Sistema ABC Jobs	ID	VC-004	Elaboración	Equipo 1	Versión	1.0	Convención
Vista	Funcional		Modelo	Componente-Conector				UML





Vista de Componentes

Proyecto	Cloud Conversion Tool	ID	VC-005	Elaboración	Equipo 1	Versión	1.0
Vista	Componentes		Modelo	Componentes			

Razonamiento sobre las principales decisiones de arquitectura tomadas en este modelo

Este diagrama refleja la organización de los componentes según su función en la arquitectura de la aplicación Cloud Conversion Tool. Cada componente está conectado de manera que refleja cómo interactúan entre sí para proporcionar las funcionalidades requeridas. La comunicación entre componentes se realiza a través de servicios REST y la cola de mensajes.

Vista de Presentación:

- API Consumer: Este componente representa la interfaz de usuario del cliente (Usuario) que interactúa con la aplicación. Puede ser una aplicación web o una aplicación móvil.

API Gateway:

- Actúa como un punto de entrada único para las solicitudes del cliente y dirige las solicitudes a los servicios apropiados en el servidor.

Nivel de Lógica de Negocio:

- Autenticador: Encargado de gestionar la autenticación de usuarios y la generación de tokens de sesión.
- AdminUsuario: Administra las operaciones relacionadas con los usuarios, como la creación y gestión de cuentas de usuario.
- Cola de Mensajes: Puede utilizarse para procesar tareas en segundo plano y comunicarse con componentes como el Conversor y el Admin Archivos.
- Conversor: Realiza las conversiones de archivos, procesando las tareas de conversión en la cola de mensajes.
- AdminArchivos: Administra las operaciones relacionadas con archivos, incluyendo la gestión de archivos originales y convertidos.
- Carpeta Local: Puede ser un componente que almacena temporalmente archivos durante el proceso de conversión.

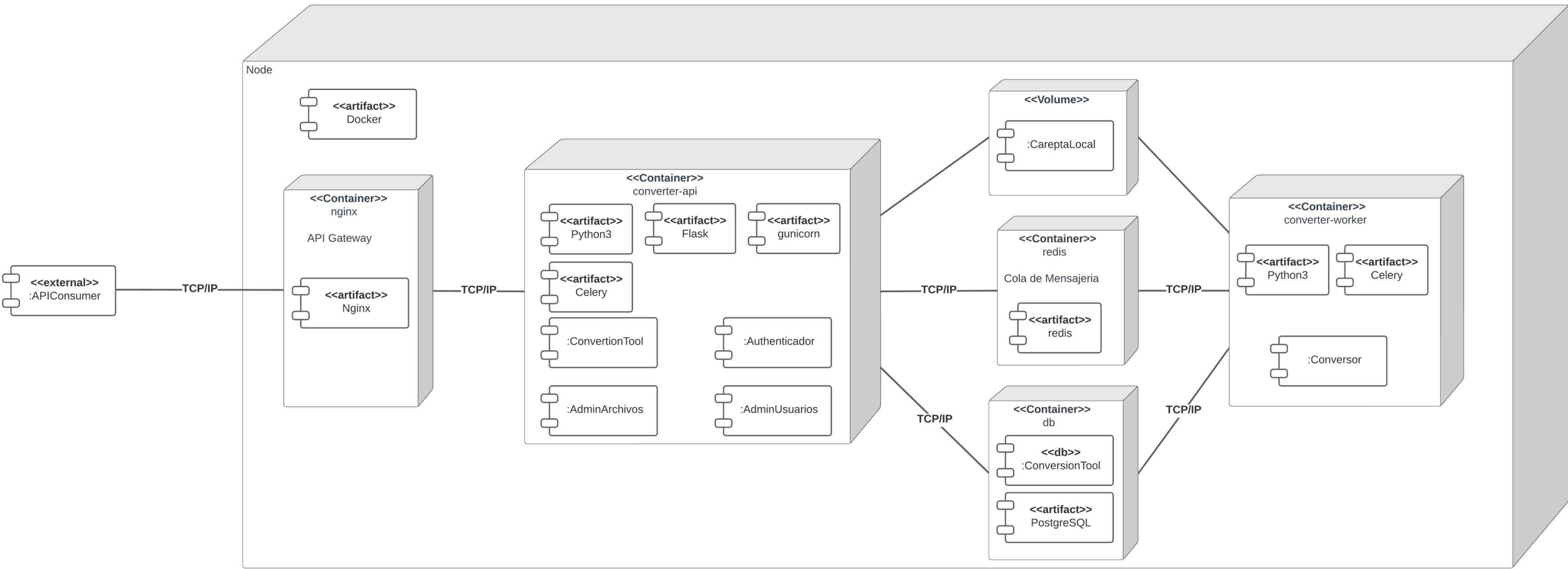
Nivel de Persistencia:

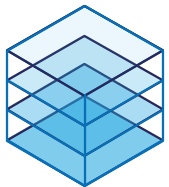
- BD Conversion Tools: La base de datos donde se almacenan los datos de la aplicación, incluyendo información de usuarios, tareas de conversión y detalles de archivos.



Vista de despliegue

Proyecto	Sistema ABC Jobs	ID	VC-005	Elaboración	Equipo 12	Versión	1.0	Convención
Vista	Despliegue		Modelo	Despliegue				UML





Vista de despliegue

Proyecto	Cloud Conversion Tool	ID	VC-005	Elaboración	Equipo 1	Versión	1.0
Vista	Despliegue		Modelo	Despliegue			

Razonamiento sobre las principales decisiones de arquitectura tomadas en este modelo

Esta arquitectura de despliegue utiliza contenedores Docker para implementar los componentes de la aplicación. Se hace un uso eficiente de componentes como Redis para la cola de mensajes y PostgreSQL para la base de datos. La comunicación entre los componentes se realiza a través de conexiones TCP/IP, lo que permite que la aplicación funcione de manera distribuida.

APIConsumer (externo)

- Este componente actúa como un cliente externo que se conecta a través de TCP/IP. Se comunica con la aplicación a través de una red externa y utiliza el API Gateway como punto de entrada.

Nodo con Docker Engine y Docker Compose: Este nodo es donde se despliegan los contenedores Docker. Docker Engine es la tecnología que permite la creación y gestión de contenedores. Es un entorno aislado para ejecutar aplicaciones y servicios.

- **Contenedor API Gateway:** El API Gateway es responsable de dirigir las solicitudes del cliente al componente adecuado dentro de la aplicación. Actúa como una puerta de entrada para las solicitudes entrantes. Se realiza con el software Nginx.
- **Contenedor converter-api:**
 - **Python 3, Flask, gunicorn:** Estos son los componentes esenciales para ejecutar la aplicación web. Python 3 es el lenguaje de programación, Flask es el marco web, y gunicorn es un servidor web HTTP.
 - **Celery:** Este es un componente para la gestión de tareas en segundo plano y se utiliza para procesar tareas de conversión y otras operaciones asincrónicas.
 - **ConversionTool, Autenticador, Admin Archivos, Admin Usuarios:** Estos son componentes de la lógica de negocio que se ejecutan en el contenedor y proporcionan las funcionalidades específicas de la aplicación.
- **Volumen Carpeta Local:** La carpeta local es un volumen que se utiliza para almacenar temporalmente archivos durante el proceso de conversión.
- **Contenedor Redis:** Este contenedor se utiliza para alojar Redis, un sistema de cola de mensajes. Se conecta a través de TCP/IP para que otros componentes, como Celery, puedan utilizarlo para gestionar tareas en segundo plano de manera eficiente.
- **Contenedor con DB en PostgreSQL:** Este contenedor alberga la base de datos PostgreSQL que almacena los datos de la aplicación, como información de usuarios, tareas de conversión y detalles de archivos. La conexión TCP/IP se utiliza para acceder a la base de datos.
- **Contenedor converter-worker:** Este contenedor se encarga de ejecutar los trabajadores de Celery que procesan tareas en segundo plano. Utiliza Python 3, Celery y ffmpeg para realizar estas tareas.

© - **Derechos Reservados:** la presente obra, y en general todos sus contenidos, se encuentran protegidos por las normas internacionales y nacionales vigentes sobre propiedad Intelectual, por lo tanto su utilización parcial o total, reproducción, comunicación pública, transformación, distribución, alquiler, préstamo público e importación, total o parcial, en todo o en parte, en formato impreso o digital y en cualquier formato conocido o por conocer, se encuentran prohibidos, y solo serán lícitos en la medida en que se cuente con la autorización previa y expresa por escrito de la Universidad de los Andes.

De igual manera, la utilización de la imagen de las personas, docentes o estudiantes, sin su previa autorización está expresamente prohibida. En caso de incumplirse con lo mencionado, se procederá de conformidad con los reglamentos y políticas de la universidad, sin perjuicio de las demás acciones legales aplicables.
