

IBM Maximo PQI SaaS Visual Insights

Guía del usuario



Nota

Antes de utilizar esta información y el producto al que da soporte, lea la información del apartado “Avisos” en la página 121.

Primera edición (Marzo 2019)

Contenido

Capítulo 1. Visión general del producto.....	1
Roles.....	1
¿Cómo se hace una copia de seguridad y una restauración de los datos?.....	1
Novedades de este release.....	2
Características de accesibilidad.....	4
Capítulo 2. Creación de sistemas perimetrales.....	5
Requisitos de los sistemas perimetrales.....	5
Abrir puertos perimetrales.....	5
Instalación de paquetes de NVIDIA GPU para Ubuntu.....	6
Instalación de paquetes de GPU de NVIDIA para Linux en servidores de NVIDIA Tesla K80 Power Systems.....	7
Instalación de paquetes de GPU de NVIDIA para Linux en servidores de NVIDIA Tesla V100 Power Systems.....	7
Instalación de Caffe para Ubuntu.....	9
Instalación de TensorFlow para Ubuntu.....	10
Instalación de Caffe para Linux en Power Systems Servers.....	10
Resolución de problemas de la instalación de Caffe.....	12
Instalación de Open CV.....	14
Instalación de bibliotecas de detección de objetos.....	15
Resolución de problemas de la instalación de biblioteca Python Faster-RCNN.....	16
Configuración del servidor de imágenes.....	17
Configuración del almacén de modelos.....	18
Instalación de módulos Python.....	19
Instalación de Paramiko para Linux en Power Systems Servers.....	19
Registro del sistema perimetral en la aplicación central.....	19
Registro de un sistema conectado en la aplicación central.....	20
Registro de un sistema autónomo en la aplicación central.....	20
Instalación de un sistema perimetral autónomo.....	21
Comprobar el estado de los servicios en los sistemas perimetrales.....	24
Actualización de sistemas perimetrales.....	25
Capítulo 3. Creación y uso de modelos.....	27
Estructura de archivos de imagen comprimidos.....	27
Adición de imágenes históricas para grupos de imágenes	28
Herramienta de etiquetado de imágenes.....	29
Creación de un grupo de imágenes no etiquetadas.....	29
Etiquetado manual de imágenes de muestreo.....	30
Etiquetado automático de imágenes de muestreo.....	30
Creación de modelo.....	30
Creación de modelos.....	31
Utilización del catálogo de modelos.....	32
Prueba de modelos en el catálogo de modelos.....	32
Creación de modelos utilizando el catálogo de modelos.....	33
Modelos de formación.....	33
Modelos formados.....	34
Estructura de los archivos de modelo.....	34
Modelos validados.....	40
Distribución de modelos formados en sistemas perimetrales.....	41
Modelos de repetición de formación	41

Utilización de modelos con un sistema perimetral autónomo.....	41
Publicación de modelos.....	42
Despliegue de una instancia de modelo.....	42
Anular el despliegue de modelos.....	42
Capítulo 4. Comprobación de los resultados de la inspección.....	43
Imágenes.....	43
Filtrado de defectos.....	43
Comprobación de defectos.....	43
Cargar imágenes utilizando el simulador.....	44
Capítulo 5. Panel de control del indicador clave de rendimiento.....	45
Capítulo 6. Integración con Prescriptive Quality.....	47
Capítulo 7. Interfaz de programas de aplicación.....	49
Flujos de trabajo de API.....	49
Preparación para el uso de llamadas de API.....	50
Respuestas del servicio.....	50
Servicios de grupos de datos.....	51
Obtener todos los grupos de datos.....	51
Obtener todos los grupos de datos con archivos de datos.....	52
Obtener un grupo de datos.....	54
Crear un grupo de datos.....	55
Actualizar un grupo de datos.....	56
Suprimir grupos de datos.....	57
Servicios de archivos de datos.....	58
Obtener todos los archivos de datos que pertenecen a un grupo de datos.....	58
Obtener un archivo de datos.....	59
Descargar el contenido binario del archivo de datos.....	61
Cargar archivos de datos en un grupo de datos.....	61
Suprimir un archivo de datos.....	62
Servicios de grupos de datos sin etiqueta.....	63
Cargar un archivo de imagen comprimido no etiquetado.....	63
Crear un grupo de datos sin etiquetar.....	64
Obtener un grupo de datos sin etiqueta.....	66
Servicios de modelo.....	67
Obtener todos los modelos.....	67
Obtener un modelo.....	69
Crear un modelo.....	71
Actualizar un modelo.....	73
Suprimir un modelo.....	74
Obtener todos los modelos compartidos.....	75
Servicios de instancias de modelos.....	76
Crear una instancia de modelo.....	76
Obtener la instancia de modelo perteneciente al modelo.....	79
Obtener una instancia de modelo.....	80
Obtener el resultado de validación de la instancia del modelo.....	81
Servicios de resultados de inspección.....	83
Obtener la lista de resultados de inspección.....	83
Obtener detalles de resultados de inspección.....	84
Obtener visión general de celdas del resultado de inspección.....	86
Confirmar resultados de inspección.....	87
Suprimir resultados de inspección.....	88
Servicios de acciones de instancias de modelos.....	89
Formar instancia de modelo.....	89
Descargar un archivo de registro de formación.....	90

Validar instancia del modelo.....	91
Rechazar instancia de modelo.....	92
Desplegar instancia del modelo.....	93
Volver a formar instancia de modelo.....	94
Anular despliegue de instancia de modelo.....	95
Servicios de sistemas perimetrales.....	96
Crear sistema perimetral.....	96
Obtener sistema perimetral.....	97
Suprimir sistema perimetral.....	98
Actualizar el sistema perimetral.....	99
Servicio de puntuación.....	100
Puntuar imagen.....	100
Servicio de integración de QEWS.....	102
Obtener archivo de tasa de imágenes de defectos.....	102
Servicio compuesto.....	103
Registrar modelo.....	103
Servicios del sistema perimetral autónomo.....	104
Obtener los modelos disponibles.....	104
Desplegar un modelo.....	105
Anular el despliegue de un modelo.....	106
Cargar y puntuar imagen en el sistema perimetral.....	107
Sincronizar el resultado de inspección del sistema perimetral con la aplicación central.....	108
Limpiar el resultado de inspección que ha sincronizado con la aplicación central.....	109

Capítulo 8. Registrar, desplegar y probar un modelo utilizando la API..... 111

Capítulo 9. Resolución de problemas.....113

Mensajes.....	113
Marcas registradas.....	122
Términos y condiciones de la documentación de producto.....	123
Declaración de privacidad en línea de IBM.....	123

Capítulo 1. Visión general del producto

IBM® Maximo PQI SaaS Visual Insights es una solución de alertas y supervisión de la calidad que puede tomar imágenes de productos y ensamblajes, tanto en curso como finalizados, y clasificarlas en categorías de defectos.

Roles

Para comprender Maximo PQI SaaS Visual Insights, resulta útil comprender cómo interactúan los distintos roles con el producto.

Rol	Descripción
Administrador de modelos	Gestiona modelos y tipos de defectos, carga conjuntos de imágenes para tipos de defectos específicos, forma modelos y distribuye modelos ejecutables a sistemas perimetrales.
Inspector	Verifica los resultados de la inspección que produce el producto, cambia los tipos de defectos si es necesario, señala los tipos de defectos desconocidos y los envía al Supervisor de inspectores para su evaluación adicional.
Supervisor de inspectores	Vuelve a comprobar los resultados de la inspección del Inspector. Revisa y clasifica los tipos de defectos desconocidos. Revisa el panel de control KPI, que incluye defecto por unidad y tasa de defectos.

¿Cómo se hace una copia de seguridad y una restauración de los datos?

La redundancia de IBM Open Platform se utiliza para proteger los datos del cliente en un entorno de Big Data. Además, Tivoli Storage Manager se utiliza para la copia de seguridad de datos en los entornos de producción, lo que incluye los archivos de Linux. Los archivos Linux incluyen archivos de cliente cargados y archivos de configuración y registro de middleware y aplicaciones.

En la tabla siguiente se muestra la planificación de copia de seguridad para diversos aspectos de la solución.

Datos	Tipo de copia de seguridad	Frecuencia	Hora (Hora central)	Periodo de retención
Archivos	Completa	Quincenal	00:00 - 03:00 el día 1 y el día 16 de cada mes	5 semanas
Archivos	Incremental	Dos veces al día	Entre las 00:00 - 03:00 y las 12:00 - 15:00	14 días

Durante el intervalo de copia de seguridad, se puede acceder a la solución. Sin embargo, es posible que el rendimiento se vea afectado.

En caso de que se produzca un error del sistema que provoque daños o pérdida de los datos, IBM le ayudará a restaurar los datos en los puntos de tiempo de recuperación, según su política de copia de seguridad.

Novedades de este release

Las siguientes nuevas características están disponibles en IBM Maximo PQI SaaS Visual Insights.

Novedades de este release

Marzo de 2019

- Soporte para la carga del modelo TensorFlow y la puntuación de la imagen

Novedades de los releases anteriores

Enero de 2019

- El nombre del producto se ha cambiado a IBM Maximo PQI SaaS Visual Insights.
- Se ha solucionado un problema de la curva de formación al formar modelos de clasificación en paralelo.
- Se ha solucionado un problema de formación de modelos al volver a entrenar modelos YOLO.
- Se ha solucionado un problema al validar y puntuar modelos FRCNN-VGG16.
- Se han actualizado las propiedades de conexión para mejorar la simultaneidad de la puntuación.

Noviembre de 2018

- Soporte de varios clústeres de sistema perimetral en un arrendatario. Los modelos se pueden implementar en varios clústeres de sistemas perimetrales.
- Soporte de clasificación de clústeres de formación para escalar durante varias presentaciones de trabajos de formación y reducir el tiempo de espera.
- Soporte de la gestión de la versión del sistema perimetral. La información de la versión se conserva cuando se registra o se actualiza un sistema perimetral. Puede comprobar la versión del sistema perimetral cuando despliegue un modelo.
- Se ha añadido una política de archivado de imágenes para especificar las clases de imágenes que se especificarán y usarán para volver a entrenar un modelo.
- El producto devuelve detalles de error cuando falla la puntuación.
- Se ha resuelto el problema de seguridad con la apropiación de sesión.
- Se han actualizado las reglas de precios para hacer un recuento del número de archivos de entrenamiento, el número de archivos de puntuación y el almacenamiento utilizado.

Septiembre de 2018

- Soporte para el epoch como unidad para la formación de modelos de clasificación. Esto hace que sea más fácil definir parámetros.
- Habilidad de clústeres de servidores de formación al formar modelos de detección de objetos. Esta agrupación en clústeres admite escalado si un servidor de formación no puede manejar muchas solicitudes en paralelo.
- Soporte para la ejecución del sistema perimetral en la plataforma Linux en Power Systems Servers.
- Soporte para el empaquetado de archivos zip en MacOS al cargar archivos zip de imágenes y archivos zip de modelos.
- Mejoras en la herramienta de etiquetado automático.

Junio 2018

- Posibilidad de etiquetar automáticamente las imágenes de formación mediante un pequeño conjunto de datos anotado, lo que disminuye el tiempo necesario para la anotación de los datos.
- Mejoras de sistemas perimetrales basadas en los comentarios del sector, que incluyen la supresión del acceso NFS desde el centro, el soporte de alta disponibilidad del sistema perimetral maestro y el soporte de ejecución de un modelo en un sistema perimetral específico y GPU específico.
- Mejoras en la formación de modelos realizadas por nuestros científicos expertos en datos, basadas en una red de formación previa para dar soporte a pequeñas iteraciones con una alta precisión.

Marzo de 2018

- Soporte para examinar modelos en un catálogo de modelos y probar las imágenes propias con modelos compartidos.
- Soporte para utilizar modelos compartidos en un catálogo de modelos y formar el propio modelo con imágenes específicas.
- Soporte para seleccionar diferentes tipos de modelos, algoritmos, redes e hiperparámetros cuando crea un nuevo modelo.
- Soporte para formación de modelos en línea. Los archivos de modelos se generan automáticamente sin necesidad de utilizar otras herramientas de formación.
- Panel de control de formación que muestra un chat en vivo de valores de pérdida de modelo y precisión con iteraciones y archivos de registro detallados.
- Lista de instantáneas disponibles durante el proceso de formación. Soporte para utilizar una instantánea como un modelo formado.
- La guía de API de tres pasos que ayuda a la integración de servicios para el registro, el despliegue y prueba de modelos.
- Se puede utilizar un simulador para enviar imágenes al servidor y revisar los resultados de la inspección.

Diciembre de 2017

- Interfaz de usuario que permite al experto en datos descargar conjuntos de datos y adjuntar modelos formados.
- Soporte de repetición de formación de un modelo de detección de objetos.
- Interfaz de usuario que permite al administrador de modelos supervisar la cola de repetición de formación.
- Soporte del modo fuera de línea para sistemas perimetrales. En la modalidad fuera de línea, los sistemas perimetrales almacenan localmente los resultados de la inspección y envían los resultados a la aplicación central por lotes.
- Soporte de un clúster de sistemas perimetrales para manejar las solicitudes de puntuación con equilibrio de carga.
- Soporte de muestreo de conjuntos de datos para la repetición de modelos de clasificación.
- Posibilidad de exportar archivos CSV para la integración con IBM Prescriptive Quality.
- API REST pública que permite realizar funciones, tales como obtener todos los grupos de datos existentes, obtener todos los archivos de datos pertenecientes a un grupo de datos y obtener todos los modelos existentes.

Septiembre de 2017

- Ahora puede utilizar grupos de imágenes para representar el mismo tipo de imágenes utilizando uno o varios archivos de imágenes comprimidos.
- Se ha añadido soporte para varias versiones de modelo que comparten los mismos grupos de imágenes, pero utilizan distintos archivos de imágenes para dar formación al modelo.

- Se ha añadido el proceso de repetición de formación de modelos. Puede repetir la formación manual o automáticamente de una nueva versión de modelo utilizando distintos archivos de imágenes.
- Se ha añadido soporte para la validación de modelos. El proceso de validación calcula y muestra un informe de precisión de modelo basado en los archivos de imágenes de validación.
- Ahora puede utilizar recuadros de defectos y tipos de defectos en una imagen para marcar la ubicación del defecto para el inspector.
- Ahora puede mostrar varias ubicaciones de defectos en una imagen. Puede añadir, ajustar y suprimir recuadros de defectos en una imagen.
- Se ha actualizado el panel de control de KPI para incluir los defectos por unidad y la tasa de defectos para el supervisor de inspectores.
- Se ha añadido soporte para que el modelo de detección de objetos pueda detectar varios defectos en una imagen con el modelo CNN.

Características de accesibilidad

Las características de accesibilidad ayudan a los usuarios que tienen una discapacidad física, por ejemplo movilidad restringida o visión limitada, a utilizar productos de tecnología de la información.

Para obtener información acerca del compromiso que IBM tiene con la accesibilidad, visite el [IBM Accessibility Center](http://www.ibm.com/able) (www.ibm.com/able).

La documentación HTML tiene funciones de accesibilidad. Los documentos PDF son suplementarios y, como tales, no incluyen características de accesibilidad adicionales.

Capítulo 2. Creación de sistemas perimetrales

Maximo PQI SaaS Visual Insights consta de la aplicación central y los perimetrales. Los sistemas perimetrales son sistemas de Linux utilizados para ejecutar una detección de defectos en tiempo de ejecución.

Los sistemas perimetrales utilizan la infraestructura de aprendizaje profundo de Caffe. Caffe es un entorno de formación de red neuronal artificial (ANN) dedicado. El aprendizaje profundo requiere una cantidad significativa de recursos de procesamiento. El aprendizaje profundo se puede ejecutar de forma eficaz utilizando una unidad de procesamiento gráfico (GPU). Aunque la mayoría de las infraestructuras de aprendizaje profundo son también compatibles con el procesamiento de la CPU, el procesamiento de la GPU ofrece un rendimiento aceptable en entornos de producción.

Los sistemas perimetrales se agrupan en clúster para el equilibrio de carga. Un clúster consta de un sistema perimetral maestro y varios sistemas perimetrales esclavos. Cuando una celda envía una imagen para su proceso, el sistema perimetral maestro recibe la imagen y, a continuación, la envía a un sistema perimetral esclavo que tiene disponibles recursos de GPU.

La arquitectura del clúster de sistemas perimetrales es escalable. Si necesita más recursos de proceso, puede añadir más sistemas perimetrales.

Cuando cree un sistema perimetral, especifique si se trata del sistema perimetral maestro o esclavo. El primer sistema perimetral que crea debe ser el maestro. Una vez creado un sistema perimetral, no puede cambiar el tipo de sistema perimetral. Solo puede crear un sistema perimetral maestro por arrendatario. El sistema perimetral maestro no se puede suprimir hasta que se hayan suprimido todos los esclavos.

Los sistemas perimetrales maestros pueden estar conectados o pueden ser autónomos. Conectado significa que cuando se puntúa la imagen, inmediatamente se envía el resultado de la inspección a la aplicación central. Autónomo significa que el resultado de la inspección se almacena en el sistema perimetral hasta que selecciona **Extraer resultados** en el recuadro de diálogo **Vista previa de sistema perimetral** o cuando invoca el servicio que se ha desplegado en el sistema perimetral.

Requisitos de los sistemas perimetrales

Antes de crear un sistema perimetral, asegúrese de que el sistema cumpla los requisitos.

- Uno de los sistemas operativos siguientes:
 - Ubuntu 16.04 en x86_64
 - Red Hat Enterprise Linux 7.5 en x86_64
 - Red Hat Enterprise Linux 7.5 en IBM Power System
- Arquitectura de CPU: x86_64 o IBM Power System
- procesador de 4 núcleos
- 64 GB de memoria
- unidad de disco duro de 2 TB
- Una o varias tarjetas NVIDIA GPU

Abrir puertos perimetrales

Antes de utilizar sistemas perimetrales, debe abrir los puertos de cortafuegos que utilizan los sistemas perimetrales.

Acerca de esta tarea

Si UFW (Uncomplicated Firewall) está habilitado y activo en los sistemas perimetrales, utilice los mandatos siguientes para abrir UFW para el servicio NFS (Network File System) y abra los siguientes puertos de cortafuegos en los sistemas perimetrales:

```
sudo ufw enable
sudo ufw allow nfs
sudo ufw allow 22
sudo ufw allow 5005
sudo ufw allow 5070:5090/tcp
sudo ufw allow 6005
sudo ufw allow 8449
```

Instalación de paquetes de NVIDIA GPU para Ubuntu

Utilice esta tarea para instalar paquetes de NVIDIA GPU para sistemas Ubuntu. Para habilitar el procesamiento de la GPU, hay que instalar los paquetes de NVIDIA GPU necesarios.

Procedimiento

1. Descargue e instale los controladores de NVIDIA GPU. La lista de controladores de NVIDIA para Ubuntu está disponible en el enlace siguiente: [Binary Driver How to - Nvidia](#). El mandato siguiente es un ejemplo:

```
sudo apt-get install ubuntu-drivers-common
sudo ubuntu-drivers devices
sudo apt-get install nvidia-384
```
2. Utilice el mandato siguiente para comprobar si el controlador de NVIDIA se ha instalado correctamente:

```
sudo nvidia-smi
```
3. Descargue e instale el kit de herramientas de CUDA de NVIDIA y al biblioteca CUDNN correspondiente. Hay soporte para CUDA 8.0, CUDA 9.0 y CUDA 10.0. El mandato siguiente es un ejemplo para CUDA 8.0.

```
wget
http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/
cuda-repo-ubuntu1604_8.0.61-1_amd64.deb
```
4. Utilice los mandatos siguientes para instalar el archivo de CUDA en el servidor de destino:

```
sudo dpkg -i cuda-repo-ubuntu1604_8.0.61-1_amd64.deb
sudo apt-get update
sudo apt-get install cuda
```

Dado que NVIDIA actualiza el kit de herramientas, puede obtener una versión más reciente del kit de herramientas utilizando este mandato. Se recomienda que instale CUDA 8.0 dado que esta es la versión probada. Si encuentra que una versión más reciente se ha instalado puede degradar a CUDA 8.0 utilizando los mandatos siguientes:

```
sudo apt-get remove cuda
sudo apt-get install cuda-8-0
sudo ln -s /usr/local/cuda-8.0 /usr/local/cuda
```
5. Descargue la biblioteca NVIDIA CUDA Deep Neural Network cudnn-8.0-linux-x64-v6.0.tgz del enlace siguiente: https://developer.nvidia.com/compute/machine-learning/cudnn/secure/v6/prod/8.0_20170307/cudnn-8.0-linux-x64-v6.0-tgz. Es posible que necesite crear una cuenta e iniciar la sesión antes de descargar el archivo.
6. Desempaque el archivo cudnn-8.0-linux-x64-v6.0.tgz en el directorio de instalación cuda con el mandato siguiente:

```
sudo tar -xvf cudnn-8.0-linux-x64-v6.0.tgz -C /usr/local
```
7. Establezca la variable de entorno utilizando los mandatos siguientes:

```
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:$LD_LIBRARY_PATH
export PATH=/usr/local/cuda/bin:$PATH
```

Añada también estos mandatos al script `~/ .bashrc`.

8. Utilice los mandatos siguientes para instalar el paquete NCCL de NVIDIA:

```
git clone https://github.com/NVIDIA/ncccl.git
cd ncccl
sudo make install -j4
```

Instalación de paquetes de GPU de NVIDIA para Linux en servidores de NVIDIA Tesla K80 Power Systems

Utilice esta tarea para instalar paquetes de GPU de NVIDIA para Linux en servidores de NVIDIA Tesla K80 Power Systems. Para habilitar el procesamiento de la GPU, hay que instalar los paquetes de NVIDIA GPU necesarios.

Acerca de esta tarea

Para sistemas Power8, utilice CUDA 8 tal como se describe en la tarea siguiente. Para sistemas Power9, sustituya CUDA 10 por CUDA 8 en la tarea siguiente.

Procedimiento

1. Descargue e instale los controladores de NVIDIA GPU. La lista de controladores de NVIDIA para Linux en Power Systems Servers está disponible en el enlace siguiente: [NVIDIA Driver Downloads](#). Siga las instrucciones de instalación de la página de descarga.
2. Descargue el archivo de repositorio CUDA e instale CUDA 8 utilizando los mandatos siguientes:

```
wget https://developer.download.nvidia.com/compute/cuda/repos/rhel7/ppc64le/cuda-repo-rhel7-8.0.61-1.ppc64le.rpm
rpm -i cuda-repo-rhel7-8.0.61-1.ppc64le.rpm
yum clean all
yum install cuda
```
3. Para CUDA 8, descargue la biblioteca NVIDIA CUDA Deep Neural Network `cuda-8.0-linux-ppc64le-v6.0.tgz` del URL siguiente: https://developer.nvidia.com/compute/machine-learning/cudnn/secure/v6/prod/8.0_20170307/cudnn-8.0-linux-ppc64le-v6.0.tgz. Para CUDA 10, descargue la biblioteca NVIDIA CUDA Deep Neural Network `cuda-10.0-linux-ppc64le-v7.3.1.20.tgz` del URL siguiente: <https://developer.download.nvidia.com/compute/cuda/repos/rhel7/ppc64le/cuda-10.0.130-1.ppc64le.rpm>. Es posible que necesite crear una cuenta e iniciar la sesión antes de descargar el archivo.
4. Desempaquete el archivo `cuda-8.0-linux-ppc64le-v6.0.tgz` en el directorio de instalación `cuda` utilizando el mandato siguiente:

```
sudo tar -xvf cuda-8.0-linux-ppc64le-v6.0.tgz -C /usr/local
```
5. Establezca la variable de entorno utilizando el mandato siguiente:

```
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:$LD_LIBRARY_PATH
```

Añada también este mandato al script `~/ .bashrc`.
6. Instale el paquete NVIDIA NCCL utilizando los mandatos siguientes:

```
git clone https://github.com/NVIDIA/ncccl.git
cd ncccl
sudo make install -j4
```

Instalación de paquetes de GPU de NVIDIA para Linux en servidores de NVIDIA Tesla V100 Power Systems

Utilice esta tarea para instalar paquetes de GPU de NVIDIA para Linux en servidores de NVIDIA Tesla V100 Power Systems. Para habilitar el procesamiento de la GPU, hay que instalar los paquetes de NVIDIA GPU necesarios.

Procedimiento

1. Descargue e instale NVIDIA CUDA 9.2.148 desde developer.nvidia.com/cuda-92-download-archive.
 - a) Seleccione Sistema operativo: Linux.
 - b) Seleccione Arquitectura: ppc64le.
 - c) Seleccione Distribución: RHEL.
 - d) Seleccione Versión: 7.
 - e) Seleccione Tipo de instalador: rpm (local). El rpm local se prefiere a la red rpm, ya que asegura que la versión instalada es la versión que se descarga. Con el rpm de red, el mandato `yum install cuda` instala siempre la versión más reciente de CUDA Toolkit.
 - f) Pulse **Descargar** para descargar el instalador base.
 - g) Haga clic en **Descargar** para descargar el parche 1.
 - h) Siga las instrucciones de instalación de Linux en POWER en la Guía de inicio rápido de CUDA, incluidos los pasos que describen cómo configurar el entorno de desarrollo de CUDA actualizando PATH y LD_LIBRARY_PATH.
2. Descargue el controlador 410.104 de NVIDIA desde <http://www.nvidia.com/Download/index.aspx>.
 - a) Seleccione Tipo de producto: Tesla.
 - b) Seleccione Series de productos: V-Series.
 - c) Seleccione Producto: Tesla V100.
 - d) Seleccione Sistema operativo: Linux POWER LE RHEL 7.
 - e) Seleccione Kit de herramientas de CUDA: 10.0.
 - f) Haga clic en **Buscar** para ir al enlace de descarga y, a continuación, haga clic en **Descargar**.
3. **Nota:** Con los sistemas IBM Power System AC922, se necesitan actualizaciones del sistema operativo y de firmware del sistema para instalar el controlador GPU más reciente.
Instale el CUDA y el controlador GPU:
 - a) Instale el rpm de repositorio base de CUDA.
 - b) Instale el rpm del repositorio Parche 1 de CUDA.
 - c) Instale el rpm del repositorio del controlador GPU.
 - d) Ejecute el mandato siguiente para instalar el controlador de CUDA, de parche y GPU:
`sudo yum install cuda`
 - e) Reinicie el sistema para activar el controlador.
4. Utilice el siguiente mandato para habilitar los servicios persistentes de sistema NVIDIA:
`systemctl enable nvidia-persistenced`
5. Compruebe los controladores NVIDIA utilizando el siguiente mandato de shell:
`nvidia-smi`
6. Descargue NVIDIA cuDNN v7.4.2 para CUDA 10.0 (cuDNN v7.4.2 Library for Linux (Power8/Power9)) desde developer.nvidia.com/cudnn. Se requiere el registro en el programa NVIDIA Accelerated Computing Developer Program.
7. Descargue NVIDIA NCCL v2.3.7 para CUDA 10.0 (NCCL 2.3.7 O/S agnostic y CUDA 10.0 e IBM Power) desde developer.nvidia.com/nccl. Se requiere registrarse en el programa NVIDIA Accelerated Computing Developer Program.
8. Instale los paquetes de cuDNN v7.4.2 y NCCL v2.3.7 y, a continuación, actualice la memoria caché de la biblioteca compartida utilizando los mandatos siguientes:
`sudo tar -C /usr/local --no-same-owner -xzf cudnn-9.2-linux-ppc64le-v7.4.2.tgz`
`sudo tar -C /usr/local --no-same-owner -xzf nccl_2.3.7+cuda10.0_ppc64le.tgz`
`sudo ldconfig`

Instalación de Caffe para Ubuntu

Debe instalar la infraestructura de aprendizaje profundo de Caffe y los paquetes relacionados. Caffe se utiliza para la formación de modelos y la clasificación de defectos.

Procedimiento

1. Utilice los mandatos siguientes para instalar los paquetes necesarios para Caffe:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install -y build-essential cmake git pkg-config
sudo apt-get install -y libprotobuf-dev libleveldb-dev libsnpappy-dev
libhdf5-serial-dev protobuf-compiler
sudo apt-get install -y libatlas-base-dev libjasper-dev
sudo apt-get install -y --no-install-recommends libboost-all-dev
sudo apt-get install -y libgflags-dev libgoogle-glog-dev liblmdb-dev
sudo apt-get install -y python-pip
sudo apt-get install -y python-dev
sudo apt-get install -y python-numpy python-scipy
sudo apt-get install -y libopencv-dev
sudo pip install opencv-python
sudo pip install flask_httpauth
sudo pip install gevent
sudo pip install pyinotify
sudo pip install tornado
```

2. Utilice el mandato siguiente para descargar el código fuente de Caffe:

```
wget https://github.com/BVLC/caffe/archive/1.0.zip
```

3. Utilice los mandatos siguientes para desempaquetar el paquete y especificar el directorio del paquete:

```
unzip 1.0.zip
cd ./caffe-1.0
```

4. Utilice el mandato siguiente para realizar una copia del archivo de configuración make:

```
cp Makefile.config.example Makefile.config
```

5. Añada las variables siguientes en el archivo Makefile.config:

```
USE_CUDA := 1
CUDA_DIR := /usr/local/cuda
PYTHON_INCLUDE := /usr/include/python2.7 \
/usr/lib/python2.7/dist-packages/numpy/core/include
PYTHON_LIB := /usr/lib/x86_64-linux-gnu
WITH_PYTHON_LAYER := 1
INCLUDE_DIRS := $(PYTHON_INCLUDE) /usr/local/include \
/usr/include/hdf5/serial
LIBRARY_DIRS := $(PYTHON_LIB) /usr/local/lib /usr/lib \
/usr/lib/x86_64-linux-gnu /usr/lib/x86_64-linux-gnu/hdf5/serial
```

6. En el directorio caffe-1.0, ejecute el mandato siguiente:

```
find . -type f -exec sed -i -e 's^"hdf5.h"^"hdf5/serial/hdf5.h"^g' -e
's^"hdf5_h1.h"^"hdf5/serial/hdf5_h1.h"^g' '{} ' \;
```

7. Ejecute los mandatos siguientes:

```
cd /usr/lib/x86_64-linux-gnu
sudo ln -s libhdf5_serial.so.10.1.0 libhdf5.so
sudo ln -s libhdf5_serial_h1.so.10.0.2 libhdf5_h1.so
```

8. Instale los paquetes Python necesarios en el directorio caffe-1.0/python mediante los mandatos siguientes:

```
cd {caffe-installation-path}/caffe-1.0/python
```

- ```
for req in $(cat requirements.txt); do sudo -H pip install $req --upgrade;
done
```
- donde {ruta\_instalación\_caffe} es la ruta de despliegue de Caffe.
- Abra el archivo make en el directorio {ruta\_instalación\_caffe} y cambie el parámetro NVCCFLAGS al valor siguiente:  
 NVCCFLAGS += -D\_FORCE\_INLINES -ccbin=\$(CXX) -Xcompiler -fPIC \$(COMMON\_FLAGS)
  - En el directorio principal de Caffe caffe-1.0, utilice los mandatos siguientes para iniciar la creación y la instalación de Caffe:  

```
make all
make test
make runtest
make pycaffe
make distribute
```
  - Añada la línea siguiente al script ~/.bashrc:  

```
export PYTHONPATH="/usr/lib/python2.7:{caffe-installation-path}/caffe-1.0/python:$PYTHONPATH"
```

 donde {ruta\_instalación\_caffe} es la ruta de despliegue de Caffe.

## Instalación de TensorFlow para Ubuntu

---

Si desea puntuar una imagen basada en un modelo TensorFlow, debe preparar el entorno de TensorFlow en el sistema perimetral.

### Procedimiento

- Instale tensorflow-gpu 1.4.0 utilizando el mandato siguiente. Asegúrese de que no se notifique ningún error durante la instalación y de que las versiones de todos los paquetes de dependencias sean correctas.  

```
pip install tensorflow-gpu==1.4.0
```
- Importe TensorFlow en Python para verificar que TensorFlow se haya instalado correctamente. Si no hay errores, TensorFlow se instala correctamente.  

```
import tensorflow
```

## Instalación de Caffe para Linux en Power Systems Servers

---

Utilice esta tarea para instalar Caffe para Linux en sistemas Power Systems Servers. Debe instalar la infraestructura de aprendizaje profundo de Caffe y los paquetes relacionados. Caffe se utiliza para la formación de modelos y la clasificación de defectos.

### Procedimiento

- Utilice los mandatos siguientes para instalar los paquetes necesarios para Caffe:  

```
sudo yum clean all
sudo yum update
sudo yum install upgrade
sudo yum install -y libboost-*
sudo yum install -y gflags-devel glog-devel lmdb-devel
sudo yum install -y python-pip
sudo yum install -y python-devel
sudo yum install -y opencv-devel
sudo yum makecache
sudo yum install -y protobuf-devel leveldb-devel lmdb-devel snappy-devel
opencv-devel boost-devel hdf5-devel atlas-devel glog-devel gflags-devel
sudo yum install libpng-devel
```

```

sudo yum install freetype-devel
sudo yum install libjpeg-turbo-devel
sudo yum install opencv-python
sudo rpm -e --nodeps numpy
sudo pip install numpy
pip install --upgrade pip
sudo pip install flask_httppauth
sudo pip install gevent
sudo pip install pyinotify
ln -s /usr/local/cuda-10.0 /usr/local/cuda
pip install scikit-image
sudo pip install tornado

```

2. Enlace la biblioteca Atlas utilizando los mandatos siguientes:

```

ln -fs /usr/lib64/atlas/libsatlas.so /usr/lib64/libatlas.so
ln -fs /usr/lib64/atlas/libsatlas.so /usr/lib64/libcblas.so

```

3. Utilice el mandato siguiente para descargar el código fuente de Caffe:

```
wget https://github.com/BVLC/caffe/archive/1.0.zip
```

4. Utilice los mandatos siguientes para desempaquetar el paquete y especificar el directorio del paquete:

```

unzip 1.0.zip
cd ./caffe-1.0

```

5. Sustituya los archivos Caffe siguientes:

- En el directorio `/include/caffe/util/cudnn.hpp`, sustituya el archivo `cudnn.hpp` por el archivo `cudnn.hpp` más reciente, que está en el repositorio de Caffe en GitHub: [github.com/BVLC/caffe.git](https://github.com/BVLC/caffe.git)
- Sustituya todos los archivos `cudnn` que se encuentran en la carpeta `/src/caffe/layers` por los archivos `cudnn` más recientes que están en el repositorio de Caffe en GitHub: [github.com/BVLC/caffe.git](https://github.com/BVLC/caffe.git)

Por ejemplo, ejecute los siguientes mandatos:

```

cp -rf /root/source/caffe-git/caffe-master/include/caffe/util/
cudnn.hpp /usr/local/caffe-1.0/include/caffe/util/
cp -rf /root/source/caffe-git/caffe-master/src/caffe/layers/cudnn_* /usr/
local/caffe-1.0/src/caffe/layers/
cp -rf /root/source/caffe-git/caffe-master/include/caffe/layers/
cudnn_* /usr/local/p/usr/local/caffe-1.0/include/caffe/layers/

```

6. Utilice el mandato siguiente para realizar una copia del archivo de configuración `make`:

```
cp Makefile.config.example Makefile.config
```

7. Añada las variables siguientes en el archivo `Makefile.config`:

```

USE_CUDNN := 1
CUDA_DIR := /usr/local/cuda
PYTHON_INCLUDE := /usr/include/python2.7 \
 /usr/lib64/python2.7/site-packages/numpy/core/include/
PYTHON_LIB := /usr/lib/gcc/ppc64le-redhat-linux/4.8.5/
WITH_PYTHON_LAYER := 1
INCLUDE_DIRS :=
$(PYTHON_INCLUDE) /usr/local/include
/usr/local/cuda-10.0/targets/ppc64le-linux/include/
LIBRARY_DIRS
:= $(PYTHON_LIB) /usr/local/lib /usr/lib \
 /usr/lib64 /usr/local/lib64

```

Cambie `CUDA_ARCH` por el texto siguiente:

```

CUDA_ARCH := -gencode arch=compute_30,code=sm_30
-gencode arch=compute_35,code=sm_35
-gencode arch=compute_50,code=sm_50
-gencode arch=compute_52,code=sm_52
-gencode arch=compute_60,code=sm_60

```

```
-gencode arch=compute_61,code=sm_61
-gencode arch=compute_61,code=compute_61
```

8. Instale los paquetes Python necesarios en el directorio `caffe-1.0/python` utilizando los mandatos siguientes:  
`cd caffe-installation-path/caffe-1.0/python`  
`for req in $(cat requirements.txt); do sudo -H pip install $req --upgrade;`  
`done`  
donde `caffe-installation-path` es la ruta de despliegue de Caffe.
9. Abra el archivo `make` en el directorio `caffe-installation-path` y cambie el parámetro `NVCCFLAGS` por el valor siguiente:  
`NVCCFLAGS += -D_FORCE_INLINES -ccbin=$(CXX) -Xcompiler -fPIC $`  
`(COMMON_FLAGS)`
10. En el directorio principal de Caffe `caffe-1.0`, utilice los mandatos siguientes para iniciar la creación y la instalación de Caffe:  
`make all`  
`make test`  
`make runtest`  
`make pycaffe`  
`make distribute`
11. Añada la línea siguiente al script `~/bashrc`:  
`export PYTHONPATH="/usr/lib/python2.7:caffe-installation-path/caffe-1.0/python:$PYTHONPATH"`  
donde `caffe-installation-path` es la ruta de despliegue de Caffe.
12. Ejecute las siguientes pruebas posteriores a la instalación:
  - a) `make runtest | tee -a runtest.out`
  - b) `grep -i OK runtest.out | wc -l`  
El resultado de la prueba de Caffe debe ser 2101
  - c) `python -c "import caffe"`  
para probar la instalación de Pycaffe
  - d) `tail -n 2 runtest.out`  
El contenido de `runtest.out` debe incluir el texto siguiente:  
[=====] 2101 tests from 277 test cases ran. (291548 ms total)  
[ PASSED ] 2101 tests

## Resolución de problemas de la instalación de Caffe

Si aparece un mensaje de error en el registro cuando inicia la compilación y la instalación de Caffe, puede seguir estos pasos para intentar resolver el problema.

### Síntomas 1

Cuando inicia la compilación y la instalación de Caffe, aparece el siguiente mensaje:

```
1. In file included from ./include/caffe/util/device_alternate.hpp:40:0,
2. from ./include/caffe/common.hpp:19,
3. from src/caffe/common.cpp:7:
4. ./include/caffe/util/cudnn.hpp: In function 'void
caffe::cudnn::createPoolingDesc(cudnnPoolingStruct**,
caffe::PoolingParameter_PoolMethod, cudnnPoolingMode_t*, int, int, int, int,
int)':
5. ./include/caffe/util/cudnn.hpp:127:41: error: too few arguments to function
'cudnnStatus_t
cudnnSetPooling2dDescriptor(cudnnPoolingDescriptor_t, cudnnPoolingMode_t,
cudnnNanPropagation_t, int,
int, int, int, int, int)'
6. pad_h, pad_w, stride_h, stride_w));
7. ^
8. ./include/caffe/util/cudnn.hpp:15:28: note: in definition of macro 'CUDNN_CHECK'
```

```

9. cudnnStatus_t status = condition; \
10. ^
11. In file included from ./include/caffe/util/cudnn.hpp:5:0,
12. from ./include/caffe/util/device_alternate.hpp:40,
13. from ./include/caffe/common.hpp:19,
14. from src/caffe/common.cpp:7:
15. /usr/local/cuda-7.5//include/cudnn.h:803:27: note: declared here
16. cudnnStatus_t CUDNNWINAPI cudnnSetPooling2dDescriptor(
17. ^
18. make: *** [.build_release/src/caffe/common.o] Error 1
19.

```

## Resolución del problema 1

Para solucionar el error, consulte los siguientes pasos:

1. En el directorio `/include/caffe/util/cudnn.hpp`, sustituya el archivo `cudnn.hpp` por el archivo `cudnn.hpp` más reciente, que está en el repositorio de Caffe en GitHub.
2. En la carpeta `/src/caffe/layers`, sustituya todos los archivos `cudnn` que hay en la carpeta `/src/caffe/layers` por los archivos `cudnn` más recientes, que están en el repositorio de Caffe en GitHub.

## Síntomas 2

Al instalar los paquetes de Python necesarios en el directorio `caffe-1.0/python`, se visualiza el siguiente mensaje:

```

Traceback (most recent call last):
 File "/usr/bin/pip", line 11, in <module>
 sys.exit(main())
 File "/usr/lib/python2.7/dist-packages/pip/__init__.py", line 215, in main
 locale.setlocale(locale.LC_ALL, '')
 File "/usr/lib/python2.7/locale.py", line 581, in setlocale
 return _setlocale(category, locale)
locale.Error: unsupported locale setting
Traceback (most recent call last):
 File "/usr/bin/pip", line 11, in <module>
 sys.exit(main())
 File "/usr/lib/python2.7/dist-packages/pip/__init__.py", line 215, in main
 locale.setlocale(locale.LC_ALL, '')
 File "/usr/lib/python2.7/locale.py", line 581, in setlocale
 return _setlocale(category, locale)
locale.Error: unsupported locale setting

```

## Resolución del problema 2

Para resolver este error, ejecute el mandato siguiente:  
`export LC_ALL=C`

## Síntomas 3

Al empezar la compilación e instalación de Caffe, se visualiza el mensaje siguiente:

```

nvcc fatal : Unsupported gpu architecture 'compute_20'
Makefile:595: recipe for target '.build_release/cuda/src/caffe/layers/prelu_layer.o' failed
make: *** [.build_release/cuda/src/caffe/layers/prelu_layer.o] Error 1

```

## Resolución del problema 3

Comente `-gencode arch=compute_20` en `Makefile.config`.

## Síntomas 4

Al empezar la compilación e instalación de Caffe, se visualiza el mensaje siguiente:

```

PROTOC src/caffe/proto/caffe.proto
make: protoc: Command not found

```

```
Makefile:639: recipe for target '.build_release/src/caffe/proto/caffe.pb.cc' failed
make: *** [.build_release/src/caffe/proto/caffe.pb.cc] Error 127
```

#### Resolución del problema 4

Ejecute el mandato siguiente para instalar el programa protoc:

```
sudo apt install protobuf-compiler
```

#### Síntomas 5

Al empezar la compilación e instalación de Caffe, se visualiza el mensaje siguiente:

```
src/caffe/layers/hdf5_data_layer.cpp:13:30: fatal error: hdf5/serial/hdf5.h: No such file or
directory
compilation terminated.
Makefile:582: recipe for target '.build_release/src/caffe/layers/hdf5_data_layer.o' failed
make: *** [.build_release/src/caffe/layers/hdf5_data_layer.o] Error 1
```

#### Resolución del problema 5

Es posible que necesite instalar un paquete de dependencias de Caffe utilizando los mandatos siguientes:

```
sudo apt-get install libprotobuf-dev libleveldb-dev libsnpappy-dev libopencv-dev
libhdf5-serial-dev protobuf-compiler
sudo apt-get install --no-install-recommends libboost-all-dev
sudo apt-get install libopenblas-dev liblapack-dev libatlas-base-dev
sudo apt-get install libgflags-dev libgoogle-glog-dev liblmdb-dev
```

## Instalación de Open CV

Puede instalar la biblioteca OpenCV (Open Source Computer Vision) 3.2 si necesita personalizar modelos con características de OpenCV 3.2.

#### Procedimiento

1. Obtenga el código fuente de OpenCV de Github:  
`wget https://github.com/opencv/opencv/archive/3.2.0.zip`
2. Desempaque el paquete descargado y cambie al directorio del paquete:  
`unzip 3.2.0.zip`  
`cd opencv-3.2.0`
3. Cree un subdirectorio de compilación y cambie al directorio:  
`mkdir build`  
`cd build`
4. Prepare y genere la configuración de compilación:  
`cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D WITH_TBB=ON -D WITH_V4L=ON ..`
5. Compile y cree el paquete:  
`make -j $((nproc) + 1)`
6. Instale el paquete:  
`sudo make install`
7. Registre las bibliotecas y módulos para el sistema:  
`sudo /bin/bash -c 'echo "/usr/local/lib" > /etc/ld.so.conf.d/opencv.conf'`  
`sudo ldconfig`
8. Si es necesario, desinstale la versión de opencv antigua para evitar el conflicto de versión:  
`sudo apt-get autoremove libopencv-dev`

## Instalación de bibliotecas de detección de objetos

Debe instalar una biblioteca de detección de objetos para poder ejecutar el modelo de detección de objetos en un sistema perimetral.

### Acerca de esta tarea

IBM Maximo PQI SaaS Visual Insights da soporte a las siguientes bibliotecas de detección de objetos: YOLO (You Only Look Once), Faster R-CNN y SSD (Single Shot MultiBox Detector).

### Procedimiento

1. Utilice los mandatos siguientes para instalar los paquetes de Python relacionados:

```
sudo apt-get install python-numpy
sudo apt-get install python-scipy
sudo pip install cython
sudo pip install easydict
sudo pip install uuid
sudo pip install multiprocessing
```

2. Instale todas las bibliotecas indicadas a continuación:

| Biblioteca                           | Instrucciones de instalación                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Biblioteca YOLO versión 2</b>     | <ol style="list-style-type: none"><li>a. Ejecute los mandatos siguientes para obtener el código fuente de YOLO:<br/><b>git clone --recursive https://github.com/pjreddie/darknet.git</b><br/><b>cd darknet</b><br/><b>git checkout 691debd</b></li><li>b. Edite el archivo make habilitando <b>GPU</b> y seleccione el parámetro <b>GPU ARCH</b> correcto según la configuración de máquina:<br/><b>vi Makefile</b><br/><b>GPU=1</b></li><li>c. Ejecute el siguiente mandato para compilar YOLO:<br/><b>make</b></li></ol>                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Biblioteca Python Faster-RCNN</b> | <ol style="list-style-type: none"><li>a. Ejecute el siguiente mandato para obtener el código fuente de Faster R-CNN:<br/><b>git clone --recursive https://github.com/rbgirshick/py-faster-rcnn.git</b></li><li>b. En el directorio <code>py-faster-rcnn</code>, en la carpeta <code>lib</code>, ejecute el siguiente mandato para compilar Cython:<br/><b>make</b></li><li>c. Vaya al directorio <code>caffe-fast-rcnn</code> bajo el directorio <code>py-faster-rcnn</code> y realice una copia del archivo de configuración <code>make</code> utilizando los mandatos siguientes:<br/><b>cd caffe-fast-rcnn</b><br/><b>cp Makefile.config.example Makefile.config</b></li><li>d. Añada las variables siguientes al archivo <code>Makefile.config</code>:<pre>USE_CUDNN := 1 CUDA_DIR := /usr/local/cuda PYTHON_INCLUDE := /usr/include/python2.7 \ /usr/lib /python2.7/dist-packages/numpy/core/include PYTHON_LIB := /usr/lib/x86_64-linux-gnu WITH_PYTHON_LAYER := 1</pre></li></ol> |

| Biblioteca            | Instrucciones de instalación                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       | <pre data-bbox="597 195 1133 373">INCLUDE_DIRS := \$(PYTHON_INCLUDE) /usr/local/include \ /usr/include/hdf5 /serial LIBRARY_DIRS := \$(PYTHON_LIB) /usr/local/lib /usr/lib \ /usr/lib/x86_64-linux-gnu /usr/lib/x86_64-linux-gnu/hdf5/serial</pre> <p data-bbox="558 405 1175 432">e. Ejecute el siguiente mandato para compilar Caffe:</p> <p data-bbox="583 436 651 464"><b>make</b></p> <p data-bbox="558 478 1443 537">f. Ejecute el mandato siguiente para compilar pycaffe utilizando la capa de Python:</p> <p data-bbox="583 541 776 569"><b>make pycaffe</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Biblioteca SSD</b> | <p data-bbox="558 606 1382 665">a. Ejecute el siguiente mandato para obtener el código fuente de SSD:</p> <pre data-bbox="583 638 1382 762"><b>git clone --recursive https://github.com/weiliu89/caffe.git ~/ssd-caffe</b> <b>cd ~/ssd-caffe</b> <b>git checkout ssd</b></pre> <p data-bbox="558 779 1469 837">b. Realice una copia del archivo de configuración make utilizando el mandato siguiente:</p> <pre data-bbox="583 842 1256 869"><b>cp Makefile.config.example Makefile.config</b></pre> <p data-bbox="558 886 1469 974">c. Edite el archivo de configuración Makefile y cambie los parámetros <b>CUDA_ARCH</b>, <b>BLAS</b> y <b>PYTHON_INCLUDE</b> de acuerdo con la configuración de máquina.</p> <p data-bbox="558 991 1084 1018">d. Ejecute el siguiente mandato en una línea:</p> <pre data-bbox="583 1022 1365 1115"><b>find . -type f -exec sed -i -e 's^"hdf5.h"^"hdf5/serial/hdf5.h"^g' -e 's^"hdf5_h1.h"^"hdf5/serial/hdf5_h1.h"^g' '{} ' \;</b></pre> <p data-bbox="558 1131 1105 1159">e. Compile el código con el siguiente mandato:</p> <p data-bbox="583 1163 716 1190"><b>make -j8</b></p> <p data-bbox="558 1207 1284 1234">f. Compile la capa de Python utilizando el siguiente mandato:</p> <p data-bbox="583 1239 699 1266"><b>make py</b></p> <p data-bbox="558 1283 1195 1310">g. Compile la prueba utilizando el siguiente mandato:</p> <p data-bbox="583 1314 792 1341"><b>make test -j8</b></p> |

3. Añada las siguientes variables de entorno en el archivo `~/.bashrc`: `YOLO_HOME`, `FRCNN_HOME` y `SSD_HOME`. El siguiente texto es un ejemplo de adición de variables de entorno: `YOLO_HOME=~/.darknet/` `FRCNN_HOME=~/.py-faster-rcnn/` `SSD_HOME=~/.ssd-caffe/`.

## Resolución de problemas de la instalación de biblioteca Python Faster-RCNN

Si se visualiza un mensaje de error al instalar la biblioteca Python Faster-RCNN, puede seguir unos pasos para resolver el problema.

### Síntomas

Al instalar la biblioteca Python Faster-RCNN, se visualiza el siguiente error de compilación de cudnn:

```
CXX src/caffe/layers/hdf5_data_layer.cpp
./include/caffe/util/cudnn.hpp: In function 'const char* cudnnGetErrorString(cudnnStatus_t)':
./include/caffe/util/cudnn.hpp:21:10: warning: enumeration value
'CUDNN_STATUS_RUNTIME_PREREQUISITE_MISSING' not handled in switch
./include/caffe/util/cudnn.hpp:15:28: note: in definition of macro 'CUDNN_CHECK'
 cudnnStatus_t status = condition;
Makefile:563: recipe for target '.build_release/src/caffe/layers/hdf5_data_layer.o' failed
make: *** [.build_release/src/caffe/layers/hdf5_data_layer.o] Error 1
```



En sistemas Power9 con CUDA 10, aparece el error de compilación cudnn siguiente:

```
./include/caffe/util/cudnn.hpp:127:41: error: too few arguments to function
'cudnnStatus_t cudnnSetPooling2dDescriptor(cudnnPoolingDescriptor_t, cudnnPoolingMode_t,
cudnnNanPropagation_t, int, int, int, int, int, int)' pad_h, pad_w, stride_h, stride_w));
^./include/caffe/util/cudnn.hpp:15:28: note: in definition of macro 'CUDNN_CHECK'
cudnnStatus_t status = condition; \ ^In file included from ./include/caffe/util/
cudnn.hpp:5:0, from ./include/caffe/util/device_alternate.hpp:40, from ./include/
caffe/common.hpp:19, from ./include/caffe/blob.hpp:8, from src/caffe/blob.cpp:4:/
usr/local/cuda/include/cudnn.h:991:1: note: declared here cudnnSetPooling2dDescriptor
(cudnnPoolingDescriptor_t poolingDesc, ^
```

## Resolución del problema

Para solucionar el error, busque la instalación de Caffe y ejecute los mandatos siguientes:

```
cp -rf ~/caffe-1.0/include/caffe/util/cudnn.hpp ~/py-faster-rcnn/caffe-fast-rcnn/include/caffe/util/
cp -rf ~/caffe-1.0/src/caffe/layers/cudnn_* ~/py-faster-rcnn/caffe-fast-rcnn/src/caffe/layers/
cp -rf ~/caffe-1.0/include/caffe/layers/cudnn_* ~/py-faster-rcnn/caffe-fast-rcnn/include/caffe/layers/
```

Para sistemas Power9 con CUDA 10, edite el archivo Makefile.config utilizando el mandato siguiente:

```
vim ~/py-faster-rcnn/caffe-fast-rcnn/Makefile.config
```

Cambie el valor de CUDA\_ARCH según se indica a continuación:

```
CUDA_ARCH := -gencode arch=compute_30,code=sm_30 \
-gencode arch=compute_35,code=sm_35 \
-gencode arch=compute_50,code=sm_50 \
-gencode arch=compute_50,code=compute_50
```

## Configuración del servidor de imágenes

Puede configurar el servidor de imágenes en la máquina perimetral maestra, de modo que la máquina perimetral maestra pueda almacenar las imágenes capturadas por una cámara industrial. El controlador del sistema perimetral es quien supervisa al servidor de imágenes. Cuando se añade una nueva imagen, se puntúa, y el resultado de la inspección se envía a la aplicación central para que un inspector pueda evaluar la imagen y los resultados de la inspección.

### Procedimiento

1. Instale e inicie el servicio NFS en el sistema perimetral maestro utilizando el mandato siguiente:  

```
sudo apt-get install nfs-kernel-server
```

Para Linux en Power Systems Servers, utilice los mandatos siguientes:  

```
sudo yum install -y nfs-utils
sudo yum install -y rpcbind
systemctl start nfs
systemctl start nfslock
systemctl start rpcbind
```
2. Cree un directorio denominado `imageserver` y cambie el propietario de carpeta utilizando los mandatos siguientes:  

```
sudo mkdir /imageserver
sudo chown usuario /imageserver
```

donde *usuario* es el usuario que se especifica cuando se registra el sistema perimetral en la aplicación central.
3. Opcional: Complete los pasos siguientes sólo si tiene tanto un sistema perimetral maestro y un sistema perimetral esclavo.
  - a. Para exportar el directorio `imageserver`, edite el archivo `/etc/exports` utilizando el mandato siguiente:  

```
vi /etc/exports
```

- b. Añada la línea siguiente al archivo de exportación. La red de dirección IP es la subred IP que puede acceder a la carpeta compartida. El formato es:  
*dirección IP/máscara de subred*. Todos los sistemas perimetrales esclavos deben poder acceder a la carpeta compartida:  
`/imageserver dirección IP/red(rw, sync, no_root_squash, no_all_squash)`  
Por ejemplo:  
`/imageserver 10.173.0.0/29(rw, sync, no_root_squash, no_all_squash)`
- c. Reinicie el servicio NFS utilizando el mandato siguiente:  
`sudo service nfs-server restart`
4. En cada máquina perimetral esclava, monte el directorio `/imageserver` en la carpeta compartida del sistema perimetral maestro:
  - a) Ejecute el mandato siguiente en la máquina perimetral esclava:  
**mkdir /imageserver**
  - b) Edite el archivo `/etc/fstab` utilizando el mandato siguiente:  
**vi /etc/fstab**
  - c) Añada la línea siguiente al archivo, especifique la dirección IP del sistema perimetral maestro:  
*dirección\_IP:/imageserver /imageserver nfs defaults 0 0*
  - d) Ejecute el mandato siguiente en la máquina perimetral esclava para que el montaje entre en vigor:  
**mount -a**

## Configuración del almacén de modelos

---

Debe configurar el almacén de modelos en la máquina perimetral maestra. El almacén de modelos es un repositorio para modelos ejecutables distribuidos desde la aplicación central.

### Procedimiento

1. Cree un directorio `modelstore` y cambie el propietario de carpeta en el sistema perimetral maestro utilizando los mandatos siguientes:  
`sudo mkdir /modelstore`  
`sudo chown usuario /modelstore`  
donde *usuario* es el usuario que se especifica cuando se registra el sistema perimetral en la aplicación central.
2. Opcional: Complete los pasos siguientes sólo si tiene tanto un sistema perimetral maestro y un sistema perimetral esclavo.
  - a. Para exportar el directorio `modelstore`, edite el archivo `/etc/exports` utilizando el mandato siguiente:  
`vi /etc/exports`
  - b. Añada la línea siguiente al archivo de exportación. La red de dirección IP es la subred IP que puede acceder a la carpeta compartida. El formato es:  
*dirección IP/máscara de subred*. Todos los sistemas perimetrales esclavos deben poder acceder a la carpeta compartida:  
`/modelstore dirección IP/red(rw, sync, no_root_squash, no_all_squash)`  
Por ejemplo:  
`/modelstore 10.173.0.0/29(rw, sync, no_root_squash, no_all_squash)`
  - c. Reinicie el servicio NFS utilizando el mandato siguiente:  
`sudo service nfs-server restart`
3. En cada máquina perimetral esclava, monte el directorio `/modelstore` en la carpeta compartida del sistema perimetral maestro:
  - a) Ejecute el mandato siguiente en la máquina perimetral esclava:  
**mkdir /modelstore**
  - b) Edite el archivo `/etc/fstab` utilizando el mandato siguiente:

**vi /etc/fstab**

- c) Añada la línea siguiente al archivo, especifique la dirección IP del sistema perimetral maestro:  
`dirección_IP:/modelstore /modelstore nfs defaults 0 0`
- d) Ejecute el mandato siguiente en la máquina perimetral esclava para que el montaje entre en vigor:  
**mount -a**

## Instalación de módulos Python

---

Complete esta tarea para instalar los módulos Python de requisito previo.

### Procedimiento

Ejecute los siguientes mandatos en el sistema perimetral maestro y en cada sistema perimetral esclavo para instalar paquetes de requisitos previos:

```
sudo pip install flask
sudo pip install gevent
sudo pip install requests
sudo pip install pyinotify
sudo pip install opencv-python
sudo pip install lmbd
Para Ubuntu: sudo apt-get install dos2unix
```

Para Linux en Power Systems Servers: `sudo yum install dos2unix`  
`sudo apt install curl`  
`sudo pip install flask_httpauth`  
`sudo pip install paramiko`  
`sudo pip install tornado`

## Instalación de Paramiko para Linux en Power Systems Servers

Si tiene problemas para instalar Paramiko para sistemas Linux en Power Systems Servers, utilice esta tarea para instalar Paramiko manualmente.

### Procedimiento

1. Ponga el archivo `rhel-75-server.repo` en `/etc/yum.repos.d`
2. Ejecute el siguiente mandato:  
`sudo yum install redhat-rpm-config gcc libffi-devel python-devel openssl-devel`
3. Ejecute el siguiente mandato:  
`pip install cryptography --no-binary cryptography`
4. Ejecute el siguiente mandato:  
`sudo pip install bcrypt`
5. Ejecute el siguiente mandato:  
`sudo pip install pynacl`
6. Ejecute el siguiente mandato:  
`sudo pip install paramiko`

## Registro del sistema perimetral en la aplicación central

---

Debe registrar los sistemas perimetrales en la aplicación central. El procedimiento es diferente si los sistemas perimetrales están conectados o son autónomos.

Hay dos tipos de sistemas perimetrales: conectados e autónomos. Los sistemas perimetrales conectados tienen una dirección IP pública que permite a la aplicación central conectarse directamente con el

sistema perimetral. Este envía los resultados de la inspección a la aplicación central en tiempo real. El sistema perimetral autónomo no tiene una dirección IP pública y, en algunas ocasiones, no puede conectarse a la aplicación central. Los resultados de la inspección deben almacenarse localmente en el sistema perimetral autónomo y el administrador del sistema perimetral debe llamar al servicio de este sistema de forma explícita para comunicarse con la aplicación central cuando el sistema perimetral se conecta a Internet.

## Registro de un sistema conectado en la aplicación central

Después de configurar el sistema perimetral conectado, debe registrarlo en la aplicación central. Puede crear un sistema perimetral o editar uno ya existente. Los sistemas perimetrales se utilizan para ejecutar un modelo de puntuación.

### Procedimiento

1. En el **Administrador de modelos**, seleccione **Datos > Sistemas perimetrales**.
2. Seleccione **Crear sistema perimetral nuevo** e indique el nombre del sistema perimetral.
3. Indique la dirección IP, el nombre de usuario de SSH y la contraseña del sistema perimetral y especifique el tipo perimetral como **Maestro** o **Esclavo**. Es necesario que la aplicación central pueda acceder a la dirección IP. El nombre de usuario de SSH y la contraseña se utilizan para iniciar la sesión en el sistema perimetral y desplegar el controlador del sistema perimetral.
4. Si está creando un sistema perimetral maestro, especifique conectado como modo de conexión. Si va a crear un sistema perimetral esclavo, especifique el sistema perimetral maestro correspondiente.
5. Si ha seleccionado la modalidad de conexión, seleccione **Siguiente** y especifique la política de supresión.
6. Seleccione **Crear** para crear el sistema perimetral. El controlador perimetral y el motor de puntuación se despliegan en el sistema perimetral y se añade el sistema perimetral a la lista registrada.

## Registro de un sistema autónomo en la aplicación central

Después de configurar un sistema perimetral autónomo, debe registrarlo en la aplicación central. Puede crear un sistema perimetral o editar uno ya existente. Los sistemas perimetrales se utilizan para ejecutar un modelo de puntuación.

### Acerca de esta tarea

Un sistema perimetral autónomo le permite realizar algunas funciones del producto cuando está desconectado del sistema central, como puntuar imágenes en la línea de producción, sincronizar datos con el sistema central y limpiar datos cargados. Para instalar un sistema perimetral autónomo, debe instalar PostgreSQL y los módulos Python necesarios. También debe instalar el sistema perimetral en el sistema manualmente.

### Procedimiento

1. En el **Administrador de modelos**, seleccione **Datos > Sistemas perimetrales**.
2. Seleccione **Crear sistema perimetral nuevo** e indique el nombre del sistema perimetral.
3. Indique la dirección IP, el nombre de usuario de SSH y la contraseña del sistema perimetral y especifique el tipo perimetral como **Maestro** o **Esclavo**. Es necesario que la aplicación central pueda acceder a la dirección IP. El nombre de usuario de SSH y la contraseña se utilizan para iniciar la sesión en el sistema perimetral y desplegar el controlador del sistema perimetral.
4. Si está creando un sistema perimetral maestro, especifique autónomo como modo de conexión.
5. Si ha seleccionado la modalidad de conexión, seleccione **Siguiente** y especifique la política de supresión.
6. Seleccione **Crear** para crear el sistema perimetral. Para un sistema perimetral autónomo, el administrador del sistema perimetral debe realizar un inicio de sesión SSH en el sistema perimetral e instalar el sistema perimetral manualmente. Para obtener más información, consulte el tema **Instalación de un sistema perimetral autónomo**.

## Instalación de un sistema perimetral autónomo

Un sistema perimetral autónomo le permite realizar algunas funciones del producto cuando está desconectado del sistema central, como puntuar imágenes en la línea de producción, sincronizar datos con el sistema central y limpiar datos cargados. Para instalar un sistema perimetral autónomo, debe instalar PostgreSQL y el módulo Python necesario. También debe instalar el sistema perimetral en la máquina perimetral manualmente.

Para crear un sistema perimetral autónomo, debe seleccionar la modalidad de conexión autónoma al crear el sistema perimetral.

Debe crear el sistema perimetral autónomo en el mismo sistema que el sistema perimetral maestro.

### Instalación de PostgreSQL para Ubuntu

Utilice esta tarea para instalar PostgreSQL para sistemas Ubuntu. El servidor de formación y el sistema perimetral autónomo utilizan una base de datos PostgreSQL.

### Procedimiento

1. Instale PostgreSQL, PHPPgadmin y Apache2 utilizando el mandato siguiente:  

```
sudo apt-get -y install postgresql postgresql-contrib phppgadmin
```
2. Configure el usuario de PostgreSQL.
  - a) Inicie sesión como el usuario de PostgreSQL utilizando los mandatos siguientes:  

```
sudo su
su - postgres
psql
```
  - b) Configure la contraseña del usuario postgres utilizando los mandatos siguientes:  

```
password postgres
contraseña
\q
```
3. Configure Apache2 editando el archivo `nano phppgadmin.conf`:  

```
cd /etc/apache2/conf-available/
nano phppgadmin.conf
```

Suprima la línea siguiente: `Require local`. Añada la línea siguiente al archivo:  

```
Require all granted
```
4. Configure PHPPgadmin editando el archivo `config.inc.php`:  

```
cd /etc/phppgadmin/
nano config.inc.php
```

Busque la línea siguiente en el archivo:  

```
$conf['extra_login_security'] = true
```

Cambie `true` por `false`.
5. Reinicie PostgreSQL y Apache2 utilizando los mandatos siguientes:  

```
systemctl restart postgresql
systemctl restart apache2
```
6. Verifique que puede acceder a la interfaz de usuario en el sistema perimetral autónomo accediendo al URL siguiente:  

```
http://IP_sistema_perimetral_autonomo/phppgadmin
```

donde `IP_sistema_perimetral_autonomo` es la dirección IP del sistema perimetral autónomo.
7. Cree el esquema de base de datos en PostgreSQL.
  - a) Ejecute el mandato siguiente en la consola SQL en PHPPgadmin:  

```
create database edge with owner postgres encoding='UTF-8'
lc_collate='en_US.utf8' lc_ctype='en_US.utf8' template template0;
```
  - b) En la base de datos, cree las tablas siguientes:  

```
CREATE TABLE vi_arrendatario_inspectionresult(id text, info jsonb);
CREATE TABLE vi_arrendatario_notification(id text, info jsonb);
CREATE TABLE vi_arrendatario_defectsummary(id text, info jsonb);
CREATE TABLE vi_arrendatario_uploaddataset(id text, info jsonb);
```

```
CREATE TABLE vi_arrendatario_syncprocess(id text, info jsonb);
CREATE TABLE vi_arrendatario_model(id text, info jsonb);
CREATE TABLE vi_arrendatario_datagroup(id text, info jsonb);
```

donde *arrendatario* es el arrendatario para el usuario de la operación en el sistema central de Maximo PQI SaaS Visual Insights. Obtenga el valor de arrendatario a partir del perfil de usuario de la interfaz de usuario de la aplicación central.

## Instalación de PostgreSQL para Linux en Power Systems Servers

Utilice esta tarea para instalar PostgreSQL para Linux en Power Systems Servers. El servidor de formación y el sistema perimetral autónomo utilizan una base de datos PostgreSQL.

### Procedimiento

1. Haga su en root utilizando el mandato siguiente:  
`sudo su`
2. Descargue el código fuente de PostgreSQL:  
`wget https://ftp.postgresql.org/pub/source/v9.5.13/postgresql-9.5.13.tar.gz`
3. Instale PostgreSQL utilizando los mandatos siguientes:  
`tar -zxvf postgresql-9.5.13.tar.gz`  
`cd postgresql-9.5.13/`  
`yum -y install readline-devel`  
`./configure --prefix=/usr/local/postgresql`  
`make`  
`make install`
4. Cree el usuario postgres y cambie el propietario del directorio postgres:  
`useradd postgres`  
`chown -R postgres:postgres /usr/local/postgresql/`
5. Cambie al usuario postgres:  
`su postgres`
6. Configure la vía de acceso del sistema para postgres:  
`vi ~/.bashrc`  
`PGHOME=/usr/local/postgresql`  
`export PGHOME`  
`PGDATA=/usr/local/postgresql/data`  
`export PGDATA`  
`PATH=$PATH:$HOME/.local/bin:$HOME/bin:$PGHOME/bin`  
`export PATH`
7. Obtenga el código fuente de la configuración:  
`source ~/.bashrc`
8. Inicialice la base de datos PostgreSQL:  
`initdb`
9. Configure la base de datos. Abra `postgresql.conf` en vi:  
`vi /usr/local/postgresql/data/postgresql.conf`  
Cambie:

```
#listen_address='localhost'
#port = 5432
```

a:

```
listen_address='*'
port = 5432
```

Abra el archivo `pg_hba.conf` en vi:  
`vi /usr/local/postgresql/data/pg_hba.conf`  
Añada la línea siguiente al archivo:  
`host all all 0.0.0.0/0 trust`

10. Reinicie postgresql:

```
pg_ctl -D /usr/local/postgresql/data -l logfile restart
```

11. Cambie la contraseña del usuario postgres en la base de datos PostgreSQL:

```
psql
ALTER USER postgres WITH PASSWORD 'password';
\q
```

Si el servicio postgresql no se ha iniciado, ejecute los mandatos siguientes:

```
su postgres
```

```
vi ~/.bashrc
```

Añada /usr/local/pgsql/bin/ al archivo:

```
export PATH=/usr/local/cuda-8.0/bin:$PATH:/usr/local/pgsql/bin/
```

Ejecute el mandato siguiente:

```
source ~/.bashrc
```

12. Cree el esquema de base de datos en PostgreSQL. Ejecute el mandato siguiente en la consola psql:

```
create database edge with owner postgres encoding='UTF-8'
lc_collate='en_US.utf8' lc_ctype='en_US.utf8' template template0;
```

En la base de datos, cree las tablas siguientes:

```
CREATE TABLE vi_tenant_inspectionresult(id text, info jsonb);
CREATE TABLE vi_tenant_notification(id text, info jsonb);
CREATE TABLE vi_tenant_defectsummary(id text, info jsonb);
CREATE TABLE vi_tenant_uploaddataset(id text, info jsonb);
CREATE TABLE vi_tenant_syncprocess(id text, info jsonb);
CREATE TABLE vi_tenant_model(id text, info jsonb);
CREATE TABLE vi_tenant_datagroup(id text, info jsonb);
```

donde *arrendatario* es el arrendatario para el usuario de la operación en el sistema central de Maximo PQI SaaS Visual Insights. Puede obtener el valor de arrendatario a partir del perfil de usuario de la interfaz de usuario de la aplicación central.

### Instalación de módulos Python

Para crear un servidor de formación o un sistema perimetral autónomo, debe instalar el módulo Python necesario.

### Procedimiento

Instale el módulo Python utilizando los mandatos siguientes:

```
sudo apt-get update
sudo apt -y install postgresql
sudo apt -y install libpq-dev
sudo pip install PyGreSQL
sudo pip install DBUtils
```

### Instalación del sistema perimetral autónomo

Para el sistema perimetral autónomo, debe instalar el sistema perimetral en la máquina perimetral manualmente.

### Descarga del archivo de compilación y el script de instalación

Debe descargar el archivo de compilación y el script de instalación del sistema perimetral autónomo para instalar un sistema perimetral autónomo.

### Procedimiento

1. En la aplicación Maximo PQI SaaS Visual Insights, seleccione **Datos > Sistemas perimetrales**. Seleccione el sistema perimetral maestro y vea los detalles del sistema perimetral.
2. Seleccione y descargue el archivo de compilación para el sistema operativo necesario. Hay soporte para los sistemas operativos siguientes: Ubuntu, Red Hat Enterprise Linux e IBM Power System.
3. Seleccione y descargue el archivo de script de shell de instalación.

### **Instalación del archivo de compilación en el sistema perimetral autónomo**

Para instalar el sistema perimetral en la máquina perimetral, debe utilizar un nombre de usuario SSH y realizar un inicio de sesión SSH en la máquina perimetral.

#### **Procedimiento**

1. Tras el inicio de sesión SSH, cambie el directorio de trabajo a la vía de acceso de despliegue del cuadro de diálogo de creación de sistema perimetral.
2. Añada el permiso de ejecución al archivo de script de shell utilizando el mandato siguiente:  
`chmod +x ./edgeDeployed.sh`
3. Ejecute el archivo de script de shell para instalar el sistema perimetral autónomo. Utilice el mandato siguiente:  
`./edgeDeployed.sh`
4. Actualice el archivo `deploy_path/vi_edge-bin_vi/vi_edge/postgres/DBConfig.json` con la información de PostgreSQL para corregir la conexión de base de datos. A continuación, ejecute los siguientes mandatos para reiniciar el sistema perimetral:  
`cd deploy_path/vi_edge-bin_vi/vi_edge/`  
`./restartController.sh`

### **Comprobar el estado de los servicios en los sistemas perimetrales**

Después de registrar el sistema perimetral en la aplicación central, compruebe que se estén ejecutando los procesos Python correctos en los sistemas perimetrales.

#### **Procedimiento**

1. Después de haber instalado el sistema perimetral, realice un inicio de sesión SSH en el sistema perimetral.
2. Ejecute el mandato siguiente en el sistema perimetral maestro para buscar el proceso Python del motor maestro:  
`ps aux | grep python`  
El resultado debe incluir el siguiente proceso Python:  
`python carpeta_despliegue/vi_edge-bin_vi/vi_task_manager/run.py.`  
Si no encuentra el proceso Python en el sistema perimetral, compruebe los archivos de registro en:  
`carpeta_despliegue/vi_edge-bin_vi/vi_task_manager/master.log.`  
Intente iniciar el motor maestro ejecutando el mandato siguiente:  
`carpeta_despliegue/vi_edge-bin_vi/vi_task_manager/restartMaster.sh`  
`IP_de_nodo_maestro`
3. Ejecute el mandato siguiente en el sistema perimetral maestro para buscar el proceso Python del controlador del sistema perimetral:  
`ps aux | grep python`  
El resultado debe incluir los siguientes procesos de Python:  
`python carpeta_despliegue/vi_edge-bin_vi/vi_edge/runMonitor.py`  
`python carpeta_despliegue/vi_edge-bin_vi/vi_edge/runService.py.`  
Si no encuentra el proceso Python `runMonitor` en el sistema perimetral maestro, consulte los archivos de registro en `carpeta_despliegue/vi_edge-bin_vi/vi_edge/Event.log.`  
Si no encuentra el proceso Python `runService` en el sistema perimetral maestro, consulte los archivos de registro en `carpeta_despliegue/vi_edge-bin_vi/vi_edge/Service.log.`  
Intente iniciar el controlador perimetral ejecutando el mandato siguiente:  
`carpeta_despliegue/vi_edge-bin_vi/vi_edge/restartController.sh`
4. Ejecute el mandato siguiente en el sistema perimetral maestro y los sistemas perimetrales esclavos para buscar procesos Python para el motor de puntuación del modelo de clasificación:  
`ps aux | grep python`  
El resultado debe incluir los siguientes procesos Python:  
`python carpeta_despliegue/vi_edge-bin_vi/vi_score_engine_restful/`  
`front_run.py 5005 6005`



```
python carpeta_despliegue/vi_edge-bin_vi/vi_score_engine_restful/back_run.py
6005
```

Si no encuentra el proceso Python en los sistemas perimetrales, compruebe los archivos de registro:  
*carpeta\_despliegue/vi\_edge-bin\_vi/vi\_score\_engine\_restful/front\_log.txt*  
*carpeta\_despliegue/vi\_edge-bin\_vi/vi\_score\_engine\_restful/back\_log.txt*  
Después de resolver los problemas, el motor maestro debe iniciar el motor de puntuación automáticamente. Para iniciar manualmente el motor de puntuación, ejecute los mandatos siguientes:  
source ~/.bashrc  
*carpeta\_despliegue/vi\_edge-bin\_vi/vi\_score\_engine\_restful/restartEngine.sh*

5. Ejecute el mandato siguiente en el sistema perimetral maestro y en los sistemas perimetrales esclavos para buscar el proceso Python del motor de puntuación del modelo de detección de objetos:

```
ps aux | grep python
```

El resultado debe incluir los siguientes procesos Python:

```
python deployment_folder/vi_edge-bin_vi/vi_obj_detection/RESTAPI/model/
run.py port gpu_id FRCNN
```

```
python deployment_folder/vi_edge-bin_vi/vi_obj_detection/RESTAPI/model/
run.py port gpu_id SSD
```

Si no encuentra los procesos Python en el sistema perimetral, consulte los siguientes archivos de registro:

```
carpeta_despliegue/vi_edge-bin_vi/vi_obj_detection/RESTAPI/model/
FRCNN_gpu_id.log
```

```
carpeta_despliegue/vi_edge-bin_vi/vi_obj_detection/RESTAPI/model/
SSD_gpu_id.log
```

Si ve un mensaje de error "No hay ningún módulo denominado *iotmyolo*", copie el archivo *iotmyolo.so* o *iotmyolo.py* del directorio */home/usuario/vi\_edge-bin\_vi/vi\_obj\_detection/model\_library/yolo* en el directorio *YOLO\_HOME*.

Después de resolver los problemas, el motor maestro debe iniciar el motor de puntuación automáticamente.

6. Vuelva a crear la biblioteca YOLO con el componente perimetral:

- a. Vaya al directorio */home/usuario/vi\_edge-bin\_vi/vi\_obj\_detection/model\_library/yolo*.

- b. Añada el archivo *detectorobj.c* a la carpeta *darknet/examples*.

- c. Edite el archivo *darknet/Makefile* e indique *EXECOBJA=detectorobj.o*. El siguiente código es un ejemplo del código en el archivo *Makefile*:

```
EXECOBJA=detectorobj.o captcha.o lsd.o super.o voxel.o art.o tag.o cifar.o go.o rnn.o
rnn_vid.o compare.o segmenter.o regressor.o classifier.o coco.o dice.o yolo.o
detector.o writing.o nightmare.o swag.o darknet.o
```

- d. En el archivo *Makefile*, añada *\$(EXECOBJ)* para los objetos *\$(SLIB)* y *\$(ALIB)*. El siguiente código es un ejemplo de código en el archivo *Makefile*:

```
$(ALIB): $(EXECOBJ) $(OBJS)
$(AR) $(ARFLAGS) $@ $^
$(SLIB): $(EXECOBJ) $(OBJS)
$(CC) $(CFLAGS) -shared $^ -o $@ $(LDFLAGS)
```

- e. Ejecute el siguiente mandato:  
make

## Actualización de sistemas perimetrales

Actualice los sistemas perimetrales si son de una versión anterior. Puede descartar el script de instalación y el archivo de compilación del sistema perimetral para actualizarlo.

## Acerca de esta tarea

Se actualiza la aplicación central de Maximo PQI SaaS Visual Insights, pero no actualiza el sistema perimetral, se producirá una incoherencia entre ambos y es posible que los modelos no puedan implementarse. Es posible que vea un mensaje de error similar a este; "El sistema perimetral debe actualizarse para desplegar el nuevo modelo."

## Procedimiento

1. Ejecute el siguiente mandato para obtener el ID del sistema perimetral.  
`/ibm/iotm/vi/service/edge?user=xxx&solution=vi`
2. Para obtener el script de shell de instalación, llame al siguiente servicio:  
`/ibm/iotm/vi/service/edgeFile?edgeId=xxx&version=shell&user=xxx&solution=vi`  
Sustituya `edgeId=xxx` por el ID del sistema perimetral que haya obtenido en el paso anterior. Guarde el archivo como `edgeDeployed.sh`. Vea este comando de muestra:  
`curl -k -H "apikey:yourapikey" "https://iotm.predictivesolutionsapps.ibmcloud.com/ibm/iotm/vi/service/edgeFile?edgeId=1508123458000&version=shell&user=youruser&solution=vi" > edgeDeployed.sh`
3. Compruebe el archivo `edgeDeployed.sh`. Compruebe los valores de `username`, que es el usuario SSH que se ha utilizado para iniciar los servicios del sistema perimetral, `password`, que es la contraseña del usuario SSH y `basefolder`, que es la ruta de despliegue de los servicios del sistema perimetral. Introduzca el valor de `<to edit>`.
4. Para obtener el archivo de compilación del sistema perimetral, llame al siguiente servicio:  
`/ibm/iotm/vi/service/edgeFile?edgeId=xxx&version=xxx&user=xxx&solution=vi`  
donde `version` es `ubuntu`, `redhat` o `power` en función del sistema. Guarde el archivo como `vi_edge-bin_vi.zip`. Vea este comando de muestra:  
`curl -k -H "apikey:yourapikey" -o vi_edge-bin_vi.zip "https://iotm.predictivesolutionsapps.ibmcloud.com/ibm/iotm/vi/service/edgeFile?edgeId=1508123458000&version=ubuntu&user=youruser&solution=vi"`
5. Coloque los archivos `edgeDeployed.sh` y `vi_edge-bin_vi.zip` en el sistema perimetral.
6. Inicie sesión en el sistema perimetral como el usuario SSH que se usó al crear el sistema perimetral en la aplicación central.
7. Coloque los archivos `edgeDeployed.sh` y `vi_edge-bin_vi.zip` en el directorio `deployPath`.
8. Cierre el servicio del sistema perimetral actual.
9. Ejecute los mandatos siguientes:  
`dos2unix edgeDeployed.sh`  
`chmod +x edgeDeployed.sh`  
`./edgeDeployed.sh`
10. Compruebe si se ha iniciado el servicio Python del sistema perimetral.

## Tareas relacionadas

[Comprobar el estado de los servicios en los sistemas perimetrales](#)

Después de registrar el sistema perimetral en la aplicación central, compruebe que se estén ejecutando los procesos Python correctos en los sistemas perimetrales.

---

## Capítulo 3. Creación y uso de modelos

Los modelos se crean para recopilar información histórica sobre imágenes y defectos. La información se utiliza para la formación del modelo. Tras la formación, el modelo debe validarse para poder desplegarse en un sistema perimetral. La validación del modelo proporciona información de precisión del modelo. Puede haber varias versiones de un modelo. Los modelos pueden compartir información de defectos, pero tienen distintos archivos de imagen de distintas líneas de producto. Puede repetir la formación de un modelo para intentar obtener una mayor precisión del modelo, para que la versión del modelo pueda sustituirse por una versión del modelo más reciente. Hay dos tipos de implementaciones de modelos: El modelo de clasificación y el modelo de detección de objetos.

---

### Estructura de archivos de imagen comprimidos

Para poder añadir imágenes históricas de grupos de imágenes, debe tener archivos que contengan los archivos de imagen que necesita para el modelo de clasificación o el modelo de detección de objetos.

#### Modelo de clasificación

Añada las imágenes a los archivos comprimidos. Un archivo comprimido debe contener todas las imágenes que pertenecen al mismo grupo de imágenes. Debe colocar todas las imágenes en una estructura plana sin subcarpetas en el archivo comprimido. Se admiten los tipos de imágenes siguientes: PNG, JPEG y JPG. Los archivos de imagen deben tener todos extensiones de archivo en mayúsculas, por ejemplo, PNG, JPEG y JPG, o deben tener todos extensiones de archivo en minúsculas, por ejemplo, png, jpeg y jpg.

#### Modelo de detección de objetos

La estructura del modelo de detección de objetos debe contener dos carpetas en un archivo comprimido. Una carpeta debe denominarse JPEGImages y la otra debe denominarse Annotations. Además de las dos carpetas, el archivo comprimido también debe contener un archivo labels.txt.

Añada todos los archivos de imagen a la carpeta JPEGImages. Se admiten los tipos de imágenes siguientes: PNG, JPEG y JPG. Los archivos de imagen deben tener todos extensiones de archivo en mayúsculas, por ejemplo, PNG, JPEG y JPG, o deben tener todos extensiones de archivo en minúsculas, por ejemplo, png, jpeg y jpg. Añada todos los archivos de anotaciones a la carpeta Annotations. Un archivo de anotaciones debe tener el mismo nombre de archivo que el archivo de imagen. Los archivos debe tener un formato XML. Si utiliza la herramienta de etiquetado de imágenes de Maximo PQI SaaS Visual Insights, la salida es un archivo de imagen comprimido que contiene el archivo de imágenes y el archivo de anotaciones. Si utiliza una herramienta de etiquetado de imágenes externa, debe asegurarse de que el archivo de imágenes y el archivo de anotaciones tienen la estructura esperada. La siguiente información es un ejemplo de un archivo de anotaciones:

```
<annotation>
 <folder>JPEGImages</folder>
 <filename>000001.jpg</filename>
 <source>
 <database>Unknown</database>
 </source>
 <size>
 <width>864</width>
 <height>1296</height>
 <depth>3</depth>
 </size>
 <segmented>0</segmented>
 <object>
 <name>defect1</name>
 <pose>Unspecified</pose>
 <truncated>0</truncated>
 <difficult>0</difficult>
 <bndbox>
 <xmin>474</xmin>
```

```

 <ymin>368</ymin>
 <xmax>540</xmax>
 <ymin>448</ymin>
 </bndbox>
</object>
<object>
 <name>defect2</name>
 <pose>Unspecified</pose>
 <truncated>0</truncated>
 <difficult>0</difficult>
 <bndbox>
 <xmin>303</xmin>
 <ymin>387</ymin>
 <xmax>369</xmax>
 <ymin>452</ymin>
 </bndbox>
</object>
</annotation>

```

El archivo labels.txt contiene los nombres de todos los tipos de defectos que hay en la carpeta de anotaciones. Cada defecto debe estar en una línea distinta, como se muestra en el siguiente ejemplo:

```

defect1
defect2
defect3

```

## Adición de imágenes históricas para grupos de imágenes

El administrador de modelos utiliza imágenes históricas para formar el modelo.

### Acerca de esta tarea

Puede cargar archivos de imagen comprimidos (.zip) en el grupo de imágenes. Prepare los archivos de imágenes comprimidos con el tamaño de archivo correcto. Si un archivo es demasiado grande, puede dividir el archivo en varios archivos. Basándose en la velocidad de carga de la red en la aplicación, asegúrese de que cada archivo de imagen comprimido pueda cargarse en un plazo de 30 minutos. Por ejemplo, si la velocidad de subida de la red en la aplicación es de 100 KB/s, el tamaño máximo de un archivo de imagen comprimido debe ser de 200 MB.

### Procedimiento

1. Seleccione **Datos > Grupos de imágenes > Nuevo grupo de imágenes**. Seleccione **Nuevo grupo de imágenes no etiquetadas** si desea utilizar la herramienta de etiquetado de imágenes de Maximo PQI SaaS Visual Insights para etiquetar las imágenes. Seleccione **Nuevo grupo de imágenes no etiquetadas** si ya se han etiquetado las imágenes mediante una herramienta externa.
2. Añada un nombre de grupo de imágenes exclusivo y una descripción, seleccione el tipo de grupo de imágenes y seleccione **Siguiente**.

Para tipo del grupo de imágenes:

- Single characteristics (características individuales) implica que las imágenes del grupo pertenecen a un tipo de defecto. La característica individual utiliza el modelo de clasificación.
- Not a defect (no es un defecto) hace referencia a características individuales donde las imágenes del grupo no tienen un defecto. No es un defecto utiliza el modelo de clasificación.
- Multiple characteristics (características múltiples) implica que las imágenes contienen uno o más defectos cada una que pertenecen al mismo tipo de defecto o a otro distinto. Característica múltiple utiliza el modelo de detección de objetos.

**Nota:** Una vez cargado el conjunto de imágenes, el tipo de grupo de imágenes que se selecciona no se puede cambiar.

3. En el panel **Conjuntos de imágenes**, añada imágenes y seleccione **Añadir grupo de imágenes**.

## Resultados

En el panel **Grupos de imágenes**, puede seleccionar un grupo de imágenes y seleccione Editar. Puede añadir o suprimir los conjuntos de imágenes, actualizar el nombre del grupo de imágenes o la descripción, o cambiar el tipo de grupo de imágenes.

**Nota:** Si el conjunto de imágenes se cita en una instancia de modelo, el conjunto de imágenes no puede suprimirse.

## Herramienta de etiquetado de imágenes

---

La herramienta de etiquetado de imágenes se puede utilizar para etiquetar imágenes de forma semiautomática. Puede cargar archivos de imágenes no etiquetadas, etiquetar manualmente algunas de las imágenes y, a continuación, activar el etiquetado automático para etiquetar las imágenes que no están etiquetadas. Puede comprobar el resultado del etiquetado automático. En función de la información de resumen, puede aceptar el resultado del etiquetado. Las imágenes se etiquetan y empaquetan en los archivos zip de imágenes correspondientes, los cuales se asocian automáticamente a grupos de imágenes, de modo que se puedan utilizar en el proceso de formación del modelo.

### Creación de un grupo de imágenes no etiquetadas

Puede cargar archivos de imágenes comprimidos (.zip) que contienen imágenes sin etiqueta para crear un grupo de imágenes sin etiquetas.

#### Acerca de esta tarea

Cuando cree un grupo de imágenes no etiquetadas, puede utilizar un archivo comprimido que contenga imágenes no etiquetadas o un archivo comprimido que contenga imágenes etiquetadas y no etiquetadas. Para archivos de características individuales, cualquier imagen etiquetada debe estar incluida en una subcarpeta, en la que el nombre de la carpeta es el nombre del defecto. En los archivos de varias características, el archivo de imágenes etiquetadas debe tener un archivo XML de anotación correspondiente en la misma carpeta.

Prepare los archivos de imágenes comprimidos con el tamaño de archivo correcto. Si un archivo es demasiado grande, puede dividir el archivo en varios archivos. Basándose en la velocidad de carga de la red en la aplicación, asegúrese de que cada archivo de imagen comprimido pueda cargarse en un plazo de 30 minutos. Por ejemplo, si la velocidad de subida de la red en la aplicación es de 100 KB/s, el tamaño máximo de un archivo de imagen comprimido debe ser de 200 MB.

#### Procedimiento

1. Seleccione **Datos > Nuevo grupo de imágenes > Nuevo grupo de imágenes no etiquetadas**.
2. En el recuadro de diálogo **Nuevo grupo de imágenes no etiquetadas**, especifique un nombre para el grupo de imágenes.
3. Especifique el tipo del grupo de imágenes:
  - Las características individuales hacen referencia a las imágenes del grupo que solo tienen un tipo de defecto.
  - Las características múltiples hacen referencia a las imágenes del grupo que tienen uno o varios defectos.
4. Pulse el enlace para cargar un archivo comprimido que contiene imágenes no etiquetadas.
5. Pulse **Añadir grupo de imágenes** para crear el nuevo grupo de imágenes no etiquetadas.
6. En el recuadro de diálogo **Reglas de muestreo**, establezca la proporción de imágenes que se han de etiquetar manualmente y pulse **Confirmar**.

#### Resultados

Una vez completada esta tarea, comience a etiquetar manualmente las imágenes de muestreo.

## Etiquetado manual de imágenes de muestreo

Puede etiquetar manualmente imágenes incluidas en un grupo de imágenes no etiquetadas.

### Procedimiento

1. En la ventana **Todos los grupos de imágenes no etiquetadas**, pulse **Ver** junto al grupo que contiene las imágenes que desea etiquetar.
2. Pulse el icono de reglas de muestreo para establecer la tasa de muestreo, que indica el número de imágenes que desea etiquetar manualmente. Pulse **Confirmar**.
3. Para cada imagen, etiquete manualmente la imagen seleccionando la etiqueta correspondiente y, a continuación, guarde la imagen. Puede seleccionar etiquetas existentes o crear nuevas etiquetas. Para grupos de características individuales, una etiqueta corresponde a un grupo de datos. En el caso de múltiples características, se crea un grupo de datos con el mismo nombre de grupo. Todas las etiquetas que se utilizan en las imágenes del grupo no etiquetado se guardan como etiquetas en el grupo de datos correspondientes.
4. Una vez etiquetadas todas las imágenes de muestreo, puede pulsar **Ejecutar etiqueta automática** para iniciar el proceso de etiquetado automático.

## Etiquetado automático de imágenes de muestreo

Maximo PQI SaaS Visual Insights utiliza técnicas de aprendizaje basadas en Convolutional Neural Network para etiquetar automáticamente las imágenes.

### Procedimiento

1. En la ventana **Todos los grupos de imágenes no etiquetadas**, pulse **Ver** junto al grupo que contiene las imágenes que desea etiquetar.
2. Pulse **Ejecutar etiqueta automática** para iniciar el proceso de etiquetado automático. Puede pulsar **Ejecutar como proceso de fondo** para ejecutar el proceso de etiquetado automático como proceso de fondo.
3. Una vez completado el proceso de etiquetado automático, pulse **Ver** para revisar los resultados.
4. Para cada imagen etiquetada automáticamente, confirme o modifique la etiqueta.
5. Si es necesario, vuelva a ejecutar el proceso de etiquetado automático.
6. Pulse **Ver resumen** para ver la información sobre la precisión de la etiqueta y el modelo.
7. Cuando esté satisfecho con la precisión del modelo, pulse **Aceptar**.  
Todas las imágenes etiquetadas se guardan como un archivo comprimido etiquetado. Puede encontrar el archivo en el panel **Datos**.

## Creación de modelo

---

Para obtener una buena precisión de modelo, necesita establecer hiperparámetros cuidadosamente basándose en los conjuntos de imágenes y el tipo de modelo seleccionado.

Al crear un modelo, puede elegir los valores de modelo recomendados o elegir personalizarlos. Los valores de modelo recomendados no son adecuados para todos los escenarios. Son sólo una referencia. Es posible que necesite personalizar los valores de modelo.

Al formar un modelo, tiene que elegir los conjuntos de imágenes necesarios. Algunas imágenes se utilizan como conjuntos de datos de formación y otras se utilizan como conjuntos de datos de validación. La proporción depende de la regla de muestreo de validación y formación. De forma predeterminada, el 80% de imágenes se utilizan para la formación y el 20% de las imágenes se utilizan para la validación.

Resulta de ayuda comprender la siguiente terminología de red neuronal:

### Epoch

Un epoch equivale a un ciclo de formación completo de todas las imágenes de formación.

### Tamaño de lote

El número de ejemplos de formación en un paso. Como más grande sea el tamaño de lote, más espacio de memoria se necesita.

### Iteración

El número de iteraciones equivale al número de pasos, cada paso utiliza el número de imágenes que indica el valor de *tamaño\_lote*.

Por ejemplo, si tiene 1000 imágenes de formación y el tamaño de lote es de 50, se requieren 20 iteraciones para completar 1 epoch. Si se define epoch a 10, se necesitan 200 iteraciones para completar la formación.

Si selecciona el tipo de modelo de clasificación, el tamaño de lote se predefine en la definición de red. Para GoogLeNet, el tamaño de lote de formación es de 32 y el tamaño de lote de prueba es de 16. Para AlexNet, el tamaño de lote de formación es de 128 y el tamaño de lote de prueba es de 32. Para LeNet, el tamaño de lote de formación es de 64 y el tamaño de lote de prueba es de 32. Puede ajustar los hiperparámetros basándose en el tamaño de lote, el número total de imágenes y la regla de muestreo.

## Creación de modelos

---

Una vez añadidos los tipos de defectos, el administrador de modelos crea un modelo. El modelo se puede guardar como borrador o aplicar la formación de forma inmediata.

### Acerca de esta tarea

Puede ver los detalles de un modelo en el separador **Todos los modelos**. Los detalles del modelo incluyen las versiones del modelo. Cada conjunto de imágenes diferente crea distintas versiones del modelo.

Los nombres de modelos deben ser únicos.

### Procedimiento

1. Seleccione **Crear nuevo modelo > Crear nuevo modelo**.
2. En el separador **General**, actualice la información.

El tipo de producto se utiliza para correlacionar el modelo. Puede seleccionar el tipo de producto fuera de Maximo PQI SaaS Visual Insights según los estándares de su propia empresa. El tipo de producto es el enlace entre el modelo y la línea de producción. El controlador del sistema perimetral selecciona el modelo en función de la información del tipo de producto que aparece en la imagen.

3. En la pestaña **Tipo de modelo**, actualice la información del tipo de modelo. En **Tipo de modelo**, puede especificar la clasificación o detección de múltiples objetos, o puede elegir cargar un modelo formado existente. Si selecciona clasificación o detección de varios objetos, puede seleccionar los valores de modelo recomendados o elegir personalizarlos.

Si opta por personalizar los valores del modelo, especifique el método de detección, hiperparámetros y reglas de muestreo. Cada hiperparámetro tiene un ámbito de valor válido. Si especifica un valor fuera de ese ámbito, se visualiza un mensaje de error. Debido a las limitaciones de recursos, hay un valor máximo para epoch, iteración y tamaño de lote. El valor máximo de iteración está relacionado con el tipo de pago del arrendatario. Para usuarios de pago, el valor máximo de iteración es 40.000 para FRCNN, 20.000 para YOLO o 5.000 para SSD. Para usuarios de prueba, el valor máximo de iteración es 3.000 para FRCNN, 2.000 para YOLO o 500 para SSD. El tamaño de lote de YOLO y SSD es 32. Para CNN, el valor máximo de epoch es 30,000 para usuarios de pago y 5.000 para usuarios de prueba.

Si carga un archivo comprimido de modelo formado existente, los valores de modelo se leen en el archivo y se visualizan en el cuadro de diálogo. Si el valor de iteración o de epoch y el tamaño de lote exceden el valor máximo, los valores se sustituyen por los valores máximos.

4. En la pestaña **Conjunto de imágenes**, seleccione los conjuntos de imágenes que desee utilizar para la formación del modelo.

5. En el separador **Políticas globales**, establezca la política de repetición de formación, la política de muestreo y los valores de política de inspección manual.
6. En la pestaña **Resumen**, seleccione **Guardar como borrador** para guardar el modelo o **Formar** para formar el modelo.
7. En el panel **Todos los modelos**, vea el modelo. Si la versión del modelo se ha establecido en Borrador, puede pulsar **Ver** para ver los detalles de la instancia del modelo. Puede editar grupos de imágenes antes de formar el modelo.

## Utilización del catálogo de modelos

El catálogo de modelos es una biblioteca de modelos compartidos que se pueden reutilizar para soluciones de industria y que los arrendatarios pueden compartir.

El catálogo de modelos contiene los modelos siguientes.

Dominio	Nombre de modelo	Descripción
Electrónica - Fabricación de hardware	Inspección de defectos de hardware de servidor	Este modelo de detección de objetos detecta que las piezas que faltan para los servidores de IBM. Soporta siete defectos principales que incluyen missingChinLabel, missingPowerLabel, missingIBMLogo, missingDASD, incorrectDASD, missingCover y missingPowerButton.
Automoción - Desgaste de automóvil	Inspección de defectos de desgaste de automóvil	Este modelo de detección de objetos detecta situaciones de desgaste de automóvil. Los fabricantes de automóviles prueban decenas de miles de actividades abiertas y cerradas y detectan cuándo el desgaste está a un nivel inaceptable. Este modelo se centra en detectar rasguños de pintura.
Automoción - Asiento de automóvil	Inspección de defectos de asiento de automóvil	Este modelo de detección de objetos detecta defectos de arrugas en los asientos de automóvil. El resultado de la inspección identifica si hay defecto de arrugas en un asiento de automóvil y localiza el defecto en la imagen.

### Prueba de modelos en el catálogo de modelos

Puede probar modelos del catálogo de modelos cargando su propia imagen y comprobando el resultado de puntuación. Puede cargar varias imágenes y comprobar el resultado de la puntuación individualmente.

#### Procedimiento

1. Seleccione **Crear modelo nuevo > Examinar catálogo de modelos**.
2. En la página **Catálogo de modelos**, seleccione el modelo compartido que desea utilizar.
3. Pulse **Cargar más** para cargar una imagen. El modelo puntúa la imagen y muestra el resultado.



## Creación de modelos utilizando el catálogo de modelos

Puede crear modelos nuevos utilizando modelos compartidos en el catálogo de modelos. Para utilizar un modelo compartido, importe la definición del modelo y utilice sus propios conjuntos de datos para formar su modelo.

### Procedimiento

1. Seleccione **Crear modelo nuevo** > **Examinar catálogo de modelos**.
2. En la página **Modelos compartidos**, seleccione el modelo compartido que desea utilizar.
3. Pulse **Usar modelo**.
4. En el separador **General**, actualice la información. El tipo de producto se utiliza para correlacionar el modelo.
5. En la pestaña **Tipo de modelo**, actualice la información del tipo de modelo y el valor del modelo.
6. En la pestaña **Conjunto de imágenes**, seleccione los conjuntos de imágenes que desee utilizar para la formación del modelo.
7. En la pestaña **Políticas globales**, establezca la política de repetición de formación y la política de inspección manual, si es necesario.
8. En la pestaña **Resumen**, pulse **Guardar como borrador**. El estado del modelo se establece en borrador.

## Modelos de formación

---

Maximo PQI SaaS Visual Insights utiliza técnicas de aprendizaje basadas en Convolutional Neural Network para la formación de modelos. Durante la formación de un modelo, éste aprende las características de las imágenes etiquetadas y crea un modelo ejecutable que incorpora estos conocimientos. Un modelo con formación puede realizar la clasificación o detección de objetos en imágenes desde la línea de productos.

### Acerca de esta tarea

Puede formar un modelo recién creado o un modelo que se ha creado utilizando un modelo existente en el catálogo de modelos. Puede formar modelos cuyo estado sea borrador, anómalo o rechazado. Durante la formación, puede consultar el estado de formación y los archivos de registro. Una vez completada la formación, puede comprobar la curva de formación incluidos los valores de pérdida de formación, pérdida de prueba y precisión de prueba.

### Procedimiento

1. En el panel **Todos los modelos**, seleccione un modelo.
2. Pulse **Ver** para ver los detalles de la instancia del modelo.
3. Opcional: Cambie los grupos de imágenes que utiliza el modelo.
4. Pulse **Formación** para iniciar la formación.  
El estado de formación cambia a En espera, mientras se preparan los datos y se pone en cola el trabajo de formación en el servidor. Cuando se inicia la formación, el estado cambia Formación. Cuando finaliza la formación, el estado cambia a Formado.
5. Una vez completada la formación, revise los resultados de la formación en el panel de control de **Formación**. Si el modelo admite instantáneas, se muestran las instantáneas en el panel de control. Puede comprobar la información de cada instantánea. Puede utilizar una instantánea seleccionando la instantánea y pulsando **Usar**. Puede seleccionar una instantánea intermedia para utilizarla como el archivo de modelo final. También puede ver y descargar los archivos de registro de formación.

## Modelos formados

Después de crear o editar un modelo, administrador de modelos puede formar el modelo. Una vez formado el modelo, su estado cambia a Formado. El administrador de modelos valida el modelo formado y acepta o rechaza el modelo. El administrador de modelos valida el modelo utilizando conjuntos de imágenes de validación.

El administrador de modelos selecciona cargar el modelo formado cuando se crea un nuevo modelo, después del proceso de carga, el estado del modelo cambia a Aceptado.

### Estructura de los archivos de modelo

Maximo PQI SaaS Visual Insights da soporte a la clasificación CNN (Convolutional Neural Network) y a los tipos de modelo de detección de objetos.

#### Modelos de clasificación de CNN para Caffe

El modelo de clasificación CNN debe tener un único archivo comprimido y contener la estructura de directorios y los archivos correctos.

#### El archivo de modelo comprimido

El archivo de modelo comprimido debe contener los siguientes directorios y archivos.

- `model.config` (archivo, necesario)
- `sink.config` (archivo, necesario)
- `parameter.config` (archivo, opcional)
- `cnet1` (directorio). El directorio `cnet1` debe contener los archivos siguientes:
  - `labels.txt` (archivo, necesario)
  - `deploy.prototxt` (archivo, necesario)
  - `mean.binaryproto` (archivo, necesario)
  - `info.json` (archivo, opcional)
  - `snapshot.caffemodel` (archivo, necesario)
  - `solver.prototxt` (archivo, necesario)
  - `train_val.prototxt` (archivo, necesario)

Cada archivo debe tener la estructura y las palabras clave correctas. Los archivos se describen en las siguientes secciones.

#### `model.config`

El texto siguiente es un ejemplo del contenido del archivo `model.config`. Las palabras clave aparecen en negrita.

```
submodel{
 module {
 type:"ChipROIExtractor"
 ref_file:"parameter.config"
 }
 module {
 type:"ClassificationNet"
 net_name:"cnet1"
 }
}
sink{
 type:"SinkFilter"
 ref_file:"sink.config"
}
```

Este archivo debe tener al menos un módulo que tenga el tipo **ClassificationNet** en el submodelo. La palabra clave `ref_file` apunta a los otros archivos de configuración en el archivo comprimido del modelo de

clasificación CNN. La palabra clave `net_name` hace referencia al nombre de carpeta que contiene el modelo CNN. No necesita cambiar la información de `sink` a menos que tenga un nombre diferente para el archivo `sink.config`.

### **sink.config**

El contenido del archivo `sink.config` es el siguiente. No es necesario editar el contenido.

```
keyword:"position"
keyword:"probableTypes"
```

### **labels.txt**

El archivo `labels.txt` contiene los nombres de clase con los que se clasifica este modelo de clasificación. Cada nombre de clase debe estar en una línea distinta, como se muestra en el siguiente ejemplo.

```
defect1
defect2
defect3
```

### **Otros archivos**

Todos los archivos `nombre_archivo.prototxt` son archivos de definición de modelo que son necesarios para dar formación a un modelo CNN.

Los archivos `snapshot.caffemodel` y `mean.binaryproto` son archivos de salida que se crean cuando finaliza la formación de modelos.

### **Modelos de clasificación de CNN para TensorFlow**

El modelo de clasificación de CNN que se ha entrenado con TensorFlow debe ser un solo archivo comprimido y contener la estructura de directorios y los archivos correctos.

### **El archivo de modelo comprimido**

El archivo de modelo comprimido debe contener los siguientes directorios y archivos.

- `model.config` (archivo, necesario)
- `source.config` (archivo, necesario)
- `cnet1.config` (archivo, necesario)
- `cnet1` (directorio). El directorio `cnet1` debe contener los archivos siguientes:
  - `labels.txt` (archivo, necesario)
  - `checkpoint` (archivo, necesario)
  - `model.ckpt-10000.data-00000-of-00001` (archivo, necesario)
  - `model.ckpt-10000.index` (archivo, opcional)
  - `model.ckpt-10000.meta` (archivo, necesario)

Cada archivo debe tener la estructura y las palabras clave correctas. Los archivos se describen en las siguientes secciones.

### **model.config**

El texto siguiente es un ejemplo del contenido del archivo `model.config`. Las palabras clave aparecen en negrita.

```
ipc_buffer_size:80000000
use_ipc_buffer:1
gpu_runnable:1
input_field_mapper {
```

```

entry {
 app_field: "imageFile"
 model_field: "image"
}
}
source {
 type: "utility.ImageFileSource"
 name: "source"
}
sink {
 type: "utility.Sink"
 name: "sink"
 port_num_config {
 input_num: 1
 }
}
}
module {
 type: "tf_classification.TF_ClassificationNet"
 name: "cnet1"
}
}
connection {
 from_module: "source"
 from_port: "0"
 to_module: "cnet1"
 to_port: "0"
}
}
connection {
 from_module: "cnet1"
 from_port: "0"
 to_module: "sink"
 to_port: "0"
}
}
process {
 module_group {
 module: "cnet1" }
 gpu_runnable: 1
}
}

```

Este archivo debe tener al menos un módulo que tenga el tipo **tf\_classification.TF\_ClassificationNet** en **input\_field\_mapper**. El nombre de módulo, por ejemplo cnet1, hace referencia al nombre de carpeta que contiene el modelo CNN. Puede añadir módulos de acuerdo con sus necesidades. No necesita cambiar la información de origen a menos que tenga un nombre diferente para el archivo source.config. Si cambia el nombre del origen, es necesario cambiar el nombre del módulo en la conexión correspondiente.

### source.config

El contenido del archivo source.config es el siguiente. No es necesario editar el contenido.

```

param {
 name: "image"
 value: "color"
}
}
param {
 name: "gray_image"
 value: "gray"
}
}

```

### cnet1.config

El contenido del archivo cnet1.config es el siguiente. El nombre **param** de este archivo de configuración no se puede modificar. El valor es el nombre de variable y el tamaño de imagen que se establece cuando se entrena al modelo.

```

param {
 name: "input_name"
 value: 'input'
}
}
param {
 name: "output_name"
 value: 'InceptionV3/Predictions/Softmax'
}
}
param {
 name: "resize_width"
 value: '299'
}
}

```

```
}
param {
 name: "resize_height"
 value: '299'
}
```

### labels.txt

El archivo `labels.txt` contiene los nombres de clase con los que se clasifica este modelo de clasificación. Cada nombre de clase debe estar en una línea distinta, como se muestra en el siguiente ejemplo.

```
defect1
defect2
defect3
```

### checkpoint

El contenido del archivo `checkpoint` es el siguiente. Este archivo se genera automáticamente cuando se entrena al modelo.

```
model_checkpoint_path: "model.ckpt-10000"
all_model_checkpoint_paths: "model.ckpt-10000"
```

### Modelo de detección de objetos

El modelo de detección de objetos debe tener un único archivo comprimido y contener la estructura de directorios y los archivos correctos. Se soportan los siguientes algoritmos de detección de objetos: Faster Region-based Convolutional Neural Network (Faster R-CNN), You Only Look Once (YOLO) V2 y Single Shot Multibox Detector (SSD).

#### ***Faster R-CNN***

El modelo de detección de objetos Faster R-CNN debe tener un único archivo comprimido y contener la estructura de directorios y los archivos correctos.

#### **Archivo de modelo comprimido**

El archivo de modelo comprimido debe contener los siguientes archivos:

- `labels.txt`
- `faster_rcnn_final.caffemodel`
- `model.config`
- `stage1_fast_rcnn_solver30k40k.pt`
- `stage1_fast_rcnn_train.pt`
- `stage1_rpn_solver60k80k.pt`
- `stage1_rpn_train.pt`
- `stage2_fast_rcnn_solver30k40k.pt`
- `stage2_fast_rcnn_train.pt`
- `stage2_rpn_solver60k80k.pt`
- `stage2_rpn_train.pt`
- `faster_rcnn_test.pt`
- `rpn_test.pt`

Cada archivo debe tener la estructura y las palabras clave correctas. Los archivos se describen en las siguientes secciones.

## labels.txt

El archivo `labels.txt` contiene los nombres de todos los objetos que detecta este modelo de detección de objetos. Cada objeto debe estar en una línea distinta, como se muestra en el siguiente ejemplo.

```
defect1
defect2
defect3
```

## faster\_rcnn\_final.caffemodel

Este archivo contiene la salida después de formar el modelo Faster R-CNN. El nombre de archivo debe coincidir con la definición de **model** en `model.config`. En la actualidad, dos redes son compatibles con Faster R-CNN: ZF y VGG16. En el caso de ZF-net, el nombre debe comenzar por ZF; por ejemplo, `ZF_mobile_final.caffemodel`. En el caso de VGG16-net, el nombre debe comenzar por VGG16; por ejemplo, `VGG16_mobile_final.caffemodel`.

## model.config

Las palabras clave en el ejemplo siguiente se muestran en **negrita**:

```
{
 "modelType": "FRCNN",
 "model": "faster_rcnn_final.caffemodel",
 "solvers":
"stage1_rpn_solver60k80k.pt,stage1_fast_rcnn_solver30k40k.pt,stage2_rpn_solver60k80k.pt,stage2_f
ast_rcnn_solver30k40k.pt",
 "net_file":
"stage1_rpn_train.pt,stage1_fast_rcnn_train.pt,stage2_rpn_train.pt,stage2_fast_rcnn_train.pt",
 "deploy_net": "faster_rcnn_test.pt",
 "parameters": {
 "iteration": "40000,80000,40000,80000",
 "learningRate": 0.001,
 "stepsize": "10000",
 "gamma": "0.1"
 }
}
```

El valor de `modelType` siempre es FRCNN. El valor del modelo es el nombre del archivo de modelo. El valor de `solvers` es la lista de archivos solver que se utilizan durante la formación del modelo. El valor de `net_file` es la lista de archivos de definición de red. El valor de `deploy_net` es el nombre de la definición de red de puntuación. Los valores en los parámetros son todos hiperparámetros del modelo Faster R-CNN. La iteración es un valor de serie que consta de cuatro números separados por comas, presentando el número de iteraciones de cuatro fases en el proceso de formación.

## Archivos \*.pt

Los archivos con la extensión `.pt` son archivos de definición de modelo. El modelo `pascol_voc` que proporciona Faster R-CNN está soportado. El archivo de plantilla puede encontrarse en el directorio de instalación de Faster R-CNN `models/pascal_voc/nombre_red/faster_rcnn_alt_opt/`, donde `nombre_red` es ZF o VGG16.

## YOLO V2

El modelo de detección de objetos YOLO V2 debe tener un único archivo comprimido y contener la estructura de directorios y los archivos correctos.

## Archivo de modelo comprimido

El archivo de modelo comprimido debe contener los siguientes archivos:

- `labels.txt`
- `model.config`
- `yolo_final.weights`
- `Yolo.cfg`

Cada archivo debe tener la estructura y las palabras clave correctas. Los archivos se describen en las siguientes secciones.

### labels.txt

El archivo labels.txt contiene los nombres de todos los objetos que detecta este modelo de detección de objetos. Cada objeto debe estar en una línea distinta, como se muestra en el siguiente ejemplo.

```
defect1
defect2
defect3
```

### model.config

Las palabras clave en el ejemplo siguiente se muestran en negrita:

```
{
 "modelType": "YOLO",
 "modelCfg": "Yolo.cfg",
 "model": "yolo_final.weights",
 "parameters": {
 "iteration": "40000",
 "batchSize": 16,
 "learningRate": 0.001,
 "subBatchSize": 2,
 "steps": "100,15000,25000,35000",
 "scales": "1,10,0.1,0.1"
 }
}
```

El valor de modelType siempre es YOLO. El valor de modelCfg es el nombre del archivo de definición de red de aprendizaje profundo. El valor del modelo es el nombre del archivo de modelo real. Los valores en los parámetros son hiperparámetros del modelo YOLO V2. El valor de batchSize en los parámetros no puede ser superior a 32.

### yolo\_final.weights

Este archivo contiene la salida después de formar el modelo YOLO. El nombre de archivo debe coincidir con la definición de **model** en model.config.

### Yolo.cfg

Este archivo de definición de modelo incluye definiciones de red, hiperparámetros y valores de ancla. Puede encontrar una plantilla de este archivo en el directorio de instalación de YOLO darknet/cfg/. Este archivo debe coincidir con el archivo weights.

### SSD

El modelo de detección de objetos SSD (Single Shot MultiBox Detector) debe tener un único archivo comprimido y contener la estructura de directorios y los archivos correctos.

### Archivo de modelo comprimido

El archivo de modelo comprimido debe contener los siguientes archivos:

- labels.txt
- solver.prototxt
- deploy.prototxt
- model.config
- SSD.caffemodel

Cada archivo debe tener la estructura y las palabras clave correctas. Los archivos se describen en las siguientes secciones.

### labels.txt

El archivo labels.txt contiene los nombres de todos los objetos que detecta este modelo de detección de objetos. Cada objeto debe estar en una línea distinta, como se muestra en el siguiente ejemplo.

```
defect1
defect2
defect3
```

### solver.prototxt

Este archivo contiene todos los hiperparámetros para el modelo SSD.

### deploy.prototxt

Este archivo contiene la definición de red del modelo formado.

### model.config

Las palabras clave en el ejemplo siguiente se muestran en negrita:

```
{
 "modelType": "SSD",
 "parameters": {
 "learningRate": 0.01,
 "iteration": 10000,
 "steps": "6000,8000",
 "batchSize": 16,
 "learningRatePolicy": "multistep",
 "display": 10,
 "snapshot": 1000
 }
}
```

El valor de modelType siempre es SSD. Los valores de los parámetros son hiperparámetros del modelo SSD.

### SSD.caffemodel

Este archivo contiene la salida después de formar el modelo SSD.

## Modelos validados

Una vez formado el modelo, su estado cambia a Formado. El administrador de modelos puede validar la versión del modelo y aceptar o rechazar el modelo formado. El administrador de modelos valida el modelo utilizando conjuntos de imágenes de validación. Una vez validado un modelo, se puede utilizar y desplegar.

Puede validar una versión de modelo que tenga un estado Formado. Una versión de modelo formada o para la que se ha repetido la formación puede desencadenar el proceso de validación. Cuando valida un informe, debe gestionar los conjuntos de imágenes. Cada grupo de imágenes debe tener al menos un conjunto de imágenes de validación para poder validar el modelo. Debe utilizar conjuntos de imágenes diferentes y conjuntos de imágenes de formación para poder validar la versión de modelo. Para empezar el proceso de validación, seleccione **Validar**.

Una vez finalizado el proceso de validación, se genera un informe que muestra la precisión del modelo. Puede crear los tipos de informes siguientes:

#### Informe de modelo de clasificación

En el informe de modelo de clasificación, una imagen tiene un defecto como máximo. La matriz de confusión se utiliza para generar un informe donde cada columna representa un tipo de grupo de imágenes real en los conjuntos de datos de validación. Cada fila representa el tipo de grupo de imágenes previsto. La última fila del gráfico representa los resultados de agregación.



### Informe de modelo de detección de objetos

En el informe de modelo de detección de objetos, una imagen tiene varios defectos. En este informe, se calculan la precisión media y la rellamada para indicar la precisión del modelo.

Para ver el informe, seleccione **Mostrar informe**. En la ventana **Informe de validación**, puede revalidar, rechazar o aceptar y desplegar el modelo.

Si se rechaza la primera versión del modelo, puede cambiar los parámetros de formación y, a continuación, repetir la formación del modelo. Si se rechaza una versión de modelo que no es la primera versión, puede repetir la formación de una nueva versión del modelo utilizando nuevos conjuntos de imágenes.

## Distribución de modelos formados en sistemas perimetrales

---

Una vez que el administrador de modelos haya aceptado la versión del modelo formado, este se distribuye a los sistemas perimetrales para su inspección.

### Procedimiento

1. Seleccione una versión de modelo con un estado Validado. La primera vez que implemente un modelo, haga clic en **Ver informe** para ver el informe de validación.
2. Pulse **Aceptar y publicar** en la página Informe de validación.
3. Se mostrará el cuadro de diálogo Publicar e implementar. Seleccione los sistemas perimetrales que desee implementar en el modelo y haga clic en **Publicar e implementar** para implementar el modelo.
4. Si el modelo se ha implementado en algunos sistemas perimetrales, pulse **Gestionar despliegue** para abrir el cuadro de diálogo de despliegue. Seleccione los sistemas perimetrales que desee desplegar o anular su despliegue y pulse **Desplegar** o **Anular despliegue**.

## Modelos de repetición de formación

---

Cuando repite la formación de un modelo, se crea una nueva versión del modelo. Cuando crea una solicitud de modelo, define la política de repetición de formación. La política de repetición de formación es la condición que desencadena la repetición de formación automática. Si hay algún problema sobre la precisión del modelo actual, el administrador de modelos puede seleccionar manualmente los archivos de imagen para desencadenar el proceso de repetición de formación.

### Procedimiento

1. Seleccione una versión de modelo que tenga un estado Desplegado y seleccione **Repetir formación**.
2. Gestione los conjuntos de imágenes. Cada grupo de imágenes debe tener al menos un conjunto de imágenes de repetición de formación para poder repetir la formación del modelo.  
**Nota:** Para repetir la formación de la versión del modelo, es mejor utilizar conjuntos de imágenes diferentes que contengan más imágenes.
3. Seleccionar **Repetir formación** para comenzar el proceso de repetición de formación.

### Qué hacer a continuación

El administrador de modelos valida si el modelo para el que se ha repetido la formación se ha aceptado. El administrador de modelos puede desplegar la nueva versión y la versión de modelo anterior ya no se despliega.

## Utilización de modelos con un sistema perimetral autónomo

---

Puede utilizar modelos con un sistema perimetral autónomo para puntuar imágenes en la línea de producción, sincronizar datos con el sistema central y limpiar datos cargados. Para utilizar modelos con

un sistema perimetral autónomo, debe publicar los modelos y desplegarlos en el sistema perimetral autónomo.

## Publicación de modelos

Antes de desplegar manualmente un sistema perimetral autónomo, debe publicar el modelo en la aplicación central.

### Acerca de esta tarea

Para publicar un modelo, debe estar ya validado.

### Procedimiento

1. Inicie sesión en la interfaz de usuario de la aplicación central y seleccione una instancia de modelo.
2. Seleccione **Ver informe** para ver el informe de validación.
3. Seleccione **Aceptar y publicar** para publicar la instancia de modelo.

### Resultados

Tras publicar el modelo, el estado del modelo cambiará a Listo para el despliegue.

## Despliegue de una instancia de modelo

Utilice la API para desplegar una instancia de modelo en el sistema perimetral autónomo.

### Procedimiento

1. Obtenga todos los modelos publicados en la aplicación central utilizando el mandato siguiente:  
GET `https://<host de máquina perimetral>:8449/api/getAvailableModels`
2. Despliegue una instancia de modelo en el sistema perimetral autónomo utilizando el mandato siguiente:  
POST `https://<host de máquina perimetral>:8449/api/deployModel`  
Utilice el cuerpo que se devuelve desde la API `getAvailableModels`.

### Qué hacer a continuación

Para obtener más información sobre los detalles de la API, vaya a los temas de servicios del sistema perimetral autónomo.

## Anular el despliegue de modelos

Utilice la API para anular el despliegue de modelos en el sistema perimetral autónomo.

### Procedimiento

Anule el despliegue de modelos utilizando el mandato siguiente:  
POST `https://<host de máquina perimetral>:8449/api/undeployModel`  
Utilice el cuerpo siguiente:

```
{"model_id": "id_modelo", "model_instance_id": "id_instancia_modelo"}
```

### Qué hacer a continuación

Para obtener más información sobre los detalles de la API, vaya a los temas de servicios del sistema perimetral autónomo.

---

## Capítulo 4. Comprobación de los resultados de la inspección

Después de enviar los resultados de la inspección a la aplicación central, el inspector y el supervisor de inspectores pueden ir al separador Comprobación de defectos para ver y filtrar los resultados de la inspección y realizar los cambios pertinentes.

### Imágenes

---

El inspector y el supervisor de inspectores pueden ver las imágenes para comprobar si están clasificadas como defectos existentes o no, y para averiguar si alguien más las ha revisado. La visualización de las imágenes determina lo que deben hacer el inspector o el supervisor cuando comprueban los defectos.

El inspector ve las imágenes que se han comprobado y las que no. Que una imagen no esté comprobada quiere decir que el modelo ha señalado la imagen, pero esta no ha sido comprobada por un inspector. Las imágenes comprobadas han sido señaladas por el modelo y comprobadas por un inspector.

El inspector ve las imágenes que se han confirmado y las que no. Que una imagen no esté confirmada quiere decir que el modelo ha señalado la imagen, pero esta no ha sido confirmada por un inspector. Las imágenes confirmadas han sido señaladas por el modelo y confirmadas por un inspector.

El supervisor de inspectores puede ver los objetos y los objetos desconocidos. Los objetos desconocidos implican que la imagen ha sido marcada por un inspector como defecto desconocido, ya que el inspector no ha clasificado la imagen como defecto existente. Estas imágenes aparecen resaltadas en la lista del supervisor de inspectores.

### Filtrado de defectos

---

El inspector y el supervisor de inspectores pueden aplicar filtros a la visión general de la celda y la lista de defectos.

#### Procedimiento

1. En la ventana **Todas las estaciones de trabajo**, seleccione una estación de trabajo para ver la lista de objetos no confirmados, confirmado y desconocidos.
2. Seleccione el icono de filtro.
3. Escriba un valor para una condición para establecer el filtro y, a continuación, seleccione el icono de añadir. El filtro se aplica a la lista inmediatamente.

### Comprobación de defectos

---

El inspector y el supervisor de inspectores revisan los resultados de la inspección y aplican los cambios necesarios.

#### Acerca de esta tarea

Al seleccionar una imagen, se visualizan el defecto candidato y su confianza correspondiente. Aparece seleccionado el primer defecto de forma predeterminada. El inspector puede seleccionar "desconocido" como tipo de defecto si no conoce el tipo de defecto.

#### Procedimiento

1. Seleccione una imagen para ver los detalles de la imagen y los resultados de la inspección.

2. Seleccione **Editar zoom** para acercar y alejar la imagen o arrastrar la imagen para localizar una posición.
3. Seleccionar **Establecer zoom** para volver al modo de edición. Puede añadir, cambiar el tamaño, mover y ver los detalles de un recuadro de defecto.
4. Seleccione el recuadro de defecto que desea confirmar y vea los detalles del tipo de defecto y el nivel de confianza. Puede cambiar el tipo de defecto o suprimir la posición.
5. Seleccione **Confirmar**.
6. Si la imagen no pertenece a ningún defecto existente, el supervisor de inspectores puede crear un nuevo grupo de imágenes. El nuevo defecto se añade a la lista de candidatos para aquellas imágenes que se encuentren en el mismo modelo.

## Cargar imágenes utilizando el simulador

---

Puede utilizar el simulador para enviar manualmente las imágenes a un sistema perimetral. Este proceso simula el envío de imágenes desde la línea de productos al sistema perimetral. Puede enviar sus propias imágenes o imágenes predefinidas en el servidor.

### Acerca de esta tarea

#### Procedimiento

1. Seleccione **Simulador** en el menú principal.
2. Especifique un tipo de producto y una celda.
3. Para cargar su propia imagen, seleccione **Usar mi propia imagen** y examine para ir a su imagen. O para utilizar imágenes desde el servidor, deseleccione **Usar mi propia imagen** y especifique el número total de imágenes y la frecuencia de envío.
4. Pulse **Iniciar**.
5. Pulse **Ver resultado del análisis**.

---

## Capítulo 5. Panel de control del indicador clave de rendimiento

El supervisor de inspectores utiliza el panel de control de KPI para comprobar la tasa de defectos de nivel de imagen y la tasa de defectos de nivel de ubicación. Las métricas proporcionan información que permiten al usuario pedir al equipo de TI que repitan la formación del modelo o ajusten el procedimiento de fabricación.

El panel de control del KPI se encuentra en el separador **KPI**. Puede seleccionar todas las estaciones de trabajo o una estación de trabajo específica. Esta selección tiene repercusión en el ámbito en el que está trabajando. También puede conmutar entre las vistas en tiempo real e históricas. En la vista en tiempo real, los datos de KPI se renuevan cada 5 segundos. Los valores de KPI incluyen los defectos por unidad y la tasa de defectos. Los defectos por unidad se calculan como un determinado número de defectos dividido por el número de imágenes total. El valor de defectos por unidad representan tasa de apariciones de un tipo de defecto. La tasa de defectos se calcula como el número de imágenes con uno o varios defectos dividido por el número de imágenes total. La tasa de defectos representa la tasa de defectos del producto. Cada línea del gráfico representa el valor de KPI en el intervalo de 5 segundos actual. La vista histórica muestra los datos de KPI históricos. Puede editar la fecha de inicio y la fecha de finalización para determinar el rango de tiempo. Los KPI en la vista histórica incluyen los defectos por unidad y la tasa de defectos.

La vista histórica tiene tres granularidades: por hora, a diario y mensual. En el gráfico por horas, cada punto representa 1 hora. Por ejemplo, un valor de KPI de 24 puntos equivale a 24 horas. En la gráfica diaria, cada punto representa un día. Por ejemplo, un valor de KPI de 30 puntos equivale a 30 días. En la gráfica mensual, cada punto representa un mes. Por ejemplo, un valor de KPI de 12 puntos equivale a 12 meses. Hay tres relaciones entre el rango de tiempo seleccionado y la granularidad. Si hay demasiados puntos en una granularidad para un rango de tiempo seleccionado, esa granularidad está inhabilitada hasta que acorta el rango de tiempo. Si desea renovar el gráfico, puede cambiar el rango de tiempo o pulsar **Renovar**. El servidor calcula periódicamente los datos de KPI históricos, por lo tanto, existe un periodo de tiempo de retardo basado en la configuración. El periodo predeterminado para calcular el KPI histórico es de 1 hora.

De forma predeterminada, sólo se visualizan los cinco primeros tipos de defectos en el gráfico Defectos por unidad. Si desea comprobar otros tipos de defectos, puede seleccionar uno o más tipos de defectos en el gráfico y, a continuación, pulsar **Mostrar tendencia**. Un nuevo gráfico de KPI muestra los tipos de defectos seleccionados. Si el valor de KPI de un tipo de defecto es 0, no se puede seleccionar para mostrar una tendencia.



## Capítulo 6. Integración con Prescriptive Quality

Puede exportar los datos de tasa de defectos históricos desde Maximo PQI SaaS Visual Insights en formato CSV e importar los datos a Prescriptive Quality.

Puede exportar los datos desde el panel de control KPI cuando visualiza las tasas de defectos históricos. Para ver estos datos, seleccione **Histórico** en el panel de control KPI y, a continuación, seleccione **Tasa de defectos**. Cuando selecciona **Exportar**, los datos se exportan en función de la celda seleccionada, el rango de tiempo y la granularidad temporal.

El archivo CSV contiene las columnas siguientes.

Columna	Descripción
ATTRIBUTE_NAME	Este valor siempre es VI_DEFECT_IMAGE_RATE.
DATE	La fecha con el formato aaaa-MM-dd cuando el intervalo es diario o mensual, y el formato aaaa-MM-dd HH:mm:ss cuando el intervalo es por horas.
CELL_NAME	El nombre de la celda donde se ha producido el defecto. Es una de las dimensiones de Prescriptive Quality.
PRODUCED_QTY	El número total de imágenes de la celda seleccionada y el rango de tiempo.
TESTED_QTY	El número total de imágenes probadas.
FAILED_QTY	El número total de imágenes que muestran defectos.





---

# Capítulo 7. Interfaz de programas de aplicación

La interfaz de programas de aplicación le permite realizar acciones, tales como gestionar grupos de datos, archivos de datos, modelos, resultados de la inspección y sistemas perimetrales.

## Flujos de trabajo de API

---

Muchas de las llamadas de API de Maximo PQI SaaS Visual Insights tienen requisitos previos o correlaciones con otras llamadas de API. Asimismo, solo determinados roles de usuario tienen acceso a cada llama a API. Por lo tanto, es importante comprender el flujo de trabajo de API para cada rol.

### Administrador de modelos

Generalmente, el administrador de modelos utiliza las llamadas de API en la siguiente secuencia de acciones:

- Crear un grupo de datos. Consulte [Crear un grupo de datos](#) en Servicios de grupos de datos.
- Cargar un archivo de datos en un grupo de datos. Consulte [Cargar archivos de datos en un grupo de datos](#) en Servicios de archivos de datos.
- Crear un modelo. Consulte [Crear un modelo](#) en Servicios de modelos.
- Crear una instancia de modelo. Consulte [Crear una instancia de modelo](#) en Servicios de instancias de modelos.
- Formar el modelo. Consulte la instancia de [Formar modelo](#). El proceso de formación puede tardar de varios minutos a horas, en función del número de imágenes y los parámetros de formación. Para comprobar el estado de la instancia de modelo, consulte [Obtener una instancia de modelo](#).
- Una vez completada la formación en línea, el administrador de modelos puede validar el modelo. Consulte [Validar instancia de modelo](#).
- Cree un sistema perimetral. Consulte [Crear sistema perimetral](#) en Servicios de sistemas perimetrales.
- Despliegue el modelo en el sistema perimetral o rechace el modelo. Consulte [Desplegar instancia de modelo](#) o [Rechazar instancia de modelo](#) en los servicios de acciones de instancias de modelos.
- Volver a formar o anular el despliegue del modelo desplegado. Consulte [Volver a formar instancia de modelo](#) o [Anular despliegue de instancia de modelo](#) en Servicios de acciones de instancias de modelos.

### Inspector

Generalmente, el inspector utiliza las llamadas de API en la siguiente secuencia de acciones:

- Puntuar una imagen. Consulte [Puntuar una imagen](#) en Servicios de puntuación.
- Obtener la visión general de celdas del resultado de inspección. Consulte [Obtener visión general de celdas del resultado de inspección](#) en Servicios de resultados de inspección.
- Obtener la lista de resultados de inspección. Consulte [Obtener lista de resultados de inspección](#) en Servicios de resultados de inspección.
- Confirmar resultados de inspección. Consulte [Confirmar resultados de inspección](#) en Servicios de resultados de inspección.

### Supervisor

Generalmente, el supervisor utiliza las llamadas de API en la siguiente secuencia de acciones:

- Puntuar una imagen. Consulte [Puntuar imagen](#) en Servicios de puntuación.
- Obtener la visión general de celdas del resultado de inspección. Consulte [Obtener visión general de celdas del resultado de inspección](#) en Servicios de resultados de inspección.

- Obtener la lista de resultados de inspección. Consulte Obtener lista de resultados de inspección en Servicios de resultados de inspección.
- Confirmar resultados de inspección. Consulte Confirmar resultados de inspección en Servicios de resultados de inspección.
- Obtener archivo de tasa de imágenes de defectos. Consulte Obtener archivo de tasa de imágenes de defectos en el Servicio de integración de QEWS.

### **Administrador del sistema perimetral autónomo**

El administrador del sistema perimetral autónomo tiene credenciales para la máquina perimetral autónoma. Generalmente, el administrador del sistema perimetral autónomo utiliza las llamadas de API en la siguiente secuencia de acciones:

- Obtener modelos disponibles. Consulte Obtener modelos disponibles en Servicios del sistema perimetral autónomo.
- Desplegar un modelo. Consulte Desplegar un modelo en Servicios del sistema perimetral autónomo.
- Desplegar todos los modelos. Consulte Desplegar todos los modelos en Servicios del sistema perimetral autónomo.
- Cargar y puntuar una imagen en el sistema perimetral. Consulte Cargar y puntuar imagen en el sistema perimetral en Servicios del sistema perimetral autónomo. La línea de producción o los servicios externos utilizan normalmente esta API.
- Sincronizar el resultado de inspección del sistema perimetral con la aplicación central. Consulte Sincronizar el resultado de inspección del sistema perimetral con la aplicación central en Servicios del sistema perimetral autónomo.
- Limpiar el resultado de inspección que ha sincronizado con la aplicación central. Consulte Limpiar el resultado de inspección que ha sincronizado con la aplicación central en Servicios del sistema perimetral autónomo.
- Anular el despliegue de un modelo. Consulte Anular el despliegue de un modelo en Servicios del sistema perimetral autónomo.

## **Preparación para el uso de llamadas de API**

---

Antes de utilizar cualquier llamada de API, asegúrese de que el usuario y la solución estén preparados y se ha obtenido una clave de API.

Realice las tareas siguientes para la preparación del usuario y la solución:

- Asegúrese de que el usuario haya iniciado sesión tenga acceso a las llamadas de API deseadas.
- En el caso de las llamadas de API que requieren un ID de solución, especifique *v1* como la solución.
- Asegúrese de que el usuario tenga la clave de API que ha creado IBM para el usuario.

## **Respuestas del servicio**

---

Cuando se completa una llamada de API, esta proporciona una respuesta de éxito o de error.

Los servicios de API proporcionan las respuestas siguientes.

- Respuestas satisfactorias:
  - Código de estado 201 para servicios de creación
  - Código de estado 200 para otros servicios
- Respuestas de error de base de datos:
  - Código de estado 500

- Mensajes de error, por ejemplo, "Faltan parámetros o su formato no es válido durante la creación de un nuevo grupo de datos".
- Errores de autorización
  - Código de estado 401
  - Mensajes de error, por ejemplo, "Se debe establecer la autorización o APIKEY en la cabecera".

## Servicios de grupos de datos

---

Los servicios de grupos de datos le permiten realizar tareas como obtener todos los grupos de datos, obtener un grupo de datos y crear un grupo de datos nuevo.

### Obtener todos los grupos de datos

Obtiene todos los grupos de datos existentes. El administrador de modelos tiene autorización para utilizar esta API.

#### URL

/ibm/iotm/service/dataGroup

#### Método

El GET del tipo de solicitud

#### Parámetros de URL

user: *Serie*. Se utiliza para identificar al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

#### Cabeceras

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

#### Parámetros de datos

Ninguno

#### Cuerpo de ejemplo

Ninguno

#### Respuesta satisfactoria

```
200
[
 {
 "id": "892d05cd-84e5-4bae-b57f-edbbc3f8c598",
 "groupName": "Mist",
 "createdBy": "auto_mm",
 "description": "test_description",
 "dataFormat": "jpg",
 "parameters": {
 "isHybrid": "false",
 "isDefect": "true"
 },
 "updatedAt": "2017-11-29T16:47:54.039+08:00"
 },
 {
 "id": "9bebc890-0237-4a3b-8908-9f5f53fdb3a2",
 "groupName": "NoDefect",
 "createdBy": "auto_mm",
 "description": "test_description",
 "dataFormat": "jpg",
 "parameters": {
 "isHybrid": "false",
 "isDefect": "true"
 },
 "updatedAt": "2017-11-29T16:47:54.463+08:00"
 }
]
```

```

{
 "id": "b989aae2-f1ae-4a77-93bb-4e9e4f18264d",
 "groupName": "Defect",
 "createdBy": "auto_mm",
 "description": "test_description",
 "dataFormat": "jpg",
 "parameters": {
 "isHybrid": "false",
 "isDefect": "true"
 },
 "updatedAt": "2017-11-29T16:47:53.536+08:00"
}
]

```

### Elementos de respuesta

**id:** *Serie*. El ID del grupo de datos.

**groupName:** *Serie*. El nombre del grupo de datos.

**createdBy:** *Serie*. El propietario del grupo de datos.

**description:** *Serie*. La descripción del grupo de datos.

**dataFormat:** *Serie*. El formato del grupo de datos.

**isHybrid:** *Booleano*. True indica la detección de objetos. False indica la clasificación.

**isDefect:** *Booleano*. True indica que es un defecto. False indica que no es un defecto.

**updatedAt:** *Hora*. La última vez que se ha actualizado el grupo de datos.

### Llamada de ejemplo

```

curl -k -H
"APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aa
b2c77d4f0579e533dd34ggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5"
"https://9.112.229.91:9447/ibm/iotm/service/dataGroup?user=auto_mm&solution=vi"

```

### Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Obtener todos los grupos de datos con archivos de datos

Obtiene todos los grupos de datos existentes que tienen archivos de datos para cada grupo. El administrador de modelos tiene autorización para utilizar esta API.

### URL

/ibm/iotm/service/dataGroup

### Método

El GET del tipo de solicitud

### Parámetros de URL

**user:** *Serie*. Se utiliza para identificar al usuario. Obligatorio.

**solution:** *Serie*. Se utiliza para identificar la solución. Obligatorio.

**dataFiles:** *[]*. Se utiliza para identificar y obtener archivos de datos. Obligatorio.

**tenant:** *Serie*. Se utiliza para identificar al arrendatario. Opcional.

### Cabeceras

**APIKEY:** *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

### Parámetros de datos

Ninguno

## Cuerpo de ejemplo

Ninguno

## Respuesta satisfactoria

```
200
[
 {
 "id": "3e420dbc-88fb-463a-83ee-a317688b02e0",
 "groupName": "test",
 "createdBy": "auto_mm",
 "description": "test_description",
 "dataFormat": "jpg",
 "parameters": {
 "isHybrid": "false",
 "isDefect": "true"
 },
 "dataFiles": [
 {
 "id": "3e420dbc-88fb-463a-83ee-a317688b02e0_539bf24a-afc9-4f43-9ca9-b7ec0648e781",
 "groupId": "3e420dbc-88fb-463a-83ee-a317688b02e0",
 "createdBy": "auto_mm",
 "fileName": "MisT.zip",
 "fileCount": "3",
 "updatedAt": "1511949371332",
 "fileUrl": "/user/AUTO/VI/datagroup/3e420dbc-88fb-463a-83ee-a317688b02e0/1511949371332.zip"
 },
 {
 "id": "3e420dbc-88fb-463a-83ee-a317688b02e0_f76b31fb-f803-4d0e-995e-b597bf3c761d",
 "groupId": "3e420dbc-88fb-463a-83ee-a317688b02e0",
 "createdBy": "auto_mm",
 "fileName": "Defect.zip",
 "fileCount": "3",
 "updatedAt": "1511949368998",
 "fileUrl": "/user/AUTO/VI/datagroup/3e420dbc-88fb-463a-83ee-a317688b02e0/1511949368998.zip"
 }
],
 "updatedAt": "2017-11-29T16:59:43.873+08:00"
 }
]
```

## Elementos de respuesta

**id:** *Serie*. El ID del grupo de datos.

**groupName:** *Serie*. El nombre del grupo de datos.

**createdBy:** *Serie*. El propietario del grupo de datos.

**description:** *Serie*. La descripción de los datos.

**dataFormat:** *Serie*. El formato del grupo de datos.

**isHybrid:** *Booleano*. True indica la detección de objetos. False indica la clasificación.

**isDefect:** *Booleano*. True indica que es un defecto. False indica que no es un defecto.

**dataFiles:** *Objeto JSON*. Contiene información sobre los archivos de datos.

**id:** *Serie*. El ID del archivo de datos.

**groupId:** *Serie*. El ID del grupo de datos.

**createdBy:** *Serie*. El propietario del archivo de datos.

**fileName:** *Serie*. El nombre del archivo.

**fileCount:** *Serie*. El recuento de archivos.

**updatedAt:** <Time>. La última vez que se ha actualizado el archivo de datos.

**fileUrl:** *Serie*. La vía de acceso del archivo que se almacena en el servidor.

updatedAt: *Hora*. La última vez que se ha actualizado el grupo de datos.

### Llamada de ejemplo

```
curl -k -H
"APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77d
4f0579e533dd34ggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5" "https://9.112.229.91:
9447/ibm/iotm/service/dataGroup?user=auto_mm&solution=vi &dataFiles=[]"
```

### Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Obtener un grupo de datos

Obtiene un grupo de datos. El administrador de modelos tiene autorización para utilizar esta API.

### URL

/ibm/iotm/service/dataGroup/*groupId*

### Método

El GET del tipo de solicitud

### Parámetros de URL

groupId: *Serie*. El ID del grupo de datos.

user: *Serie*. Se utiliza para identificar al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

### Cabeceras

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

### Parámetros de datos

Ninguno

### Cuerpo de ejemplo

Ninguno

### Respuesta satisfactoria

```
200
[
 {
 "tags": "",
 "id": "3e420dbc-88fb-463a-83ee-a317688b02e0",
 "groupName": "test",
 "createdBy": "auto_mm",
 "description": "test_description",
 "dataFormat": "jpg",
 "parameters": {
 "isHybrid": "false",
 "isDefect": "true"
 },
 "updatedAt": "2017-11-29T16:59:43.873+08:00"
 }
]
```

### Elementos de respuesta

tags: *Serie*. La etiqueta del grupo de datos.

id: *Serie*. El ID del grupo de datos.

groupName: *Serie*. El nombre del grupo de datos.

createdBy: *Serie*. El propietario del grupo de datos.

`description`: *Serie*. La descripción del grupo de datos.  
`dataFormat`: *Serie*. El formato del grupo de datos.  
`isHybrid`: *Booleano*. True indica la detección de objetos. False indica la clasificación.  
`isDefect`: *Booleano*. True indica que es un defecto. False indica que no es un defecto.  
`updatedAtTime`: *Hora*. La última vez que se ha actualizado el grupo de datos.

### Llamada de ejemplo

```
curl -k -H
"APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77
d4f0579e533dd34gg959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5"
"https://9.112.229.91:9447/ibm/iotm/service/dataGroup/3e420dbc-88fb-463a-8
3ee-a317688b02e0?user=auto_mm&solution=vi"
```

### Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Crear un grupo de datos

Crea un grupo de datos. El administrador de modelos tiene autorización para utilizar esta API.

### URL

`/ibm/iotm/service/dataGroup`

### Método

El POST del tipo de solicitud

### Parámetros de URL

`user`: *Serie*. Se utiliza para identificar al usuario. Obligatorio.  
`solution`: *Serie*. Se utiliza para identificar la solución. Obligatorio.  
`tenant`: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

### Cabeceras

`Content-type`: `application/json`.  
`APIKEY`: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio

### Parámetros de datos

`groupName`: *Serie*. El nombre del grupo de datos.  
`description`: *Serie*. La descripción del grupo de datos.  
`dataFormat`: *Serie*. El formato del grupo de datos.  
`isHybrid`: *Booleano*. True indica la detección de objetos. False indica la clasificación.  
`isDefect`: *Booleano*. True indica que es un defecto. False indica que no es un defecto.

### Cuerpo de ejemplo

```
[
 {
 "groupName": "test",
 "description": "test_description",
 "parameters": {
 "isHybrid": "false",
 "isDefect": "true"
 }
 }
]
```

## Respuesta satisfactoria

```
201
[
 {
 "id": "7dac9493-4d8c-4472-a4ca-ae450ccaea5d",
 "groupName": "test",
 "tags": null,
 "dataFormat": "jpg",
 "createdBy": "auto_mm",
 "updatedAt": "2017-11-30T09:59:09.597+08:00",
 "parameters": {
 "isHybrid": "false",
 "isDefect": "true"
 },
 "description": "test_description"
 }
]
```

## Elementos de respuesta

**id:** *Serie*. El ID del grupo de datos.

**groupName:** *Serie*. El nombre del grupo de datos.

**tags:** *Serie*. La etiqueta del grupo de datos.

**createdBy:** *Serie*. El propietario del grupo de datos.

**description:** *Serie*. La descripción del grupo de datos.

**dataFormat:** *Serie*. El formato del grupo de datos.

**isHybrid:** *Booleano*. True indica la detección de objetos. False indica la clasificación.

**isDefect:** *Booleano*. True indica que es un defecto. False indica que no es un defecto.

**updatedAt:** *Hora*. La última vez que se ha actualizado el grupo de datos.

## Llamada de ejemplo

```
curl -k -H
"APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77d4f05
79e533dd34ggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5" -H "Content-Type:
application/json;charset=UTF-8" "https://9.112.229.91:9447/ibm/iotm/service/
dataGroup?user=auto_mm&solution=vi" --data '{"groupName":"test","dataFormat":
"jpg","description":"test_description","parameters":{"isHybrid":"false","isDefect"
:"true"}}'
```

## Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Actualizar un grupo de datos

Actualiza un grupo de datos. El administrador de modelos tiene autorización para utilizar esta API.

### URL

`/ibm/iotm/service/dataGroup/groupId`

### Método

El PUT del tipo de solicitud

### Parámetros de URL

**groupId:** *Serie*. El ID del grupo de datos. Obligatorio

**user:** *Serie*. Se utiliza para identificar al usuario. Obligatorio.

**solution:** *Serie*. Se utiliza para identificar la solución. Obligatorio.

**tenant:** *Serie*. Se utiliza para identificar al arrendatario. Opcional.



### Cabeceras

Content-type: application/json.

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio

### Parámetros de datos

description: *Serie*. La descripción del grupo de datos.

isHybrid: *Booleano*. True indica la detección de objetos. False indica la clasificación.

isDefect: *Booleano*. True indica que es un defecto. False indica que no es un defecto.

### Cuerpo de ejemplo

```
200
[
 {
 "description": "test",
 "parameters": {
 "isDefect": "true",
 "isHybrid": "true" }
 }
]
```

### Respuesta satisfactoria

```
[{}]
```

### Elementos de respuesta

Ninguno

### Llamada de ejemplo

```
curl -k -H
"APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77d4f0579e533d
d34ggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5" -H "Content-Type:
application/json;charset=UTF-8" -X PUT "https://9.112.229.91:9447/ibm/iotm/service
/dataGroup/7dac9493-4d8c-4472-a4ca-ae450cca5d?user=auto_mm&solution=vi" --data
' [{"description": "test", "parameters": {"isHybrid": "true", "isDefect": "true"}}]'
```

### Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Suprimir grupos de datos

Suprime grupos de datos existentes. El administrador de modelos tiene autorización para utilizar esta API.

### URL

/ibm/iotm/service/dataGroup/*groupId*

### Método

El DELETE del tipo de solicitud

### Parámetros de URL

groupId: *Serie*. El ID del grupo de datos. Obligatorio

user: *Serie*. Se utiliza para identificar al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

### Cabeceras

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

### Parámetros de datos

Ninguno

### Cuerpo de ejemplo

```
[{}]
```

### Respuesta satisfactoria

```
200
[{}]
```

### Elementos de respuesta

Ninguno

### Llamada de ejemplo

```
curl -k -X DELETE -H "APIKEY:8b796582
25de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77d4f05
79e533dd34ggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5"
"https://9.112.229.91:9447/ibm/iotm/service/dataGroup/
ed568556-1462-45e9-9319-4bfa8186d2cd?user=auto_mm&solution=
vi" -d '[{}]
```

### Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Servicios de archivos de datos

---

Los servicios de archivos de datos le permiten realizar tareas como obtener todos los archivos de datos que pertenecen a un grupo de datos, obtener un archivo de datos y descargar un archivo binario.

### Obtener todos los archivos de datos que pertenecen a un grupo de datos

Obtiene todos los archivos de datos que pertenecen a un grupo de datos. El administrador de modelos tiene autorización para utilizar esta API.

#### URL

/ibm/iotm/service/dataFile

#### Método

El GET del tipo de solicitud

#### Parámetros de URL

groupId: *Serie*. El ID del grupo de datos. Obligatorio.

user: *Serie*. Se utiliza para identificar al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

### Cabeceras

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

### Parámetros de datos

Ninguno

## Cuerpo de ejemplo

Ninguno

## Respuesta satisfactoria

```
200
[
 {
 "id": "06c870e5-1701-419a-8a84-0f40ee82fe33_a7f1772f-07a0-4d33-9efe-f2e45a2dc904",
 "groupId": "06c870e5-1701-419a-8a84-0f40ee82fe33",
 "createdBy": "modelmanager1",
 "fileName": "NG.zip",
 "updatedAt": "1502185276752",
 "fileCount": "52",
 "fileUrl": "/user/T1/VIQ/datagroup/06c870e5-1701-419a-8a84-0f40ee82fe33/1502185276752.zip"
 },
 {
 "id": "06c870e5-1701-419a-8a84-0f40ee82fe33_b3af214c-b603-43f4-944d-40277b9a6fd9",
 "groupId": "06c870e5-1701-419a-8a84-0f40ee82fe33",
 "createdBy": "modelmanager1",
 "fileName": "IoT4M_March2017_VI_Source.zip",
 "updatedAt": "1502184389077",
 "fileCount": "52",
 "fileUrl": "/user/T1/VIQ/datagroup/06c870e5-1701-419a-8a84-0f40ee82fe33/1502184389077.zip"
 },
 {
 "id": "06c870e5-1701-419a-8a84-0f40ee82fe33_c901e409-d9db-4f02-8d9c-fa6002929701",
 "groupId": "06c870e5-1701-419a-8a84-0f40ee82fe33",
 "createdBy": "modelmanager1",
 "fileName": "NG.zip",
 "updatedAt": "1502185248433",
 "fileCount": "52",
 "fileUrl": "/user/T1/VIQ/datagroup/06c870e5-1701-419a-8a84-0f40ee82fe33/1502185248433.zip"
 }
]
```

## Elementos de respuesta

id: *Serie*. El ID del archivo de datos.

groupId: *Serie*. El ID del grupo de datos.

createdBy: *Serie*. El propietario del archivo de datos.

fileName: *Serie*. El nombre del archivo.

fileCount: *Serie*. El recuento de archivos.

updatedAt: *Hora*. La última vez que se ha actualizado el archivo de datos.

fileUrl: *La vía de acceso del archivo que se almacena en el servidor.*

## Llamada de ejemplo

```
curl -k -H
"APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77d4f0579e533dd34ggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5"
"https://9.112.229.91:9447/ibm/iotm/service/dataFile?user=auto_mm&solution=vi&groupId=ed568556-1462-45e9-9319-4bfa8186d2cd"
```

## Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Obtener un archivo de datos

Obtiene un archivo de datos. El administrador de modelos tiene autorización para utilizar esta API.

## URL

/ibm/iotm/service/dataFile

## Método

El GET del tipo de solicitud

## Parámetros de URL

**fileId:** *Serie*. El ID del archivo de datos. Obligatorio.  
**user:** *Serie*. Se utiliza para identificar al usuario. Obligatorio.  
**solution:** *Serie*. Se utiliza para identificar la solución. Obligatorio.  
**tenant:** *Serie*. Se utiliza para identificar al arrendatario. Opcional.

## Cabeceras

**APIKEY:** *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

## Parámetros de datos

Ninguno

## Cuerpo de ejemplo

Ninguno

## Respuesta satisfactoria

```
200
[
 {
 "id": "06c870e5-1701-419a-8a84-0f40ee82fe33_c901e409-d9db-4f02-8d9c-fa6002929701",
 "groupId": "c901e409-d9db-4f02-8d9c-fa6002929701",
 "createdBy": "modelmanager1",
 "fileName": "NG.zip",
 "updatedAt": "1502185248433",
 "fileCount": "52",
 "fileUrl": "/user/T1/VIQ/datagroup/06c870e5-1701-419a-8a84-0f40ee82fe33/1502185248433.zip"
 }
]
```

## Elementos de respuesta

**id:** *Serie*. El ID del archivo de datos.  
**groupId:** *Serie*. El ID del grupo de datos.  
**createdBy:** *Serie*. El propietario del archivo de datos.  
**fileName:** *Serie*. El nombre del archivo.  
**updatedAt:** *Hora*. La última vez que se ha actualizado el archivo de datos.  
**fileUrl:** *La vía de acceso del archivo que se almacena en el servidor.*

## Llamada de ejemplo

```
curl -k -H
"APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77d4f0579e533dd34ggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5" "https://9.112.229.91:9447/ibm/iotm/service/dataFile/06c870e5-1701-419a-8a84-0f40ee82fe33_c901e409-d9db-4f02-8d9c-fa6002929701?user=auto_mm&solution=vi"
```

## Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Descargar el contenido binario del archivo de datos

Descarga el contenido binario de un archivo de datos. El administrador de modelos y el experto en datos tienen autorización para utilizar esta API.

### URL

/ibm/iotm/service/dataFileBinary

### Método

El GET del tipo de solicitud

### Parámetros de URL

fileId: *Serie*. El ID del archivo de datos. Obligatorio.

user: *Serie*. Se utiliza para identificar al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

### Cabeceras

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

### Parámetros de datos

Ninguno

### Cuerpo de ejemplo

Ninguno

### Respuesta satisfactoria

El contenido binario del archivo de datos.

### Elementos de respuesta

Ninguno

### Llamada de ejemplo

```
curl -k -H
"APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77d4f05
79e533dd34ggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5" "https://9.
112.229.91:9447/ibm/iotm/service/ dataFileBinary ?fileId=06c870e5-1701-419a
-8a84-0f40ee82fe33_c901e409-d9db-4f02-8d9c-fa6002929701&user=auto_mm&solution
=vi"
```

### Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Cargar archivos de datos en un grupo de datos

Carga uno o varios archivos de datos en un grupo de datos. El administrador de modelos tiene autorización para utilizar esta API.

### URL

/ibm/iotm/service/dataFileBinary

### Método

El POST del tipo de solicitud

### Parámetros de URL

groupId: *Serie*. El ID del grupo de datos. Obligatorio.

user: *Serie*. Se utiliza para identificar al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

### Cabeceras

content-type: multipart/form-data.

Content-Disposition: form-data; name="files[]"; filename=*el nombre de su archivo*.

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

### Parámetros de datos

Ninguno

### Cuerpo de ejemplo

El contenido binario del archivo de datos.

### Respuesta satisfactoria

```
200
{"result":{"Defect.zip":{"id":"9f7aa9d6-24d6-4611-8d4d-79a52d2bab02_76e8b2cb-057a-4904-8fad-bbca93abd80b","count":"3","name":"Defect.zip","errorMsg":"","updatedAtTime":"1512021663498","url":"/user/AUTO/VI/datagroup/9f7aa9d6-24d6-4611-8d4d-79a52d2bab02/1512021663498.zip"},"error_message":{"}}
```

### Elementos de respuesta

id: *Serie*. El ID del archivo de datos.

name: *Serie*. El nombre del archivo.

count: *Serie*. El recuento de imágenes.

updatedAtTime: *Hora*. La última vez que se ha actualizado el archivo de datos.

fileUrl: . La vía de acceso del archivo que se almacena en el servidor.

error\_message: . El mensaje de error de una respuesta de carga.

### Llamada de ejemplo

```
curl -k -X POST -H
"APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77d4f0579e53
3dd34ggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5"
-H "Content-Type:multipart/form-data" -H "Content-Disposition:
form-data; name=\"files[]\"; filename=\"Defect.zip\" \"https://9.112.229.91:9447
/ibm/iotm/service/dataFileBinary?user=auto_mm&solution=vi&groupId=9f7aa9d6-24d6-
4611-8d4d-79a52d2bab02\" --connect-timeout 600 -F file=@C:\\code\\Automation_98\\
API\\vi\\VI_API\\file\\Defect.zip
```

### Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Suprimir un archivo de datos

Suprime un archivo de datos. El administrador de modelos tiene autorización para utilizar esta API.

### URL

/ibm/iotm/service/dataFile/*fileId*

### Método

El DELETE del tipo de solicitud

### Parámetros de URL

fileId: *Serie*. El ID del archivo de datos. Obligatorio.

user: *Serie*. Se utiliza para identificar al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

#### Cabeceras

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

#### Parámetros de datos

Ninguno

#### Cuerpo de ejemplo

```
[{}]
```

#### Respuesta satisfactoria

```
200
[{}]
```

#### Elementos de respuesta

Ninguno

#### Llamada de ejemplo

```
curl -k -X DELETE -H "APIKEY:8b79658
225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77d4f057
9e533dd34ggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5"
"https://9.112.229.91:9447/ibm/iotm/service/dataFile/
9f7aa9d6-24d6-4611-8d4d-79a52d2bab02_e2cc3d4a-bae9-
4600-9dff-d80fa3ee8832?user=auto_mm&solution=vi" -d '[{}]
```

#### Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Servicios de grupos de datos sin etiqueta

Los servicios de grupos de datos sin etiquetar le permiten realizar tareas, como cargar archivos de imagen comprimidos sin etiqueta, crear grupos de datos sin etiqueta y obtener grupos de datos sin etiqueta.

### Cargar un archivo de imagen comprimido no etiquetado

Carga un archivo comprimido no etiquetado en el servidor. El administrador de modelos tiene autorización para utilizar esta API.

#### URL

/ibm/iotm/service/unLabeledImageZipServlet

#### Método

El POST del tipo de solicitud

#### Parámetros de URL

groupType: *Serie*. El tipo de grupo de datos. El valor debe ser *classification* u *objectdetection*. Obligatorio.

user: *Serie*. Se utiliza para identificar al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

#### Cabeceras

content-type: multipart/form-data.

Content-Disposition: form-data; name="files[]"; filename=*el nombre de su archivo*.

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

### Parámetros de datos

Ninguno

### Cuerpo de ejemplo

El contenido binario del archivo de datos.

### Respuesta satisfactoria

```
200 {"message":{},"result":{"imageZip":{"imageZipId":"b76b33b0-26d6-4f85-83ea-e520edbe5ff8",
"labeledImageCount":0,"labelCount":0,"imageZipName":"images.zip","count":85,"updatedAt":
"2019-03-19 00:18:53.304","url":"\\user\\TENANT\\VI\\unlabeledgroup\\b76b33b0-26d6-4f85-
83ea-e520edbe5ff8.zip"}}}
```

### Elementos de respuesta

imageZip: *objeto JSON*. La información del archivo de imagen comprimido cargado. La información incluye imageZipId, LabeledImageCount, labelCount, imageZipName, count, updatedAt, y URL.

imageZipId: *Serie*. El ID que se ha asignado al archivo de imagen comprimido cargado.

LabeledImageCount: *Entero*. El número de imágenes que se han etiquetado previamente en el archivo de imagen comprimido.

labelCount: *Entero*. El número de etiquetas que se asignan a las imágenes en el archivo de imagen comprimido.

imageZipName: *Serie*. El nombre del archivo de imagen comprimido.

count: *Entero*. El número total de imágenes en el archivo de imagen comprimido.

updatedAt: *Hora*. La hora a la que se ha cargado el archivo de imagen comprimido.

url: *Serie*. La vía de acceso del archivo que se almacena en el servidor.

mensaje\_error: *objeto JSON*. El mensaje de error de una respuesta de carga.

### Llamada de ejemplo

```
curl -k -X POST -H"APIKEY:apikey"
-H "Content-Type:multipart/form-data" -H "Content-Disposition: form-data;
name=\"files[]\"; filename=\"image.zip\"" "https://iotm.predictivesolutionsapps
.ibmcloud.com/ibm/iotm/service/unLabeledImageZipServlet?user=auto_mm&solution=vi&
groupType=classification" --connect-timeout 600
-F file=@C:\\code\\Automation_98\\API\\vi\\VI_API\\file\\image.zip
```

### Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Crear un grupo de datos sin etiquetar

Crea un grupo de datos sin etiqueta con el archivo de imagen comprimido que se ha cargado. El administrador de modelos tiene autorización para utilizar esta API.

### URL

/ibm/iotm/service/unLabeledGroup

### Método

El POST del tipo de solicitud



## Parámetros de URL

**user:** *Serie*. Se utiliza para identificar al usuario. Obligatorio.

**solution:** *Serie*. Se utiliza para identificar la solución. Obligatorio.

**tenant:** *Serie*. Se utiliza para identificar al arrendatario. Opcional.

## Cabeceras

**content-type:** multipart/form-data.

**APIKEY:** *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

## Parámetros de datos

**groupName:** *Serie*. El nombre del grupo de datos.

**groupType:** *Serie*. El tipo de grupo de datos. El valor debe ser *classification* u *objectdetection*.

**imageZip:** *JSON Array*. La información del archivo de imagen comprimido cargado. Puede obtener la información de la respuesta cuando cargue un archivo de imagen comprimido sin etiquetar. La información incluye *imageZipId*, *LabeledImageCount*, *labelCount*, *imageZipName*, *count*, *updatedAtTime*, y *URL*.

**imageZipId:** *Serie*. El ID que se ha asignado al archivo de imagen comprimido cargado.

**LabeledImageCount:** *Entero*. El número de imágenes que se han etiquetado previamente en el archivo de imagen comprimido.

**labelCount:** *Entero*. El número de etiquetas que se asignan a las imágenes en el archivo de imagen comprimido.

**imageZipName:** *Serie*. El nombre del archivo de imagen comprimido.

**count:** *Entero*. El número total de imágenes en el archivo de imagen comprimido.

**updatedAtTime:** *Hora*. La hora a la que se ha cargado el archivo de imagen comprimido.

**Url:** *Serie*. La vía de acceso del archivo que se almacena en el servidor.

## Cuerpo de ejemplo

```
[{"groupName":"DI","groupType":"classification","imageZip":[{"imageZipId":"b76b33b0-26d6-4f85-83ea-e520edbe5ff8","labeledImageCount":0,"labelCount":0,"imageZipName":"image.zip","count":85,"updatedAtTime":"2019-03-19 00:18:53.304","url":"/user/TENANT/VI/unlabeledgroup/b76b33b0-26d6-4f85-83ea-e520edbe5ff8.zip"}]}
```

## Respuesta satisfactoria

```
200
[{"groupName":"DI","id":"04f4eaf3-dc93-4ad8-b2e2-d3fbd202c758","groupType":"classification","imageZip":[{"imageZipId":"b76b33b0-26d6-4f85-83ea-e520edbe5ff8","labeledImageCount":0,"labelCount":0,"imageZipName":"image.zip","count":85,"updatedAtTime":"2019-03-19 00:18:53.304","url":"/user/TENANT/VI/unlabeledgroup/b76b33b0-26d6-4f85-83ea-e520edbe5ff8.zip"}]}
```

## Elementos de respuesta

**id:** *Serie*. El ID del grupo de datos.

**groupName:** *Serie*. El nombre del grupo de datos.

**groupType:** *Serie*. El tipo de grupo de datos.

**createdBy:** *Serie*. El propietario del grupo de datos.

**groupStatus:** *Serie*. El estado del grupo de datos no etiquetado.

**updatedAtTime:** *Hora*. La última vez que se ha actualizado el grupo de datos.

imageZip: *matriz JSON*. La información de los archivos de imagen comprimidos asociados. Cada elemento incluye imageZipId, LabeledImageCount, labelCount, imageZipName, count, updateTime, y URL.

imageZipId: *Serie*. El ID que se ha asignado al archivo de imagen comprimido cargado.

LabeledImageCount: *Entero*. El número de imágenes que se han etiquetado previamente en el archivo de imagen comprimido.

labelCount: *Entero*. El número de etiquetas que se asignan a las imágenes en el archivo de imagen comprimido.

imageZipName: *Serie*. El nombre del archivo de imagen comprimido.

count: *Entero*. El número total de imágenes en el archivo de imagen comprimido.

updateTime: *Hora*. La hora a la que se ha cargado el archivo de imagen comprimido.

url: *Serie*. La vía de acceso del archivo que se almacena en el servidor.

## Llamada de ejemplo

```
curl -k -X POST -H "APIKEY:apikey" -H
"Content-Type:application/json;charset=UTF-8"
"https://iotm.predictivesolutionsapps.ibmcloud.com/
ibm/iotm/service/unLabeledGroup?user=auto_mm&solution=vi"
--data '[{"groupName":"DI","groupType":"classification",
"imageZip":{"imageZipId":"b76b33b0-26d6-4f85-83ea-e520edbe5ff8",
"labeledImageCount":0,"labelCount":0,"imageZipName":"image.zip",
"count":85,"updateTime":"2019-03-19 00:18:53.304","url":
"/user/TENANT/VI/unlabeledgroup/b76b33b0-26d6-4f85-83ea-
e520edbe5ff8.zip"}}]'
```

## Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Obtener un grupo de datos sin etiqueta

Obtiene un grupo de datos sin etiqueta. El administrador de modelos tiene autorización para utilizar esta API.

### URL

/ibm/iotm/service/unLabeledGroup/*ID\_grupo*

### Método

El GET del tipo de solicitud

### Parámetros de URL

groupId: *Serie*. El ID del grupo de datos.

user: *Serie*. Se utiliza para identificar al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

### Cabeceras

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

### Parámetros de datos

Ninguno

### Cuerpo de ejemplo

Ninguno

## Respuesta satisfactoria

```
200
[{"groupName": "DI", "id": "04f4eaf3-dc93-4ad8-b2e2-d3fbd202c758",
"groupType": "classification", "imageZip": [{"imageZipId": "b76b33b0-
26d6-4f85-83ea-e520edbe5ff8", "labeledImageCount": 0, "labelCount": 0,
"imageZipName": "image.zip", "count": 85, "updatedAt": "2019-03-19
00:18:53.304", "url": "/user/TENANT/VI/unlabeledgroup/b76b33b0-26d6-
4f85-83ea-e520edbe5ff8.zip"}]}]
```

### Elementos de respuesta

**id:** *Serie*. El ID del grupo de datos.

**groupName:** *Serie*. El nombre del grupo de datos.

**groupType:** *Serie*. El tipo de grupo de datos.

**createdBy:** *Serie*. El propietario del grupo de datos.

**groupStatus:** *Serie*. El estado del grupo de datos no etiquetado.

**updatedAt:** *Hora*. La última vez que se ha actualizado el grupo de datos.

**imageZip:** *matriz JSON*. La información de los archivos de imagen comprimidos asociados. Cada elemento incluye **imageZipId**, **LabeledImageCount**, **labelCount**, **imageZipName**, **count**, **updatedAt** y **URL**.

**imageZipId:** *Serie*. El ID que se ha asignado al archivo de imagen comprimido cargado.

**LabeledImageCount:** *Entero*. El número de imágenes que se han etiquetado previamente en el archivo de imagen comprimido.

**labelCount:** *Entero*. El número de etiquetas que se asignan a las imágenes en el archivo de imagen comprimido.

**imageZipName:** *Serie*. El nombre del archivo de imagen comprimido.

**count:** *Entero*. El número total de imágenes en el archivo de imagen comprimido.

**updatedAt:** *Hora*. La hora a la que se ha cargado el archivo de imagen comprimido.

**Url:** *Serie*. La vía de acceso del archivo que se almacena en el servidor.

### Llamada de ejemplo

```
curl -k -H "APIKEY:apikey"
"https://iotm.predictivesolutionsapps.ibmcloud.com/ibm/iotm/service
/unLabeledGroup/04f4eaf3-dc93-4ad8-b2e2-d3fbd202c758?user=auto_mm&
solution=vi"
```

### Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Servicios de modelo

---

Los servicios de modelo le permiten realizar tareas como obtener todos los modelos, obtener un modelo y crear un modelo.

### Obtener todos los modelos

Obtiene todos los modelos existentes. El administrador de modelos tiene autorización para utilizar esta API.

#### URL

/ibm/iotm/service/model

## Método

El GET del tipo de solicitud

## Parámetros de URL

user: *Serie*. Se utiliza para identificar al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

## Cabeceras

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

## Parámetros de datos

Ninguno

## Cuerpo de ejemplo

Ninguno

## Respuesta satisfactoria

```
200
[{"modelId": "a9150695-5dc5-4bef-937d-c71fc186bc14",
"groupIds": "e83f9faf-2b08-44d1-81b0-7c69e6407329",
"modelName": "appscan test01",
"createdBy": "demoadm@cn.ibm.com",
"modelType": "classification",
"statusStatistics": {},
"dataFormat": "png",
"retrainPolicy": {
"scheduler": "Weekly, Sunday",
"ratio": "",
"reuseDayNumber": "",
"maxPieceNumber": "",
"archiveDataType": "0",
"imageNumber": 1000,
"accuracy": "70"
},
"parameters": {
"imageSize": "100*100",
"confidence": 100,
"trainParam": {
"epoch": 50,
"ratioTrain": 80,
"stepsize": 30,
"test_iter": 20,
"snapshot": 100,
"momentum": 0.9,
"ratioVal": 20,
"learningRate": 0.0001,
"display": 1,
"learningRatePolicy": "step",
"weight_decay": 0.00001,
"test_epoch": 10,
"network": "GoogLeNet",
"algorithm": "CNN",
"test_interval": 5,
"gamma": 0.1,
"solver_type": "SGD",
"maxIter": 600
}
},
"updatedAt": "2018-11-16T00:16:20.032-06:00",
"productType": "appscan test01"
}
...
]
```

## Elementos de respuesta

modelId: *Serie*. ID del modelo.

modelName: *Serie*. El nombre del modelo.

createdBy: *Serie*. El propietario del modelo.  
updatedAt: *Hora*. La última vez que se ha actualizado el modelo.  
productType: *Serie*. El tipo de producto del modelo.  
description: *Serie*. La descripción del modelo.  
StatusStatistics: *Serie*. El estado del modelo.  
groupIds: *Serie*. El ID del grupo de datos del modelo.  
dataFormat: *Serie*. El formato de datos del modelo.  
retrainPolicy: *Objeto JSON*. La política de repetición de formación del modelo.  
scheduler: *Fecha y hora*. La fecha de repetición de formación del modelo, de lunes a domingo.  
imageNumber: *Entero*. Un número de imagen superior al número cuya formación se puede repetir.  
accuracy: *Doble*. Un valor de precisión del modelo inferior al número cuya formación se puede repetir.  
parameters: *Objeto JSON*. Contiene el tamaño de imagen y la información de confianza del modelo y el campo trainParam.  
imageSize: *Doble \* Doble*. El tamaño de la imagen.  
confidence: *Doble*. El valor de confianza definido para el modelo.  
trainParam: *Objeto JSON*. Contiene la información de los parámetros de entrenamiento.

### Llamada de ejemplo

```
curl -k -H
"APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77d4f0579e
533dd34gggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5" "https://9.112
.229.91:9447/ibm/iotm/service/model?user=auto_mm&solution=vi"
```

### Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Obtener un modelo

Obtiene un modelo existente. El administrador de modelos tiene autorización para utilizar esta API.

### URL

/ibm/iotm/service/model/<modelID>

### Método

El GET del tipo de solicitud

### Parámetros de URL

modelId: *Serie*. ID del modelo. Obligatorio.  
user: *Serie*. Se utiliza para identificar al usuario. Obligatorio.  
solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.  
tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

### Cabeceras

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

### Parámetros de datos

Ninguno

## Cuerpo de ejemplo

Ninguno

## Respuesta satisfactoria

```
200
[
 {
 "modelId": "f39df888-7559-4283-ae9d-b3504118dddd",
 "groupIds": "9e744bc4-0a22-4bf3-9e63-fa39135842de",
 "modelName": "arthuryolo22000v2",
 "createdBy": "demoadm@cn.ibm.com",
 "modelType": "objectdetection",
 "description": "arthuryolo22000v2",
 "statusStatistics": {
 "trained": "1"
 },
 "dataFormat": "png",
 "retrainPolicy": {
 "scheduler": "Weekly, Sunday",
 "ratio": "",
 "reuseDayNumber": "",
 "maxPieceNumber": "",
 "archiveDataType": "0",
 "imageNumber": 1000,
 "accuracy": "70"
 },
 "trainedBy": "demoadm@cn.ibm.com",
 "parameters": {
 "JobInstanceMap": {
 "4a9605a2-03e3-43ff-9a47-d7638a0145d9": "f39df888-7559-4283-ae9d-b3504118dddd_1541744213466"
 },
 "odRetrainUrl": "/user/Q3T1/VI/modellist/f39df888-7559-4283-ae9d-b3504118dddd_1541761094044.zip",
 "imageSize": "100*100",
 "confidence": 100,
 "trainParam": {
 "ratioTrain": 80,
 "subBatchSize": 2,
 "scales": "1,1",
 "ratioVal": 20,
 "learningRate": 0.0001,
 "batchSize": 16,
 "steps": "6000,10000",
 "recommend": 0,
 "network": "YoloV2",
 "iteration": 10000,
 "algorithm": "YOLO"
 }
 },
 "updatedAt": "2018-11-09T00:24:19.949-06:00",
 "productType": "arthuryolo22000v2"
 }
]
```

## Elementos de respuesta

modelId: *Serie*. ID del modelo.

modelName: *Serie*. El nombre del modelo.

createdBy: *Serie*. El propietario del modelo.

updatedAt: *Hora*. La última vez que se ha actualizado el modelo.

productType: *Serie*. El tipo de producto del modelo.

description: *Serie*. La descripción del modelo.

StatusStatistics: *Serie*. El estado del modelo.

groupIds: *Serie*. El ID del grupo de datos del modelo.

dataFormat: *Serie*. El formato de datos del modelo.

retrainPolicy: *Objeto JSON*. La política de repetición de formación del modelo.

**scheduler:** *Fecha y hora.* La fecha de repetición de formación del modelo, de lunes a domingo.

**imageNumber:** *Entero.* Un número de imagen superior al número cuya formación se puede repetir.

**accuracy:** *Doble.* Un valor de precisión del modelo inferior al número cuya formación se puede repetir.

**parameters:** *Objeto JSON.* Contiene el tamaño de imagen y la información de confianza del modelo, así como los campos `odRetrainUrl`, `JobInstanceMap` y `trainParam`.

**imageSize:** *Doble \* Doble.* El tamaño de la imagen.

**confidence:** *Doble.* El valor de confianza definido para el modelo.

**trainParam:** *Objeto JSON.* Incluye el algoritmo e información sobre la red y el parámetro `hyper`.

### Llamada de ejemplo

```
curl -k -H
"APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77d4f0579e533
dd34ggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5" "https://9.112.229.91:
9447/ibm/iotm/service/model/ab219b13-16c0-4c7b-ae74-721f4719e314?user=auto_mm&so
lution=vi"
```

### Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Crear un modelo

Crea un modelo. El administrador de modelos tiene autorización para utilizar esta API.

### URL

`/ibm/iotm/service/model`

### Método

El POST del tipo de solicitud

### Parámetros de URL

**user:** *Serie.* Se utiliza para identificar al usuario. Obligatorio.

**solution:** *Serie.* Se utiliza para identificar la solución. Obligatorio.

**tenant:** *Serie.* Se utiliza para identificar al arrendatario. Opcional.

### Cabeceras

**Content-type:** `application/json`.

**APIKEY:** *clave cifrada.* La clave de la API utilizada para autenticación. Obligatorio.

### Parámetros de datos

**modelName:** *Serie.* El nombre del modelo.

**modelType:** *Serie.* El tipo de modelo. Las opciones son "classification" y "objectdetection".

**createdBy:** *Serie.* El propietario del modelo.

**description:** *Serie.* La descripción del modelo.

**groupIds:** *Serie.* El ID del grupo de datos del modelo.

**dataFormat:** *Serie.* El formato de datos del modelo.

**retrainPolicy:** *Objeto JSON.* La política de repetición de formación del modelo.

**scheduler:** *Fecha y hora.* La fecha de repetición de formación del modelo, de lunes a domingo.

**imageNumber:** *Entero.* Un número de imagen superior al número cuya formación se puede repetir.

accuracy: *Doble*. Un valor de precisión del modelo inferior al número cuya formación se puede repetir.

parameters: *Objeto JSON*. Contiene el tamaño de la imagen, la información de confianza del modelo y las propiedades de los parámetros de formación, incluida la información sobre algoritmo, red e hiperparámetro.

Los parámetros de formación, los hiperparámetros y los valores de tipo de modelo tienen relaciones correspondientes. Si el valor del tipo de modelo es "classification", el valor de "algorithm" es "CNN" y el valor de "network" tiene 3 opciones: "GoogLeNet", "LeNet" o "AlexNet". Los hiperparámetros correspondientes incluyen: "learningRate", que es el número real entre 0 y 1; "maxIter" o "epoch" que es un entero positivo; "stepsize" que es un entero positivo; "gamma" que es un número real entre 0 y 1; "learningRatePolicy" cuyo valor es "step"; "test\_iter" o "test\_epoch" que es un entero positivo; "test\_interval" que es un entero positivo; "snapshot", que es un entero igual o mayor que maxIter/20; "ratioTrain", el índice de imágenes utilizadas para la formación, cuyo valor son enteros entre 0 y 100; "ratioVal", el índice de imágenes utilizadas para la formación, cuyo valor son enteros entre 0 y 100.

Si el valor de tipo de modelo es "objectdetection", los valores de "algorithm" son: "FRCNN", "YOLO", "SSD". Si el valor de "algorithm" es "FRCNN", el valor de "network" correspondiente tiene dos opciones: "ZfNet" y "VGG16". Los hiperparámetros correspondientes incluyen: "learningRate", que es un número real entre 0 y 1; "iteration", que consta de enteros positivos separados por comas, por ejemplo: 10000, 10000, 10000, 10000; "stepsize", que es un entero positivo; "gamma", que es un número real entre 0 y 1; "ratioTrain", que es un entero entre 0 y 100; y "ratioVal", que es un entero entre 0 y 100.

Si el valor de "algorithm" es "YOLO", el valor de "network" correspondiente tiene 3 opciones: "YoloV2", "TinyYolo" y "YoloV1". Los hiperparámetros correspondientes son: "learningRate", que es un número real entre 0 y 1; "iteration", que es un entero positivo; "steps", que consta de enteros positivos separados por comas, por ejemplo: 100, 1000, 5000, 8000; "batchSize", que es un entero positivo; "scales", que son números reales separados por comas, por ejemplo: 0.1, 0.1, 0.1, 0.1; "subBatchSize", que es un entero positivo; "ratioTrain", que es un entero entre 0 y 100; "ratioVal", que es un entero entre 0 y 100.

Si el valor de "algorithm" es "SSD", el valor de "network" correspondiente es "SSD300". Los hiperparámetros correspondientes son "learningRate", que es un número real entre 0 y 1; "iteration", que es un entero positivo; "steps", que consta de enteros positivos separados por comas, por ejemplo: 100, 1000, 5000, 8000; "batchSize", cuyo valor es un entero positivo; "learningRatePolicy", cuyo valor es "multistep"; "snapshot" que es un entero igual o mayor que iteration/20; "ratioTrain", que es un entero entre 0 y 100; y "ratioVal", que es un entero entre 0 y 100.

confidence: *Doble*. El valor de confianza definido para el modelo.

productType: *Serie*. El tipo de producto del modelo.

## Cuerpo de ejemplo

```
[{"modelName": "walkermodel1", "productType": "walkermodel1", "dataFormat": "png", "modelType": "classification", "retrainPolicy": {"imageNumber": "1000", "accuracy": "70", "scheduler": "Weekly, Sunday", "reuseDayNumber": "", "maxPieceNumber": "", "ratio": ""}, "description": "this is a test", "groupIds": "b5164942-370c-418d-9fc2-f6ffd860b79d", "parameters": {"imageSize": "100*100", "confidence": 100, "trainParam": {"ratioTrain": 80, "ratioVal": 20, "gamma": 0.1, "stepsize": 33, "maxIter": 100, "learningRate": 0.01, "learningRatePolicy": "step", "test_iter": 1, "test_interval": 2, "snapshot": 10, "network": "GoogLeNet", "algorithm": "CNN"}}}]
```

## Respuesta satisfactoria

201

```
[{"modelId": "4d0dfe01-7a26-4071-92fb-ec597e7863ce", "modelName": "walkermodel1", "description": "this is a test", "productType": "walkermodel1", "groupIds": "b5164942-370c-418d-9fc2-f6ffd860b79d", "dataFormat": "png", "createdBy": "automm@cn.ibm.com", "updatedAt": "2018-02-13T00:44:31.288-06:00", "parameters": {"imageSize": "100*100", "confidence": 100, "trainParam": {"ratioTrain": 80, "stepsize": 33, "test_iter": 1, "snapshot": 10, "momentum": 0.9, "ratioVal": 20,
```



```
"learningRate":0.01,"display":1,"learningRatePolicy":"step",
"weight_decay":1.0E-5,"network":"GoogLeNet","test_interval":
2,"algorithm":"CNN","gamma":0.1,"solver_type":"SGD","maxIter":100}},
"retrainPolicy":{"ratio":"","scheduler":"Weekly, Sunday",
"maxPieceNumber":"","reuseDayNumber":"","imageNumber":"1000",
"accuracy":"70"}}]
```

## Elementos de respuesta

modelId: *Serie*. ID del modelo.

modelName: *Serie*. El nombre del modelo.

productType: *Serie*. El tipo de producto.

createdBy: *Serie*. El propietario del modelo.

updatedAt: *Hora*. La última vez que se ha actualizado el modelo.

productType: *Serie*. El tipo de producto del modelo.

description: *Serie*. La descripción del modelo.

groupIds: *Serie*. El ID del grupo de datos del modelo.

dataFormat: *Serie*. El formato de datos del modelo.

retrainPolicy: *Objeto JSON*. La política de repetición de formación del modelo.

scheduler: *Fecha y hora*. La fecha de repetición de formación del modelo, de lunes a domingo.

imageNumber: *Entero*. Un número de imagen superior al número cuya formación se puede repetir.

accuracy: *Doble*. Un valor de precisión del modelo inferior al número cuya formación se puede repetir.

parameters: *Objeto JSON*. Contiene el tamaño de imagen y la información de confianza del modelo.

confidence: *Doble*. El valor de confianza definido para el modelo.

## Llamada de ejemplo

```
curl -k -H
"APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c
77d4f0579e533dd34ggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5"
-H "Content-Type:application/json;charset=UTF-8" "https://9.112.229.91/
ibm/iotm/service/model?&tenant=VITest&solution=vi&user=auto_mm"
--data '[{"modelName":"testcreatemodel11","imagegroupList":[{"id\
":\
\b5164942-370c-418d-9fc2-f6ffd860b79d\
","dataFiles":[{"id\
":\
\b5164942-370c-418d-9fc2-f6ffd860b79d-\
b5164942-370c-418d-9fc2-f6ffd860b79d-74674410-06df-4a9e-a9e9-12f5038efc22\
","name\
":\
"Good_val.zip\
","count\
":5,"updatedAt\
":\
"12/27/2017,
10:38:13 AM\
"}],"name\
":\
"Good\
","updatedAt\
":\
"12/27/2017,
10:36:10 AM\
"}],"productType":"testcreatemodel11","dataFormat":
"png","modelType":"classification","retrainPolicy":{"imageNumber":
"1000","accuracy":"70","scheduler":"Weekly, Sunday","reuseDayNumber":
":"","maxPieceNumber":"","ratio":"","description":"this is a test",
"groupIds":"b5164942-370c-418d-9fc2-f6ffd860b79d","parameters":
{"imageSize":"100*100","confidence":100,"trainParam":{"ratioTrain":
80,"ratioVal":20,"gamma":0.1,"stepsize":33,"maxIter":100,"learningRate":
0.01,"learningRatePolicy":"step","test_iter":1,"test_interval":2,
"snapshot":10,"network":"GoogLeNet","algorithm":"CNN"}}}]'
```

## Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Actualizar un modelo

Actualizar un modelo existente. El administrador de modelos tiene autorización para utilizar esta API.

### URL

/ibm/iotm/service/model/*modelId*

## Método

El PUT del tipo de solicitud

## Parámetros de URL

modelId: *Serie*. ID del modelo. Obligatorio.

user: *Serie*. Se utiliza para identificar al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

## Cabeceras

Content-type: application/json.

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

## Parámetros de datos

groupIds: *Serie*. El ID del grupo de datos.

## Cuerpo de ejemplo

```
[{"groupIds": "45805b3c-f234-4f18-a55e-61ce203a7db4,678a3f42-f9d4-48a2-bb29-c226d61a08ea"}]
```

## Respuesta satisfactoria

```
200
[{}]
```

## Elementos de respuesta

Ninguno

## Llamada de ejemplo

```
curl -k -H
"APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77d4f0579e533dd
34ggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5" -H "Content-Type:application
/json;charset=UTF-8" -X PUT "https://9.112.229.91:9447/ibm/iotm/service/model/106ef
cb7-1338-4ed2-b8ca-27b45e11e865?user=auto_mm&solution=vi" --data '{"groupIds": "9f
7aa9d6-24d6-4611-8d4d-79a52d2bab02"}'
```

## Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Suprimir un modelo

Suprime un modelo existente y todas las instancias de modelo relacionadas. El administrador de modelos tiene autorización para utilizar esta API.

## URL

/ibm/iotm/service/model/*modelId*

## Método

El DELETE del tipo de solicitud

## Parámetros de URL

modelId: *Serie*. ID del modelo. Obligatorio.

user: *Serie*. Se utiliza para identificar al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

## Cabeceras

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

## Parámetros de datos

Ninguno

## Cuerpo de ejemplo

[{}]

## Respuesta satisfactoria

```
200
[{}]
```

## Elementos de respuesta

Ninguno

## Llamada de ejemplo

```
curl -k -X DELETE -H "APIKEY:8b79658225de5
3488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77d4f0579e53
3dd34ggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5"
"https://9.112.229.91:9447/ibm/iotm/service/model/
106efcb7-1338-4ed2-b8ca-27b45e11e865?user=auto_mm&solution=
vi" -d '[{}]
```

## Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Obtener todos los modelos compartidos

Obtiene todos los modelos compartidos. El administrador de modelos tiene autorización para utilizar esta API.

## URL

/ibm/iotm/service/model/

## Método

El GET del tipo de solicitud

## Parámetros de URL

category: 'shared'. Se utiliza para identificar el catálogo de modelos. Obligatorio.

user: *Serie*. Se utiliza para identificar al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

## Cabeceras

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

## Parámetros de datos

Ninguno

## Cuerpo de ejemplo

Ninguno

## Respuesta satisfactoria

```
[{"modelId": "0f9c8d47-5ccc-4e52-8e0d-0596e792a2cd",
"groupIds": "e5e83a2d-5814-45b3-af40-74096198cd39", "modelName":
"Car Seat Defect Inspection", "createdBy": "vimodelmanager@163.com",
"modelType": "objectdetection", "description": "This is the object
detection model we build to detect the wrinkle defect for car seat."}]
```

```
The inspection result will identify whether there is wrinkle defect
on the car and localize the defect on the image.,"statusStatistics":
{"deployed":"1"},"dataFormat":"jpg","retrainPolicy":{"scheduler":
"Weekly, Sunday","ratio":"","reuseDayNumber":"","maxPieceNumber":
"", "imageNumber":"1000","accuracy":"70"},"trainedBy":
"vimodelmanager@163.com","parameters":{"odRetrainUrl":
"\user\VI\VI\modellist\carseatfrcnn_1516957839865.zip",
"lastRetrainData":
"[{"groupID":"e5e83a2d-5814-45b3-af40-74096198cd39"}]"},
"imageSize":"100*100","confidence":"100"},"updatedAtTime":
"2018-01-26T03:11:05.804-06:00","productType":"carseat"}]
```

## Elementos de respuesta

modelId: *Serie*. ID del modelo.

groupIds: *Serie*. El ID del grupo de datos del modelo.

modelName: *Serie*. El nombre del modelo.

createdBy: *Serie*. El propietario del modelo.

modelType: *Serie*. El tipo de modelo.

description: *Serie*. La descripción del modelo.

updatedAtTime: *Hora*. La última vez que se ha actualizado el modelo.

productType: *Serie*. El tipo de producto del modelo.

StatusStatistics: *Serie*. El estado del modelo.

dataFormat: *Serie*. El formato de datos del modelo.

retrainPolicy: *Objeto JSON*. La política de repetición de formación del modelo.

scheduler: *Fecha y hora*. La fecha de repetición de formación del modelo, de lunes a domingo.

imageNumber: *Entero*. Un número de imagen superior al número cuya formación se puede repetir.

accuracy: *Doble*. Un valor de precisión del modelo inferior al número cuya formación se puede repetir.

parameters: *Objeto JSON*. Contiene el tamaño de imagen y la información de confianza del modelo.

imageSize: *Doble \* Doble*. El tamaño de la imagen.

confidence: *Doble*. El valor de confianza definido para el modelo.

## Llamada de ejemplo

```
curl -k -H
"APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830a5afb
8c149f6aab2c77d4f0579e533dd34ggb959317ac69ff73f886fc3713
bf305a158fdf485cc6f275f5" " https://9.112.229.91:9447/ibm/
iotm/service/model?category=shared&tenant=Q3T1&solution=
vi&user=testuser"
```

## Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Servicios de instancias de modelos

Los servicios de instancias de modelos le permiten realizar tareas como crear una instancia de modelo, devolver una instancia de modelo perteneciente a un modelo y devolver una instancia de modelo.

### Crear una instancia de modelo

Crea una instancia de modelo. El administrador de modelos tiene autorización para utilizar esta API.

## URL

/ibm/iotm/service/modelInstance

## Método

El POST del tipo de solicitud

## Parámetros de URL

user: *Serie*. Se utiliza para identificar al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

## Cabeceras

Content-type: application/json.

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

## Parámetros de datos

modelID: *Serie*. ID del modelo.

trainData: *Objeto JSON*. Los datos de formación del modelo.

groupId: *Serie*. El ID del grupo de datos.

fileIds: *Serie*. El ID del archivo de datos.

## Cuerpo de ejemplo

```
[
 {
 "modelId": "91a21e63-17e6-4e0e-bbe8-5c6b9d599a35",
 "trainData": [
 {
 "groupId": "9f7aa9d6-24d6-4611-8d4d-79a52d2bab02",
 "fileIds": "9f7aa9d6-24d6-4611-8d4d-79a52d2bab02_76e8b2cb-057a-4904-8fad-bbca93abd80b"
 },
 {
 "groupId": "d84c028e-235a-4c13-8a77-894500b8a736",
 "fileIds": "d84c028e-235a-4c13-8a77-894500b8a736_fe201c83-b138-4dca-b918-ae1b9f7db894"
 },
 {
 "groupId": "c5f11ed9-db49-403a-972c-cd6cb1d0f8eb",
 "fileIds": "c5f11ed9-db49-403a-972c-cd6cb1d0f8eb_bd0c8246-7208-44d8-a2fa-e74c1578eebe"
 }
]
 }
]
```

## Respuesta satisfactoria

```
200
[
 {
 "status": "deployed",
 "validateData": [
 {
 "groupId": "1a7d4c81-36ff-4a3e-a614-9e996fb380ba"
 }
],
 "instanceId": "4cc40ab4-d3b3-4911-b7a0-90281248c075_1542340917709",
 "validateResult": {
 "matrix": [
 ["", "scratch", "Chris"],
 ["mAP", "-", "-"],
 ["recall", "-", "-"]
],
 "type": "objectdetection",
 "accuracy": "-",
 }
 }
]
```

```

 "fileCount": "0"
 },
 "trainedBy": "demoadm@cn.ibm.com",
 "validateTime": "2018-11-15T22:01:57.726-06:00",
 "fileCount": "0",
 "modelId": "4cc40ab4-d3b3-4911-b7a0-90281248c075",
 "createdBy": "demoadm@cn.ibm.com",
 "instanceName": "1",
 "parameters": {
 "edges": [
 {
 "id": "1542102975873"
 },
 {
 "id": "1508476898000"
 }
],
 "modelType": "objectdetection",
 "attached": true,
 "edgesDeploying": []
 },
 "isDeletable": false,
 "modelUrl": "/user/Q3T1/VI/modellist/frcnnmodel_1542340914671.zip",
 "trainData": [
 {
 "groupId": "1a7d4c81-36ff-4a3e-a614-9e996fb380ba"
 }
],
 "accuracy": "-",
 "updatedAt": "2018-11-16T04:21:34.656-06:00"
}
]

```

### Elementos de respuesta

**instanceId:** *Serie.* El ID de instancia del modelo.

**modelId:** *Serie.* ID del modelo.

**instanceName:** *Entero.* El nombre de instancia del modelo.

**createdBy:** *Serie.* El propietario de la instancia del modelo.

**updatedAt:** *Hora.* La última vez que se ha actualizado la instancia del modelo.

**status:** *Serie.* El estado de la instancia del modelo.

**trainData:** *Objeto JSON.* Los datos de formación del modelo.

**groupId:** *Serie.* El ID del grupo de datos.

**fileIds:** *Serie.* El ID del archivo de datos.

**parameters:** *Objeto JSON.* Contiene información sobre el sistema perimetral implementado.

**edges:** *Matriz JSON.* Contiene información sobre el ID del sistema perimetral implementado.

**libModelUrl:** *Serie.* modelURL de la instancia de modelo en la biblioteca de modelos.

**lastAutoTrainTime:** *Serie.* La hora de la última formación automática, en formato yyyy-mm-dd hh:mm:ss.SSS.

**trainType:** *Serie.* El tipo de formación: online\_train, retrain\_manual, or retrain\_auto.

**onDataPreparing:** *Serie.* 1 significa que se están preparando los datos. 0 significa que la preparación de los datos ha finalizado.

**snapshot:** *Serie.* La información de instantánea que se ha usado para la formación en línea.

**unlabeledGroupId:** *Serie.* El ID de grupo sin etiqueta.

### Llamada de ejemplo

```

curl -k -H
"APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77d4f0579e533dd
34gg959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5" -H "Content-Type:application/
json;charset=UTF-8" " https://9.112.229.91:9447/ibm/iotm/service/modelInstance?user
=auto_mm&solution=vi" --data '{
 "modelId": "91a21e63-17e6-4e0e-bbe8-5c6b9d59
9a35",
 "trainData": [
 {
 "groupId": "9f7aa9d6-24d6-4611-8d4d-79a52d2ba
b02",
 "fileIds": "9f7aa9d6-24d6-4611-8d4d-79a52d2bab02_76e8b2cb-057a-4904-8fa
d-bbca93abd80b"
 },
 {
 "groupId": "d84c028e-235a-4c13-8a77-894500b8a73
6",
 "fileIds": "d84c028e-235a-4c13-8a77-894500b8a736_fe201c83-b138-4dca-b918-a
e1b9f7db894"
 },
 {
 "groupId": "c5f11ed9-db49-403a-972c-cd6cb1d0f8eb",

```

```
"fileIds": "c5f11ed9-db49-403a-972c-cd6cb1d0f8eb_bd0c8246-7208-44d8-a2fa-e74c1578eebe" }] }
```

## Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Obtener la instancia de modelo perteneciente al modelo

Obtiene la instancia de modelo que pertenece al modelo. El administrador de modelos y el experto en datos tienen autorización para utilizar esta API.

### URL

/ibm/iotm/service/modelInstance

### Método

El GET del tipo de solicitud

### Parámetros de URL

modelId: *Serie*. ID del modelo. Obligatorio.

user: *Serie*. Se utiliza para identificar al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

### Cabeceras

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

### Parámetros de datos

Ninguno

### Cuerpo de ejemplo

Ninguno

### Respuesta satisfactoria

```
200
[
 {
 "modelId": "91a21e63-17e6-4e0e-bbe8-5c6b9d599a35",
 "createdBy": "auto_mm",
 "status": "draft",
 "instanceName": "1",
 "instanceId": "91a21e63-17e6-4e0e-bbe8-5c6b9d599a35_1512029557869",
 "trainData": [
 {
 "fileIds": "9f7aa9d6-24d6-4611-8d4d-79a52d2bab02_76e8b2cb-057a-4904-8fad-bbca93abd80b",
 "groupId": "9f7aa9d6-24d6-4611-8d4d-79a52d2bab02"
 },
 {
 "fileIds": "d84c028e-235a-4c13-8a77-894500b8a736_fe201c83-b138-4dca-b918-ae1b9f7db894",
 "groupId": "d84c028e-235a-4c13-8a77-894500b8a736"
 },
 {
 "fileIds": "c5f11ed9-db49-403a-972c-cd6cb1d0f8eb_bd0c8246-7208-44d8-a2fa-e74c1578eebe",
 "groupId": "c5f11ed9-db49-403a-972c-cd6cb1d0f8eb"
 }
],
 "fileCount": "9",
 "updatedAt": "2017-11-30T16:12:37.869+08:00"
 }
]
```

## Elementos de respuesta

`modelId`: *Serie*. ID del modelo.  
`createdBy`: *Serie*. El propietario de la instancia del modelo.  
`status`: *Serie*. El estado de la instancia del modelo.  
`instanceId`: *Serie*. El ID de instancia del modelo.  
`instanceName`: *Entero*. El nombre de instancia del modelo.  
`trainData`: *Objeto JSON*. Los datos de formación del modelo.  
`groupId`: *Serie*. El ID del grupo de datos.  
`fileIds`: *Serie*. El ID del archivo de datos.  
`fileCount`: *Serie*. El número del recuento total de imágenes del modelo.  
`updatedAtTime`: *Hora*. La última vez que se ha actualizado la instancia del modelo.

## Llamada de ejemplo

```
curl -k -H
"APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77d4f0579e533d
d34gggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5" "https://9.112.229.91:9447
/ibm/iotm/service/modelInstance?modelId=91a21e63-17e6-4e0e-bbe8-5c6b9d599a35&user=
auto_mm&solution=vi"
```

## Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Obtener una instancia de modelo

Obtiene una instancia de modelo. El administrador de modelos y el experto en datos tienen autorización para utilizar esta API.

### URL

`/ibm/iotm/service/modelInstance/modelInstanceId`

### Método

El GET del tipo de solicitud

### Parámetros de URL

`modelInstanceId`: *Serie*. El ID de instancia del modelo. Obligatorio.  
`user`: *Serie*. Se utiliza para identificar al usuario. Obligatorio.  
`solution`: *Serie*. Se utiliza para identificar la solución. Obligatorio.  
`tenant`: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

### Cabeceras

`APIKEY`: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

### Parámetros de datos

Ninguno

### Cuerpo de ejemplo

Ninguno

### Respuesta satisfactoria

```
200
[
 {
```



```

"modelId": "91a21e63-17e6-4e0e-bbe8-5c6b9d599a35",
"createdBy": "auto_mm",
"status": "draft",
"instanceName": "1",
"instanceId": "91a21e63-17e6-4e0e-bbe8-5c6b9d599a35_1512029557869",
"trainData": [
 {
 "fileIds": "9f7aa9d6-24d6-4611-8d4d-79a52d2bab02_76e8b2cb-057a-4904-8fad-bbca93abd80b",
 "groupId": "9f7aa9d6-24d6-4611-8d4d-79a52d2bab02"
 },
 {
 "fileIds": "d84c028e-235a-4c13-8a77-894500b8a736_fe201c83-b138-4dca-b918-ae1b9f7db894",
 "groupId": "d84c028e-235a-4c13-8a77-894500b8a736"
 },
 {
 "fileIds": "c5f11ed9-db49-403a-972c-cd6cb1d0f8eb_bd0c8246-7208-44d8-a2fa-e74c1578eebe",
 "groupId": "c5f11ed9-db49-403a-972c-cd6cb1d0f8eb"
 }
],
"updatedAt": "2017-11-30T16:12:37.869+08:00"
}
]

```

### Elementos de respuesta

`instanceId`: *Serie*. El ID de instancia del modelo.

`modelId`: *Serie*. ID del modelo.

`instanceName`: *Entero*. El nombre de instancia del modelo.

`createdBy`: *Serie*. El propietario de la instancia del modelo.

`updatedAt`: *Hora*. La última vez que se ha actualizado la instancia del modelo.

`status`: *Serie*. El estado de la instancia del modelo.

`trainData`: *Objeto JSON*. Los datos de formación del modelo.

`groupId`: *Serie*. El ID del grupo de datos.

`fileIds`: *Serie*. El ID del archivo de datos.

### Llamada de ejemplo

```

curl -k -H
"APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c
77d4f0579e533dd34gggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5"
"https://9.112.229.91:9447/ibm/iotm/service/modelInstance/91a21e63-17e6-4e0e-bbe8-5c6b9d599a35_1512029557869?user=auto_mm&solution=vi"

```

### Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Obtener el resultado de validación de la instancia del modelo

Obtiene el resultado de validación de la instancia del modelo para el informe. El administrador de modelos y el experto en datos tienen autorización para utilizar esta API.

### URL

`/ibm/iotm/service/modelInstance/modelInstanceId/validateResult`

### Método

El GET del tipo de solicitud

### Parámetros de URL

`modelInstanceId`: *Serie*. El ID de instancia del modelo. Obligatorio.

`user`: *Serie*. Se utiliza para identificar al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

### Cabeceras

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

### Parámetros de datos

Ninguno

### Cuerpo de ejemplo

Ninguno

### Respuesta satisfactoria

```
200
[
 {
 "instanceName": "FirstBlood_2017-08-08 19:21:53.888",
 "validateResult": {
 "matrix": [
 [
 "",
 "type1",
 "type2",
 "type3"
],
 [
 "type1",
 "77,35,45",
 "0,0,40",
 "6,3,50"
],
 [
 "type2",
 "17,8,45",
 "100,40,40",
 "4,2,50"
],
 [
 "type3",
 "4,2,45",
 "0,0,40",
 "90,45,50"
],
 [
 "Total",
 "45",
 "40",
 "50"
]
],
 "accuracy": "84.56"
 },
 "validateTime": "2017-08-09T22:58:46.416+08:00",
 "fileCount": "200"
 }
]
```

### Elementos de respuesta

instanceName: *Entero*. El nombre de instancia del modelo.

validateResults: *Objeto JSON*. El resultado de la validación de la instancia del modelo.

validateTime: *Hora*. La hora de validación de la instancia del modelo.

fileCount: *Serie*. El número de imagen de la instancia del modelo.

accuracy: *Doble*. El valor de precisión de la instancia del modelo.

### Llamada de ejemplo

```
curl -k -H
"APIKEY:8b79658225de53488321fb7b657f9f161acd2ea830a5afb8c149f6aab2c77d4f0579e53
```

```
3dd34gg959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5" "https://9.112.229.91:9447/
ibm/iotm/service/modelInstance/91a21e63-17e6-4e0e-bbe8-5c6b9d599a35_1512029557869/
validateResult?user=auto_mm&solution=vi"
```

## Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Servicios de resultados de inspección

Los servicios de resultados de inspección le permiten realizar tareas como obtener una lista de resultados de la inspección, obtener detalles de los resultados de la inspección y obtener una visión general de las celdas de resultados de la inspección.

### Obtener la lista de resultados de inspección

Obtiene la lista de resultados de inspección. El inspector y el supervisor tienen autorización para utilizar esta API.

#### URL

/ibm/iotm/service/inspectResult

#### Método

El GET del tipo de solicitud

#### Parámetros de URL

**user:** *Serie*. Se utiliza para identificar al usuario. Obligatorio.

**solution:** *Serie*. Se utiliza para identificar la solución. Obligatorio.

**tenant:** *Serie*. Se utiliza para identificar al arrendatario. Opcional.

**isConfirmed:** *Entero*. Un 0 indica que el resultado de la inspección no se ha confirmado. Un 1 indica que el resultado de la inspección se ha confirmado. Un 2 indica que el resultado de la inspección se desconoce. Opcional.

**cell:** *Serie*. Se utiliza para identificar la celda. Opcional.

**inspectTime:** *Hora*. Se utiliza para identificar la hora de inspección.

**reverse:** *Booleano*. Se utiliza para identificar el orden de la lista de resultados de inspección.

**pageCount:** *Entero*. El recuento de páginas.

**sampleRate:** *Entero*. La tasa de comprobación de muestreo.

#### Cabeceras

**APIKEY:** *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

#### Parámetros de datos

Ninguno

#### Cuerpo de ejemplo

Ninguno

#### Respuesta satisfactoria

```
200
[{"id": "plant1,line1,cell1_1511768220935_AUTO|auto_1511768206483.jpg",
"isConfirmed": "0", "inspectFileUrl": "imageserver/911222954/\\AUTO|
auto_plant1_line1_cell1_1511768206483.jpg", "isUpdated": "0",
"defectTypeCount": "1", "cell": "plant1,line1,cell1", "instanceId": "ebf9
4e98-958c-48b0-8377-826241ffc8d8_1511765289053", "defectCount": "1",
"confidence": "99.87", "inspectId": "auto_1511768206483.jpg",
"inspectTime": "1511768220935"}, {"id": "plant1,line1,
```

```
cell1_1511761780903_AUTO|auto_1511761766470.jpg", "isConfirmed": "0", "
inspectFileUrl": "imageserver/911222954/\\AUTO|
auto_plant1_line1_cell1_1511761766470.jpg", "isUpdated": "0",
"defectTypeCount": "1", "cell": "plant1,line1,cell1", "instanceId":
"9ab990d6-465a-48b0-838c-416c8e4a9879_1511761328538", "defectCount":
"1", "confidence": "99.79", "inspectId": "auto_1511761766470.jpg",
"inspectTime": "1511761780903"}},
{"id": "plant1,line1,cell1_1511761559536_AUTO|auto_1511761541254.jpg", "isConfirmed": "0", "i
nspectFileUrl": "imageserver/911222954/\\AUTO|
auto_plant1_line1_cell1_1511761541254.jpg", "isUpdated": "0",
"defectTypeCount": "1", "cell": "plant1,line1,cell1", "instanceId":
"9ab990d6-465a-48b0-838c-416c8e4a9879_1511761328538", "defectCount":
"1", "confidence": "99.87", "inspectId": "auto_1511761541254.jpg",
"inspectTime": "1511761559536"}}, {"id": "plant1,line1,
cell1_1511421414687_AUTO|auto_1511421410914.jpg", "isConfirmed": "0",
"inspectFileUrl": "imageserver/911222954/\\AUTO|
auto_plant1_line1_cell1_1511421410914.jpg", "isUpdated": "0",
"defectTypeCount": "1", "cell": "plant1,line1,cell1", "instanceId":
"ae509ec0-e1d6-4827-afd1-a8af598eb28c_1511419442550",
"defectCount": "1", "confidence": "99.99", "inspectId":
"auto_1511421410914.jpg", "inspectTime": "1511421414687"}]
```

## Elementos de respuesta

id: *Serie*. El ID de inspección.

isConfirmed: *Entero*. Un 0 indica que el resultado de la inspección no se ha confirmado. Un 1 indica que el resultado de la inspección se ha confirmado. Un 2 indica que el resultado de la inspección se desconoce.

cell: *Serie*. La información de la celda.

inspectFileUrl: *Serie*. El URL del archivo de inspección.

defectTypeCount: *Entero*. El número de tipos de defectos.

defectCount: *Entero*. El número de defectos.

confidence: *Doble*. El nivel de confianza del defecto.

inspectId: *Serie*. El ID de inspección.

inspectTime: *Hora*. La hora de inspección.

## Llamada de ejemplo

```
curl -k -H
"APIKEY: 8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77d4f0579
e533dd34ggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5" "https://9.112.22
9.91:9447/ibm/iotm/service/inspectResult?user=auto_inspector&solution=vi&isCon
firmed=0&cell=plant1,line1,cell1&sampleRate=10&pageCount=1&reverse=true"
```

## Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Obtener detalles de resultados de inspección

Obtiene detalles acerca de los resultados de la inspección. El inspector y el supervisor tienen autorización para utilizar esta API.

### URL

/ibm/iotm/service/inspectResult/*inspectResultId*

### Método

El GET del tipo de solicitud

### Parámetros de URL

inspectResultId: *Serie*. Se utiliza para identificar el ID de inspección.

user: *Serie*. Se utiliza para identificar al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

### Cabeceras

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

### Parámetros de datos

Ninguno

### Cuerpo de ejemplo

Ninguno

### Respuesta satisfactoria

```
200
[
 {
 "isConfirmed": "0",
 "inspectFileUrl": "imageserver/edge1/image1.jpg",
 "isUpdated": "0",
 "cell": "plant1,line1,cell1",
 "instanceId": "6e4bfc6b-0ddb-4c85-81e7-0ea789681ae8_775e39b9-fedd-4418-a5b3-70de6b86fbec",
 "inspectResult": [
 {
 "position": {
 "height": 331,
 "width": 467,
 "y": 1653,
 "x": 897
 },
 "probableTypes": [
 {
 "confidence": 100,
 "type": "NoDefect"
 },
 {
 "confidence": 0,
 "type": "Defect"
 },
 {
 "confidence": 0,
 "type": "Mist"
 }
]
 }
]
 }
]
```

### Elementos de respuesta

isConfirmed: *Entero*. Un 0 indica que el resultado de la inspección no se ha confirmado. Un 1 indica que el resultado de la inspección se ha confirmado. Un 2 indica que el resultado de la inspección se desconoce.

inspectFileUrl: *Serie*. El URL del archivo de inspección.

inspectId: *Serie*. El ID de inspección.

inspectResult: *Objeto JSON*. El resultado de la inspección.

position: *Matriz JSON*. La altura, anchura de la inspección junto con la información de y/x.

probableTypes: *Objeto JSON*. Cada tipo de defecto y su nivel de confianza.

### Llamada de ejemplo

```
curl -k -H
"APIKEY: 8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2
c77d4f0579e533dd34gggb959317ac69ff73f886fc3713bf305a158fd485cc6f275f5"
"https://9.112.229.91:9447/ibm/iotm/service/inspectResult/plant1,line1,
```

```
cell1_1511166988402_AUTO|auto_1511166972283.jpg?user=auto_inspector&
solution=vi"
```

### Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Obtener visión general de celdas del resultado de inspección

Obtiene la visión general de celdas del resultado de la inspección. El inspector y el supervisor tienen autorización para utilizar esta API.

### URL

/ibm/iotm/service/inspectResultCell

### Método

El GET del tipo de solicitud

### Parámetros de URL

user: *Serie*. Se utiliza para identificar al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

### Cabeceras

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

### Parámetros de datos

Ninguno

### Cuerpo de ejemplo

Ninguno

### Respuesta satisfactoria

```
200
[
 {
 "confirmed": 1,
 "cell": "plant1,line1,cell1",
 "unknown": 0,
 "unconfirmed": 10
 }
]
```

### Elementos de respuesta

confirmed: *Entero*. Un número de inspector confirmado.

unconfirmed: *Entero*. Un número de inspector no confirmado.

unknown: *Entero*. Un inspector desconocido.

cell: *String*. La celda de la imagen.

### Llamada de ejemplo

```
curl -k -H
"APIKEY: 8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c7
7d4f0579e533dd34gg959317ac69ff73f886fc3713bf305a158fdf485ccc6f275f5" "
https://9.112.229.91:9447/ibm/iotm/service/inspectResultCell?user=auto_
inspector&solution=vi"
```

### Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Confirmar resultados de inspección

Confirma el resultado de la inspección. El inspector y el supervisor tienen autorización para utilizar esta API.

### URL

/ibm/iotm/service/inspectResult/inspectResultId

### Método

El PUT del tipo de solicitud

### Parámetros de URL

inspectResultId: *Serie*. Se utiliza para identificar el ID de inspector.

user: *Serie*. Se utiliza para identificar al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

### Cabeceras

Content-type: application/json.

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

### Parámetros de datos

position: *Matriz JSON*. La altura, anchura de la inspección junto con la información de y/x.

probableTypes: *Objeto JSON*. Cada tipo de defecto y su nivel de confianza.

logs: *Objeto JSON*. Contiene la acción de confirmación, el operador y la hora de actualización.

description: *Serie*. La descripción del inspector de confirmación.

### Cuerpo de ejemplo

```
[
 {
 "confirmedResult": [
 {
 "position": {
 "height": 3231,
 "width": 467,
 "y": 1653,
 "x": 897
 },
 "probableTypes": [
 {
 "confidence": 100,
 "type": "NoDefect"
 },
 {
 "confidence": 0,
 "type": "Defect"
 },
 {
 "confidence": 0,
 "type": "MisT"
 }
],
 "logs": [{"action": "confirm",
 "operator": "inspector1",
 "updateTime": "1503657054321"}]
 }
],
 "description": "VI result is OK"
 }
]
```

## Respuesta satisfactoria

```
200
[{}]
```

## Elementos de respuesta

Ninguno

## Llamada de ejemplo

```
curl -k -H"APIKEY: 8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8
c149f6aab2c77d4f0579e533dd34ggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5" -H
"Content-Type:application/json;charset=UTF-8" -X PUT "https://9.112.229.91:9447/ibm
/iotm/service/inspectResult/plant1,line1,cell1_1511507025284_AUTO|auto_1511507012702
.jpg?user=auto_inspector&solution=vi" --data '[{ "confirmedResult": [{
"position": { "height": 430, "width": 552, "x": 1254, "y": 2123 },
"probableTypes": [{ "confidence": 99.87, "type": "Defect" }, {
"confidence": 0.1, "type": "MisT" }, { "confidence": 0.03, "type": "NoDefect"
}], "logs": [{"action": "confirm", "operator": "auto_super",
"updateTime": "1503657054321"}] }], "description": "VI result is OK" }]'
```

## Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Suprimir resultados de inspección

Suprime los resultados de inspección. El inspector y el supervisor tienen autorización para utilizar esta API.

## URL

/ibm/iotm/service/inspectResult

## Método

El DELETE del tipo de solicitud

## Parámetros de URL

**user:** *Serie.* Se utiliza para identificar al usuario. Obligatorio.

**solution:** *Serie.* Se utiliza para identificar la solución. Obligatorio.

**tenant:** *Serie.* Se utiliza para identificar al arrendatario. Opcional.

**cell:** *Serie.* Se utiliza para identificar la celda. Obligatorio.

**inspectTime:** *Hora.* Se utiliza para identificar la hora de inspección. Los resultados de inspección que se producen antes de esta hora se suprimen. Opcional.

**instanceId:** *Serie.* Se utiliza para identificar el ID de instancia. Los resultados de inspección producidos por este ID de instancia se suprimen. Opcional.

## Cabeceras

**APIKEY:** *clave cifrada.* La clave de la API utilizada para autenticación. Obligatorio.

## Parámetros de datos

Ninguno.

## Cuerpo de ejemplo

Ninguno.

## Respuesta satisfactoria

```
200
[{}]
```



## Elementos de respuesta

Ninguno.

## Llamada de ejemplo

```
curl -k -H "APIKEY: 8b79658225de53488321fb
7bb657f9f161acd2ea830a5afb8c149f6aab2c77d4f0579e533dd34
ggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5"
-H "Content-Type:application/json;charset=UTF-8" -X DELETE
"https:// 9.112.229.91:9447/ibm/iotm/service/inspectResult?instanceId=
04b61494-99ec-49f4-9a0f-0125650c40dd_1520316716077&
tenant=Q3T1&solution=vi&user=demoadm@cn.ibm.com&cell=plant2,line2,cell2&inspectTime=
1520324778324"
```

## Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Servicios de acciones de instancias de modelos

---

Los servicios de acciones de instancias de modelos le permiten realizar tareas como enviar una instancia de modelo, adjuntar un archivo de modelo y validar una instancia de modelo.

### Formar instancia de modelo

Forme una instancia de modelo utilizando un modelo de aprendizaje profundo. El administrador de modelos tiene autorización para utilizar esta API.

#### URL

`/ibm/iotm/service/modelInstance/modelInstanceId`

#### Método

El PUT del tipo de solicitud

#### Parámetros de URL

`modelInstanceId`: *Serie*. El ID de instancia del modelo. Obligatorio.

`action`: *Serie*. La acción. Establecida como `train`. Obligatorio.

`user`: *Serie*. Identifica al usuario. Obligatorio.

`solution`: *Serie*. Se utiliza para identificar la solución. Obligatorio.

`tenant`: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

#### Cabeceras

`Content-Type`: `application/json`.

`APIKEY`: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

#### Parámetros de datos

Ninguno.

#### Cuerpo de ejemplo

```
[{}]
```

#### Respuesta satisfactoria

```
200
[{"status":"waiting","instanceId":
"1e760614-ab0e-4c2a-be2e-435505087fe4_1518060853002",
"parameters":{"modelType":"classification","trainType":
"online_train"}}]
```

## Elementos de respuesta

`status`: *Serie*. El estado de la repetición de formación de la instancia del modelo.

`instanceId`: *Serie*. El ID de instancia del modelo.

`updatedAtTime`: *Hora*. La última vez que se ha actualizado la instancia del modelo.

`parameters`: *Objeto JSON*. Información acerca del tipo de modelo y el tipo de formación.

## Llamada de ejemplo

```
curl -k -X PUT -H
"APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830a5af
b8c149f6aab2c77d4f0579e533dd34ggb959317ac69ff73f886fc37
13bf305a158fdf485cc6f275f5" -H "Content-Type:
application/json;charset=UTF-8"
"https://int_iotm.predictivesolutionsapps.ibmcloud.com/
ibm/iotm/service/modelInstance/1e760614-ab0e-4c2a-be2e-
435505087fe4_1518060853002?action=train&tenant=Q3T1&
solution=vi&user=admin@example.com
" --data ' [{}]
```

## Notas

Este servicio prepara imágenes y realiza la formación en línea. Puede comprobar el estado de la instancia del modelo utilizando la llamada a la API "Obtener una instancia de modelo".

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Descargar un archivo de registro de formación

Descargue un archivo de registro de formación para la formación de un modelo. El administrador de modelos tiene autorización para utilizar esta API.

### URL

/ibm/iotm/vi/service/logFile

### Método

El GET del tipo de solicitud

### Parámetros de URL

`instanceId`: *Serie*. El ID de instancia del modelo. Obligatorio.

`user`: *Serie*. Identifica al usuario. Obligatorio.

`solution`: *Serie*. Se utiliza para identificar la solución. Obligatorio.

`tenant`: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

### Cabeceras

`APIKEY`: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

### Parámetros de datos

Ninguno.

### Cuerpo de ejemplo

Ninguno.

### Respuesta satisfactoria

Contenido binario del archivo de registro.

### Elementos de respuesta

Ninguno.

### Llamada de ejemplo

```
curl -k -H
"APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77d
```

```
4f0579e533dd34ggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5"
"https://9.112.229.91:9447/ibm/iotm/vi/service/logFile?instanceId=06c87
0e5-1701-419a-8a84-0f40ee82fe33_11260029297&user=auto_mm&solution=vi"
```

## Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Validar instancia del modelo

Validar una instancia del modelo con formación. El administrador de modelos tiene autorización para utilizar esta API.

### URL

/ibm/iotm/service/modelInstance/*modelInstanceId*

### Método

El PUT del tipo de solicitud

### Parámetros de URL

*modelInstanceId*: *Serie*. El ID de instancia del modelo. Obligatorio.

*action*: *Serie*. La acción. Se establece como *validate*. Obligatorio.

*user*: *Serie*. Identifica al usuario. Obligatorio.

*solution*: *Serie*. Se utiliza para identificar la solución. Obligatorio.

*tenant*: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

### Cabeceras

Content-type: application/json.

Content-Disposition: form-data; name="files[]"; filename="model.zip".

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

### Parámetros de datos

*validateData*: *Objeto JSON*. Datos de validación del modelo.

*groupId*: *Serie*. El ID del grupo de datos.

*fileIds*: *Serie*. El ID del archivo de datos.

### Cuerpo de ejemplo

```
[{"validateData": [{"fileIds": "9f7aa9d6-24d6-4611-8d4d-79a52d2bab02_76e8b2cb-057a-4904-8fad-bbca93abd80b", "groupId": "9f7aa9d6-24d6-4611-8d4d-79a52d2bab02"}, {"fileIds": "d84c028e-235a-4c13-8a77-894500b8a736_fe201c83-b138-4dca-b918-ae1b9f7db894", "groupId": "d84c028e-235a-4c13-8a77-894500b8a736"}, {"fileIds": "c5f11ed9-db49-403a-972c-cd6cb1d0f8eb_bd0c8246-7208-44d8-a2fa-e74c1578eebe", "groupId": "c5f11ed9-db49-403a-972c-cd6cb1d0f8eb"}] }
```

### Respuesta satisfactoria

```
200
[{"validateData": [{"fileIds": "9f7aa9d6-24d6-4611-8d4d-79a52d2bab02_76e8b2cb-057a-4904-8fad-bbca93abd80b", "groupId": "9f7aa9d6-24d6-4611-8d4d-79a52d2bab02"}, {"fileIds": "d84c028e-235a-4c13-8a77-894500b8a736_fe201c83-b138-4dca-b918-ae1b9f7db894", "groupId": "d84c028e-235a-4c13-8a77-894500b8a736"}, {"fileIds": "c5f11ed9-db49-403a-972c-cd6cb1d0f8eb_bd0c8246-7208-44d8-a2fa-e74c1578eebe", "groupId": "c5f11ed9-db49-403a-972c-cd6cb1d0f8eb"}], "status": "validating", "instanceId": "ef12294e-2d09-4acd-9946-00a09bdeeba2_1512092239371"}]
```

### Elementos de respuesta

*validateData*: *Objeto JSON*. Datos de validación del modelo.

groupId: *Serie*. El ID del grupo de datos.

fileIds: *Serie*. El ID del archivo de datos.

### Llamada de ejemplo

```
curl -k -H "APIKEY:8b79658225de53488321fb7bb657f9f161ac
d2ea830a5afb8c149f6aab2c77d4f0579e533dd34ggb959317ac69ff73f886fc3713bf30
5a158fdf485cc6f275f5" -H "Content-Type:application/json;charset=UTF-8" -X PUT
"https://9.112.229.91:9447/ibm/iotm/service/modelInstance/ef12294e-2d09-
4acd-9946-00a09bdeeba2_1512092239371?action=validate&user=auto_mm&solution=vi"
--data '[{"validateData": [{"fileIds": "9f7aa9d6-24d6-4611-8d4d-
79a52d2bab02_76e8b2cb-057a-4904-8fad-bbca93abd80b"}, {"groupId":
"9f7aa9d6-24d6-4611-8d4d-79a52d2bab02"}], {"fileIds":
"d84c028e-235a-4c13-8a77-894500b8a736_fe201c83-b138-4dca-b918-ae1b9f7db894"},
{"groupId": "d84c028e-235a-4c13-8a77-894500b8a736"}],
{"fileIds": "c5f11ed9-db49-403a-972c-cd6cb1d0f8eb_
bd0c8246-7208-44d8-a2fa-e74c1578eebe"},
{"groupId": "c5f11ed9-db49-403a-972c-cd6cb1d0f8eb"}]]'
```

### Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Rechazar instancia de modelo

Rechazar una instancia de modelo validada. El administrador de modelos tiene autorización para utilizar esta API.

### URL

/ibm/iotm/service/modelInstance/*modelInstanceId*

### Método

El PUT del tipo de solicitud

### Parámetros de URL

modelInstanceId: *Serie*. El ID de instancia del modelo. Obligatorio.

action: *Serie*. La acción. Se establece como reject. Obligatorio.

user: *Serie*. Identifica al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

### Cabeceras

Content-type: application/json.

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

### Parámetros de datos

Ninguno.

### Cuerpo de ejemplo

```
[{}]
```

### Respuesta satisfactoria

```
200
[
 {
 "status": "rejected",
 "instanceId": "e23b6796-4f4a-4c5d-8fb7-fa3f729f591c_1510812725969"
 }
]
```

## Elementos de respuesta

status: *Serie*. El estado de la instancia del modelo.

instanceId: *Serie*. El ID de instancia del modelo.

## Llamada de ejemplo

```
curl -k -H "APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77d4f0579e533dd34ggg959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5" -H "Content-Type:application/json; charset=UTF-8" -X PUT "https://9.112.229.91:9447/ibm/iotm/service/modelInstance/ef12294e-2d09-4acd-9946-00a09bdeeba2_1512092239371?action=reject&user=auto_mm&solution=vi" --data '[{}]'
```

## Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Desplegar instancia del modelo

Desplegar una instancia de modelo aceptada. El administrador de modelos tiene autorización para utilizar esta API.

### URL

/ibm/iotm/service/modelInstance/*modelInstanceId*

### Método

El PUT del tipo de solicitud

### Parámetros de URL

modelInstanceId: *Serie*. El ID de instancia del modelo. Obligatorio.

action: *Serie*. La acción. Se establece como publish. Obligatorio.

user: *Serie*. Identifica al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

Parameters: *Objeto JSON*. El ID de los sistemas perimetrales en los que se desplegará el modelo.

### Cabeceras

Content-type: application/json.

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

### Parámetros de datos

Ninguno

### Cuerpo de ejemplo

```
[{"parameters": {"edges": [{"id": "1508476898000"}, {"id": "1542102975873"}, ...]}}
```

### Respuesta satisfactoria

```
[{"status": "deployed", "instanceId": "96b651a9-2ba0-4392-aca4-1463320a19c2_1542354262146"}]
```

## Elementos de respuesta

status: *Serie*. El estado de la instancia del modelo.

instanceId: *Serie*. El ID de instancia del modelo.

parameters: *Objeto JSON*. El nombre y la dirección IP del sistema perimetral.

modelType: *Serie*. El tipo de modelo.

### Llamada de ejemplo

```
curl -k -H
"APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab
2c77d4f0579e533dd34ggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5"
-H "Content-Type:application/json;charset=UTF-8" -X PUT
"https://9.112.229.91:9447/ibm/iotm/service/modelInstance/ef12294e-
2d09-4acd-9946-00a09bdeeba2_1512092239371?action=publish&user=
auto_mm&solution=vi" --data '[{"parameters": {"edges":[{"id":
"1508476898000"}, {"id": "1542102975873"}, ...}]}'
```

### Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Volver a formar instancia de modelo

Volver a formar una instancia de modelo desplegada. El administrador de modelos tiene autorización para utilizar esta API.

### URL

/ibm/iotm/service/modelInstance/*modelInstanceId*

### Método

El PUT del tipo de solicitud

### Parámetros de URL

modelInstanceId: *Serie*. El ID de instancia del modelo. Obligatorio.

action: *Serie*. La acción. Se establece como *retrain*. Obligatorio.

user: *Serie*. Identifica al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

### Cabeceras

Content-Type: *application/json*.

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

### Parámetros de datos

retrainData: *objeto\_JSON*. Los datos de repetición de formación para el modelo. *model*.

groupId: *Serie*. El ID del grupo de datos.

fileIds: *Serie*. El ID del archivo de datos.

### Cuerpo de ejemplo

```
[{"trainData": [{"fileIds": "9f7aa9d6-24d6-4611-8d4d-79a52d2bab02_76e8b2cb-057a-4904-8fad-bbca93abd80b",
"groupId": "9f7aa9d6-24d6-4611-8d4d-79a52d2bab02" },
{"fileIds": "d84c028e-235a-4c13-8a77-894500b8a736_fe201c83-b138-4dca-b918-ae1b9f7db894",
"groupId": "d84c028e-235a-4c13-8a77-894500b8a736" },
{"fileIds": "c5f11ed9-db49-403a-972c-cd6cb1d0f8eb_bd0c8246-7208-44d8-a2fa-e74c1578eebe",
"groupId": "c5f11ed9-db49-403a-972c-cd6cb1d0f8eb"
}] }
```

### Respuesta satisfactoria

```
200
[{"modelId": "ef12294e-2d09-4acd-9946-00a09bdeeba2", "status":
"waiting", "instanceId": "ef12294e-2d09-4acd-9946-00a09bdeeba2_
```

```
1512095854903", "trainData": [{"fileIds": "9f7aa9d6-24d6-4611-8d4d-79a52d2bab02_76e8b2cb-057a-4904-8fad-bbca93abd80b", "groupId": "9f7aa9d6-24d6-4611-8d4d-79a52d2bab02"}, {"fileIds": "d84c028e-235a-4c13-8a77-894500b8a736_fe201c83-b138-4dca-b918-ae1b9f7db894", "groupId": "d84c028e-235a-4c13-8a77-894500b8a736"}, {"fileIds": "c5f11ed9-db49-403a-972c-cd6cb1d0f8eb_bd0c8246-7208-44d8-a2fa-e74c1578eebe", "groupId": "c5f11ed9-db49-403a-972c-cd6cb1d0f8eb"}]}
```

### Elementos de respuesta

modelId: *Serie*. ID del modelo.

status: *Serie*. El estado de la repetición de formación de la instancia del modelo.

instanceId: *Serie*. El ID de instancia del modelo.

trainData: *Objeto JSON*. Datos de formación del modelo.

groupId: *Serie*. El ID del grupo de datos.

fileIds: *Serie*. El ID del archivo de datos.

### Llamada de ejemplo

```
curl -k -H "APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77d4f0579e533dd34gggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5" -H "Content-Type:application/json;charset=UTF-8" -X PUT "https://9.112.229.91:9447/ibm/iotm/service/modelInstance/ef12294e-2d09-4acd-9946-00a09bdeeba2_1512092239371?action=retrain&user=auto_mm&solution=vi" --data '[{"trainData": [{"fileIds": "9f7aa9d6-24d6-4611-8d4d-79a52d2bab02_76e8b2cb-057a-4904-8fad-bbca93abd80b", "groupId": "9f7aa9d6-24d6-4611-8d4d-79a52d2bab02"}, {"fileIds": "d84c028e-235a-4c13-8a77-894500b8a736_fe201c83-b138-4dca-b918-ae1b9f7db894", "groupId": "d84c028e-235a-4c13-8a77-894500b8a736"}, {"fileIds": "c5f11ed9-db49-403a-972c-cd6cb1d0f8eb_bd0c8246-7208-44d8-a2fa-e74c1578eebe", "groupId": "c5f11ed9-db49-403a-972c-cd6cb1d0f8eb"}]}]'
```

### Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Anular despliegue de instancia de modelo

Anular una instancia de modelo desplegada. El administrador de modelos tiene autorización para utilizar esta API.

### URL

/ibm/iotm/service/modelInstance/*modelInstanceId*

### Método

El PUT del tipo de solicitud

### Parámetros de URL

modelInstanceId: *Serie*. El ID de instancia del modelo. Obligatorio.

action: *Serie*. La acción. Se establece como undeploy. Obligatorio.

user: *Serie*. Identifica al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

Parameters: *Objeto JSON*. El ID del sistema perimetral del que se anulará la instancia de modelo.

## Cabeceras

Content-Type: application/json.

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

## Parámetros de datos

Ninguno.

## Cuerpo de ejemplo

```
[{"parameters": {"edges": [{"id": "1508476898000"}]}}
```

## Respuesta satisfactoria

```
200
[{"status": "undeployed", "instanceId": "ef12294e-2d09-4acd-9946-00a09bdeeba2_1512092239371"}]
```

## Elementos de respuesta

status: *Serie*. El estado de la repetición de formación de la instancia del modelo.

instanceId: *Serie*. El ID de instancia del modelo.

## Llamada de ejemplo

```
curl -k -H "APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830
a5afb8c149f6aab2c77d4f0579e533dd34ggb959317ac69ff73f886fc3713bf305a158fdf485cc6
f275f5" -H "Content-Type:application/json;charset=UTF-8"
-X PUT "https://9.112.229.91:9447/ibm/iotm/service/modelInstance/ef12294e-2d09-
4acd-9946-00a09bdeeba2_1512092239371?action=undeploy&user=auto_mm&solution=vi"
--data '[{"parameters": {"edges": [{"id": "1508476898000"}]}]'
```

## Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Servicios de sistemas perimetrales

---

Los servicios de sistemas perimetrales le permiten realizar tareas como crear un sistema perimetral, obtener un sistema perimetral y actualizar un sistema perimetral.

### Crear sistema perimetral

Crear un sistema perimetral. El administrador de modelos tiene autorización para utilizar esta API.

#### URL

/ibm/iotm/vi/service/edge

#### Método

El POST del tipo de solicitud

#### Parámetros de URL

user: *Serie*. Identifica al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

#### Cabeceras

Content-Type: application/json.

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.



## Parámetros de datos

name: *Serie*. El nombre del sistema perimetral. Obligatorio.

ip: *IP*. La dirección IP del sistema perimetral. Obligatorio.

port: *Entero*. El puerto del sistema perimetral. Obligatorio.

userName: *Serie*. El nombre de usuario de la máquina virtual perimetral. Obligatorio.

passWord: *Serie*. La contraseña de la máquina virtual perimetral. Obligatorio.

clusterType: *Serie*. El tipo de clúster del sistema perimetral. Especifique slave o master. Obligatorio.

online: *Serie*. Indica si el sistema perimetral está en línea. Especifique true si el sistema perimetral está en línea o false si está fuera de línea. Obligatorio.

## Cuerpo de ejemplo

```
{ "name": "edge_auto", "ip": "9.112.229.54",
 "port": "22", "userName": "root", "passWord": "U2FtcGx1QDY2NiE=",
 "clusterType": "mater", "online": "true" }
```

## Respuesta satisfactoria

```
{ "port": "22", "clusterType": "master", "deployPath": "\\home\\pmqopsadmin", "hierarchyList":
 ["plant
 1, line1, cell1", "plant2, line2, cell2"], "servicePass": "passw0rd@", "deletePolicy":
 { "schedule": false, "rate
 ": 0.2, "onDay": 7, "afterDay": "1", "frequency": "weekly" }, "version": "2018Q4.01", "ip": "viedge1", "o
nline
 ": "true", "id": "1508476898000", "upgradeFlag": "false", "createdBy": "demoadm@cn.ibm.com", "servic
eUser":
 "admin", "name": "master", "sshUser": "pmqopsadmin", "updatedAt": "2018-11-16T03:54:13.382-06:00
 " }
```

## Elementos de respuesta

id: *Serie*. El ID del sistema perimetral.

deletePolicy: *Objeto JSON*. La política para eliminar los resultados de la inspección en el sistema perimetral.

version: *Serie*. La información de la versión del sistema perimetral.

upgradeFlag: *Serie*. Indica si el sistema perimetral necesita una actualización.

## Llamada de ejemplo

```
curl -k -H "APIKEY:8b79658225d
e53488321fb7bb657f9f161acd2ea830a5afb8c149
f6aab2c77d4f0579e533dd34ggb959317ac69ff
73f886fc3713bf305a158fdf485cc6f275f5" -H
"Content-Type:application/json;charset=UTF-8"
"https://9.112.229.91:9447/ibm/iotm/vi/service/edge?
user=auto_mm&solution=vi " --data '{ "name":
"edge_auto", "ip": "9.112.229.54", "port": "22",
"userName": "root", "passWord": "U2FtcGx1QDY2NiE=",
"clusterType": "master", "online": "true" }'
```

## Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Obtener sistema perimetral

Obtener un sistema perimetral. El administrador de modelos tiene autorización para utilizar esta API.

### URL

/ibm/iotm/vi/service/edge

## Método

El GET del tipo de solicitud

## Parámetros de URL

user: *Serie*. Identifica al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

hierarchyList: La información de la jerarquía de celdas, por ejemplo, plant1, line1, cell1. Opcional.

## Cabeceras

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

## Parámetros de datos

Ninguno.

## Cuerpo de ejemplo

Ninguno.

## Respuesta satisfactoria

```
200
[{"port": "22", "id": "1512098793912", "clusterType": "slave", "hierarchyList":
["plant1, line1, cell1"], "createdBy": "auto_mm", "description": "test edge",
"name": "edge_auto", "updatedAt": "2017-12-01T10:58:26.559+08:00", "online":
"false", "ip": "9.112.229.54"}]
```

## Elementos de respuesta

name: *Serie*. El nombre del sistema perimetral.

ip: *IP*. La dirección IP del sistema perimetral.

port: *Entero*. El puerto del sistema perimetral.

clusterType: *Serie*. El tipo de clúster del sistema perimetral. Especifique slave o master.

online: *Serie*. El estado en línea del sistema perimetral: true significa que el sistema perimetral está en línea y false que está fuera de línea.

hierarchyList: *Serie*. La jerarquía del sistema perimetral.

description: *Serie*. Una descripción del sistema perimetral.

createdBy: *Serie*. El propietario del sistema perimetral.

updatedAt: *Hora*. La última vez que se ha actualizado el sistema perimetral.

## Llamada de ejemplo

```
curl -k -H "APIKEY:8b79658225de53488321fb7bb657
f9f161acd2ea830a5afb8c149f6aab2c77d4f0579e533dd34ggb959317ac69ff
73f886fc3713bf305a158fdf485cc6f275f5" "https://9.112.229.91:9447/ibm/iotm/vi/
service/edge?user=auto_mm&solution=vi"
```

## Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Suprimir sistema perimetral

Suprimir un sistema perimetral. El administrador de modelos tiene autorización para utilizar esta API.

## URL

/ibm/iotm/vi/service/edge/edgeId

## Método

El DELETE del tipo de solicitud

## Parámetros de URL

edgeId: . El ID del sistema perimetral. Obligatorio.

user: *Serie*. Identifica al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

## Cabeceras

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

## Parámetros de datos

Ninguno.

## Cuerpo de ejemplo

[{}]

## Respuesta satisfactoria

```
200
[{}]
```

## Elementos de respuesta

Ninguno.

## Llamada de ejemplo

```
curl -k -X DELETE -H "APIKEY:8b7965
8225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab
2c77d4f0579e533dd34ggb959317ac69ff73f886fc3713bf305a
158fd485cc6f275f5" "https://9.112.229.91:9447/ibm/iotm/
vi/service/edge/1512098793912?user=auto_mm&solution=
vi" -d '[{}]
```

## Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Actualizar el sistema perimetral

Actualizar un sistema perimetral. Los usuarios con autoridad para crear un sistema perimetral tienen permiso para usar esta API.

Para actualizar un sistema perimetral, debe usar una llamada a la API para obtener los archivos del sistema perimetral necesarios, incluidos `edgeDeployed.sh` y `edge-vi.zip`.

Para actualizar un sistema perimetral en el lado del sistema perimetral, el usuario SSH especificado al crear el sistema perimetral debe ejecutar el script de shell en el sistema perimetral.

## URL

/ibm/iotm/vi/service/edgeFile

## Método

El GET del tipo de solicitud

## Parámetros de URL

edgeID: el ID del sistema perimetral. Obligatorio.

user: *Serie*. Identifica al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

Version: *Serie*. El valor puede ser `shell` para una descarga de script de shell, o `ubuntu`, `redhat`, o `power` para el tipo de Linux en el sistema perimetral.

#### Cabeceras

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

#### Parámetros de datos

Ninguno

#### Respuesta satisfactoria

```
200
```

Los archivos `edgeDeployed.sh` o `vi_edge-bin_vi.zip`.

#### Elementos de respuesta

Ninguno

#### Llamada de ejemplo

```
curl -k -X GET -H "APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77d4f0579e533dd34ggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5" "https://int_iotm.predictivesolutionsapps.ibmcloud.com/ibm/iotm/vi/service/edgeFile?tenant=Q3T1&solution=vi&user=modelmanager1&version=shell&edgeId=1508476898000"
```

#### Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Servicio de puntuación

---

El servicio de puntuación le permite puntuar una imagen.

### Puntuar imagen

Puntuar una imagen. El inspector y el supervisor tienen autorización para utilizar esta API.

#### URL

`/ibm/iotm/vi/service/uploadScoreImage`

#### Método

El POST del tipo de solicitud

#### Parámetros de URL

productType: *Serie*. El tipo de producto del modelo. Obligatorio.

cell: *Serie*. La celda de la imagen. Obligatorio.

user: *Serie*. Identifica al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

tenant: *Serie*. Se utiliza para identificar al arrendatario. Opcional.

#### Cabeceras

Content-Type: `application/binary`

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

#### Parámetros de datos

Ninguno.

#### Cuerpo de ejemplo

El contenido binario del archivo de imagen.

## Respuesta satisfactoria

```
200
//for object detection example:
{
 "detections": [
 {
 "position": {
 "height": 60,
 "width": 253,
 "x": 1594,
 "y": 773
 },
 "probableTypes": [
 {
 "confidence": 99.798703193664551,
 "type": "scratch"
 }
],
 "properties": []
 }
],
 "timestamp": "2017-09-28 10:32:13.528415"
}

//for classification example:
{
 "Timestamp": "2017-09-12 18:43:52",
 "detections": [
 {
 "position": {
 "height": 436,
 "width": 537,
 "x": 1574,
 "y": 1588
 },
 "probableTypes": [
 {
 "confidence": 100.0,
 "type": "Defect"
 },
 {
 "confidence": 0.0,
 "type": "MisT"
 },
 {
 "confidence": 0.0,
 "type": "NoDefect"
 }
]
 }
]
}
}
```

## Elementos de respuesta

position: *Matriz JSON*. La altura, anchura de la inspección junto con la información de y/x.

probableTypes: *Objeto JSON*. El nivel y tipo de confianza de cada defecto.

Timestamp: *Hora*. La hora en que se ha puntuado la imagen.

## Llamada de ejemplo

```
curl -k -X POST -H "APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77d4f0579e533dd34ggg959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5" -H "Content-Type: application/binary" "https://9.112.229.91:9447/ibm/iotm/vi/service/uploadScoreImage?user=auto_super&productType=auto&cell=plant1_line1_cell1&solution=vi" --connect-timeout 600 --data-binary @C:\code\Automation_98\API\vi\VI_API\file\test.JPG
```

## Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Servicio de integración de QEWS

---

El servicio de integración de QEWS le permite obtener un archivo de tasas de imágenes de defectos.

### Obtener archivo de tasa de imágenes de defectos

Obtener un archivo que contiene datos acerca de la tasa de imágenes de defectos. El supervisor tiene autorización para utilizar esta API.

#### URL

/ibm/iotm/service/defectImageRateFile

#### Método

El GET del tipo de solicitud

#### Parámetros de URL

**user:** *Serie.* Identifica al usuario. Obligatorio.

**solution:** *Serie.* Se utiliza para identificar la solución. Obligatorio.

**tenant:** *Serie.* Se utiliza para identificar al arrendatario. Opcional.

**date from:** La fecha de inicio con el formato aaaa-MM-dd o aaaa-MM-dd HH:mm:ss. Opcional.

**date to:** La fecha de finalización con el formato aaaa-MM-dd o aaaa-MM-dd HH:mm:ss. Opcional.

**cell name:** El nombre de la celda. Opcional.

**interval:** El intervalo de la tasa de imágenes de defectos. Especifique **diario**, **por horas** o **mensual**. Opcional.

#### Cabeceras

**APIKEY:** *clave cifrada.* La clave de la API utilizada para autenticación. Obligatorio.

#### Parámetros de datos

Ninguno.

#### Cuerpo de ejemplo

Ninguno.

#### Respuesta satisfactoria

200. El contenido binario del archivo de datos. El archivo de datos es un archivo CSV de valores separados por comas.

#### Elementos de respuesta

**ATTRIBUTE\_NAME:** VI\_DEFECT\_IMAGE\_RATE, un valor fijo.

**DATE:** La fecha con el formato aaaa-MM-dd cuando el intervalo es diario y mensual, y el formato aaaa-MM-dd HH:mm:ss cuando el intervalo es por horas.

**CELL\_NAME:** El nombre de la celda. En este elemento de respuesta, el carácter de coma (,) se sustituye por el carácter de subrayado (\_), debido a que no se admiten comas.

#### Llamadas de ejemplo

```
curl -k -H "APIKEY:8b79658225de53488
321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77d4f0579e533
dd34ggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5"
"https://9.112.229.89:9447/ibm/iotm/service/
defectImageRateFile?user=supervisor1&solution=
vi&tenant=T1" --output "C:\Users\IBM_ADMIN\Downloads\a.csv"
```

```
curl -k -H "APIKEY:8b79658225de53488
321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77d4f0579e533
dd34ggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5"
"https://9.112.229.89:9447/ibm/iotm/service/
```

```
defectImageRateFile?user=supervisor1&solution=
vi&tenant=T1&dateFrom=2017-10-23%2011:00:00&dateTo=
2017-11-14%2018:40:00&cellName=plant1,line1,cell1&
interval=hourly" --output "C:\Users\IBM_ADMIN\
Downloads\cell1_hourly.csv"
```

### Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Servicio compuesto

---

El servicio compuesto le permite obtener un archivo comprimido que contiene un modelo formado y registrar el modelo en Maximo PQI SaaS Visual Insights.

### Registrar modelo

Obtiene un archivo comprimido que contiene un modelo formado y registra el modelo en Maximo PQI SaaS Visual Insights. El administrador de modelos tiene autorización para utilizar esta API.

#### URL

/ibm/iotm/vi/service/registerModel

#### Método

El POST del tipo de solicitud

#### Parámetros de URL

user: *Serie*. Se utiliza para identificar al usuario. Obligatorio.

solution: *Serie*. Se utiliza para identificar la solución. Obligatorio.

name: *Serie*. El nombre del modelo.

productType: *Serie*. El tipo de producto asociado al modelo.

#### Cabeceras

Content-Type: multipart/form-data.

Content-Disposition: form-data; name="files[]"; filename="model.zip".

APIKEY: *clave cifrada*. La clave de la API utilizada para autenticación. Obligatorio.

#### Parámetros de datos

Ninguno

#### Cuerpo de ejemplo

Contenido binario del archivo de modelo.

#### Respuesta satisfactoria

200

```
{ "errorMessage": "", "modelId": "3914ad2c-3b7b-47ad-974c-e5b6bc2075ef",
 "createdBy": "modelmanager1", "status": "validated", "instanceId":
 "3914ad2c-3b7b-47ad-974c-e5b6bc2075ef_1517796906688",
 "updatedAt": "2018-02-05 10:15:06.608" }
```

#### Elementos de respuesta

modelId: *Serie*. ID del modelo.

createdBy: *Serie*. El propietario del modelo.

status: *Serie*. El estado de la instancia del modelo.

instanceId: *Serie*. El ID de la instancia del modelo.

updatedAt: *Hora*. La última vez que se ha actualizado la instancia del modelo.

### Llamada de ejemplo

```
curl -k -X POST -H
"APIKEY:8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77
d4f0579e533dd34ggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5" -H
"Content-Type:multipart/form-data" -H "Content-Disposition: form-data;
name="files[]"; filename="walkermodell.zip"
"https://int_iotm.predictivesolutionsapps.ibmcloud.com/ibm/iotm/vi/
service/registerModel?user=demoadm@cn.ibm.com&solution=vi&name=
samplename&productType=sampletype" --connect-timeout 6000 -F
file=@walkermodell.zip
```

### Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Servicios del sistema perimetral autónomo

---

Los servicios del sistema perimetral autónomo le permiten realizar tareas como obtener modelos disponibles, desplegar modelos, cargar puntuaciones de imágenes, sincronizar resultados de inspección limpiar resultados de inspección.

Los servicios del sistema perimetral autónomo se ejecutan en la máquina perimetral autónoma y no en el sistema central de Maximo PQI SaaS Visual Insights. Debe utilizar diferentes hosts, puertos y credenciales de usuario.

### Obtener los modelos disponibles

Obtiene todos los modelos disponibles. El administrador del sistema perimetral autónomo tiene autorización para utilizar esta API.

#### URL

https://<edge machine host>:8449/api/getAvailableModels

#### Método

El GET del tipo de solicitud

#### Parámetros de URL

Ninguno.

#### Cabeceras

Authorization: *nombre\_usuario/contraseña*

#### Parámetros de datos

Ninguno.

#### Cuerpo de ejemplo

Ninguno.

#### Respuesta satisfactoria

```
{ "model_list": [
 {
 "coviacenter": {
 "apikey": "8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77d
4f0579e533dd34ggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5",
 "credential": "ZGVtb2FkbUBjbi5pYm0uY29tO1EzVDE=",
 "url": "https://int_iotm.predictivesolutionsapps.ibmcloud.com/ibm/iotm/
service/apiWrapper?apiName=addInspectResult",
 "username": "demoadm@cn.ibm.com"
 },
 "groupIds": "0c4f11d7-b255-4c05-9c87-694e9f912fa9",
 "model_id": "049dabe5-d865-4f09-862f-fb7430afe6d",
 "model_instance_id": "049dabe5-d865-4f09-862f-fb7430afe6d_1518401501376",
 "model_type": "classification",
 "model_url": "049dabe5-d865-4f09-862f-fb7430afe6d_1518401599389.zip",
```



```
 "product_type": "Q3T1|jianyuan45",
 "status": "deployed"
 },

]}
```

### Elementos de respuesta

model\_list: *Serie*. La lista de modelos publicados.

model\_id: *Serie*. El ID de modelo del modelo publicado.

model\_instance\_id: *Serie*. El ID de instancia de modelo del modelo publicado.

model\_type: *Serie*. El tipo de modelo del modelo publicado.

model\_url: *Serie*. El URL de modelo del modelo publicado.

product\_type: *Serie*. El tipo de producto del modelo publicado.

coviacenter: *Objeto\_JSON*. La información central necesaria para el sistema perimetral.

### Llamada de ejemplo

```
curl -k -H 'Authorization: Basic YWRtaW46cGFzc3cwcmRA' -X GET https://{localhost}:8449/api/getAvailableModel
```

## Desplegar un modelo

Despliega un modelo. El administrador del sistema perimetral autónomo tiene autorización para utilizar esta API.

### URL

https://<edge machine host>:8449/api/deployModel

### Método

El POST del tipo de solicitud

### Parámetros de URL

Ninguno.

### Cabeceras

Authorization: *nombre\_usuario/contraseña*

### Parámetros de datos

model\_list: *Serie*. La lista de modelos publicados.

model\_id: *Serie*. El ID de modelo del modelo publicado.

model\_instance\_id: *Serie*. El ID de instancia de modelo del modelo publicado.

model\_type: *Serie*. El tipo de modelo del modelo publicado.

model\_url: *Serie*. El URL de modelo del modelo publicado.

product\_type: *Serie*. El tipo de producto del modelo publicado.

coviacenter: *Objeto\_JSON*. La información central necesaria para el sistema perimetral.

edgeId: *Serie*. El ID del sistema perimetral actual.

### Cuerpo de ejemplo

```
{ "model_id": "56608a3d-df64-4ef0-85a6-f88778cf583b", "model_url": "56608a3d-df64-4ef0-85a6-f88778cf583b_1540279487176.zip", "edgeId": "1540275842203", "coviacenter": { "url": "https://int_iotm.predictivesolutionsapps.ibmcloud.com/ibm/iotm/service/apiWrapper?apiName=addInspectResult", "credential": "c3VwZXJ2aXNvcjE6UTNUMQ==", "username": "supervisor1", "apikey": "8b79658225de53488321fb7bb657f9f161acd2ea830a5afb8c149f6aab2c77d4f0579e533dd34ggb959317ac69ff73f886fc3713bf305a158fdf485cc6f275f5"}, "groupIds": "b18a8d56-7a4c-4266-9280-49258c3fd798", "model_type": "classification",
```

```
"product_type": "Q3T1|qwtest12", "model_instance_id": "56608a3d-df64-4ef0-85a6-f88778cf583b_1540279416752", "model_name": "qwtest12"}
```

### Respuesta satisfactoria

```
{ "status": "success" }
```

### Elementos de respuesta

Status: *Serie*. Success (correcta) o failed (errónea).

### Llamada de ejemplo

```
curl -k -H 'Authorization: Basic YWRtaW46cGFzc3cwcmRA'
-H 'Content-Type:application/json;charset=UTF-8' -X POST --data
'{"model_id": "56608a3d-df64-4ef0-85a6-f88778cf583b", "model_url":
"56608a3d-df64-4ef0-85a6-f88778cf583b_1540279487176.zip", "edgeId":
"1540275842203", "coviacenter": { "url": "https://int_iotm.
predictivesolutionsapps.ibmcloud.com/ibm/iotm/service/apiWrapper?
apiName=addInspectResult", "credential": "c3VwZXJ2aXNvcjE6UTNUMQ==",
"username": "supervisor1", "apikey": "8b79658225de53488321fb7bb657f9f161
acd2ea830a5afb8c149f6aab2c77d4f0579e533dd34gg959317ac
69ff73f886fc3713bf305a158fdf485cc6f275f5" }, "groupIds": "b18a8d56-7a4c-
4266-9280-49258c3fd798", "model_type": "classification", "product_type":
"Q3T1|qwtest12", "model_instance_id": "56608a3d-df64-4ef0-85a6-f88778cf5
83b_1540279416752", "model_name": "qwtest12"}' https://{localhost}:
8449/api/deployModel
```

## Anular el despliegue de un modelo

Anula el despliegue de un modelo. El administrador del sistema perimetral autónomo tiene autorización para utilizar esta API.

### URL

https://<edge machine host>:8449/api/undeployModel

### Método

El POST del tipo de solicitud

### Parámetros de URL

Ninguno.

### Cabeceras

Authorization: *nombre\_usuario/contraseña*

### Parámetros de datos

model\_id: *Serie*. El ID de modelo del modelo cuyo despliegue se ha anulado.

model\_instance\_id: *Serie*. El ID de instancia de modelo del modelo cuyo despliegue se ha anulado.

### Cuerpo de ejemplo

```
{"model_id": "5868dcd7-7032-4101-9ba8-797dea9005aa", "model_instance_id": "5868dcd7-
7032-4101-9ba8-797dea9005aa_1523253620387"}
```

### Respuesta satisfactoria

```
{ "status": "success" }
```

### Elementos de respuesta

Status: *Serie*. Success (correcta) o failed (errónea).

### Llamada de ejemplo

```
curl -k -H 'Authorization: Basic YWRtaW46cGFzc3cwcmRA'
-H 'Content-Type:application/json;charset=UTF-8' -X POST --data
'{"model_id": "8e474fb5-97dd-4f5d-8415-14c72855c6a5", "model_instance_id":
```

```
"8e474fb5-97dd-4f5d-8415-14c72855c6a5_1532404443311"}' https://{local
host}:8449/api/undeployModel
```

## Cargar y puntuar imagen en el sistema perimetral

Carga y puntuá una imagen en el sistema perimetral. La línea de producción o los servicios externos utilizan normalmente esta API.

### URL

/api/uploadScoreImage

### Método

El POST del tipo de solicitud

### Parámetros de URL

productType: *Serie*. El tipo de producto del modelo. Obligatorio.

cell: *Serie*. La celda de la imagen. Obligatorio.

### Cabeceras

Content-Type: application/binary.

Authorization: . El código de autorización base del servicio de sistema perimetral. Obligatorio.

### Parámetros de datos

Ninguno.

### Cuerpo de ejemplo

El contenido binario del archivo de imagen.

### Respuesta satisfactoria

Ejemplo de detección de objetos:

```
200
{
 "detections": [
 {
 "position": {
 "height": 60,
 "width": 253,
 "x": 1594,
 "y": 773
 },
 "probableTypes": [
 {
 "confidence": 99.798703193664551,
 "type": "scratch"
 }
],
 "properties": []
 }
],
 "timestamp": "2017-09-28 10:32:13.528415"
}
```

Ejemplo de clasificación:

```
200
{
 "Timestamp": "2017-09-12 18:43:52",
 "detections": [
 {
 "position": {
 "height": 436,
 "width": 537,
 "x": 1574,
 "y": 1588
 },
 "probableTypes": [
 {

```

```

 "confidence": 100.0,
 "type": "Defect"
 },
 "confidence": 0.0,
 "type": "MisT"
},
"confidence": 0.0,
"type": "NoDefect"
}
]
}

```

### Llamada de ejemplo

```

curl -k --cert /home/pmqopsadmin/vi_edge-bin_vi/vi_edge/https/cert.pem --key
/home/pmqopsadmin/vi_edge-bin_vi/vi_edge/https/key.pem -H 'Authorization:
Basic autorización' -T 'filePath' -X POST https://localhost
:8449/api/uploadScoreImage?productType=tipoProducto&cell=celda

```

El valor de *autorización* es la autorización base para acceder al servicio de sistema perimetral. El valor de *filePath* es la vía de acceso del archivo que desea cargar. El valor de *localhost* es la dirección IP en la que se despliega el sistema perimetral. El valor de *tipoProducto* es el tipo de producto del modelo que ha utilizado para puntuar la imagen que ha cargado. El valor de *celda* es la celda a la que desea que se envíe el resultado de la puntuación.

### Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Sincronizar el resultado de inspección del sistema perimetral con la aplicación central

Sincroniza el resultado de inspección del sistema perimetral con la aplicación central. El administrador del sistema perimetral autónomo tiene autorización para utilizar esta API.

### URL

/api/syncInspectResult

### Método

El POST del tipo de solicitud

### Parámetros de URL

Ninguno

### Cabeceras

Content-Type: application/json.

Authorization:. El código de autorización base del servicio de sistema perimetral. Obligatorio.

### Parámetros de datos

Ninguno.

### Cuerpo de ejemplo

```

{
 "percentage":100
}

```

### Respuesta satisfactoria

```

200
{
 "message": "success"
}

```

## Llamada de ejemplo

```
curl -k --cert /home/pmqopsadmin/vi_edge-bin_vi/vi_edge/https/cert.pem --key /home/pmqopsadmin/vi_edge-bin_vi/vi_edge/https/key.pem -H "Content-Type: application/json" -H 'Authorization: Basic YWRtaW46cGFzc3cwcmRA' -X POST --data '{"porcentaje":porcentaje}' https://localhost:8449/api/syncInspectResult
```

El valor de *porcentaje* es el porcentaje de resultados de inspección que desea enviar. El valor de *localhost* es la dirección IP en la que se despliega el sistema perimetral.

## Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.

## Limpiar el resultado de inspección que ha sincronizado con la aplicación central

Limpia el resultado de una inspección que ha sincronizado con la aplicación central. El administrador del sistema perimetral autónomo tiene autorización para utilizar esta API.

## URL

/api/cleanInspectResult

## Método

El POST del tipo de solicitud

## Parámetros de URL

Ninguno

## Cabeceras

Content-Type: application/json.

Authorization: . El código de autorización base del servicio de sistema perimetral. Obligatorio.

## Parámetros de datos

Ninguno.

## Cuerpo de ejemplo

```
{
 "fromDayBefore":1
}
```

## Respuesta satisfactoria

```
{
 "message": "success"
}
```

## Llamada de ejemplo

```
curl -k --cert /home/pmqopsadmin/vi_edge-bin_vi/vi_edge/https/cert.pem --key /home/pmqopsadmin/vi_edge-bin_vi/vi_edge/https/key.pem -H "Content-Type: application/json" -H 'Authorization: Basic YWRtaW46cGFzc3cwcmRA' -X POST --data '{"fromDayBefore":numeroDías}' https://localhost:8449/api/cleanInspectResult
```

El valor de *numeroDías* es el número de días que desea conservar los datos. Por ejemplo, si establece *numeroDías* en 0, se eliminarán todos los datos anteriores al día de hoy. Si establece *numeroDías* en 1, se eliminarán todos los datos anteriores al día de ayer. El valor de *localhost* es la dirección IP en la que se despliega el sistema perimetral.

## Notas

El sistema utiliza la APIKEY para la autenticación. Si no proporciona una APIKEY, el sistema rechaza su solicitud.



---

# Capítulo 8. Registrar, desplegar y probar un modelo utilizando la API

La función Guía de la API es una interfaz web que le guía en un caso de uso de la API común: el proceso de registrar, desplegar y probar un modelo formado. La Guía de la API automatiza muchos pasos del proceso para facilitar la finalización de las tareas.

## Acerca de esta tarea

El primer paso es registrar un modelo. Puede importar un archivo comprimido que contiene un modelo formado y registrarlo en el producto.

El segundo paso es desplegar el modelo. Puede desplegar un modelo en sistemas perimetrales, de modo que se pueda utilizar el modelo para inspeccionar las imágenes.

El tercer paso es probar el modelo. Puede utilizar sus propias imágenes para probar el modelo desplegado y revisar el resultado de la puntuación.

## Procedimiento

1. Seleccione **Guía de la API** en el menú desplegable.
2. Pulse **Intentar registrar modelo** para entrar en la página **Guía de la API completa**.
3. Escriba el nombre de modelo y el tipo de producto.
4. Pulse **Seleccionar** para seleccionar un archivo comprimido que contiene un modelo formado.
5. Pulse **Registrar** para registrar el modelo en el producto.  
El resultado del registro se muestra en el recuadro **Resultado**.
6. Pulse **Siguiente**.
7. Si es necesario, escriba el ID de instancia.
8. Pulse **Desplegar** para desplegar el modelo.  
El resultado del despliegue se muestra en el recuadro **Resultado**.
9. Pulse **Siguiente**.
10. Escriba el tipo de producto y la información de celda, seleccione una imagen y pulse **Puntuar** para iniciar el proceso de puntuación.  
El resultado de la puntuación se muestra en el recuadro **Resultado**.





## Capítulo 9. Resolución de problemas

Revise los archivos de registro y mensajes de error para ayudar a resolver los problemas del producto.

### Archivos de registro

Consulte los archivos de registro del producto para obtener más información sobre las condiciones de error.

Archivo de registro	Descripción
/via_acceso_Liberty/usr/servers/VICenterServer/logs/messages.log	Mensajes de WebSphere Application Server Liberty
/home/nombre_usuario/edgeDeploy_nombre_arrendatario.log	Mensajes de sistemas perimetrales
/home/nombre_usuario/vi_edge-bin_vi/vi_task_manager/master.log	Mensajes de gestor de tareas
/home/nombre_usuario/vi_edge-bin_vi/vi_edge/Service.log	Mensajes de servicio

## Mensajes

### Mensajes de sistemas perimetrales

Los siguientes mensajes aparecen durante las condiciones de error de los sistemas perimetrales.

Mensaje	Acción
El sistema perimetral maestro ya existe, sólo se soporta un sistema perimetral maestro.	Especifique un nombre de sistema perimetral que no se utilice.
Los campos necesarios no deben estar vacíos.	Incluya un valor en todos los campos necesarios.
No se puede conectar a la máquina. Utilice la dirección IP, el nombre de usuario SSH y la contraseña correctos.	Verifique que la dirección IP, el nombre de usuario SSH y la contraseña son valores válidos.
Para gestionar el sistema perimetral, debe tener los permisos correctos.	Asegúrese de que ha iniciado la sesión como un usuario con permisos de gestor de modelos. Si no puede resolver el problema, póngase en contacto con un administrador del sistema.
La dirección IP no es válido. Especifique la dirección IP correcta, por ejemplo 10.172.0.23.	Verifique el valor de la dirección IP.
El sistema perimetral no se ha creado debido a un error de base de datos.	Consulte el servicio Apache HBase de la consola de Ambari.
Existe un sistema perimetral con la misma dirección IP.	No puede crear dos sistemas perimetrales en el mismo sistema. Utilice la dirección IP de otro sistema para crear un nuevo sistema perimetral.

Tabla 4. Mensajes de error para los sistemas perimetrales (continuación)

Mensaje	Acción
No se han podido cargar scripts en la máquina de sistema perimetral. Compruebe la existencia y los permisos de la vía de acceso de despliegue.	Verifique que la vía de acceso de despliegue existe y tiene los permisos correctos.
No se han podido iniciar los servicios en la máquina perimetral.	Asegúrese de que el servicio se está ejecutando en el sistema perimetral maestro. Consulte el archivo de registro en el sistema perimetral.
No se han podido ejecutar scripts en la máquina de sistema perimetral.	Compruebe el archivo de registro del sistema perimetral.
El nombre del sistema perimetral es un duplicado. Utilice un nombre de sistema perimetral diferente.	Utilice un nombre diferente para crear un nuevo sistema perimetral.
No se puede crear el sistema perimetral esclavo si no existe un sistema perimetral maestro.	Debe crear un sistema perimetral maestro antes de crear un sistema perimetral esclavo.
No se ha podido suprimir el sistema perimetral porque lo utiliza el modelo "nombre_modelo".	El sistema perimetral no se puede suprimir si lo utiliza una versión de modelo. Detenga todas las versiones de modelo que se están ejecutando en el sistema perimetral antes de suprimir el sistema perimetral.
El sistema perimetral no se ha suprimido debido a un error de base de datos.	Consulte el servicio Apache HBase de la consola de Ambari.
No puede suprimir el sistema perimetral maestro porque hay sistemas perimetrales esclavos existentes.	El sistema perimetral no se puede suprimir si aún tiene sistemas perimetrales esclavos. Suprima todos los sistemas perimetrales esclavos antes de suprimir el sistema perimetral maestro.
No se ha podido suprimir el sistema perimetral del repositorio maestro: nombre_repositorio.	Asegúrese de que el servicio se está ejecutando en el sistema perimetral maestro. Consulte los archivos de registro de servicio.

### Mensajes de archivo de imagen comprimido

Los siguientes mensajes aparecen durante las condiciones de error para los archivos de imagen comprimidos.

Tabla 5. Mensajes de error para archivos de imagen comprimidos

Mensaje	Acción
El archivo comprimido que ha cargado no contiene una imagen en la ubicación esperada.	El archivo de imagen comprimido cargado debe contener imágenes. Todas las imágenes del archivo de imagen comprimido debe estar en una estructura plana sin subdirectorios.
El archivo no es válido. Sólo debe cargar archivos .zip.	El único formato de archivo soportado es .zip.
Faltan los archivos XML de anotación en el archivo comprimido cargado.	Añada los archivos XML de anotación al archivo de imagen comprimido y, a continuación, cárguelo de nuevo.
Falta el archivo labels.txt en el archivo comprimido cargado.	Añada el archivo labels.txt en el archivo de imagen comprimido y, a continuación, cárguelo de nuevo.

Tabla 5. Mensajes de error para archivos de imagen comprimidos (continuación)

Mensaje	Acción
La entrada no es válida. Debe especificar el ID.	No se ha proporcionado ningún ID de archivo en la solicitud de API. Proporcione el ID de archivo.
La entrada no es válida. Se puede suprimir un número máximo de 100 entidades.	No puede suprimir más de 100 archivos comprimidos al mismo tiempo.
No se ha podido suprimir el archivo de datos.	El archivo de datos no se ha podido suprimir.
El archivo de datos especificado se ha citado en el modelo <i>nombre_modelo</i> . No se puede suprimir.	Un modelo hace referencia al archivo de imagen comprimido. No se puede suprimir.

### Mensaje de grupo de imagen

Los siguientes mensajes aparecen durante las condiciones de error para los grupos de imagen.

Tabla 6. Mensajes de error para grupos de imagen

Mensaje	Acción
El nombre de grupo no puede estar vacío cuando se crea un nuevo grupo de datos.	Especifique un nombre para el nuevo grupo.
Existen archivos de datos para el grupo de datos seleccionado. La actualización ha fallado. Asegúrese de que el grupo de datos seleccionado no contiene archivos de datos.	Si el grupo de imagen se crea con archivos de datos, el tipo del grupo de imagen no se puede actualizar. Si el grupo de imagen no se crea con archivos de datos, este error puede producirlo un error de operación de Apache HBase.
No se ha podido crear un grupo de datos porque faltan parámetros o éstos tienen un formato no válido.	Asegúrese de que se especifican parámetros para el nuevo grupo de datos y que los parámetros están en un formato válido.
El nombre de grupo es un duplicado. Utilice un nombre de grupo diferente.	Especifique un nombre exclusivo para el grupo nuevo.
Error al crear el grupo de datos.	Se ha producido un error de Apache HBase mientras se estaba creando el grupo de datos. Consulte el servicio Apache HBase de la consola de Ambari.
El grupo de datos no existe.	Falta el ID de grupo en la solicitud o dicho ID no es válido.
Error al suprimir los archivos de datos en un grupo de datos especificado.	Se ha producido un error de Hadoop Distributed File System (HDFS) mientras se estaban suprimiendo archivos de datos.
El grupo de datos especificado se ha citado en un modelo. No se puede suprimir.	Si un grupo se utiliza en un modelo, el grupo no se puede suprimir.
Error al suprimir grupo de datos.	Se ha producido un error de Apache HBase mientras se estaba suprimiendo el grupo de datos. Consulte el servicio Apache HBase de la consola de Ambari.

### Mensajes de modelo

Los siguientes mensajes aparecen durante las condiciones de error para los modelos.

Tabla 7. Mensajes de error para modelos

Mensaje	Acción
El nombre de modelo no puede estar vacío cuando se crea un nuevo modelo.	Especifique un nombre para el modelo.
El nombre de modelo debe tener menos de 128 caracteres y sólo debe contener letras, números, espacios y subrayados.	Especifique un nombre de modelo válido.
El tipo de producto no puede estar vacío cuando se crea un nuevo modelo.	Especifique el tipo de producto.
El tipo de producto debe tener menos de 128 caracteres y sólo debe contener letras, números y espacios.	Especifique un tipo de producto válido.
Los ID de grupo de datos no pueden estar vacíos cuando se crea un nuevo modelo.	Especifique un valor para <code>groupIds</code> en la llamada de API.
El formato de datos no puede estar vacío cuando se crea un modelo nuevo.	Especifique un valor para <code>dateFormat</code> en la llamada de API.
La política de repetición de formación debe tener un formato válido cuando se crea un nuevo modelo.	Especifique un valor válido para <code>retrainPolicy</code> en la llamada de API.
El tipo de modelo no puede estar vacío cuando se crea un nuevo modelo.	Especifique un valor para <code>modelType</code> en la llamada de API.
El tipo de modelo debe ser clasificación o detección de objetos.	Especifique un valor de <code>classification</code> o <code>objectDetection</code> para <code>modelType</code> en la llamada de API.
Los parámetros deben tener un formato válido cuando se crea un nuevo modelo.	Especifique un valor válido para <code>parameters</code> en la llamada de API.
No se ha podido crear el modelo porque el valor <code>trainParam</code> en los parámetros tiene un valor no válido.	Especifique un valor válido para <code>trainParam</code> en la llamada de API.
Los ID de grupo de datos no son válidos.	Especifique un valor válido para <code>groupIds</code> en la llamada de API.
Se ha producido un error porque los grupos de datos deben ser todos de varios tipos de característica o todos de tipos de característica únicos.	Los grupos de <code>groupIds</code> de la llamada de API deben tener el mismo tipo de grupo. característica única o varias características.
No se ha podido crear el modelo.	Se ha producido un error de Apache HBase durante la creación del modelo. Consulte el servicio Apache HBase de la consola de Ambari.
La entrada no es válida. Se puede suprimir un número máximo de 100 entidades.	Especifique menos de 100 modelos que se vayan a suprimir.
La entrada no es válida. Debe especificar el ID.	Incluya el ID de modelo en la solicitud.
El modelo no se puede actualizar porque el ID de modelo está vacío. Especifique un ID de modelo.	Incluya el ID de modelo en la solicitud.
No se han podido actualizar los grupos de modelos porque la versión de modelo no está en estado Borrador.	Asegúrese de que los modelos que se deben actualizar están en estado de borrador.

Tabla 7. Mensajes de error para modelos (continuación)

Mensaje	Acción
No se han podido actualizar los grupos de modelos porque el valor isHybrid de los grupos de datos es diferente.	No puede cambiar el tipo de modelo después de crear el modelo.

### Mensajes de instancia de modelo

Los siguientes mensajes aparecen durante las condiciones de error para las instancias de modelo.

Tabla 8. Mensajes de error para instancias de modelo

Mensaje	Acción
El ID de modelo no puede estar vacío cuando se crea una nueva instancia de modelo.	Cree una instancia de modelo después de que se haya creado su modelo correspondiente.
Los datos de formación no pueden estar vacíos cuando se crea una nueva instancia de modelo.	Especifique un valor válido para trainData en la llamada de API.
No se ha podido registrar el modelo en el servidor de formación.	Consulte el servicio que se está ejecutando en el servidor de formación.
No se ha podido crear el modelo porque el valor trainParam en los parámetros tiene un valor no válido.	Especifique un valor válido para trainParam en la llamada de API.
Los datos de formación no son válidos. Debe existir al menos un grupo de datos.	Asegúrese de que cada grupo de datos que pertenece a la instancia tenga archivos de datos válidos.
Los datos de formación no son válidos. Compruebe la proporción de formación y validación para asegurarse de que al menos existe una imagen en el conjunto de formación y prueba para cada grupo de datos.	Asegúrese de que al menos existe una imagen en los conjuntos de formación y pruebas para cada grupo de datos.
Tiene $n$ modelos en la cola de formación. No puede enviar una nueva solicitud de formación hasta que se hayan completado o cancelado.	Cancele trabajos o espere a que finalicen los trabajos que se están ejecutando actualmente.
No se ha podido cancelar la formación.	Consulte el servicio Apache HBase de la consola de Ambari.
El cuerpo de entrada no es válido.	Compruebe la entrada de solicitud y, a continuación, vuelva a intentar la operación.
El modelo <i>nombre_modelo</i> tiene un estado de retrotracción debido a que ha excedido el tiempo de espera de validación.	Compruebe el registro en el servidor de validación. Para el modelo de clasificación, el registro es /home/pmqopsadmin/vi_edge-bin_vi/vi_score_engine_restful/back.log. Para el modelo de detección de objetos, el registro es /home/pmqopsadmin/vi_obj_detection_retrain/RESTAPI/model/frcnn_log.txt o ssd_log.txt. Corrija cualquier problema y, a continuación, vuelva a intentar la operación.

Tabla 8. Mensajes de error para instancias de modelo (continuación)

Mensaje	Acción
No se ha podido validar la versión de modelo.	Compruebe el registro en el servidor de validación. Para el modelo de clasificación, el archivo de registro es /home/pmqopsadmin/vi_edge-bin_vi/vi_score_engine_restful/back.log. Para el modelo de detección de objetos, el archivo registro es /home/pmqopsadmin/vi_obj_detection_retrain/RESTAPI/model/frcnn_log.txt o ssd_log.txt. Corrija cualquier problema y, a continuación, vuelva a intentar la operación.
El modelo no existe.	No se ha encontrado ningún modelo con este ID de modelo. Compruebe el ID de modelo.
No se ha podido rechazar la versión de modelo.	Consulte el servicio Apache HBase de la consola de Ambari.
No se ha podido desplegar la versión de modelo porque el archivo de modelo está vacío.	Se ha producido un error cuando creaba el modelo. Cree un nuevo modelo.
No hay ningún sistema perimetral maestro para desplegar el modelo.	Cree un sistema perimetral maestro antes de desplegar el modelo.
La IP <i>dirección_IP</i> del sistema perimetral no válida, sólo se puede utilizar un sistema perimetral maestro para desplegar el modelo.	Utilice la dirección IP correcta para el sistema perimetral maestro para desplegar un modelo.
No puede desplegar el modelo en el sistema perimetral debido a que el modelo <i>nombre_modelo</i> se ha desplegado con el mismo tipo de producto. No puede desplegar un modelo en un sistema perimetral cuando otro modelo con el mismo tipo de producto se está ejecutando en el sistema perimetral. Primero debe anular el despliegue del otro modelo.	No puede desplegar un modelo en un sistema perimetral cuando otro modelo con el mismo tipo de producto se está ejecutando en el sistema perimetral.
Error de base de datos al crear el archivo de datos.	Consulte el servicio Apache HBase de la consola de Ambari.
El archivo <i>nombre_archivo</i> no se ha cargado en el almacenamiento.	Compruebe Hadoop Distributed File System.
No se ha podido actualizar el grupo de datos debido a un error de base de datos.	Consulte el servicio Apache HBase de la consola de Ambari.
El estado de la instancia de modelo no soporta esta acción.	Asegúrese de que la instancia de modelo está en el estado correcto y, a continuación, vuelva a intentar la acción.
Los datos de formación no son válidos. Debe existir al menos un archivo de datos en cada grupo de datos.	Asegúrese de que cada grupo de datos que pertenece a la instancia de modelo tiene archivos de datos válidos.
Los datos de formación no son válidos. Compruebe la proporción de formación y validación para asegurarse de que al menos existe una imagen en el conjunto de formación y prueba para cada grupo de datos.	Asegúrese de que al menos existe una imagen en los conjuntos de formación y pruebas para cada grupo de datos.

<i>Tabla 8. Mensajes de error para instancias de modelo (continuación)</i>	
<b>Mensaje</b>	<b>Acción</b>
No se podido repetir la formación del modelo debido a un error de base de datos.	Consulte el servicio Apache HBase de la consola de Ambari.
No se ha podido desplegar la versión de modelo en el sistema perimetral <i>nombre_sistema_perimetral</i> .	Asegúrese de que el servicio se esté ejecutando en el sistema perimetral maestro. Consulte el archivo <code>Service.log</code> en el sistema perimetral maestro.
No se ha podido anular el despliegue de la versión de modelo del sistema perimetral <i>nombre_sistema_perimetral</i> .	Asegúrese de que el servicio se esté ejecutando en el sistema perimetral maestro. Consulte el archivo <code>Service.log</code> en el sistema perimetral maestro.
No se ha podido anular el despliegue de la versión de modelo debido a un error de base de datos.	Consulte el servicio Apache HBase de la consola de Ambari.
No se ha podido cancelar la formación.	Consulte el servicio Apache HBase de la consola de Ambari.
El valor de instantánea no es válido. No se puede utilizar la instantánea.	El valor de instantánea no existe o no es válido.
El archivo de instantánea no existe en el servidor de formación.	Sáltese la instantánea o vuelva a formar el modelo.
La instancia de modelo no se puede suprimir porque el ID de instancia está vacío. Especifique un ID de instancia.	Especifique un ID de instancia válido.
No se ha podido suprimir la instancia de modelo debido a un error de base de datos.	Consulte el servicio Apache HBase de la consola de Ambari.
La instancia de modelo con el estado de <i>estado</i> no se puede suprimir.	Una instancia de modelo con el estado especificado no se puede suprimir.

### **Mensajes de guía de API**

Los siguientes mensajes aparecen durante las condiciones de error para la guía de API.

<i>Tabla 9. Mensajes de error para la guía de API</i>	
<b>Mensaje</b>	<b>Acción</b>
No se ha encontrado ningún modelo para el tipo de producto especificado.	Si este error se produce como resultado de una llamada de API, confirme que ha enviado el tipo de producto correcto. Si este error se produce en la interfaz de usuario, intente otro tipo de producto.
No hay información del sistema perimetral en el que se ha desplegado el modelo.	Verifique que el modelo correspondiente se ha desplegado.
No se ha podido transferir la imagen al sistema perimetral <i>nombre_sistema_perimetral</i> .	Asegúrese de que el servicio se está ejecutando en el sistema perimetral maestro. Consulte el archivo <code>Service.log</code> en el sistema perimetral maestro.
No se ha podido obtener el resultado de puntuación de imagen del sistema perimetral <i>nombre_sistema_perimetral</i> .	Asegúrese de que el servicio se está ejecutando en el sistema perimetral maestro. Consulte el archivo <code>Service.log</code> en el sistema perimetral maestro.

Tabla 9. Mensajes de error para la guía de API (continuación)

Mensaje	Acción
Se ha producido una problema de tiempo de espera excedido. No se ha podido obtener el resultado de puntuación de imagen del sistema perimetral <i>nombre_sistema_perimetral</i> .	Asegúrese de que el servicio se está ejecutando en el sistema perimetral maestro. Consulte el archivo <i>Service.log</i> en el sistema perimetral maestro.
El nombre de modelo no puede estar vacío cuando se crea un nuevo modelo.	Especifique un nombre para el modelo.
El nombre de modelo debe tener menos de 128 caracteres y sólo debe contener letras, números, espacios y subrayados.	Especifique un nombre de modelo válido.
El tipo de producto no puede estar vacío cuando se crea un nuevo modelo.	Especifique el tipo de producto.
El tipo de producto debe tener menos de 128 caracteres y sólo debe contener letras, números y espacios.	Especifique un tipo de producto válido.
El nombre de modelo es un duplicado. Utilice un nombre de modelo diferente.	Especifique un nombre exclusivo para el modelo.
El archivo <i>nombre_archivo</i> no se ha cargado en el almacenamiento.	Compruebe Hadoop Distributed File System.
No se ha podido cargar el archivo de modelo.	Compruebe la conexión de red y, a continuación, vuelva a intentar la carga.

### Mensajes de simulador

Los siguientes mensajes aparecen durante las condiciones de error para el simulador.

Tabla 10. Mensajes de error para el simulador

Mensaje	Acción
No se han podido enviar las imágenes simuladas.	Asegúrese de que el servicio se está ejecutando en el sistema perimetral maestro. Compruebe el archivo <i>Service.log</i> .



## Avisos

---

Esta información se ha desarrollado para productos y servicios que se ofrecen en los Estados Unidos. Este material puede estar disponible por parte de IBM en otros idiomas. Sin embargo, es posible que tenga obligación de tener una copia del producto o de la versión del producto en dicho idioma para acceder a él.

Puede que IBM no ofrezca en algunos países los productos, servicios o características que se explican en este documento. Póngase en contacto con el representante local de IBM para obtener información sobre los productos y servicios disponibles actualmente en su área. Cualquier referencia hecha a un producto, programa o servicio de IBM no implica ni pretende indicar que sólo pueda utilizarse ese producto, programa o servicio de IBM. En su lugar, se puede utilizar cualquier producto, programa o servicio funcionalmente equivalente que no infrinja ninguno de los derechos de propiedad intelectual de IBM. Sin embargo, será responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier programa, producto o servicio que no sea de IBM.

Puede que IBM tenga patentes o solicitudes de patentes pendientes relacionadas con los temas principales que se tratan en este documento. La posesión de este documento no le otorga ninguna licencia sobre estas patentes. Puede enviar sus consultas sobre licencias, por escrito, a:

*IBM Director of Licensing  
IBM Corporation  
North Castle Drive, MD-NC119  
Armonk, NY 10504-1785  
EE.UU.*

Para consultas sobre licencias en las que se solicite información sobre el juego de caracteres de doble byte (DBCS), póngase en contacto con el departamento de propiedad intelectual de IBM de su país o envíe las consultas, por escrito, a:

*Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokio 103-8510, Japón*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL" SIN GARANTÍA DE NINGÚN TIPO, NI EXPLÍCITAS NI IMPLÍCITAS, INCLUIDAS, PERO SIN LIMITARSE A, LAS GARANTÍAS IMPLÍCITAS DE NO INFRACCIÓN, COMERCIALIZACIÓN O IDONEIDAD PARA UN PROPÓSITO CONCRETO. Algunas legislaciones no contemplan la exclusión de garantías, ni implícitas ni explícitas, por lo que puede haber usuarios a los que no afecte esta declaración.

Esta información podría incluir imprecisiones técnicas o errores tipográficos. Periódicamente se realizan cambios en la información que aquí se presenta; tales cambios se incorporarán en nuevas ediciones de la publicación. IBM podría realizar mejoras y/o cambios en el producto o productos y/o programa o programas que se describen en esta publicación en cualquier momento y sin previo aviso.

Cualquier referencia en este documento a sitios web que no son de IBM se proporciona únicamente para su comodidad y no significa en modo alguno que se recomiende visitar dichos sitios web. Los materiales de dichos sitios web no forman parte de los materiales para este producto de IBM y el uso de estos sitios web solo es responsabilidad del cliente.

IBM puede utilizar o distribuir la información que el cliente le suministre del modo que considere conveniente, sin incurrir por ello en ninguna obligación para con el cliente.

Los titulares de licencias de este programa que deseen obtener información acerca de este con el fin de permitir: (i) el intercambio de información entre programas creados independientemente y otros programas (incluido el presente) y (ii) la utilización mutua de la información que se ha intercambiado, deben ponerse en contacto con:

*IBM Director of Licensing  
IBM Corporation  
North Castle Drive, MD-NC119  
Armonk, NY 10504-1785  
EE.UU.*

Esta información podría estar disponible, de acuerdo con los términos y condiciones aplicables, incluyendo en algunos casos el pago de una tarifa.

El programa bajo licencia que se describe en este documento y todo el material bajo licencia disponible para éste los proporciona IBM en función de lo dispuesto en el IBM Customer Agreement, el Acuerdo Internacional de Licencia de Programa de IBM o cualquier otro acuerdo equivalente celebrado entre ambas partes.

Los datos de rendimiento y los ejemplos de clientes citados se presentan solamente a efectos ilustrativos. Los resultados reales de rendimiento podrían variar en función de las configuraciones específicas y las condiciones de funcionamiento.

La información relativa a los productos que no son de IBM ha sido obtenida de los proveedores de dichos productos, sus anuncios publicados u otras fuentes públicas disponibles. IBM no ha probado estos productos y no puede confirmar la precisión de rendimiento, compatibilidad ni otras afirmaciones relacionadas con productos que no son de IBM. Las preguntas sobre las prestaciones de los productos que no son de IBM deben dirigirse a los proveedores de dichos productos.

Esta documentación se suministra solo a efectos de planificación. La información aquí contenida puede cambiar antes de que se comercialicen los productos descritos.

Esta información contiene ejemplos de datos e informes que se utilizan en operaciones empresariales de uso habitual. Para ilustrarlas de la forma más completa posible, los ejemplos incluyen nombres de personas, compañías, marcas y productos. Todos estos nombres son ficticios y cualquier parecido con personas o empresas comerciales reales es mera coincidencia.

#### LICENCIA DE COPYRIGHT:

Esta información contiene programas de aplicaciones de ejemplo en código fuente, que ilustran técnicas de programación en las distintas plataformas operativas. Puede copiar, modificar y distribuir los programas de muestra de cualquier forma, sin tener que pagar a IBM, con intención de desarrollar, utilizar, comercializar o distribuir programas de aplicación que estén en conformidad con la interfaz de programación de aplicaciones (API) de la plataforma operativa para la que están escritos los programas de muestra. Los ejemplos no se han probado minuciosamente bajo todas las condiciones. Por lo tanto, IBM no puede garantizar ni dar por sentada la fiabilidad, la capacidad de servicio ni el funcionamiento de dichos programas. Los programas de ejemplo se ofrecen "TAL CUAL", sin garantía de ningún tipo. IBM no será responsable de los daños derivados del uso de los programas de muestra.

## Marcas registradas

---

IBM, el logotipo de IBM e [ibm.com](http://ibm.com) son marcas registradas o marcas comerciales registradas de International Business Machines Corp., registradas en muchas jurisdicciones en todo el mundo. Otros nombres de productos y servicios pueden ser marcas registradas de IBM o de otras compañías. Encontrará una lista actualizada de marcas registradas de IBM en el apartado "Información de copyright y marcas registradas" de la web [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Java y todas las marcas registradas y logotipos basados en Java son marcas registradas de Oracle o sus filiales.

Linux es una marca registrada de Linus Torvalds en los Estados Unidos o en otros países.

Microsoft, Windows, Windows NT y el logotipo de Windows son marcas registradas de Microsoft Corporation en EE.UU. y/o en otros países.

UNIX es una marca registrada de The Open Group en EE.UU. y en otros países.

## Términos y condiciones de la documentación de producto

---

Se otorga permiso para el uso de estas publicaciones si se cumplen estos términos y condiciones.

### Aplicabilidad

Estos términos y condiciones completan los términos y condiciones de uso del sitio web de IBM.

### Uso personal

Puede reproducir estas publicaciones para su uso personal, no comercial siempre que se conserven todos los avisos de propiedad. No puede distribuir, visualizar o realizar trabajo derivado de estas publicaciones, ni de ninguna parte de las mismas, sin el expreso consentimiento de IBM.

### Uso comercial

Puede reproducir, distribuir y mostrar estas publicaciones únicamente dentro de su empresa siempre que se preserven todos los avisos de la marca registrada. No está autorizado a realizar acciones derivadas de estas publicaciones, ni a reproducir, distribuir ni mostrar estas publicaciones o fragmentos de las mismas fuera de la empresa, sin el consentimiento expreso de IBM.

### Derechos

Excepto de la forma expresamente otorgada en este permiso, no se otorga ningún otro permiso, licencia o derecho, ni explícito ni implícito, a las publicaciones o cualquier información, datos, software u otra propiedad intelectual aquí contenida.

IBM se reserva el derecho de retirar los permisos otorgados aquí siempre que, a su discreción, la utilización de las publicaciones se considere perjudicial para sus intereses o, según determine IBM, las instrucciones anteriores no se sigan correctamente.

Queda prohibido descargar, exportar o reexportar esta información si no se cumplen íntegramente todas las leyes y regulaciones aplicables, incluidas las leyes y regulaciones de exportación de los Estados Unidos.

IBM NO PROPORCIONA NINGÚN TIPO DE GARANTÍA SOBRE EL CONTENIDO DE ESTAS PUBLICACIONES. ESTAS PUBLICACIONES SE PROPORCIONAN "TAL CUAL" SIN NINGÚN TIPO DE GARANTÍA, IMPLÍCITA NI EXPLÍCITA, INCLUIDAS, PERO SIN LIMITARSE A, LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN, IDONEIDAD PARA UN PROPÓSITO CONCRETO O NO INFRACCIÓN.

## Declaración de privacidad en línea de IBM

---

Los productos de IBM Software, incluidas las soluciones de software como servicio (“ofertas de software”), pueden utilizar cookies u otras tecnologías para recopilar información de uso del producto, ayudar a mejorar la experiencia del usuario final, adaptar las interacciones con el usuario final u otros fines. En muchos casos, las ofertas de software no recopilan información de identificación personal. Algunas de nuestras ofertas de software pueden ayudarle a recopilar información de identificación personal. Si esta oferta de software utiliza cookies para recopilar información de identificación personal, más adelante se incluirá información específica sobre el uso de cookies por parte de esta oferta.

En función de la configuración desplegada, esta oferta de software puede utilizar cookies de sesión y persistentes que recopilan los nombres, nombres de usuario, contraseñas u otra información de identificación personal de todos los usuarios con fines de gestión de sesiones, autenticación, configuración de inicio de sesión único u otros fines de seguimiento de uso o funcionales. Estas cookies se pueden inhabilitar, pero, de hacerlo, es posible que se elimine también la funcionalidad que habilitan.

Si las configuraciones desplegadas para esta oferta de software proporcionan al cliente la capacidad de recopilar información de identificación personal de los usuarios finales mediante cookies y otras

tecnologías, aquel debería obtener su propio asesoramiento legal acerca de las leyes aplicables a dicha recopilación de datos, incluidos los requisitos de aviso y consentimiento.

Para obtener más información acerca del uso de las distintas tecnologías, incluidas las cookies, para estos fines, consulte la política de privacidad de IBM en <http://www.ibm.com/privacy> y la declaración de política de privacidad de IBM en <http://www.ibm.com/privacy/details>, en la sección llamada “Cookies, Web Beacons and Other Technologies”, y la publicación "IBM Software Products and Software-as-a-Service Privacy Statement" en <http://www.ibm.com/software/info/product-privacy>.





Número Pieza

(1P) P/N: