**.**

**Ms ms-intermediario-redis.**

**Infoprenda.**

**MANUAL DE INSTALACION Y EJECUCION.**

**S P S**

**TECH IS NOW**

**Fecha: 02 /09/2021**

**DE SERVICIOS S P S**

Contabilidad Electrónica API

**Tabla de contenido**

Historial de versiones 2

Información del proyecto 3

Introducción. 4

Objetivos. 4

Consideraciones. 4

Insumos para la ejecución del manual. 4

Sobre el lenguaje. 4

Sobre el framework. 4

Sobre el entorno de despliegue. 4

Sobre el monitoreo. 6

Repositorio. 8

Configuración de la aplicación en el ambiente de IBM Cloud 9

Variables de entorno 13

Conexión del recurso con la aplicación 16

Status. 30

Iniciar. 31

Logs. 31

Detener. 31

Conclusiones. 31

# Historial de versiones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fecha** | **Versión** | **Autor** | **Solicitante** | **Descripción** |
| 02/09/2020 | 1.0 | Juan Carlos García López | Cesar Reyna Medina | Versión inicial. |

# Información del proyecto

|  |  |
| --- | --- |
| **Nombre Proveedor** | S P S |
| **Proyecto** | Infoprenda |
| **Área Solicitante de TI** | Gerencia de TI. |
| **Gerente / Líder de proyecto** | Cesar Reyna Medina |

# Introducción.

En el siguiente documento se describen el conjunto de configuraciones necesarias a realizar para construir, desplegar y ejecutar el microservicio Intermediario Redis en el PaaS de IBM Cloud, para los diversos ambientes de Nacional Monte de Piedad, considerados en el proyecto de Infoprenda.

Para la ejecución de estas configuraciones, se considera que el operador del manual está familiarizado con los conceptos que en el manual se encuentren plasmados.

# Objetivos.

Contar con una guía que establezca todas las consideraciones necesarias, para que el operador encargado de configurar, construir, desplegar y ejecutar el microservicio Intermediario Redis en un ambiente en IBM Cloud, pueda ejecutar exitosamente estas tareas.

# Consideraciones.

## Insumos para la ejecución del manual.

Con la entrega de este documento de instalación se deberán haber proporcionado:

* Credenciales para acceso a plataforma de IBM Cloud y datos referentes al espacio de trabajo correspondiente a ese ambiente.
* URL y de escaneo estático de sonarqube.
* Credenciales para repositorio de código (GitHub).

## Sobre el lenguaje.

El lenguaje utilizado para la construcción del microservicio es Java Script.

## Sobre el framework.

El framework utilizado para la construcción del microservicio es Node v12.18.3.

## Sobre el entorno de despliegue.

El microservicio será desplegado en la plataforma IBM Cloud proporcionada por Nacional Monte de Piedad.

#### Variables fijas y variables que cambian por ambiente.

La siguiente tabla lista todas las variables requeridas para el funcionamiento del microservicio. Adicional se indica cuales tienen valores fijos y cuales requieren modificarse por cada ambiente.

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre de la variable | Detalle | Valor en DEV | ¿Es valor fijo? |
| CONTEXT\_NAME | Contexto de la aplicación | api/partidas | Si, con la posibilidad de ajustar el contexto |
| CONTEXT\_VERSION | Versión con la que se identifica al componente. | v1 | Si, en caso de que esta versión no varié entre ambientes |
| LOG\_LEVEL | Nivel de reporteo de trazas en la aplicación. | debug | No, depende de cada ambiente. |
| REDIS\_URL | URL del servicio de redis | rediss://ibm\_cloud\_5bd4b948\_a4e3\_4ba9\_ad04\_a0b08af1e8e8:c071d43ff986bb79b45a08c4a1b995c01312de1561184b6a1666f00715664da4@17333051-f999-40bd-a311-05cf2d93a91f.blijs0dd0dcr4f55oehg.databases.appdomain.cloud:30560/0 | No, depende de cada ambiente. |
| REDIS\_CERT\_BASE64 | Certificado de comunicación TLS |  | No, depende de cada ambiente. |
| URL\_OAUTH\_VALIDATOR | URL del servicio para la validación del token | https://iamdr.montepiedad.com.mx:4444/NMP/oauth2/v1 | No, depende de cada ambiente. |
| NEW\_RELIC\_APP\_NAME | Nombre con el que se identifica la aplicación en la plataforma de New Relic. | dev1775-ms-intermediario-redis | No, depende de cada ambiente. |
| NEW\_RELIC\_LICENSE\_KEY | Clave de licencia para la aplicación en New Relic | No se tienen (En trámite) | No, depende de cada ambiente. |
| SONARQUBE\_PROJECT\_NAME | Nombre del proyecto para la aplicación en SonarQube | No se tienen (En trámite) | Si, es el mismo para cada ambiente |
| SONARQUBE\_PROJECT\_KEY | Llave para el proyecto en SonarQube | No se tienen (En trámite) | Si, es el mismo para cada ambiente |
| SONARQUBE\_PROJECT\_LOGIN | Token de login para proyecto en SonarQube | a2ff4af3666a2ea08d25523fb5e6d97286bee9a3 | Si, es el mismo para cada ambiente |

### Servicios externos.

#### Servicio OAuth

* Se deberá contar con la URL para la validación de los tokens para la validación de la seguridad en el ms
* Se deberá contar con la URL y el certificado en base 64 para la comunicación con Redis.

## Sobre el monitoreo.

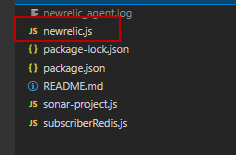
### Servicio de Health

|  |
| --- |
| http://dev1775-ms-intermediario-redis.mybluemix.net/api/partidas/v1/ |

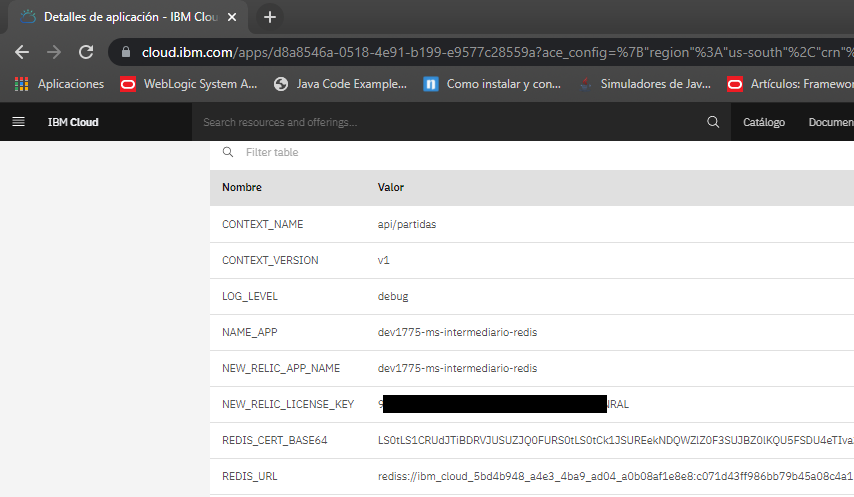
### Métricas.

El microservicio realiza él envió de métricas a una instancia de [NewRelic](https://newrelic.com/) definida por Nacional Monte de Piedad.

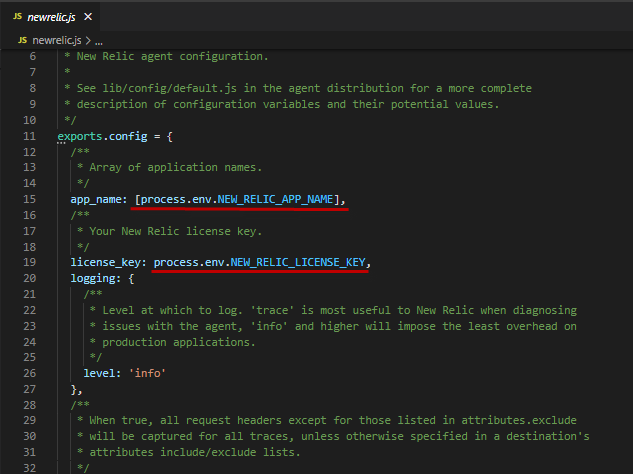
Con el despliegue de la aplicación, se incluye el archivo newrelic.js para realizar la configuración con la plataforma.



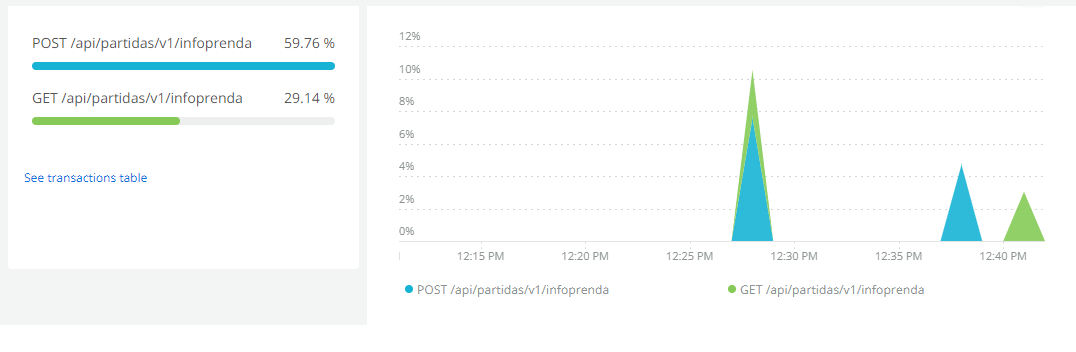
En este archivo se especifican la licencia y nombre de la aplicación para NewRelic mediante variables de ambiente, desde la consola de Bluemix en la sección de variables de entorno.



Finalmente, los parámetros de nombre y licencia se mapean hacia el archivo de configuración del agente



Una vez iniciado el agente, se podrán ver en NewRelic las métricas reportadas por el microservicio:



## 

## Repositorio.

El código para el microservicio de recomendaciones se encuentra en el siguiente repositorio de Nacional Monte de Piedad:

|  |  |
| --- | --- |
| Tipo | Valor |
| Repositorio: | <https://github.com/MontePiedadMx/MS_Intermediario_Redis.git> |
| Branch: | ftr-1.0.0-26Agt21-Infoprenda-Redis-SPS, dev, test, master |
| Usuario utilizado: | JuanCarlosGarciaLopez |

## Servicio IBM Redis.

Es necesaria la creación del servicio de redis desde el portal de Bluemix, para su configuración de debe seguir el siguiente manual “NMP\_Infoprenda\_Servicio\_Redis.docx”

**NOTA**: Este paso es requerido ya que durante la configuración del ms como aplicación será requerido la REDIS\_URL y los datos de la cuenta de conexión hacia este servicio.

## Configuración de la aplicación en el ambiente de IBM Cloud

Para crear un espacio en Bluemix y poder desplegar una aplicación se deben de seguir los siguientes pasos:

1.- Iniciar Sesión en <https://cloud.ibm.com>



Desde el menú principal seleccionar en el panel izquierdo la opción de CF en su submenú Público

Graphical user interface, application

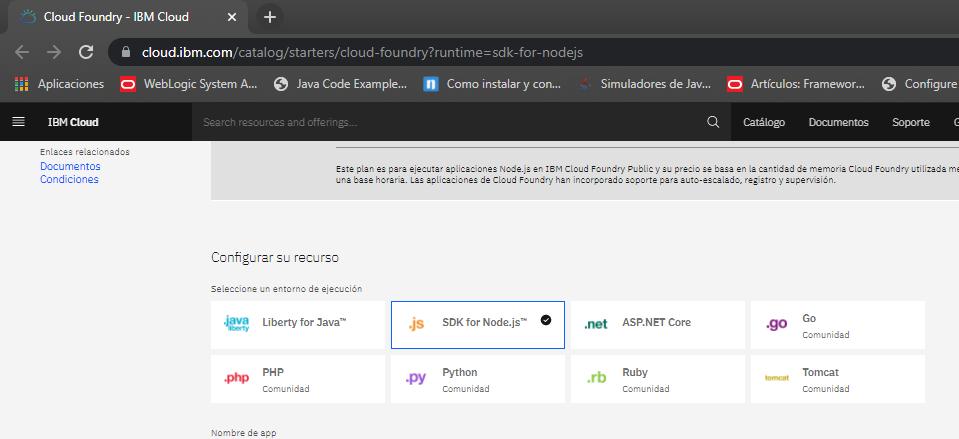
Description automatically generated

Seleccionar la opción Crear:

Graphical user interface

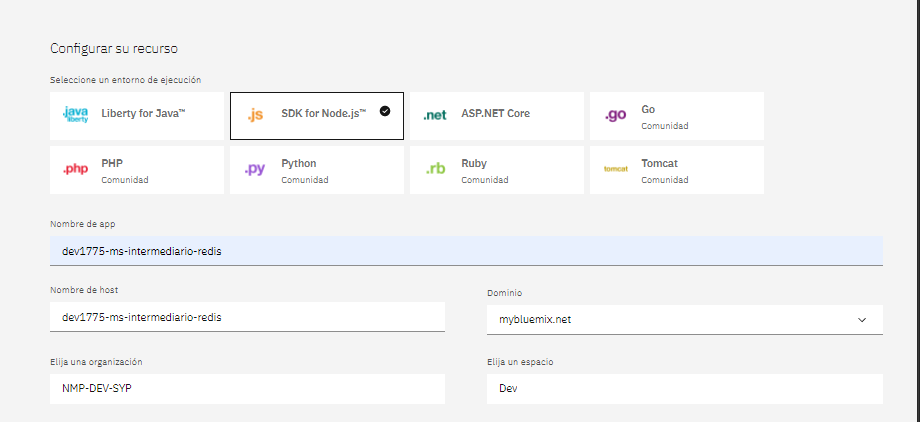
Description automatically generated

Una vez dentro, buscamos la opción **SDK for Node.js**

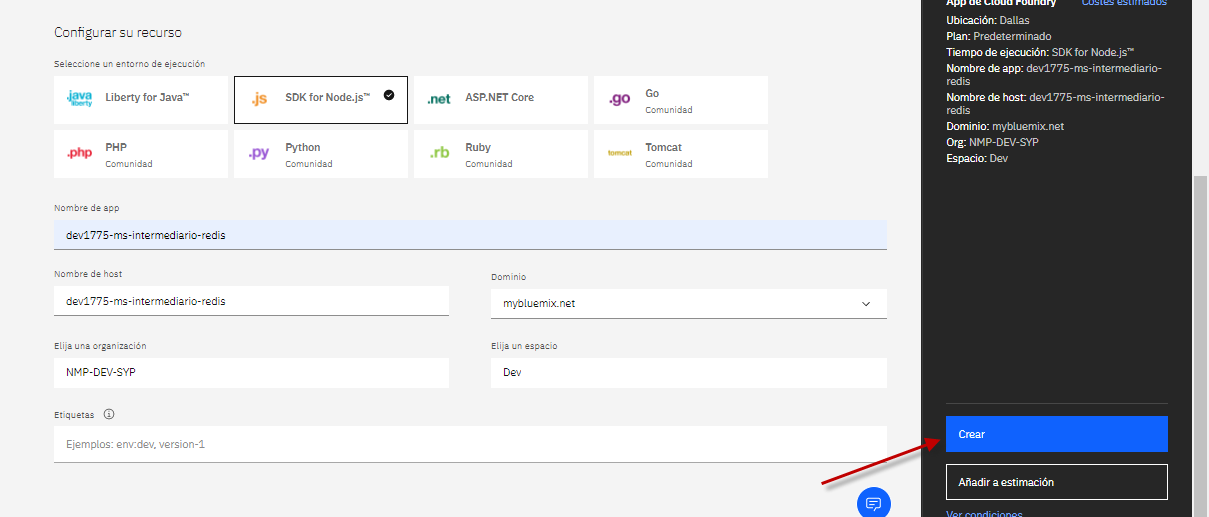


Capturamos los siguientes datos:

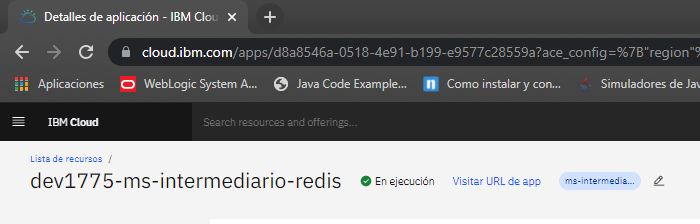
* 1. Nombre de la app: **dev1775-ms-intermediario-redis (**varía dependiendo el ambiente)
  2. Nombre del host: **dev1775-ms-intermediario-redis (**varía dependiendo el ambiente)
  3. Dominio: **mybluemix.net**
  4. Organización: **NMP\_DEV\_SYP (**varía dependiendo el ambiente)
  5. Espacio: **Dev** **(**varía dependiendo el ambiente)



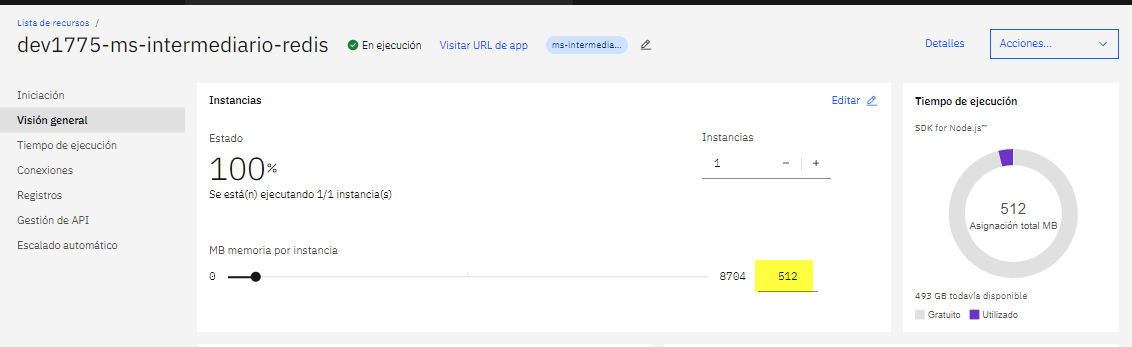
Una vez llenados los valores seleccionamos la opción **crear**.



La aplicación creada se mostrará de la siguiente manera.



Se asignarán 512 MB de memoria para la instancia.

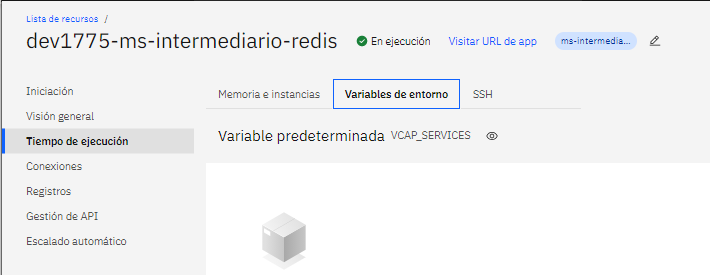


## 

## Variables de entorno

Las variables de entorno se usan en tiempo de despliegue/ejecución, para parametrizar cierta funcionalidad dentro de la misma.

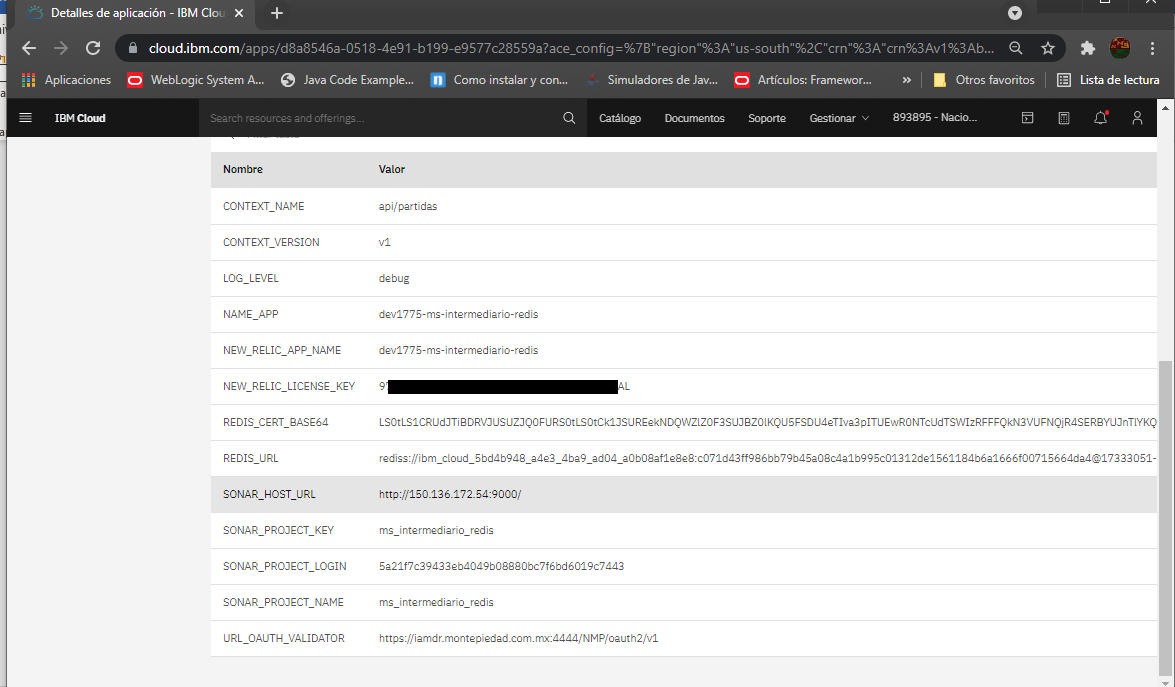
1. Seleccionar la opción **Tiempo de ejecución** del menú derecho.
2. Seleccionar la opción **Variables de entorno**.



A continuación, la lista de variables que se necesitan ingresar para el funcionamiento de la aplicación:

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre de la variable | Detalle | Valor en DEV | ¿Es valor fijo? |
| CONTEXT\_NAME | Contexto de la aplicación | api/partidas | Si, con la posibilidad de ajustar el contexto |
| CONTEXT\_VERSION | Versión con la que se identifica al componente. | v1 | Si, en caso de que esta versión no varié entre ambientes |
| LOG\_LEVEL | Nivel de reporteo de trazas en la aplicación. | debug | No, depende de cada ambiente. |
| REDIS\_URL | URL del servicio de redis | rediss://ibm\_cloud\_5bd4b948\_a4e3\_4ba9\_ad04\_a0b08af1e8e8:c071d43ff986bb79b45a08c4a1b995c01312de1561184b6a1666f00715664da4@17333051-f999-40bd-a311-05cf2d93a91f.blijs0dd0dcr4f55oehg.databases.appdomain.cloud:30560/0 | No, depende de cada ambiente. |
| REDIS\_CERT\_BASE64 | Certificado de comunicación TLS |  | No, depende de cada ambiente. |
| URL\_OAUTH\_VALIDATOR | URL del servicio para la validación del token | https://iamdr.montepiedad.com.mx:4444/NMP/oauth2/v1 | No, depende de cada ambiente. |
| NEW\_RELIC\_APP\_NAME | Nombre con el que se identifica la aplicación en la plataforma de New Relic. | dev1775-ms-intermediario-redis | No, depende de cada ambiente. |
| NEW\_RELIC\_LICENSE\_KEY | Clave de licencia para la aplicación en New Relic | No se tienen (En trámite) | No, depende de cada ambiente. |
| SONARQUBE\_PROJECT\_NAME | Nombre del proyecto para la aplicación en SonarQube | dev1775-ms-intermediario-redis | Si, es el mismo para cada ambiente |
| SONARQUBE\_PROJECT\_KEY | Llave para el proyecto en SonarQube | 1775-MS-Intermediario-Redis-90bcee873c7f | Si, es el mismo para cada ambiente |
| SONARQUBE\_PROJECT\_LOGIN | Token de login para proyecto en SonarQube | a2ff4af3666a2ea08d25523fb5e6d97286bee9a3 | Si, es el mismo para cada ambiente |
| SONAR\_HOST\_URL | Host de sonar | http://169.62.185.92:5749/ | Si, es el mismo para cada ambiente |

Una vez ingresadas las variables quedarían de la siguiente manera:

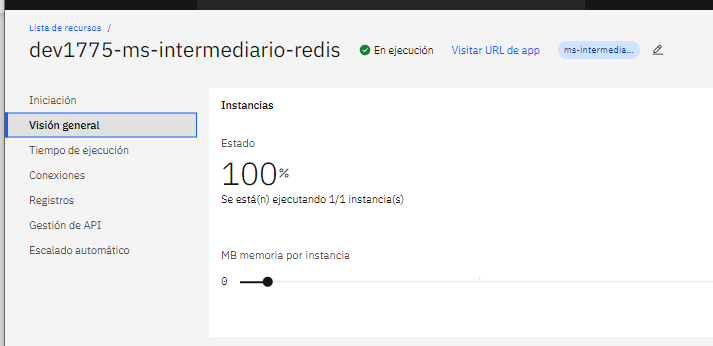


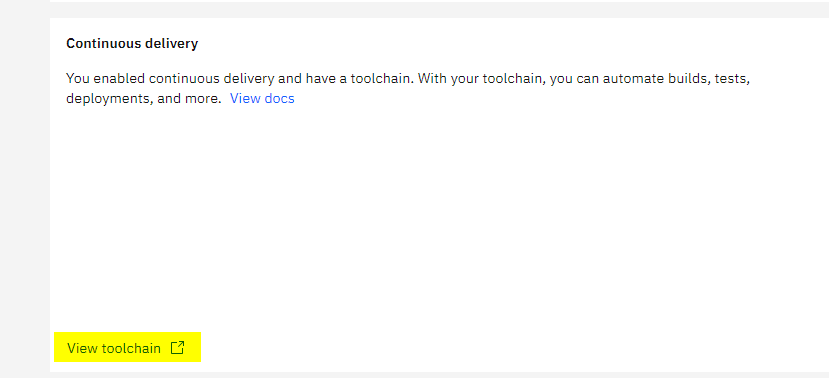
## 

## Conexión del recurso con la aplicación

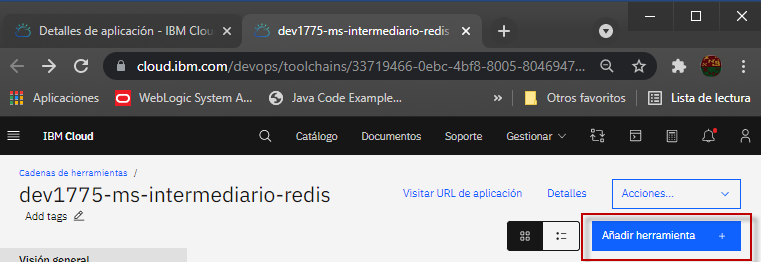
Una vez configurada la memoria y las variables de ambiente, procedemos a vincular la aplicación para poder utilizarla.

Nos ubicamos en el menú ubicado en la parte derecha de la pantalla y seleccionamos la opción **Overview,** una vez ubicados ahí nos dirigimos a la parte inferior y seleccionamos la opción **View** **ToolChain**

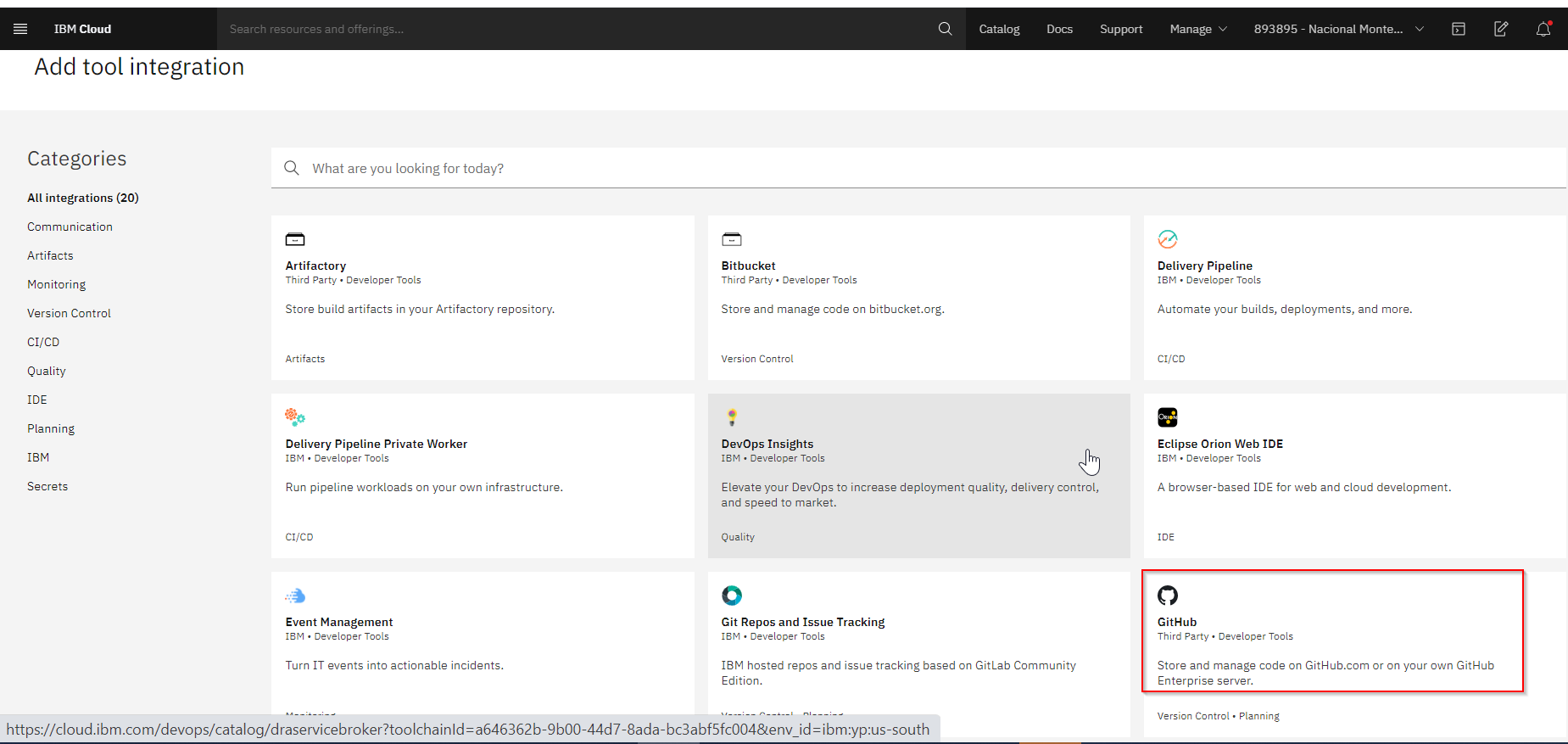




Presionar el boto ***Añadir una herramienta*** para configurar el repositorio de código



Posteriormente seleccionamos la opción GitHub, procedemos a realizar la configuración.



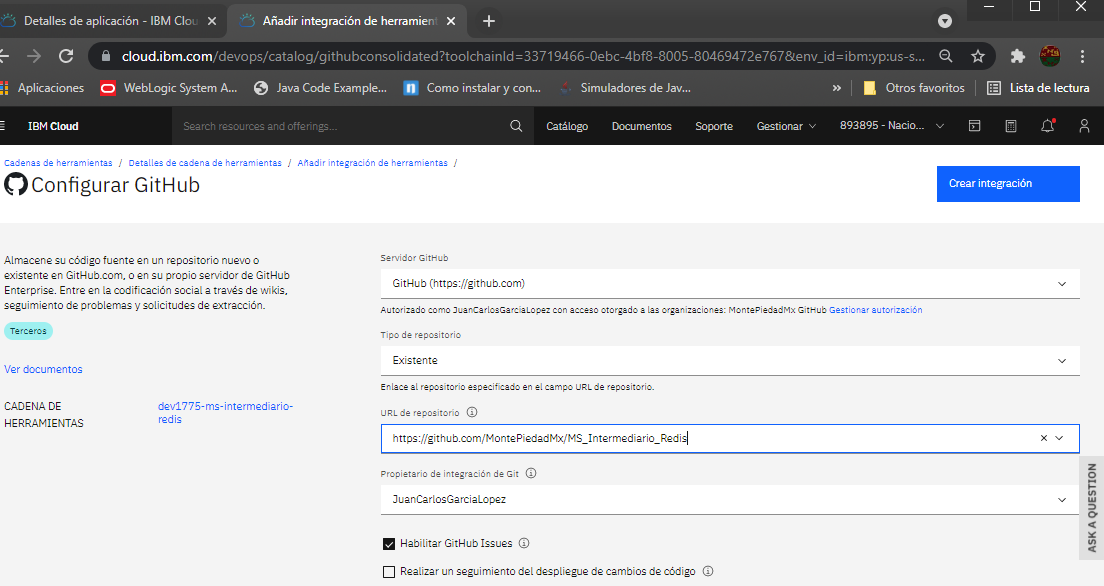
Capturamos los siguientes datos:

**GitHub Server:** Servidor GitHub

**Repository type:** (Existente)

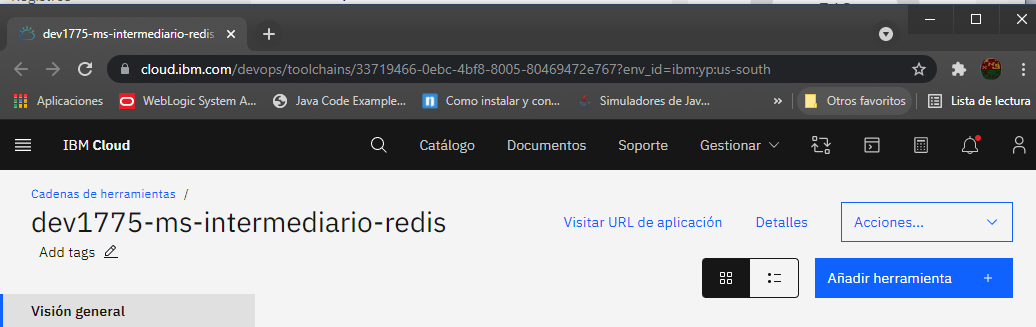
**Repository:** URL del repositorio Especificada en la parte del documento [Repositorio](#_Repositorio.).

**Integration Owner**: default

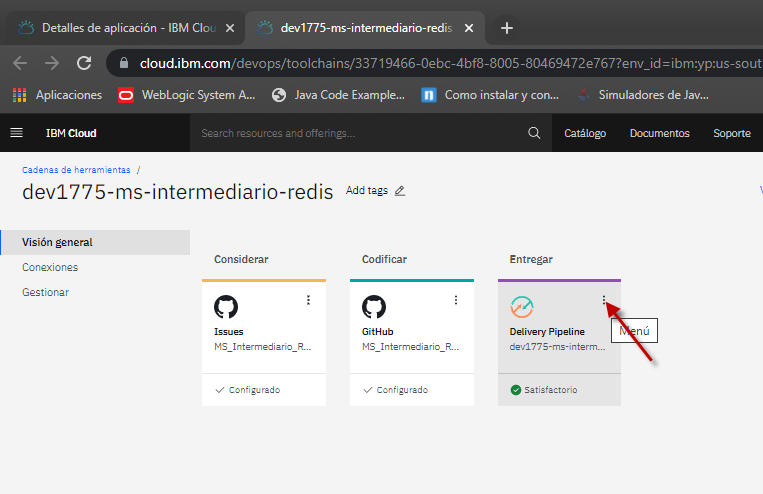


Una vez finalizada la parte del llenado de los datos dar click en **Save Integration**

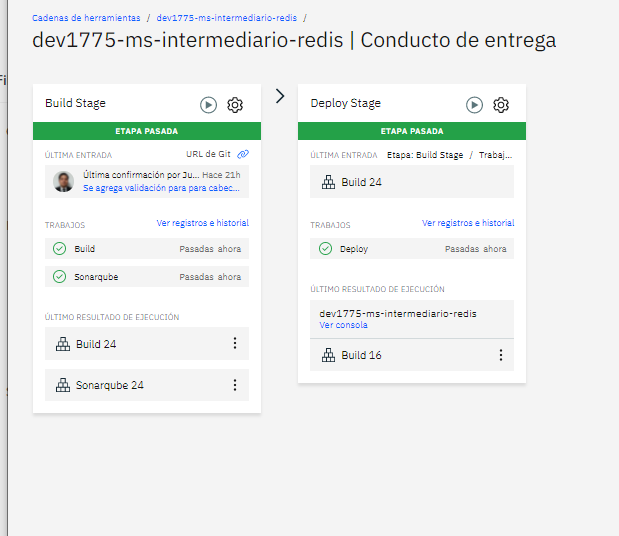
Nuevamente presionar el boto ***Añadir una herramienta*** para vincular el servidor de sonar con la aplicación.



Damos click sobre el recuadro de **deliver** o **entregar**



Una vez dentro nos mostrara los siguientes stages (las siguientes imágenes son de los stages actualmente corriendo, esto en la aplicación de dev).



Procedemos a configurar la parte del Stage **Build Stage**, seleccionamos la opción del engranaje en la parte superior derecha y le damos **configurar state**.

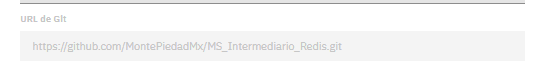


Una vez dentro procedemos a capturar los siguientes valores

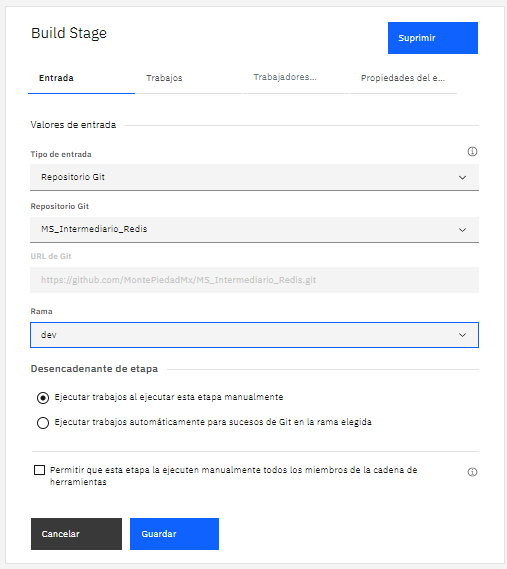
**Tipo de entrada**: Git Repository

**Repositorio Git**: Nombre del repositorio

**Git URL:** capturar la versión default dada por el sistema, en caso de marcar error seleccionar la siguiente:



Branch: según corresponda.



Cambiamos de ventana y seleccionamos la opción Jobs(trabajos), configuramos los siguientes valores. Para el proceso de **Build Stage**.

**Tipo de compilador:** nmp

**Script de compilación:**

#!/bin/bash

# Para instalar una versión específica del nodo, elimine el comentario de la línea siguiente:

# nvm install <version>

# Para cambiar a la versión de Node.js instalada, elimine el comentario de la línea siguiente:

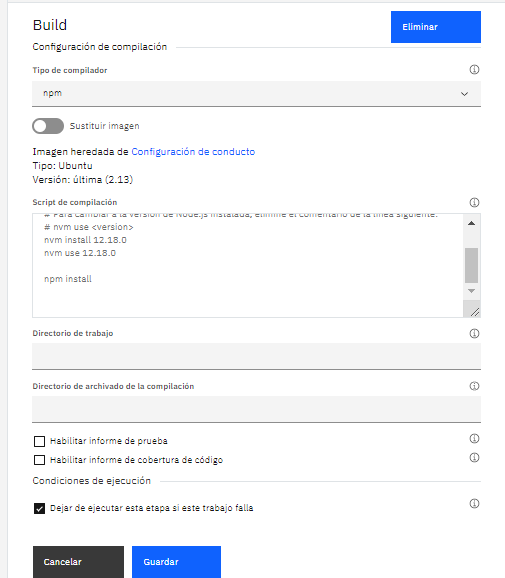
# nvm use <version>

nvm install 12.18.0

nvm use 12.18.0

npm install

**Dejar de ejecutar esta etapa si este trabajo falla:** true



Una vez finalizado el llenado de los valores, damos click en guardar.

Ahora debemos presionar la opción de **Añadir trabajo.** Para el proceso generar el proceso de **Sonarqube**.

**Tipo de compilador: nmp**

**Imagen heredada de**[**Configuración de conducto**](https://cloud.ibm.com/devops/pipelines/3ec513e1-c8cf-402d-b21e-7e74869dbd55/config?env_id=ibm:yp:us-south) **Tipo: Ubuntu  
Versión: última (2.13)**

**Script de compilación**

#!/bin/bash

# Para instalar una versión específica del nodo, elimine el comentario de la línea siguiente:

# nvm install <version>

# Para cambiar a la versión de Node.js instalada, elimine el comentario de la línea siguiente:

# nvm use <version>

export REDIS\_URL="rediss://ibm\_cloud\_5bd4b948\_a4e3\_4ba9\_ad04\_a0b08af1e8e8:c071d43ff986bb79b45a08c4a1b995c01312de1561184b6a1666f00715664da4@17333051-f999-40bd-a311-05cf2d93a91f.blijs0dd0dcr4f55oehg.databases.appdomain.cloud:30560/0"

export REDIS\_CERT\_BASE64=""

export NAME\_APP=dev1775-ms-intermediario-redis

export LOG\_LEVEL=trace

export CONTEXT\_NAME=api/partidas

export CONTEXT\_VERSION=v1

export URL\_OAUTH\_VALIDATOR=https://iamdr.montepiedad.com.mx:4444/NMP/oauth2/v1

export SONAR\_HOST\_URL=http://150.136.172.54:9000/

export SONAR\_PROJECT\_NAME=ms\_intermediario\_redis

export SONAR\_PROJECT\_KEY=ms\_intermediario\_redis

export SONAR\_PROJECT\_LOGIN=5a21f7c39433eb4049b08880bc7f6bd6019c7443

npm install

npm run test

npm run sonar

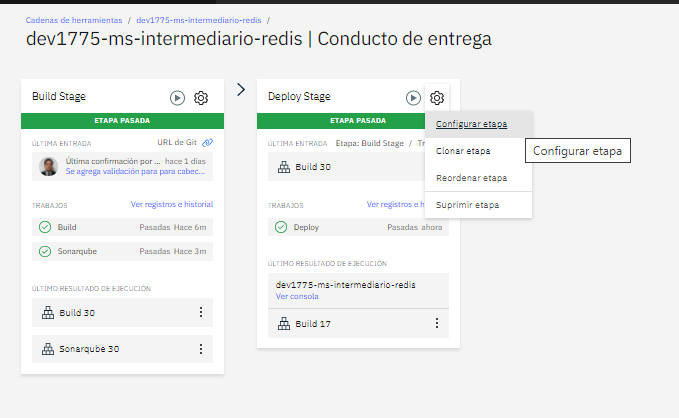
**Working Directory:** default

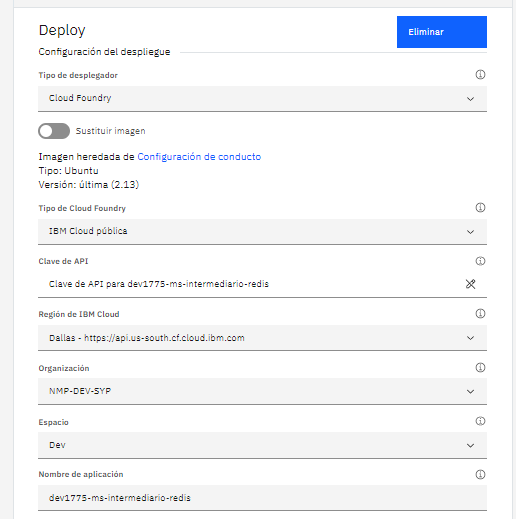
**Build Archive Directory:** default

**NOTA**: se deben remplazar los valores correspondientes al ambiente de configuración, estos valores corresponden al ambiente de desarrollo.

Una vez finalizado el llenado de los valores, damos click en guardar.

Procedemos a configurar la parte del Stage **Deaploy Stage**, seleccionamos la opción del engranaje en la parte superior derecha y le damos **configurar state**.





Nos ubicamos en la opción **trabajos**, una vez dentro procedemos a modificar los siguientes valores:

**Organización**: NMP-DEV-SYP

**Espacio**: Dev

**Nombre de la aplicación:** dev1775-ms-intermediario-redis

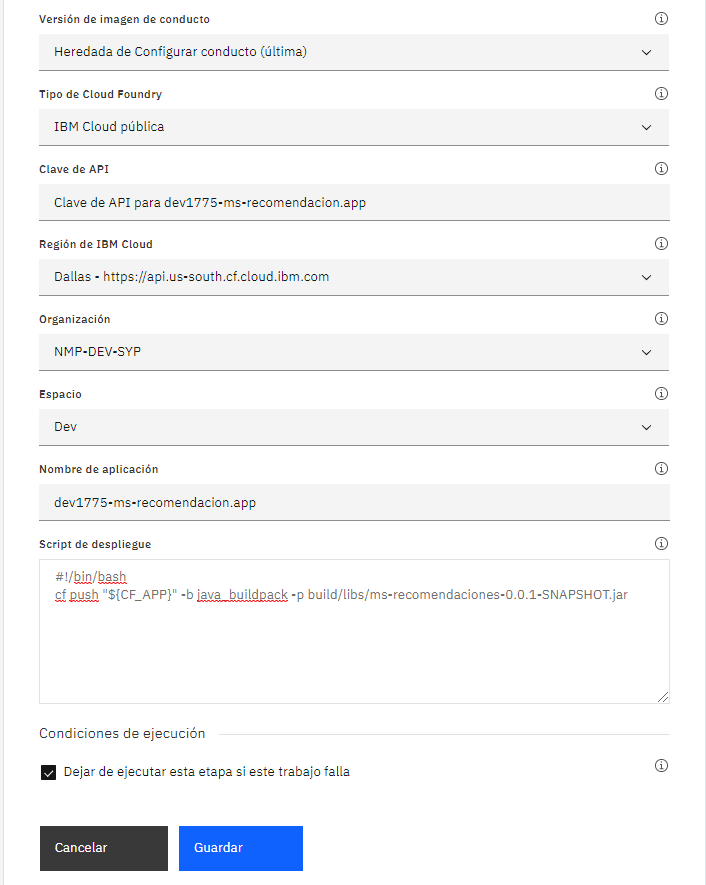
**Script de despliegue:**

#!/bin/bash

cf push "${CF\_APP}"

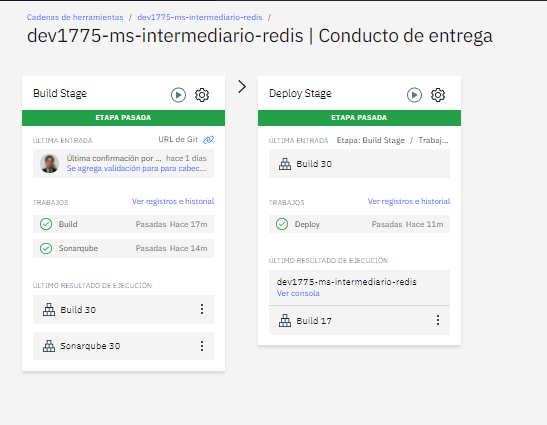
# View logs

# cf logs "${CF\_APP}" –recent



Para finalizar damos click en el botón de guardar.

Una vez finalizada la configuración de los Stages, iniciamos Build Stage, para que se inicialice la aplicación.

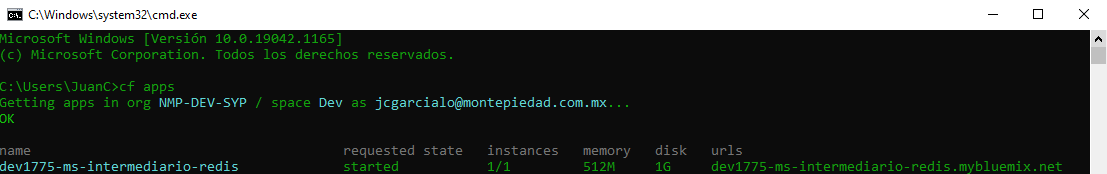


Al finalizar el despliegue nos mostrara una leyenda al inicio del recuadro de cada stage con la palabra “**ETAPA PASADA**”

## Status.

Para verificar el estado del microservicio desde línea de comandos ejecutar el siguiente comando (se requiere autenticarse en el espacio de trabajo):

|  |
| --- |
| cf apps |



Con esto comprobamos que el microservicio se encuentra en ejecución.

## Iniciar.

Para iniciar el microservicio de recomendaciones en caso de que este detenido, hacer uso del siguiente comando:

|  |
| --- |
| cf start dev1775-ms-intermediario-redis |

**Nota:** Esta acción está implícita al hacer el despliegue del microservicio en Pivotal Cloud Foundry, utilizando la automatización con Bluemix.

## Logs.

Para ver los logs del microservicio recomendaciones hacer uso del siguiente comando:

|  |
| --- |
| cf logs dev1775-ms-intermediario-redis |

## Detener.

Para detener el microservicio recomendaciones en caso de ser requerido, hacer uso del siguiente comando:

|  |
| --- |
| cf stop dev1775-ms-intermediario-redis |

# Conclusiones.

Este documento deberá ser suficiente para que el operador encargado de desplegar y dejar ejecutando el microservicio intermediario redis en IBM Cloud.

