**Qa1ki0**

**Ms intermediario redis.**

**Infoprenda.**

**MANUAL TECNICO.**

**S P S**

**TECH IS NOW**

**Fecha: 03/09/2021**

**DE SERVICIOS S P S**

Contabilidad Electrónica API

**Tabla de contenido**

Historial de versiones 3

Información del proyecto 3

Introducción. 4

Objetivos. 4

Consideraciones. 4

Sobre el dominio del problema. 4

Desglose Técnico. 15

Conclusiones. 18

# Historial de versiones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fecha** | **Versión** | **Autor** | **Solicitante** | **Descripción** |
| 03/09/2021 | 1.0 | Juan Carlos García López | Nacional Monte de Piedad | 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 presente documento se describe el microservicio “Intermediario Redis” implementado, como resultado de la necesidad de contar con un punto de comunicación entre diversas aplicaciones y redis, como primer instancia para el sistema de Infoprenda.

# Objetivos.

Los objetivos que con el requerimiento del microservicio se deben alcanzar son:

* Publicar mensajes en redis, para que sea procesados por el ms consumidor siguiendo el patrón (productor consumidor)
* Recuperar los mensajes depositados por el ms consumidor.

# Consideraciones.

## Sobre el dominio del problema.

El siguiente diagrama se muestra la solución planteada para el Ms de intermediario redis.

Imagen que contiene árbol

Descripción generada automáticamente

Figura 1. Secuencia para el almacenamiento de una partida.

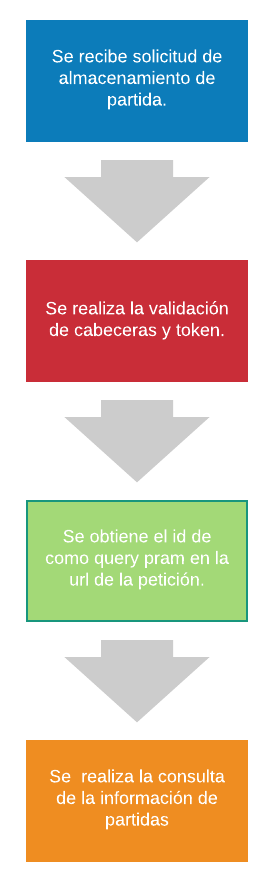


Figura 2. Secuencia para recuperar partidas.

### Vocabulario.

Los siguientes términos son necesarios conocerlos para entender el alcance y funcionamiento de la solución presentada:

|  |  |
| --- | --- |
| Palabra/Término | Significado/Acepción |
| Prenda | objetos en el empeño. |
| Partida | Colección de prendas agrupadas para su referencia. |
| redis | Redis es un motor de base de datos en memoria, basado en el almacenamiento en tablas de hashes pero que opcionalmente puede ser usada como una base de datos durable o persistente. Está escrito en ANSI C por Salvatore Sanfilippo, quien es patrocinado por Redis Labs |

### Contrato

La especificación del contrato del servicio asociado al microservicio para almacenar una partida en redis se encuentra disponible en la plataforma de <https://login.apiary.io> bajo el en la siguiente ruta [partidas](https://msclientredis.docs.apiary.io/#reference/partida), para usuarios con credenciales y permisos para acceder al espacio de trabajo “NMP-DEV” para mayor referencia.

El microservicio cuenta a manera de resumen con la siguiente definición:

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo | Valor | | |
| Endpoint: | <http://host:port/api/partidas/v1> | | |
| Path Recurso: | /Infoprenda | | |
| Método: | POST | | |
| Operación: | Generación de intermediario redis. | | |
| Headers: | **Campo** | **Tipo** | **Descripción** |
| Canal | string | Tipo de canal |
| Folio | string | Folio asociado al registro |
| Producto | string | Producto asociado a la solicitud |
| Operacion | string | Operación que se va a realizar |
| Trigger | string | Sistema que dispara la acción |
| Evento | string | Tipo de operación |
| UnidadOrganizacional | string | Unidad organizacional |
| idConsumidor | string | Origen de la petición |
| idDestino | string | Destino de la petición |
| oauthbearer | string | Token de OAuth 2.0 |
| Cuerpo: | {  "idCliente": "22170",  "nivelCliente": "DIAMANTE",  "calificacionAjustada": 10,  "calificacionSiva2": 7,  "gramaje": 1,  "rango": "F5",  "kilataje": 12,  "incremento": 25,  "desplazamiento": "5",  "ramo": "Alhajas",  "subramo": "Alhajas"  } | | |
| Detalle del cuerpo: | **Campo** | **Tipo** | **Descripción** |
| entrada | string | Cuerpo de entrada para la generación de una simulación de crédito. |
| salida | string | Respuesta de la petición |
| Mensaje de respuesta esperado: | {  "code": "NMP-API-REDIS-200",  "status": "SUCCESS",  "message": "Se ha realizado correctamente la operación"  } | | |
| Mensaje de respuesta Bad Request: | {  "code": "NMP-API-REDIS-400",  "message": "El campo idCliente es requerido",  "description": "Petición mal formada."  } | | |
| Mensaje de respuesta Unauthorized: | {  "code": "NMP-API-REDIS-401",  "message": "Se ha producido un error de autorización"  } | | |
| Mensaje de respuesta Internal Server Error: | {  "code": "NMP-API-REDIS-500",  "message": "Error interno del servidor."  } | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo | Valor | | |
| Endpoint: | <http://host:port/api/partidas/v1> | | |
| Path Recurso: | /Infoprenda?id | | |
| Método: | GET | | |
| Operación: | Generación de intermediario redis. | | |
| Headers: | **Campo** | **Tipo** | **Descripción** |
| Canal | string | Tipo de canal |
| Producto | string | Producto asociado a la solicitud |
| Operacion | string | Operación que se va a realizar |
| Trigger | string | Sistema que dispara la acción |
| Evento | string | Tipo de operación |
| Folio | string | Folio asociado al registro |
| UnidadOrganizacional | string | Unidad organizacional |
| Usuario | string | Usuario asociado a la petición |
| NombreUsuario | string | Nombre del usuario asociado a la petición |
| idConsumidor | string | Origen de la petición |
| idDestino | string | Destino de la petición |
| oauthbearer | string | Token de OAuth 2.0 |
|  |  | | |
| Detalle del cuerpo: | **Campo** | **Tipo** | **Descripción** |
| id | string | Id de la partida a consultar. |
| salida | string | Lista de preendas asociadas a la partida. |
| Mensaje de respuesta esperado: | {  "controlExcepcion": {  "codigo": "NMP-API-REDIS-200",  "mensaje": "Resultado Exitoso"  },  "listaPrendasAsociadas": [  {  "tipo": "Otros",  "lote": "",  "idPrenda": "nmp-al-al-108423424",  "numeroRefrendo": 1,  "fechaEmpenio": "17/06/2021",  "valorAncla": 1196,  "valorMonte": 6220,  "elementosComplementarios": "CON ACCION",  "gramaje": 8,  "prestamo": 5910,  "descripcionPrenda": "2 NUEVA CLASIFICACION NUEVO TIPO ESFERAS DE ORO DE 12 KILATES Y PESO DE 8.00 GRAMOS CON ACCION",  "clasificacion": "Otros",  "cantidad": 2,  "kilataje": 12,  "idTipo": "",  "disenio": "esferas",  "idDisenio": "",  "idClasificacion": "",  "folio": "Infoprenda-22170-23-1628783013127"  }  ]  } | | |
| Mensaje de respuesta Unauthorized: | {  "code": "NMP-API-REDIS-401",  "message": "Se ha producido un error de autorización"  } | | |
| Mensaje de respuesta Internal Server Error: | {  "code": "NMP-API-REDIS-500",  "message": "Error interno del servidor."  } | | |

### Diagrama de componentes.

En el siguiente diagrama se muestran los componentes que intervienen en el funcionamiento del microservicio del Intermediario redis al almacenar una partida:

* El cliente realiza la solicitud de almacenar partida
* El Ms intermediario realiza la validación de la autorización con ayuda del entpoint Oauth
* El Ms intermediario realiza la validación de la estructura de la información.
* El Ms deposita el mensaje en el canal de comunicación con redis. El canal al que se depositara el mensaje es obtenido en la cabecera “Canal”.

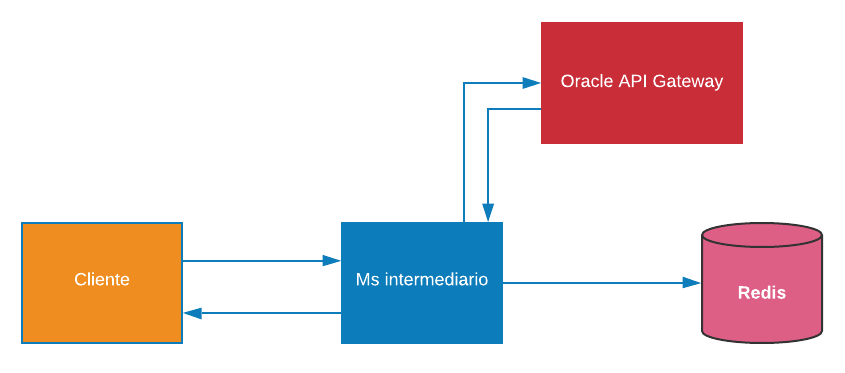


Figura 3. Componentes que intervienen en el funcionamiento de almacenar una partida.

*Ejemplo JSON:*

*Entrada:*

|  |
| --- |
| {    "idCliente": "22170",    "nivelCliente": "DIAMANTE",    "calificacionAjustada": 10,    "calificacionSiva2": 7,    "gramaje": 1,    "rango": "F5",    "kilataje": 12,    "incremento": 25,    "desplazamiento": "5",    "ramo": "Alhajas",    "subramo": "Alhajas"  } |

*Salida:*

|  |
| --- |
| {  "code": "NMP-API-REDIS-200",  "status": "SUCCESS"  } |

En el siguiente diagrama se muestran los componentes que intervienen en el funcionamiento del microservicio del Intermediario redis al recuperar una partida:

* El cliente realiza la solicitud de recuperar partidas por id.
* El Ms intermediario realiza la validación de la autorización con ayuda del entpoint Oauth
* El Ms intermediario realiza la validación para obtener el id de la partida.
* El Ms recupera la partida asociada al id de partida de la petición realizando la consulta de información en redis.

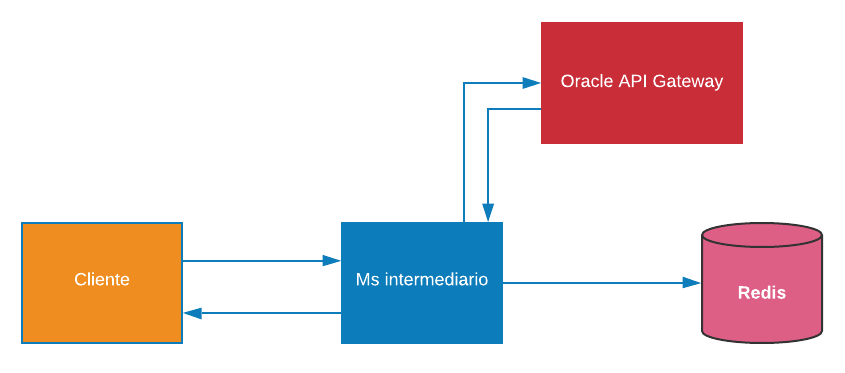


Figura 4. Componentes que intervienen en el funcionamiento de recuperar una partida.

*Ejemplo JSON:*

*Entrada:*

|  |
| --- |
| http://[host]/api/partidas/v1/infoprenda?id=[id] |

*Salida:*

|  |
| --- |
| {  "controlExcepcion": {  "codigo": "NMP-API-REDIS-200",  "mensaje": "Resultado Exitoso"  },  "listaPrendasAsociadas": [  {  "tipo": "Otros",  "lote": "",  "idPrenda": "nmp-al-al-108423424",  "numeroRefrendo": 1,  "fechaEmpenio": "17/06/2021",  "valorAncla": 1196,  "valorMonte": 6220,  "elementosComplementarios": "CON ACCION",  "gramaje": 8,  "prestamo": 5910,  "descripcionPrenda": "2 NUEVA CLASIFICACION NUEVO TIPO ESFERAS DE ORO DE 12 KILATES Y PESO DE 8.00 GRAMOS CON ACCION",  "clasificacion": "Otros",  "cantidad": 2,  "kilataje": 12,  "idTipo": "",  "disenio": "esferas",  "idDisenio": "",  "idClasificacion": "",  "folio": "Infoprenda-22170-23-1628783013127"  }  ]  } |

***Solicitud de alta de partida***

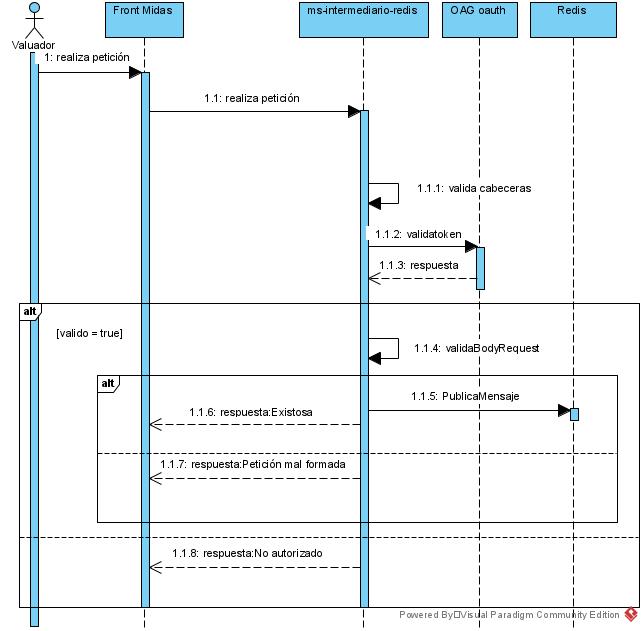
******

Figura 5. Almacenar partida.

***Recuperar partidas***

***Diagrama

Descripción generada automáticamente con confianza media***

Figura 6. Recuperar partida.

## Desglose Técnico.

Para este desglose.

### Lenguaje, framework y empaquetado.

|  |  |
| --- | --- |
| Tipo | Detalle |
| Lenguaje | Java Script |
| Framework | node 12.18.3 |
| Build | npm 6.14.6 |

### Codebase.

|  |  |
| --- | --- |
| Tipo | Detalle |
| Repositorio | https://github.com/MontePiedadMx/MS\_Intermediario\_Redis.git |
| Branch de desarrollo | ftr-1.0.0-26Agt21-Infoprenda-Redis-SPS |
| Usuario de lectura/escritura | JuanCarlosGarciaLopez |

### Dependencias.

A continuación, se desglosan las dependencias definidas en el ***package.json*** del microservicio:

|  |  |
| --- | --- |
| ArtifactId | Versión |
| jsonschema | 1.2.5 |
| log4js | 6.3.0 |
| node-fetch | 2.6.1 |
| redis | 3.1.2 |
| sonarqube-scanner | 2.6.0 |
| nock | 12.0.3 |
| newrelic | 6.14.0 |
| cfenv | 1.2.2 |

#### Configuraciones en plataforma de despliegue

La plataforma de despliegue del microservicio es Bluemix Cloud Foundry.

|  |  |  |  |
| --- | --- | --- | --- |
| 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 | URL del servicio de redis |  | 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 | e48ef5cd64e4148fa6c24131024ed3d28dec4964 | 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 |

Para verificar las variables asociadas a la aplicación en la plataforma de bluemix Cloud Foundry, hacer uso del comando ***cf env*** ***<app\_name>*** desde línea de comandos.

### Backing services.

El microservicio se comunica con los siguientes servicios externos:

* OAG ValidarToken: Para la validación del token.
* Redis: Instancia como servicio creada en bluemix.

### Construir, desplegar, ejecutar.

Para la construcción, despliegue y ejecución del microservicio generar intermediario redis. En la plataforma de Bluemix Cloud Foundry, se hará de forma automatizada.

Los pasos para ejecutar el proceso de instalación se encuentran detallados en el documento “***NMP\_Infoprenda\_MsIntermediario\_Redis\_Manual\_de\_Instalacion\_Ejecucion\_IBMCloudv1.0.docx***”.

#### Asignación de puertos.

El microservicio expone un puerto http sobre el que escucha y atiende a las peticiones, este puerto por defecto es el ***6001***, en caso de no ser asignado algún otro desde variables de ambiente.

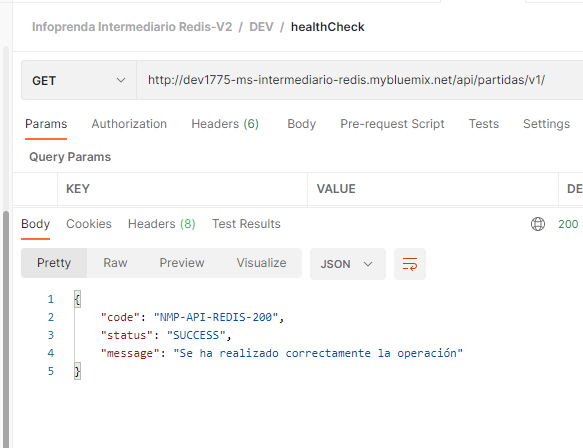
#### Concurrencia.

La concurrencia se maneja aumentando el número de procesos en ejecución, dado que el microservicio no maneja estados, este tipo de concurrencia es viable.

### Observabilidad.

El microservicio expone un entpoint para validar si este se encuentra disponible:

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



### Seguridad.

El microservicio requiere se especifique una cabecera de autorización con un ***Token*** ***Bearer y las cabeceras de idConsumidor y idDestino***. Con este token se valida sobre un servicio de OAuth que se tengan permisos para consumir los recursos expuestos por el microservicio. El servicio de OAuth varia su URL dependiendo del ambiente en el que se encuentre desplegada la aplicación.

* oauth.bearer
* idConsumidor
* idDestino

# Conclusiones.

Con este documento se cubren todos los detalles técnicos que pueden requerirse conocer sobre el microservicio generar intermediario redis.

****