MIDNI – Servicio nube
Manual de Integración

©Izenpe s.a. 2026

Este documento es propiedad de Izenpe, s.a. y su contenido es confidencial. Este documento no puede ser reproducido, en su totalidad o parcialmente, ni mostrado a otros, ni utilizado para otros propósitos que los que han originado su entrega, sin el previo permiso escrito de Izenpe, s.a.. En el caso de ser entregado en virtud de un contrato, su utilización estará limitada a lo expresamente autorizado en dicho contrato. Izenpe, s.a. no podrá ser considerada responsable de eventuales errores u omisiones en la edición del documento.

Histórico de versiones

Versión	Fecha	Resumen de los cambios producidos
1.0	20/02/2026	Primera versión
2.0	07/04/2026	Actualización con parámetros opcionales configurables en la transacción
3.0	14/05/2026	Parámetros de modo de integración.
4.0	19/05/2026	Añadido nuevo dispositivo para captura

Contenido

1 Introducción

1.1 Objeto del documento

MIDNI – Servicio nube es un servicio ofrecido por **IZENPE** para facilitar la captura, validación y extracción de datos de QR generados a través de la aplicación MIDNI[1] de la DGP.

El objetivo de este documento es describir este servicio de **IZENPE** desde el punto de vista del integrador, detallando los elementos y conceptos que es necesario conocer para poder integrar este servicio en aplicación y servicios de terceros. Es decir, el objetivo de este documento no es detallar el funcionamiento interno del servicio, sino describirlo a nivel de caja negra con la que se tiene que interactuar.

Para alcanzar el objetivo detallado en el parrafo anterior la sección *2 Integración* estara dividida en cuatro subsecciones

- ***2.1 Autenticación y autorización***: En esta sección se describirá el proceso mediante el cual un usuario es autenticado y autorizado para hacer uso del servicio
- ***2.2 Creación y configuración de transacción***: En esta sección se describira como, un usuario autenticado, puede configurar un transacción del lectura y validación de código QR.
- ***2.3 Ejecución de transacción***: En esta sección se describirá como se inicia el proceso de lectura definido en la seccion anterior
- ***2.4 Recuperación de datos***: En esta sección se describirá como recuperar los datos de una transacción una vez se haya recibido el callback de respuesta

[1] **MIDNI**: https://www.midni.gob.es/

2 Integración

En este apartado se describen los elementos del servicio que son necesarios conocer para poder interactuar con el mismo. Antes de entrar en las secciones especificas de la integración conviene aclarar unos conceptos para poder entender el correcto funcionamiento de los mecanismos que se van a describir en esta sección.

MIDNI – Servicio nube permite a usuarios registrados ejecutar transacciones de lectura y válidación de QR/MIDNI. En las frase anterior se han subrayado dos conceptos que es importante definir para poder entender correctamente los mecanismos de integración que se describirán en este documento.

Se entenderá por **usuario resgistrado** a cualquier entidad registrada por **IZENPE** a la que se le haya dados permisos para poder ejecutar transacciones. La descripción del proceso de registro de usuarios y los requisitos necesarios para el registro queda fuera del alcance de este documento.

Lo que si es necesario saber de los usuarios registrados desde el punto de vista del integrador es lo siguiente

1. Se autentican mediante un API-Key (generado por **IZENPE**) y password
2. Solo pueden pedir el retorno de control a un conjunto discreto de callbacks acordados con **IZENPE**

Por su parte, una **transacción** representa un flujo completo desde que la aplicación invocante delega el control **MIDNI – Servicio nube** (para que inicie la cámara o el lector, lea el QR, extraiga los datos) hasta que este servicio devuelve el control al callback del invocante.

2.1 Autenticación y autorización

En esta sección se detallan los mecanismos que el integrador dispone para autenticar y autorizar a los usuarios. Este proceso se basa en canjear el API-Key y el password por un token JWT. Para ello se invocará al siguiente *endpoint* propiorcionando los siguientes parametros en el cuerpo de la llamada

endpoint: POST /izwspub/midni/auth/apikey
parametros:
apikey: El API-Key del usuario
password: El password del usuario

```
POST https://servicios.izenpe.eus/izwspub/midni/auth/apikey
{
   "apikey": "b6eae8f3-258b-4c48-8d2c-3045429c4c1d"
   , "password": "secret-password"
}
```

Si este par de datos define un usuario registrado en el sistema se retornará una respuesta **HTTP 201 – CREATED** en cuyo cuerpo se proporciona un token JWT. Será necesario presentar este token en las futuras interacciones con **MIDNI – Servicio nube** para que se autorice al usuario a crear, ejecutar y recuperar los datos de las transacciones.

```
HTTP/1.1 201 CREATED
eyJraWQiOiJpendzcHViLW1pZG5pIiwi…
```

Si este par de datos no define a ningún usuario registrado en el sistema se retornará un respuesta **HTTP 401 – UNAUTORIZED**

2.2 Creación y configuración de transacción

En esta sección se detalla como se crean y configuran las transacciones que se ejecutarán en **MIDNI – Servicio nube**. Para que el usuario esté autorizado para crear la transaccion tiene que presentar el token JWT recibido durante el proceso de ***2.1 Autenticación y autorización*** como *Bearer* en la propiedad *Authorization* de la cabecera de la petición

endpoint: POST /izwspub/midni/transaccion
cabecera:
Authorization: El token JWT dek paso de autenticación
cuerpo:
*callback***:** La URL de callback a la que retornar el control al terminar de ejecutar la transacción
lang: El idioma de la transacción. A eleigir entre ES o EU
parametros: Permite ajustar configuraciones adicionales de la transacción. Su uso es opcional; en caso de no indicarse, cada parámetro aplicará su valor por defecto. Los parámetros disponibles son:

tiposPermitidos: Este parámetro define mediante una lista qué tipos de MIDNI están habilitados para ser utilizados en la validación de una transacción concreta. Si no se indica, se utilizarán todos los tipos disponibles por defecto. Los valores esperados son: **SIMPLE**, **COMPLETA**, **MAYORIA_EDAD**. Si un usuario intenta realizar la validación con un tipo de MIDNI que no está incluido en la lista, el proceso finalizará devolviendo un error controlado, sin continuar con la validación.

verDatosError: Este parámetro determina si se deben mostrar o no los datos del usuario en caso de que se haya procesado un QR válido pero se hayan detectado errores de validación. Los valores esperados son: **true** o **false**. Si no se indica, el valor por defecto es false.

dispositivo: Define el tipo de dispositivo que se va a utilizar para la captura del documento QR. Los valores esperados son: **CAMARA**, **LECTOR** o **FICHERO**. Si no se indica, el valor por defecto es CAMARA.

modoIntegración: Este parametros contienen las propiedades del modo de integración. Es decir, parametros que afectan a como se muestra o se comporta el flujo de identificación contra MIDNI. Dentro de este parametro se pueden encontrar los siguientes subparametros.

enIframe: Es un booleano que representa si el flujo se va a integrar dentro de un *iframe*. El valor por defecto si no se especifica será *false*. Si se expecifica parametro con el valor *true* las paginas HTML que se usen durante el flujo no mostrarán la cabecera y pie.

conDescargaEvidencias: Es un booleano que representa si durante el flujo de validación, en la pantalla que se muestran los datos obtenidos del QR enc aso de validación satisfactoria, se muestra el boón que permite descargar los datos de QR en bruto.

```
POST https://servicios.izenpe.eus/izwspub/midni/transaccion
Authorization: Bearer eyJraWQiOiJpendzcHViLW1pZG5pIiwi…
{
   "callback": "https://serviciosdes.local.com/izwspub/midni/view/cliente/prepararCallback"
    , "lang": "ES"
    , "parametros": {
        "tiposPermitidos": ["COMPLETA", "SIMPLE"]
        , "verDatosError": false
        , "dispositivo": "CAMARA"
        , "modosIntegracion": {
            "enIframe": false
            ,"conDescargaEvidencias": true
        }
    }
}
```

Si los datos proporcionados son correctos y se puede crear la transacción se obtendra una respuesta **HTTP 201 – CREATED** en cuyo cuerpo se proporcionarán los datos de la transacción creada.

```
HTTP/1.1 201 CREATED
{
    "uuid": "cd47c64d-1f26-479a-a537-30bc68cf7fb3",
    "idUsuario": 41,
    "creada": "2026-04-07T17:33:30.382",
    "data": {
        "callback": "https://serviciosdes.local.com/izwspub/midni/view/cliente/prepararCallback",
        "lang": "ES",
        "parametros": {
            "tiposPermitidos": [
                "COMPLETA",
                "SIMPLE"
            ],
            "verDatosError": false,
            "dispositivo": "CAMARA"
        }
    }
}
```

Se usará el UUID de la transacción que se ha retornado, cuando la transacción ha sido creada de forma satisfactoria, para iniciar la ejecución y obtener los datos de la respuesta cuando se retorne el control tras el *callback*.

Si alguno de los datos proporcionados en el cuerpo de la petición no son válidos se devolverá una respuesta **HTTP 400 – BAD REQUEST** que contiene un codigo de error detallando el motivo por el que se considera que la petición no es válida.

```
HTTP/1.1 400 BAD REQUEST
CALLBACK_NULO
```

Los potenciales errores que se pueden recibir de de este código de error pueden ser

Código error	Descripción
CALLBACK_NULO	Si no se proporciona ningun callback en el cuerpo de la petición
IDIOMA_INVALIDO	Si el idioma de la aplicaicón no esta entre los idiomas permitidos

Si el token JWT proporcionado no es válido o no define a un usuario registrado en el sistema se obtendra una respuesta **HTTP 403 – FORBIDDEN**.

```
HTTP/1.1 201 CREATED
{
    "timestamp": "2026-02-20T09:51:31.070+00:00",
    "status": 403,
    "error": "Forbidden",
    "path": "/izwspub/midni/transaccion"
}
```

Si, aun siendo la petición válida, hay algun motivo por el que no se pueda crear la transacción, se devolverá una respuesta **HTTP 409 – CONFLICT REQUEST** que contiene un codigo de error detallando el motivo por el que no se ha podido crear la transacción.

```
HTTP/1.1 409 CREATED
CALLBACK_INVALIDO
```

Los potenciales errores que se pueden recibir de de este código de error pueden ser

Código error	Descripción
CALLBACK_INVALIDA	Si el callback proporcionado en la petición no es un callback registrado para el usuario que esta creando la transacción

2.3 Ejecución de transacción

Una vez se dispone del JWT del proceso de ***2.1 Autenticación y autorización*** y del UUID de transacción de la ***2.2 Creación y configuración de transacción***, se tiene todos los elementos necesarios para delegar el control de la ejeucución a **MIDNI – Servicio nube**.

Para ello será necesario navegar a la URL de inicio del proceso proporcionando como queryParam los siguientes parametros

- **uuid**: Para identificar la transacción que se desea ejecuatr
- **jwt**: Para autorizar la ejecución
- **lang**: Para indicar el idioma en el que se desea mostrar las pantallas

```
https://servicios.izenpe.com/izwspub/midni/view/servidor/prepararLectorQr
      ?uuid=115c8f62-3d4c-45b4-8efe-4e76fa8b13e6
      &jwt=eyJraWQiOiJpendzcHViLW1pZG5pIiwi…
      &lang=eu
```

Esta nevagación se puede realizar tando desde un formulario en una pagina, como desde un redirect desde un servidor. En cualquierá de los casos, a partir de este momento y, hasta que se retorne el control al *callback*, **MIDNI – Servicio nube** tomará el control de lo que se muestra en el navegador del usuario.

AL finalizar la transacción **MIDNI – Servicio nube** devolvera el control redirigiendo el navegador a la URL proporcionada en el *callback* de la transacción.

2.4 Recuperación de datos

MIDNI – Servicio nube informará de que la transacción ha finalizado redirigiendo la navegación al *callback* proporcionado durante la configuración de la transacción. Dado que la URL del *callback* puede estar compartida por multiples transacciones, **MIDNI – Servicio nube** añadirá como queryParam el UUID de la transacción que se ha ejecutado.

Es decir, la URL a la que se devolverá el control será algo similar a

```
https://dominio.usuario.com/callback-proporcionado
      ?uuid=115c8f62-3d4c-45b4-8efe-4e76fa8b13e6
```

Se supondrá que el invocate dispone de un servicio accesible desde **MIDNI – Servicio nube** que exponiendo la ruta proporcionada en el *callback* como un *endpoint* de tipo GET preparado para recepcionar las finalizaciones de las transacciónes, recuperar los datos obtenidos de las mismas y realizar cualesquiera procesos internos que deba hacer en sus sistemas.

Para recuperar los datos de la transacción **MIDNI – Servicio nube** proporciona los siguientes *endpoint* de servicios REST.

RECUPERACIÓN DE DATOS DE TRANSACCIÓN

Tras recuperar el control tras ejeuctar una transacción, el primer caso es comprobar el estado en el que ha finalizado la misma. Para recuperar los datos de una transacción se dispone del siguiente *endpoint* REST

endpoint: GET /izwspub/midni/transaccion/:uuid
parametros:
uuid (QueryParam obligatorio): El UUID que identifica a la transacción
cabecera:
Authorization: El token JWT dek paso de autenticación

```
GET https://servicios.izenpe.eus/izwspub/midni/transaccion/115c8f62-3d4c-45b4-8efe-4e76fa8b13e6
Authorization: Bearer eyJraWQiOiJpendzcHViLW1pZG5pIiwi…
```

Si se puede identificar una transacción con el UUID proporcionado, y se dispone de permisos suficientes como para porder acceder a ella, se retornará una respuesta **HTTP 200 – OK** en cuyo cuerpo se proporcionarán los datos de la transacción en formato JSON.

```
HTTP/1.1 201 CREATED
{
    "uuid": "cd47c64d-1f26-479a-a537-30bc68cf7fb3",
    "idUsuario": 41,
    "creada": "2026-04-07T17:33:30.382",
    "status": true,
    "responseUuid": "c3f6e66e-7738-423e-b383-17b9872927b9",
    "qrUuid": "12ab600b-5434-444f-81cc-d1bb69e39c1d",
    "data": {
        "callback": "https://serviciosdes.local.com/izwspub/midni/view/cliente/prepararCallback",
        "lang": "ES",
        "parametros": {
            "tiposPermitidos": [
                "COMPLETA",
                "SIMPLE"
            ],
            "verDatosError": false,
            "dispositivo": "CAMARA"
        }
    }
}
```

El JSON contendrá la siguiente información

- **uuid**: EL UUID de la transacción
- **idUsuario**: El identificador del usuario que solicito la transacción
- **creada**: La fecha de creación de la transacción ebn formato ISO 8601[2]

[2] **ISO 8601**: https://www.iso.org/iso-8601-date-and-time-format.html

- **status**: Booleano que representa si la transacción ha finalizado correctamente. Si tiene el valor *true* se entenderá que la transacción ha finalizado sin problemas. Si tiene el valor *false* se entenderá que la transacción no ha finalizado correctamente.

- **responseUuid**: El UUID que identifica el fichero que contiene la respuesta de la transacción. El contenido del fichero puede variar en función del valor de campo **status.**
    - Si **status** es true, el contenido del fichero muestra los datos de la válidación
    - Si **status** es false, el contenido del fichero muestra el error que se ha producido
- **qrUuid**: El UUID del fichero que contiene los bytes del QR en bruto codificados en base 64
- **data**: Contiene los tados de configuración con los que se solicito la transacción en el momento en el que fue creada
    - **callback**. El *callback* al que se tendría que retornar el control
    - **lang**. El idioma de ejecución para la transacción
    - **parametros**: Contiene los parametros adicionales para la transacción. En caso de que al crear la transacción no se hayan establecido, se utilizarán los valores por defecto.
        - **tiposPermitidos**: Tipos de datos permitidos se permiten en la transacción.
        - **verDatosError**: Indica si se mostrarán los datos del usuario aunque haya errores de validación
        - **dispositivo**: Tipo de dispositivo utilizado para la captura del QR.

Si no se proporciona el token JWT del usuario en la cabecera, se retornará un código de error **HTTP 401 – UNAUTHORIZED** como respuesta.

Si se proporciona un token JWT del usuario en la cabecera, pero (1) este token no es válido o (2) se esta solicitando datos de una transación que no pertenece al usuario, se retornará un código de error **HTTP 403 – FORBIDDEN** como respuesta.

```
HTTP/1.1 201 CREATED
{
    "timestamp": "2026-02-20T09:51:31.070+00:00",
    "status": 403,
    "error": "Forbidden",
    "path": "/izwspub/midni/transacción/3c737033-d8b1-4e9b-a53c-5b26eac6ff91"
}
```

Si no se puede identificar ninguna transacción con el UUID proporcionado, se retornará un código de error **HTTP 404 – NOT FOUND** como respuesta.

RECUPERACIÓN DE FICHEROS DE TRANSACCIÓN

En la sección anterior se detalla como se pueden recuperar los datos de una transacción. En estos datos se puede comprobar que hay dos ficheros asociados a la transacción

- Los datos en bruto del QR leeido a través de la camara, lector o fichero subido (**qrUuid**).
- Los datos de validación del QR o en su defecto el error producido durante el proceso de validación (**responseUuid**)

Todos estos ficheros estan rerpesentados en formato JSON. Para leer estos ficheros se expone, entre los servicios de **MIDNI – Servicio nube** el siguiente *endpoint*.

endpoint: GET /izwspub/midni/transaccion/:uuidTransaccion/file/:uuidFichero
parametros:
uuidTransaccion (QueryParam obligatorio): El UUID que identifica a la transacción
uuidFichero (QueryParam obligatorio): El UUID de un fichero de la transaccion
cabecera:
Authorization: El token JWT dek paso de autenticación

```
GET https://servicios.izenpe.eus/izwspub/midni/transaccion/115c8f62-3d4c-45b4-8efe-4e76fa8b13e6
     /file/115c8f62-3d4c-45b4-8efe-4e76fa8b13e6
Authorization: Bearer eyJraWQiOiJpendzcHViLW1pZG5pIiwi…
```

Si se puede identificar un fichero con el UUID de transacción y el UUID de fichero proporcionado, y se dispone de permisos suficientes como para porder acceder al fichero, se retornará una respuesta **HTTP 200 – OK** en cuyo cuerpo se proporcionarán los datos del fichero en formato JSON. Al final de este apartado se detallarán los psobles ficheros que se pueden obtener.

Si no se proporciona el token JWT del usuario en la cabecera, se retornará un código de error **HTTP 401 – UNAUTHORIZED** como respuesta.

Si se proporciona un token JWT del usuario en la cabecera, pero (1) este token no es válido o (2) se esta solicitando datos de una transacción y fichero que no pertenece al usuario, se retornará un código de error **HTTP 403 – FORBIDDEN** como respuesta.

```
HTTP/1.1 201 CREATED
{
    "timestamp": "2026-02-20T09:51:31.070+00:00",
    "status": 403,
    "error": "Forbidden",
    "path": "/izwspub/midni/transacción/3c737033-d8b1-4e9b-a53c-5b26eac6ff91
                /file/3c737033-d8b1-4e9b-a53c-5b26eac6ff91"
```

Si no se puede identificar ninguna transacción con el UUID proporcionado, se retornará un código de error **HTTP 404 – NOT FOUND** como respuesta.

Mediante este metodo se pueden obtener tres tipos de ficheros distintos. A continuación se detalla la estructura de cada uno de estos tipos de fichero.

Contenido en bruto del QR. En este fichero se retorna el contenido binario en bruto del QR leido a través de la camara. El UUID de este fichero esta asignado a la propiedad **qrUuid** de los datos recibidos de la transacción.

```
HTTP/1.1 200 OK
{
    "qr":"3AN1gXWeqbU0sFJrcw5aMCduUtlrN3…"
}
```

El JSON contendrá la siguiente información

- **qr**: El contenido del QR en base64

Los datos de validación del QR. Cuando el valor de **status** retornado en los datos de la transacción tiene el valor de **true**, en la propiedad **responseUuid** se retornará la referencia al fichero que contiene los datos de válidación del QR. Dada la cantidad de datos que puede contener este JSON se detallará su estructura en el ***Anexo B: Estructura de datos de validación***.

Los datos del error producido en la transacción. Cuando el valor de **status** retornado en los datos de la transacción tiene el valor de **false**, en la propiedad **responseUuid** se retornará un JSON detallando el error que se ha producido en la transacción.

El contenido del JSON se detalla en el ***Anexo C: Estructura de errores de proceso***. Sin embargo, entre todos los datos que se pueden obtener de un error el elemento que identifica la casuistica del error viene proporcionado en la propiedad código de error. Dependiendo del error que se haya producido este código puede variar.

Código error	Descripción
UUID_TRANSACCION_NULO	No se ha proporcionado el UUID de la transacción que se desea ejecutar
QR_NULO	No se ha podido obtener los bytes del QR que se desea validar
USUARIO_NO_AUTORIZADO	(1) No se a proporcionado un token JWT, o (2) el token proporcionado no es válido o, aun siendo válido, no define a un usuario válido registrado en el sistema
TRANSACCION_NO_ENCONTRADA	El UUID de transacción proporcionado no define una transacción registrada en el sistema
TRANSACCION_NO_PERMITIDA	Aún existiendo una transacción con el UUID proporcionado, esta no pertenece al usuario definido con el JWT proporcionado

NOTA: Cabe mencionar que pueden darse errores en las invocaciones a otros servicio REST internos. En caso de error en estos servicios se detallará en cual de estos servicios se ha producido el fallo.

Anexo A: Estructura de datos de validación

```
HTTP/1.1 200 OK
{
    "datos":{
        "cabecera":{
            "magicNumber":"0xDC"
            ,"version":"V04"
            ,"pais":"ES"
            ,"identificadorFirmante":"ESPN20"
            ,"referenciaCertificado":<<Número de serie del certificado con el que se ha firmado el QR>>
            ,"fechaEmision":<<fecha de emisión del QR en formato ISO 8601>>
            ,"fechaFirma":<<fecha en la que se firmó el QR en formato ISO 8601>>
            ,"referencia":<<tipo de QR solicitado>>
            ,"categoriaTipoDocumento":"DNI_ES_MOVIL"
        },"mensaje":{
            "documento":{
                "dni":<<número del DNI>>
                ,"fechaCaducidad":<<fecha de caducidad del DNI en formato ISO 8601>>
                ,"numeroSoporte":<<número de soporte del DNI>>
            },"poseedor":{
                "nombre":<<Nombre del poseedor del DNI>>
                ,"apellidos":<<Apellidos del poseedor de DNI>>
                ,"sexo":<<Seco del poseedor>>
                ,"fechaNacimiento":<<Fecha de nacimiento del poseedor en formato ISO 8601>>
                ,"nacionalidad":"ESP"
                ,"padreMadre":<<Nombres del padre y madre del poseedor>>
                ,"foto":<<UUID del fichero con los bytes de la foto del poseedor en formato PNG
                            y codificado en base64>>
                ,"direccion":{
                    "completa":<<dirección completa del poseedor>>
                    ,"domicilio":{
                        "linea1":<<Primera línea de la dirección del domicilio del poseedor>>
                        ,"linea2":<<Segunda línea de la dirección del domicilio del poseedor>>
                        ,"linea3":<<Tercera línea de la dirección del domicilio del poseedor>>
                    },"nacimiento":{
                        "linea1":<<Primera línea de la dirección del nacimiento del poseedor>>
                        ,"linea2":<<Segunda línea de la dirección del nacimiento del poseedor>>
                        ,"linea3":<<Tercera línea de la dirección del nacimiento del poseedor>>
                    }
                }
            },"fechaCaducidadDatos":<<Fecha de caducidad de los datos del QR en formato ISO 8601>>
        },"firma":{
            "contenido":<<Contiendo de la firma del QR en base64>>
            ,"estado":<<Estado de al firma del QR>>
            ,"certificado":{
                "dn":<<DN del certificado de la firma>>
                ,"serial":<<Numero de serie del certificado de la firma>>
            }
        }
    },"errores":[
        <<Si hay errores o avisos de validación, listado de códigos de errores o avisos detectados>>
    ]
}
```

Anexo B: Estructura de errores de proceso

```
HTTP/1.1 200 OK
{
    "tipo":<<El tipo de error que se ha producido>>
    ,"aplicacionOrigen":<<Aplicación en la que se ha producido el error>>
    ,"clase":<<Clase en la se produce el error>>
    ,"version":<<Método en el que se produce el error>>
    ,"codigoError":<<Código identificativo del error que se ha producido>>
    ,"throwable":<<La excepción que origina el error>>
    ,"requestEndpoint":<<Endpoint REST invocado que ha retornado un error>>
    ,"responseStatus":<<Código HTTP de respuesta obtenida>>
    ,"responseBody":<<El cuerpo de la HTTP respuesta de error>>
    ,"cause":<<Si este error se ha producido por otro error interno,
                el error interno con la misa estructura que se esta detallando en este punto>>
}
```