



Junta de Andalucía



Cofinanciado por
la Unión Europea



GOBIERNO
DE ESPAÑA



Fondos Europeos



Junta
de Andalucía

**CONTR 2022 671562 Desarrollo y evolución del modelo de
administración digital de la Junta de Andalucía. Lote 3**

Generador de evidencias

GECI - Definición servicios REST

Versión: v01r06
Fecha: 26/08/26

HOJA DE CONTROL

Título	Generador de evidencias		
Entregable	GECI - Definición servicios REST		
Nombre del Fichero	GECI - Definicion servicios RESTv01r06.docx		
Autor	UTE		
Versión/Edición	v01r06	Fecha Versión	26/08/26
Aprobado por		Fecha Aprobación	
		Nº Total Páginas	30

REGISTRO DE CAMBIOS

Versión	Causa del Cambio	Responsable del Cambio	Area	Fecha del Cambio
v01r00	Creación del documento	UTE	UTE	08/11/2024
v01r01	Actualización documento a la versión 1.0.0.0	UTE	UTE	27/01/2025
v01r02	Actualización documento a la versión 1.0.0.1	UTE	UTE	14/05/2025
v01r03	Actualización documento a la versión 1.1.0.0	UTE	UTE	19/08/2025
v01r04	Actualización documento a la versión 1.2.0.0	UTE	UTE	16/10/2025
v01r05	Actualización documento a la versión 1.2.2.0	UTE	UTE	21/07/2026
v01r06	Actualización documento a la versión 1.2.3.0	UTE	UTE	26/08/2026

CONTROL DE DISTRIBUCIÓN

Nombre y Apellidos	Cargo	Area	Nº Copias
José Ignacio Cortés Santos	Jefe de Servicio de Administración Digital	ADA	1
José Andrés García Romero de la Osa	Gabinete estratégico	ADA	1
Almudena López Maraver	Gestor de servicio OTP	PMO	1
Rafael Castillo Saiz	Gestor de proyecto	PMO	1
José Joaquín Belmonte Delgado	Gestor de servicio	UTE	1

ÍNDICE

1	INTRODUCCIÓN.....	5
2	DEFINICIÓN INTERFAZ REST.....	5
3	DEFINICIÓN DE OBJETOS.....	8
4	ANEXO 1: Listado de servicios disponibles.....	12
5	ANEXO 2: Listado de errores.....	13
6	ANEXO 3: Seguridad.....	14
7	ANEXO 4: Ejemplo cliente Java.....	15
8	ANEXO 5: Ejemplo de peticiones Json.....	18
9	ANEXO 6: Ejemplo con Postman.....	24

1 INTRODUCCIÓN

Este documento tiene como objetivo describir la interfaz REST disponible para la integración de aplicaciones con el Generador de Evidencias (GECI), en su versión 1.2.3.0.

El Generador de Evidencias (GECI) es un sistema diseñado para generar documentos en formato PDF a partir de identificadores de consultas previamente realizadas en la plataforma SCSP (Supresión de Certificados en Soporte Papel). Este proceso puede iniciarse desde cualquiera de sus dos componentes principales: los servicios web o la herramienta para usuarios finales (SCSP-Web).

Desde el punto de vista técnico, GECI expone un servicio REST que recibe un conjunto de identificadores y genera los certificados correspondientes en formato PDF. Estos documentos son firmados digitalmente, custodiados en el sistema HCV y se devuelven los códigos CSV asociados para su posterior descarga en HCV.

La generación de certificados puede realizarse de forma individual (un PDF por identificador) o de manera agrupada por número de expediente (varios certificados en un único PDF, si pertenecen al mismo expediente).

2 DEFINICIÓN INTERFAZ REST

La API REST que proporciona el sistema GECI se compone de dos servicios principales:

1. **Generación de certificados:** este servicio permite generar evidencias en formato PDF a partir de un listado de identificadores de consultas previamente realizadas en la plataforma SCSP. La operación se ejecuta de forma asíncrona, devolviendo un identificador de generación de certificado que permitirá consultar posteriormente el estado del proceso. La generación puede realizarse de forma individual (un certificado por identificador) o agrupada por expediente (varios certificados del mismo expediente en un único PDF).
2. **Consulta del estado de generación de certificados:** este servicio permite consultar el estado de una petición de generación a partir del identificador de generación de certificado previamente obtenido. Los posibles estados son:
 - *Pendiente de atender*
 - *En proceso*
 - *Atendida*

Si el estado es *atendida*, la respuesta incluirá un listado con los resultados obtenidos. Dicho listado contendrá los códigos CSV de los certificados PDF generados y disponibles para su descarga en el sistema HCV.

Las URLs de los servicios REST según el entorno:

- Entorno de PRUEBAS:
 1. **Generación de certificados:**
<https://cancanaprun1.chap.junta-andalucia.es/geci/certificados/generarCertificado>
 2. **Consulta de estado:**
<https://cancanaprun1.chap.junta-andalucia.es/geci/certificados/consultarEstado>
- Entorno de PRODUCCIÓN:
 1. **Generación de certificados:**
<https://geci.chap.junta-andalucia.es/geci/certificados/generarCertificado>
 2. **Consulta de estado:**
<https://geci.chap.junta-andalucia.es/geci/certificados/consultarEstado>

A continuación, se detalla la interfaz REST de cada servicio:

Operación:	generarCertificado	
Objetivo:	Iniciar el proceso de generación de certificados	
Método HTTP		Encoding
generarCertificado		UTF-8
Mensaje Entrada		Mensaje Salida
JSON		JSON
Parámetros de entrada		
Tipo		Descripción
PeticonGeneracionCertificado		Objeto petición de solicitud para la generación del certificado. Debe incluir, como campo obligatorio, el listado de identificadores de las consultas SCSP para las que se genera el certificado PDF. En caso de que el tipo de generación corresponda a un certificado agrupado por expediente, será necesario establecer el parámetro "agruparPorExpediente=true". Si dicho parámetro no se especifica, se aplicará el valor por defecto (false).
Parámetros de salida		
Tipo		Descripción

RespuestaGeneracionCertificado	Objeto de respuesta de la generación de certificado. Esta objeto incluye el identificador de petición de generación de certificado, necesario para las invocaciones del servicio “consultarEstado”.
--------------------------------	---

Operación:	consultarEstado
Objetivo:	Consultar el estado en que se encuentra la generación de certificados
Método HTTP	Encoding
consultarEstado	UTF-8
Mensaje Entrada	Mensaje Salida
JSON	JSON
Parámetros de entrada	
Tipo	Descripción
PeticionConsultaEstado	Objeto petición de consulta de estado del certificado. Este objeto debe incluir, como campo obligatorio, el identificador de la petición de generación de certificado (devuelto por el servicio generarCertificado), tanto en generaciones agrupadas por expediente como en generaciones individuales.
Parámetros de salida	
Tipo	Descripción
RespuestaConsultaEstado	Objeto de respuesta de Consulta de Estado

3 DEFINICIÓN DE OBJETOS

Nombre	PeticiónGeneraciónCertificado		
Descripción	Objeto de petición de generación de certificado		
Campo	Tipo	Obl.	Descripción
idsPetición	List<String>	S	Listado de identificadores de las peticiones de consulta realizadas en SCSP para las que se desea generar los certificados PDF. Es obligatorio para cualquier modalidad de generación.
agruparPorExpediente	Boolean	N	Indica si los certificados se generan de forma individual o agrupada por expediente. Si no se especifica, se aplica el valor por defecto "false" y se realiza la generación individual.
expediente	String	N	Número de expediente, utilizado como criterio de agrupación de los certificados. El valor proporcionado se validará frente a los datos presentes en los XML de respuesta de las consultas SCSP. Si se detectan discrepancias, se generarán certificados independientes por cada número de expediente encontrado. Campo obligatorio para modalidad agrupada, se ignorará en modalidad individual.
codigoRPS	String	N	Código del RPS. Se muestra en la portada del certificado agrupado y se utiliza para identificar la finalidad de la consulta, rellenando los campos <i>descripción</i> , <i>finalidad</i> y <i>código procedimiento</i> de la portada. No se emplea como criterio de agrupación, por lo que se permite indicar un valor diferente al presente en los XML de respuesta de las consultas SCSP. Campo obligatorio para modalidad agrupada, se ignorará en modalidad individual.
titularConsulta	Object	N	Objeto que contiene los datos del titular de la consulta (NIF, nombre y apellidos). Se muestra en la portada del certificado agrupado, pero no se emplea como criterio de agrupación, por lo que se permite indicar un valor diferente al presente en los XML de respuesta de las consultas SCSP. Campo obligatorio para modalidad agrupada, se ignorará en modalidad individual.

Nombre	TitularConsulta		
Descripción	Objeto con los datos del titular de la consulta		
Campo	Tipo	Obl.	Descripción
nif	String	S	Documento identificativo del titular.
nombre	String	S	Nombre del titular.
apellidos	String	N	Apellidos del titular.

Nombre	RespuestaGeneracionCertificado		
Descripción	Objeto de respuesta de generación de certificado		
Campo	Tipo	Obl.	Descripción
idGeneracionCertificado	String	S	Identificador de petición de generación de certificado.
mensajeError	MensajeError	N	Mensaje de error, si se produce algún error que impida crear la petición de generación de certificado. Nulo si la operación es correcta.

Nombre	PeticonConsultaEstado		
Descripción	Objeto de petición de consulta estado		
Campo	Tipo	Obl.	Descripción
idGeneracionCertificado	String	S	Identificador de petición de generación de certificado.

Nombre	RespuestaConsultaEstado		
Descripción	Objeto de respuesta de consulta estado		
Campo	Tipo	Obl.	Descripción
idGeneracionCertificado	String	S	Identificador de petición de generación de certificado.
estado	EstadoGeneraciónCertificado	S	Estado en el que se encuentra la petición de generación de certificado.
mensajeError	MensajeError	N	Mensaje de error, si se produce algún error que impida realizar la consulta. Nulo si la operación es correcta.
agruparPorExpediente	Boolean	S	Indica si la generación de los certificados se realiza de forma individual o agrupada por expediente.
resultadosCertificadoIndi	List<Resultado>	N	Listado de resultados cuando el estado es "atendida".

vidual	CertificadoIndi vidual>		Este campo sólo estará relleno si no es una generación de tipo agrupada (agruparPorExpediente=false). En caso contrario, se rellenará el listado resultadosCertificadoAgrupado.
resultadosCertificadoAgr upado	List<Resultado CertificadoAgr upado>	N	Listado de resultados cuando el estado es “atendida”. Este campo sólo estará relleno si es una generación de tipo agrupada (agruparPorExpediente=true). En caso contrario, se rellenará el listado resultadosCertificadoIndividual.

Nombre	EstadoGeneraciónCertificado		
Descripción	Objeto de estado de generación de certificado		
Campo	Tipo	Obl.	Descripción
codigo	String	S	Código del estado. Valores posibles: “0” - Pendiente de atender “1” - En proceso “2” - Atendida
descripcion	String	S	Descripción del estado según el código (“pendiente de atender”, “en proceso” o “atendida”).

Nombre	MensajeError		
Descripción	Objeto de mensajes de error de generación/consulta de certificado		
Campo	Tipo	Obl.	Descripción
codigo	String	S	Código del error. Valores posibles incluidos en anexo 2.
descripcion	String	S	Descripción del error según el código.

Nombre	ResultadoCertificadoIndividual		
Descripción	Objeto de resultado de la generación de certificado individual		
Campo	Tipo	Obl.	Descripción
idGeneracionCertificado	String	S	Identificador de petición de generación de certificado.
idPetición	String	S	Identificador de la petición de consulta realizada sobre SCSP.
csv	String	N	CVS del certificado generado y disponible en HCV. La web

			de HCV utiliza este formato para recuperar el informe.
csvB64	String	N	CVS en base64 del certificado generado y disponible en HCV. Los servicios REST de HCV utilizan este formato para obtener el informe.
mensajeError	MensajeError	N	Mensaje de error, si se produce algún error que impida generar el certificado. Nulo si la operación es correcta.

Nombre	ResultadoCertificadoAgrupado		
Descripción	Objeto de resultado de la generación de certificado agrupado		
Campo	Tipo	Obl.	Descripción
idGeneracionCertificado	String	S	Identificador de petición de generación de certificado.
csv	String	N	CVS del certificado generado y disponible en HCV. La web de HCV utiliza este formato para recuperar el informe.
csvB64	String	N	CVS en base64 del certificado generado y disponible en HCV. Los servicios REST de HCV utilizan este formato para obtener el informe.
mensajeError	MensajeError	N	Mensaje de error, si se produce algún error que impida generar el certificado. Nulo si la operación es correcta.
expediente	String	S	Número del expediente para el cual se ha generado el certificado agrupado.
codigoRPS	String	S	Código RPS para el cual se ha generado el certificado agrupado.
titularConsulta	TitularConsulta	S	Titular de la consulta para el cual se ha generado el certificado agrupado.
idsPetición	List<String>	S	Listado de identificadores de las peticiones incluidas en el certificado agrupado por expediente.

4 ANEXO 1: LISTADO DE SERVICIOS DISPONIBLES

A continuación, se incluye el listado de servicios de consulta disponibles en la versión 1.2.3.0 del sistema.

- AEAT - Consulta de nivel de renta (NIVRENTI)
- AEAT - Obligaciones tributarias - Ayudas y subvenciones (ECOT103)
- CCAA - Certificado Electrónico de Discapacidad (SVDDISCIBS)
- CCAA - Corriente de pago - Ayudas (SVDCCAACPASWS01)
- DGC - Consulta de certificación de titularidad (SVDCTITWS02)
- DGP - Consulta de datos de identidad (SVDDGPCIWS02)
- IGAE - Consulta de Concesiones de subvenciones y ayudas de BDNS (SVDIGAECONCESIONWS01)
- INE - Consulta de datos de residencia (última variación patronal) (SVDREXTFECHAWS01)
- INE - Datos Convivencia Actual (SVDINECONVIVENCIAACTUALWS01)
- INSS - Consulta histórico prestaciones del Registro Prestaciones Sociales Públicas Incapacidad Temporal Maternidad y Paternidad (Q2827002CINSS002)
- SEPE - Consulta de Importes de Prestaciones Percibidas en un Periodo (SVDSEPEIPERWS02)
- SEPE - Consulta de prestaciones percibidas a fecha actual (SVDSEPEIACTWS02)
- TGSS - Consulta de estar al corriente de pago con la Seguridad Social (Q2827003ATGSS001)
- TGSS - Consulta de un periodo de la vida laboral en los últimos 5 años (SVDTGSSVIDALABORAL5ANIOSWS01)

5 ANEXO 2: LISTADO DE ERRORES

A continuación, se incluye el listado de los errores disponibles en la versión 1.2.3.0 del sistema. Este listado se actualizará/ampliará en próximas versiones del sistema.

Errores generales:

- 7000 - Error inesperado
- 7001 - Falta informar campo obligatorio

Errores generación solicitud de certificados:

- 7101 - No ha sido posible crear la solicitud de certificado
- 7102 - La solicitud supera el número máximo de identificadores de peticiones de consultas permitido
- 7103 - La solicitud de certificado contiene identificadores de peticiones duplicados
- 7104 - El NIF del titular no es válido

Errores consulta estado generación de certificados:

- 7201 - No ha sido posible obtener la solicitud de certificado indicada
- 7202 - No ha sido posible obtener el listado de resultados para la solicitud de certificado indicada

Errores generación del certificado PDF:

- 7301 - No ha sido posible generar el certificado para el identificador de petición
- 7302 - No ha sido posible generar el certificado agrupado para el expediente

Errores de usuarios/autenticación:

- 7401 - El usuario autenticado no se encuentra activo
- 7402 - La solicitud de certificado indicada no pertenece al usuario autenticado

6 ANEXO 3: SEGURIDAD

Por motivos de seguridad, todas las peticiones realizadas a la API REST deben incluir un token JWT que permita validar que el usuario/aplicación que realiza la solicitud está autorizada para obtener los certificados correspondientes a las consultas indicadas.

Para ello, las cabeceras de seguridad deberán incorporar un token JWT con la información identificativa del usuario/aplicación que realiza la petición. Dicho token deberá estar firmado utilizando la clave secreta (*secret*) proporcionada durante el proceso de alta del usuario, junto con las credenciales de acceso (*username*, *password* y *client_id*). Además, el token deberá incluir el atributo *iat* (*Issued At*), que indica la fecha y hora de emisión del mismo.

A continuación, se muestra un ejemplo de generación de un token JWT en un cliente Java:

```
Map<String, Object> claims = new HashMap<>();
claims.put("username", username);
claims.put("password", password);
claims.put("client_id", clientId);

byte[] secret = jwtSecretKey.getBytes();

String tokenJWT = Jwts.builder()
    .setClaims(claims)
    .setSubject(username)
    .setIssuedAt(new Date(Calendar.getInstance().getTimeInMillis()))
    .signWith(SignatureAlgorithm.HS256, secret)
    .compact();
```

7 ANEXO 4: EJEMPLO CLIENTE JAVA

La siguiente clase describe un ejemplo simple de cliente Java generado con Spring, para una petición de **generación y consulta de certificados individual**:

```
public RespuestaGeneracionCertificado generarCertificado(String urlGenerarCertificado, String
secretKey, String username, String password, List<String> idsPeticion) {

    RespuestaGeneracionCertificado respuesta = null;

    log.info("Inicio de generarCertificado individual para las peticiones {}", idsPeticion);

    if (idsPeticion != null && !idsPeticion.isEmpty()) {

        PeticionGeneracionCertificado peticion = new PeticionGeneracionCertificado();
        peticion.setIdsPeticion(idsPeticion);

        log.info("Incluyendo Security Header");
        HttpHeaders headers = new HttpHeaders();
        headers.setContentType(MediaType.APPLICATION_JSON_UTF8);
        headers.setAccept(Arrays.asList(MediaType.APPLICATION_JSON));

        String tokenJWT = getTokenJWT(secretKey, username, password);
        headers.set("Authorization", "Bearer " + tokenJWT);

        HttpEntity<PeticionGeneracionCertificado> entity = new HttpEntity<>(peticion,
headers);
        log.info("Security Header incluido en la cabecera");

        RestTemplate restTemplate = new RestTemplate(new
BufferingClientHttpRequestFactory(new SimpleClientHttpRequestFactory()));

        // Opcional
        List<ClientHttpRequestInterceptor> interceptors = new ArrayList<>();
        interceptors.add(new LoggingRequestInterceptor());
        restTemplate.setInterceptors(interceptors);

        restTemplate.getMessageConverters().add(0, new
StringHttpMessageConverter(Charset.forName("UTF-8")));

        log.info("Realizando petición de generación de certificados...");

        respuesta = restTemplate.postForObject(urlGenerarCertificado, entity,
RespuestaGeneracionCertificado.class);

        log.info("Petición realizada.");
    }

    return respuesta;
}
```

```
public RespuestaConsultaEstado consultarEstado(String urlConsultarEstado, String secretKey,
String username, String password, String idGeneracionCertificado) {

    RespuestaConsultaEstado respuesta = null;

    log.info("Inicio de consultarEstado para la petición {}", idGeneracionCertificado);

    if (StringUtils.isNotBlank(idGeneracionCertificado)) {

        PeticionConsultaEstado peticion = new PeticionConsultaEstado();
        peticion.setIdGeneracionCertificado(idGeneracionCertificado);

        log.info("Incluyendo Security Header...");
        HttpHeaders headers = new HttpHeaders();
        headers.setContentType(MediaType.APPLICATION_JSON_UTF8);
        headers.setAccept(Arrays.asList(MediaType.APPLICATION_JSON));

        String tokenJWT = getTokenJWT(secretKey, username, password);
        headers.set("Authorization", "Bearer " + tokenJWT);

        HttpEntity<PeticionConsultaEstado> entity = new HttpEntity<>(peticion, headers);
        log.info("Security Header incluido en la cabecera");

        RestTemplate restTemplate = new RestTemplate(new
        BufferingClientHttpRequestFactory(new SimpleClientHttpRequestFactory()));

        // Opcional
        List<ClientHttpRequestInterceptor> interceptors = new ArrayList<>();
        interceptors.add(new LoggingRequestInterceptor());
        restTemplate.setInterceptors(interceptors);

        restTemplate.getMessageConverters().add(0, new
        StringHttpMessageConverter(Charset.forName("UTF-8")));

        log.info("Realizando petición de consulta de estado...");

        respuesta = restTemplate.postForObject(urlConsultarEstado, entity,
        RespuestaConsultaEstado.class);

        log.info("Consulta realizada.");
    }

    return respuesta;
}
```

```
private String getTokenJWT(String secretKey, String username, String password, String
clientId) {

    log.info("Generando token JWT...");

    Map<String, Object> claims = new HashMap<>();
    claims.put("username", username);
    claims.put("password", password);
    claims.put("client_id", clientId);

    byte[] secret = secretKey.getBytes();

    String tokenJwt = Jwts.builder()
        .setClaims(claims)
        .setSubject(username)
        .setIssuedAt(new Date(Calendar.getInstance().getTimeInMillis()))
        .signWith(SignatureAlgorithm.HS256, secret)
        .compact();

    log.info("Token JWT generado.");

    return tokenJwt;
}
```

8 ANEXO 5: EJEMPLO DE PETICIONES JSON

A continuación, se incluyen ejemplos de peticiones para los servicios generar certificado y consultar certificado:

1. Generar Certificado:

Ejemplo mensaje de entrada – generación individual:

```
{
  "idsPetición": ["R_AFECHA000000000000000001", "DGPCDIV000000000000000002"],
  "agruparPorExpediente": "false",
  "expediente": null,
  "codigoRPS": null,
  "titularConsulta": null
}
```

Ejemplo mensaje de entrada – generación agrupada:

```
{
  "idsPetición": ["ECOTAYU_000000000000000001", "MINHAPCPA0000000000000002"],
  "agruparPorExpediente": "true",
  "expediente": "0000000SC/PRUEBA/12345",
  "codigoRPS": "123456",
  "titularConsulta": {
    "nif": "12345678Z",
    "nombre": "Juan",
    "apellidos": "Español Español"
  }
}
```

Ejemplo mensaje de salida (mismo para ambas modalidades):

```
{
  "idGeneracionCertificado": "1234",
  "mensajeError": null
}
```

2. Consultar Certificado:

Ejemplo mensaje de entrada:

```
{
  "idGeneracionCertificado": "1234"
}
```

Ejemplo mensaje de salida – solicitud pendiente:

```
{
  "idGeneracionCertificado": "1234",
  "estado": {
    "codigo": "0",
    "descripcion": "Pendiente de atender"
  },
  "mensajeError": null,
  "agruparPorExpediente": true,
  "resultadosCertificadoIndividual": null,
  "resultadosCertificadoAgrupado": null
}
```

Ejemplo mensaje de salida – solicitud en proceso:

```
{
  "idGeneracionCertificado": "1234",
  "estado": {
    "codigo": "1",
    "descripcion": "En proceso"
  },
  "mensajeError": null,
  "agruparPorExpediente": true,
  "resultadosCertificadoIndividual": null,
  "resultadosCertificadoAgrupado": null
}
```

Ejemplo mensaje de entrada:

```
{
  "idGeneracionCertificado": "1234"
}
```

Ejemplo mensaje de salida – solicitud individual atendida correcta:

```
{
  "idGeneracionCertificado": "1234",
  "estado": {
    "codigo": "2",
    "descripcion": "Atendida"
  },
  "mensajeError": null,
  "agruparPorExpediente": false,
  "resultadosCertificadoIndividual": [
    {
      "idGeneracionCertificado": "1234",
      "idPetición": "R_AFECHA000000000000000001",
      "csv": "LDVL*****9QJ",
      "csvB64": "bERW*****OVFK",
      "mensajeError": null
    },
    {
      "idGeneracionCertificado": "1234",
      "idPetición": "DGPCDIV000000000000000002",
      "csv": "LDVL*****AC9",
      "csvB64": "bERW*****QUM5",
      "mensajeError": null
    }
  ],
  "resultadosCertificadoAgrupado": null
}
```

Ejemplo mensaje de entrada:

```
{
  "idGeneracionCertificado": "1234"
}
```

Ejemplo mensaje de salida – solicitud agrupada atendida correcta:

```
{
  "idGeneracionCertificado": "1234",
  "estado": {
    "codigo": "2",
    "descripcion": "Atendida"
  },
  "mensajeError": null,
  "agruparPorExpediente": true,
  "resultadosCertificadoIndividual": null,
  "resultadosCertificadoAgrupado": [
    {
      "idGeneracionCertificado": "1234",
      "csv": "LDVL*****LZU",
      "csvB64": "bERW*****FpV",
      "mensajeError": null,
      "expediente": "000000SC/PRUEBA/12345",
      "codigoRPS": "123456",
      "titularConsulta": {
        "nif": "12345678Z",
        "nombre": "Juan",
        "apellidos": "Español Español"
      },
      "idsPetición": ["ECOTAYU_000000000000000001", "MINHAPCPA0000000000000002"]
    }
  ]
}
```

Ejemplo mensaje de entrada:

```
{  
  "idGeneracionCertificado": "1234"  
}
```

Ejemplo mensaje de salida – solicitud individual atendida con errores:

```
{  
  "idGeneracionCertificado": "1234",  
  "estado": {  
    "codigo": "2",  
    "descripcion": "Atendida"  
  },  
  "agruparPorExpediente": false,  
  "resultadosCertificadoIndividual": [  
    {  
      "idGeneracionCertificado": "1234",  
      "idPetición": "R_AFECHA000000000000000001",  
      "csv": "LDVL*****9QJ",  
      "csvB64": "bERW*****OVFK",  
      "mensajeError": null  
    },  
    {  
      "idGeneracionCertificado": "1234",  
      "idPetición": "DGPCDIV300000000000000002",  
      "csv": null,  
      "csvB64": null,  
      "mensajeError": {  
        "codigo": "7301",  
        "descripcion": "No ha sido posible generar el certificado para el identificador  
de petición DGPCDIV300000000000000002: No se ha encontrado el idPetición  
DGPCDIV300000000000000002"  
      }  
    }  
  ],  
  "resultadosCertificadoAgrupado": null  
}
```

Ejemplo mensaje de entrada:

```
{
  "idGeneracionCertificado": "1234"
}
```

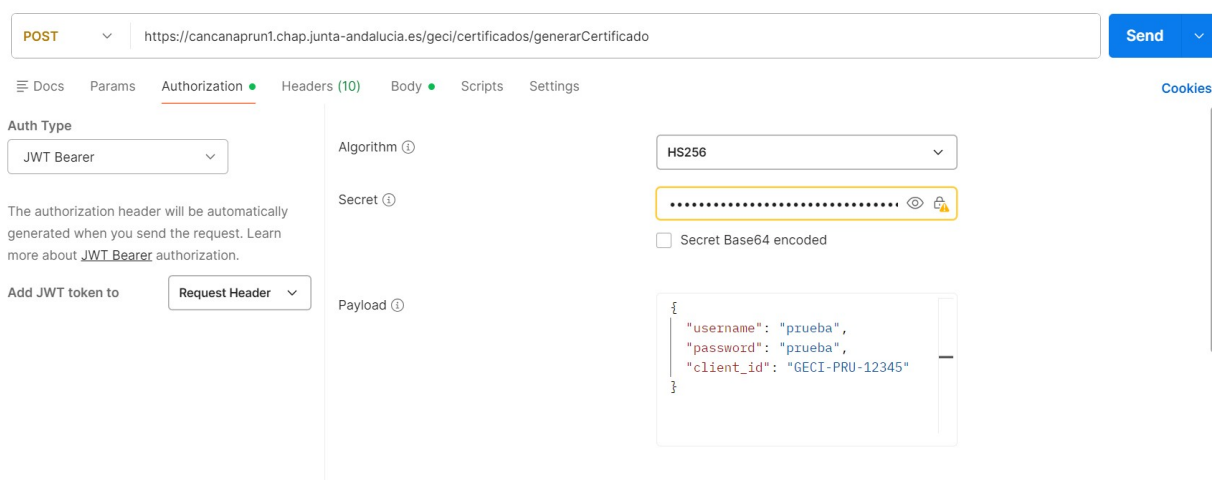
Ejemplo mensaje de salida – solicitud agrupada atendida con errores:

```
{
  "idGeneracionCertificado": "1234",
  "estado": {
    "codigo": "2",
    "descripcion": "Atendida"
  },
  "mensajeError": null,
  "agruparPorExpediente": true,
  "resultadosCertificadoIndividual": null,
  "resultadosCertificadoAgrupado": [
    {
      "idGeneracionCertificado": "1234",
      "csv": "1DVL*****LZU",
      "csvB64": "bERW*****FpV",
      "mensajeError": null,
      "expediente": "000000SC/PRUEBA/12345",
      "codigoRPS": "123456",
      "titularConsulta": {
        "nif": "12345678Z",
        "nombre": "Juan",
        "apellidos": "Español Español"
      },
      "idsPetición": ["ECOTAYU_000000000000000001", "MINHAPCPA0000000000000002"],
    },
    {
      "idGeneracionCertificado": "406",
      "csv": null,
      "csvB64": null,
      "mensajeError": {
        "codigo": "7302",
        "descripcion": "No ha sido posible generar el certificado agrupado para el expediente 000000SC/PRUEBA/09876: No se ha encontrado el idPetición 0123456789ABC; No se ha encontrado el idPetición XYZ9876543210"
      },
      "expediente": "000000SC/PRUEBA/09876",
      "codigoRPS": "123456",
      "titularConsulta": {
        "nif": "12345678Z",
        "nombre": "Juan",
        "apellidos": "Español Español"
      },
      "idsPetición": ["0123456789ABC", "XYZ9876543210"],
    }
  ]
}
```

9 ANEXO 6: EJEMPLO CON POSTMAN

A continuación, se incluyen algunos ejemplos de peticiones para los servicios generar certificado y consultar certificado utilizando la herramienta Postman:

1. Configuración de seguridad:



The screenshot shows the Postman interface for a POST request to the endpoint `https://cancanaprun1.chap.junta-andalucia.es/geci/certificados/generarCertificado`. The request is configured with the following security settings:

- Auth Type:** JWT Bearer. A note states: "The authorization header will be automatically generated when you send the request. Learn more about [JWT Bearer](#) authorization."
- Algorithm:** HS256.
- Secret:** A masked secret key is entered in the field.
- Secret Base64 encoded:** This checkbox is unchecked.
- Payload:** A JSON object is entered in the field:

```
{  "username": "prueba",  "password": "prueba",  "client_id": "GECI-PRU-12345"}
```

The interface also shows tabs for Docs, Params, Authorization, Headers (10), Body, Scripts, and Settings. A "Send" button is visible in the top right corner.

2. Generar Certificado:

2.1. Certificados individuales:

POST Send

Docs Params Authorization Headers (10) Body Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON ☐ Schema Beautify

```
1 {
2   "idsPetición": [
3     "R_AFECHA00000000000000000001",
4     "DGPCDIV00000000000000000002",
5     "TGSS_ECP00000000000000000003"
6   ],
7   "agruparPorExpediente": "false",
8   "expediente": null,
9   "codigoRPS": null,
10  "titularConsulta": null
11 }
```

Body Cookies Headers (11) Test Results 200 OK • 3.16 s • 388 B • Save Response

{ JSON Preview Visualize

```
1 {
2   "idGeneracionCertificado": "1222",
3   "mensajeError": null
4 }
```

2.2. Certificados agrupados por expediente:

POST ▼ <https://cancanapun1.chap.junta-andalucia.es/geci/certificados/generarCertificado> Send ▼

Docs Params Authorization Headers (11) **Body** Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼ Schema Beautify

```
1 {
2   "idsPeticion": ["ECOTAYU_000000000000000001", "MINHAPCPA0000000000000002"],
3   "agruparPorExpediente": "true",
4   "expediente": "00000000SC/PRUEBA/12345",
5   "codigoRPS": "123456",
6   "titularConsulta": {
7     "nif": "12345678Z",
8     "nombre": "Juan",
9     "apellidos": "Español Español"
10  }
11 }
```

Body Cookies (1) Headers (14) Test Results 🕒 **200 OK** • 1.31 s • 542 B • 🌐 📄 Save Response ⋮

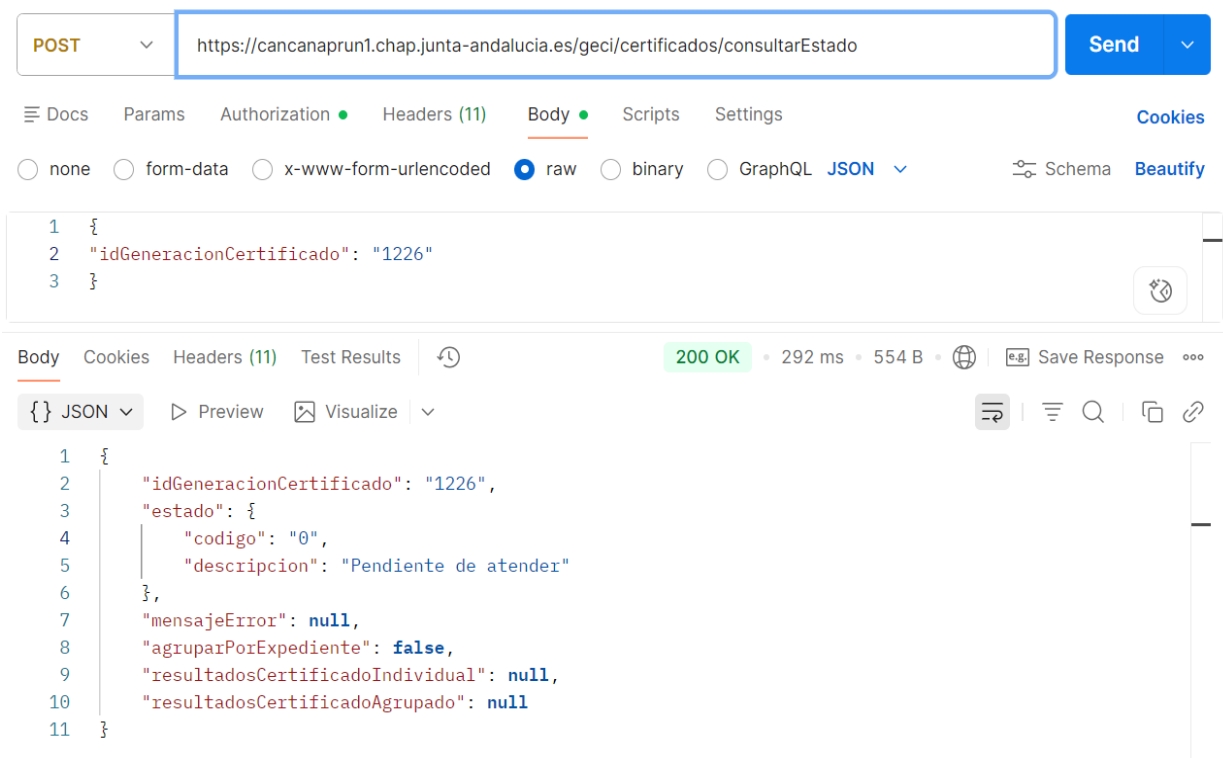
{} JSON ▼ ▶ Preview 🖼️ Visualize ▼ ↩️ ☰ 🔍 📄 🔗

```
1 {
2   "idGeneracionCertificado": "1223",
3   "mensajeError": null
4 }
```

3. Consultar Certificado:

3.1. Certificados individuales

Ejemplo 1 de consulta de certificado individual:



POST Send

Docs Params Authorization Headers (11) Body Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON ☐ Schema Beautify

```
1 {
2   "idGeneracionCertificado": "1226"
3 }
```

Body Cookies Headers (11) Test Results 200 OK • 292 ms • 554 B • Save Response

{ } JSON Preview Visualize

```
1 {
2   "idGeneracionCertificado": "1226",
3   "estado": {
4     "codigo": "0",
5     "descripcion": "Pendiente de atender"
6   },
7   "mensajeError": null,
8   "agruparPorExpediente": false,
9   "resultadosCertificadoIndividual": null,
10  "resultadosCertificadoAgrupado": null
11 }
```

Ejemplo 2 de consulta de certificado individual:

POST ▼ <https://cancanapun1.chap.junta-andalucia.es/geci/certificados/consultarEstado> Send ▼

Docs Params Authorization Headers (11) Body Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON ▼ Schema Beautify

```

1  {
2    "idGeneracionCertificado": "1226"
3  }

```

Body Cookies Headers (11) Test Results 200 OK 271 ms 1.09 KB Save Response

{} JSON Preview Visualize

```

1  {
2    "idGeneracionCertificado": "1226",
3    "estado": {
4      "codigo": "2",
5      "descripcion": "Atendida"
6    },
7    "mensajeError": null,
8    "agruparPorExpediente": false,
9    "resultadosCertificadoIndividual": [
10     {
11       "idGeneracionCertificado": "1226",
12       "idPetición": "R_AFECHA00000000000000000001",
13       "csv": "LDV*****HX6",
14       "csvB64": "bER*****Fg2",
15       "mensajeError": null
16     },
17     {
18       "idGeneracionCertificado": null,
19       "idPetición": "DGPCDIV00000000000000000002",
20       "csv": "LDV*****FYP",
21       "csvB64": "bER*****llq",
22       "mensajeError": null
23     },
24     {
25       "idGeneracionCertificado": null,
26       "idPetición": "TGSS_ECP00000000000000000003",
27       "csv": "LDV*****X8Z",
28       "csvB64": "bER*****Dha",
29       "mensajeError": null
30     }
31   ]
32 }

```

3.2. Certificados agrupados por expediente

Ejemplo 1 de consulta de certificado agrupado:

POST ▼ <https://cancanapun1.chap.junta-andalucia.es/geci/certificados/consultarEstado> Send ▼

Docs Params Authorization Headers (11) **Body** Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼ Schema Beautify

```
1 {
2   "idGeneracionCertificado": "1225"
3 }
```

Body Cookies (1) Headers (14) Test Results 200 OK • 348 ms • 625 B • Save Response

{} JSON ▼ Preview Visualize ▼

```
1 {
2   "idGeneracionCertificado": "1225",
3   "estado": {
4     "codigo": "1",
5     "descripcion": "En proceso"
6   },
7   "mensajeError": null,
8   "agruparPorExpediente": true,
9   "resultadosCertificadoIndividual": null,
10  "resultadosCertificadoAgrupado": null
11 }
```

Ejemplo 2 de consulta de certificado agrupado:

POST ▼ <https://cancanaprun1.chap.junta-andalucia.es/geci/certificados/consultarEstado> Send ▼

Docs Params Authorization Headers (11) Body Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON ▼ Schema Beautify

```

1 {
2   "idGeneracionCertificado": "1225"
3 }

```

Body Cookies Headers (11) Test Results 200 OK 233 ms 910 B Save Response

{} JSON Preview Visualize

```

1 {
2   "idGeneracionCertificado": "1225",
3   "estado": {
4     "codigo": "2",
5     "descripcion": "Atendida"
6   },
7   "mensajeError": null,
8   "agruparPorExpediente": true,
9   "resultadosCertificadoIndividual": null,
10  "resultadosCertificadoAgrupado": [
11    {
12      "idGeneracionCertificado": "1225",
13      "csv": "1DV*****BAY",
14      "csvB64": "bER*****kFZ",
15      "mensajeError": null,
16      "expediente": "00000000SC/PRUEBA/12345",
17      "codigoRPS": "123456",
18      "titularConsulta": {
19        "nif": "123456782",
20        "nombre": "JUAN",
21        "apellidos": "ESPAÑOL ESPAÑOL"
22      },
23      "idsPeticion": [
24        "ECOTAYU_000000000000000001",
25        "MINHAPCPA0000000000000002"
26      ]
27    }
28  ]
29 }

```