



**Servicio Web de
Verificación de Jugadores**

SECRETARÍA GENERAL DE
HACIENDA

DIRECCIÓN GENERAL DE
ORDENACIÓN DEL JUEGO

**Especificación del
Servicio web de Verificación de Jugadores
para operadores de juego
(SVJ-WS)**

Versión 1.8

Abril de 2015



1	Objetivos	9
1.1	Control de cambios del documento	10
1.2	Calendario de implantaciones de nuevos endpoints	15
2	Especificación funcional	16
2.1	Servicio web y operaciones	16
2.2	Parámetros de entrada	21
2.2.1	VerificarJugador, VerificarRGIAJ y VerificarIdentidad	21
2.2.2	Parámetros VerificarCambiosRGIAJ	26
2.2.3	Parámetros bajaJugador	26
2.3	Respuestas del servicio	27
2.3.1	Respuesta para VerificarJugador	27
2.3.2	Respuesta para VerificarRGIAJ	28
2.3.3	Respuesta para VerificarIdentidad	29
2.3.4	Respuesta para VerificarCambiosRGIAJ	29
2.3.5	Respuesta para bajaJugador	30
2.4	Autenticación del Operador	30
3	Descripción de los entornos	35
3.1	Producción	35
3.2	Pruebas	36
4	Anexos	37
4.1	Códigos de resultado y error	37
4.2	Juegos de ensayo en el entorno de pruebas	39
4.2.1	Descripción de los casos de Prueba	40
4.2.2	Relación de casos de error genéricos	45
4.2.2.1	Datos Obligatorios no enviados	45
4.2.2.2	Autenticación del Operador	45
4.2.2.3	DNI o NIE repetidos	46



4.2.2.4	Validación de formato del DNI/NIE	46
4.2.2.5	Validación de letra del DNI/NIE	46
4.2.2.6	Validación de Apellido2 obligatorio si es un NIF.....	46
4.2.2.7	Validación de fecha de nacimiento posterior al 01/01/1900	46
4.2.2.8	Chequeo de caracteres no válidos	46
4.2.2.9	El servidor del RGIAJ no está disponible	46
4.2.2.10	El servidor del SVDI no está disponible	47
4.2.2.11	Validación de formato del Número de soporte.....	47
4.2.2.12	Validación de número de soporte con NIF.....	47
4.2.3	Casos de Prueba de la Operación de verificarJugador.....	47
4.2.3.1	Verificar un jugador No OK-EN RGIAJ.....	47
4.2.3.2	Verificar un jugador OK- EN RGIAJ	48
4.2.3.3	Verificar dos jugadores OK- NO RGIAJ	49
4.2.3.4	Verificar dos jugador, uno OK-NO RGIAJ y otro con ERROR FORMATO.....	50
4.2.3.5	Verificar dos jugador OK- EN RGIAJ los dos	51
4.2.3.6	Verificar un jugador OK- NO RGIAJ	53
4.2.3.7	Verificar un jugador OK- NO RGIAJ (pero menor)	53
4.2.3.8	Verificar un jugador con Nif Anulado- NO RGIAJ.....	54
4.2.3.9	Verificar un jugador con Nif Retenido- NO RGIAJ.....	55
4.2.3.10	Verificar un jugador al que le faltan datos obligatorios	56
4.2.3.11	Verificar un jugador OK- EN RGIAJ.....	56
4.2.3.12	Verificar un jugador sin enviar fecha de nacimiento	57
4.2.3.13	Verificar un jugador que NO OK – NO EN RGIAJ	58
4.2.3.14	Verificar el mismo DNI dos veces en la misma petición.....	59



4.2.3.15	Verificar un jugador con un NIE y número de soporte	60
4.2.3.16	Verificar un jugador con un NIE y número de soporte con formato erróneo	63
4.2.3.17	Verificar un jugador con un DNI enviando número de soporte	64
4.2.4	Casos de Prueba de la Operación de verificarIdentidad.....	65
4.2.4.1	Verificar una Identidad OK	65
4.2.4.2	Verificar una identidad No OK.....	65
4.2.4.3	Verificar una identidad con ERROR-FORMATO.....	66
4.2.4.4	Verificar una identidad que no esta correctamente identificada	67
4.2.4.5	Verificar identidad de Nif Anulado	67
4.2.4.6	Verificar un jugador con Nif Retenido.....	68
4.2.4.7	Verificar un jugador con un NIE y número de soporte.....	69
4.2.4.8	Verificar un jugador con un NIE y número de soporte con formato erróneo	71
4.2.4.9	Verificar un jugador con un DNI enviando número de soporte	72
4.2.5	Casos de Prueba de la Operación de verificarRGIAJ	73
4.2.5.1	Verificar en RGIAJ un jugador que está inscrito.....	73
4.2.5.2	Verificar en RGIAJ un jugador que no está inscrito.....	73
4.2.5.3	Verificar un jugador que está inscrito en RGIAJ pero sin enviar todos los datos	74
4.2.5.4	Verificar en RGIAJ un jugador que no está inscrito.....	75
4.2.5.5	Verificar un RGIAJ con ERROR-FORMATO	76
4.2.5.6	Verificar un jugador con un NIE y número de soporte con formato erróneo	77
4.2.5.7	Verificar un jugador con un DNI enviando número de soporte	78



4.2.6	Casos de Prueba de la Operación de verificarCambiosRGIAJ	79
4.2.6.1	Verificar cambios en RGIAJ sin verificar jugadores.....	80
4.2.6.2	Verificar cambios en RGIAJ al ir verificando jugadores.....	81
4.2.7	Casos de Prueba de la Operación de bajaJugador	82
4.2.7.1	Baja un jugador dado de alta por el Operador.....	82
4.2.7.2	Baja de un jugador no dado de alta previamente por el Operador	82
4.2.7.3	Baja de un jugador ya dado de baja previamente por el Operador	83
4.2.7.4	Causa de la baja errónea	84
4.2.7.5	Baja de un jugador enviando un DNI/NIE erróneo	85
4.2.7.6	Baja de un jugador donde no se envía el DNI/NIE	85
4.2.7.7	Baja de un jugador donde no se envía la causa.....	86
4.2.7.8	Verificar cambios en RGIAJ después de una baja	86
4.3	Juegos de ensayo para verificar el entorno de producción	88
4.3.1	Relación de casos de error genéricos.....	88
4.3.1.1	Datos Obligatorios a null	88
4.3.1.2	Autenticación del Operador	88
4.3.1.3	Validación de formato del DNI/NIE	89
4.3.1.4	Validación de letra del DNI/NIE	89
4.3.1.5	Validación de Apellido2 obligatorio si es un NIF.....	89
4.3.1.6	Validación de fecha de nacimiento posterior al 01/01/1900	89
4.3.1.7	Validación de caracteres no permitidos.....	89
4.3.1.8	DNI/NIE repetidos	89
4.3.2	Casos de Prueba de la Operación de verificarJugador.....	90
4.3.2.1	Verificar un jugador No OK-NO RGIAJ	90
4.3.2.2	Verificar un jugador OK- EN RGIAJ	91



4.3.2.3	Verificar dos jugadores OK- NO RGIAJ	92
4.3.2.4	Verificar dos jugador, uno OK-NO RGIAJ y otro con ERROR FORMATO 93	
4.3.2.5	Verificar dos jugador OK- EN RGIAJ los dos	95
4.3.2.6	Verificar un jugador OK- NO RGIAJ	96
4.3.2.7	Verificar un jugador No OK- EN RGIAJ	97
4.3.2.8	Verificar un jugador OK- NO RGIAJ (pero menor)	98
4.3.2.9	Verificar un jugador al que le faltan datos obligatorios	99
4.3.2.10	Verificar un jugador OK- EN RGIAJ	99
4.3.2.11	Verificar un jugador sin enviar fecha de nacimiento	101
4.3.2.12	Verificar un jugador que NO EXISTE el DNI	102
4.3.2.13	Verificar un jugador que da error de Servicio (COD005)	103
4.3.2.14	Verificar el mismo DNI dos veces en una misma petición	104
4.3.3	Casos de Prueba de la Operación de verificarIdentidad	105
4.3.3.1	Verificar una Identidad OK pero menor de edad	105
4.3.3.2	Verificar una Identidad OK	106
4.3.3.3	Verificar una identidad No OK	106
4.3.3.4	Verificar una identidad No OK	107
4.3.3.5	Verificar una identidad con ERROR-FORMATO	108
4.3.3.6	Verificar una identidad con caracteres no permitidos	109
4.3.3.7	Verificar una identidad que NO EXISTE	110
4.3.3.8	Verificar una identidad que da error de Servicio (COD005)	110
4.3.4	Casos de Prueba de la Operación de verificarRGIAJ	112
4.3.4.1	Verificar en RGIAJ un jugador que está inscrito	112
4.3.4.2	Verificar en RGIAJ un jugador que no está inscrito	112



4.3.4.3	Verificar un jugador que está inscrito en RGIAJ pero sin enviar todos los datos	113
4.3.4.4	Verificar en RGIAJ un jugador que no está inscrito.....	114
4.3.4.5	Verificar un RGIAJ con ERROR-FORMATO	115
4.3.5	Casos de Prueba de la Operación de verificarCambiosRGIAJ	116
4.3.5.1	Verificarcambios en RGIAJ sin verificar jugadores.....	118
4.3.5.2	Verificarcambios en RGIAJ al ir verificando jugadores.....	118
4.3.6	Casos de Prueba de la Operación de bajaJugador	120
4.3.6.1	Baja un jugador dado de alta por el Operador.....	120
4.3.6.2	Baja de un jugador no dado de alta previamente por el Operador	120
4.3.6.3	Baja de un jugador ya dado de baja previamente por el Operador	121
4.3.6.4	Causa de la baja errónea	122
4.3.6.5	Baja de un jugador enviando un DNI/NIE erróneo	122
4.3.6.6	Baja de un jugador donde no se envía el DNI/NIE	123
4.3.6.7	Baja de un jugador donde no se envía la causa.....	124
4.3.6.8	Verificar cambios en RGIAJ después de una baja	124
4.4	Formulario de solicitud de alta en el servicio.	125
5	Test del servicio	127
5.1	Objetivos	127
5.2	Especificación funcional.....	127
5.3	Descripción del sistema	127
5.4	Conexión segura	127
5.5	Peticiones de consulta desde un explorador.....	128
5.5.1	Aplicación web.....	128
5.5.2	Mediane peticiones GET con protocolo HTTPS.....	128
5.6	Respuestas del servicio de TEST (TESTOPERADORES).	129
5.6.1	En formato WEB.	129



5.6.2	Con peticiones GET con protocolo HTTPS.....	130
5.7	Ejemplo para utilización del servicio TESTOPERADORES desde un cliente Java.	131
5.8	Requisitos.....	132
5.9	Ejemplos.....	132
5.10	Dependencias.....	133



1 Objetivos

El objetivo de este documento es describir la especificación técnica y funcional del servicio web de verificación de jugadores que ofrecerá la DGOJ a los distintos operadores de juego para verificar la identidad los datos de identidad de los residentes en España en tiempo real, en cumplimiento de lo previsto en el artículo 26.3 del Real Decreto 1613/2011, de 14 de noviembre, por el que se desarrolla la Ley 13/2011, de 27 de mayo, regulación del juego, en lo relativo a los requisitos técnicos de las actividades de juego .

Junto con los datos de identidad, se valida igualmente la fecha de nacimiento del jugador, para facilitar al operador la verificación de la mayoría de edad del participante según lo previsto en el artículo 27.3 del citado Real Decreto.

Por último, estos servicios web facilitan los procedimientos, a los que hace referencia al artículo 27.2, para permitir a los operadores el acceso telemático al Registro General de Interdicciones de Acceso al Juego (RGIAJ) antes de la activación de la cuenta de usuario del participante, así como la posterior verificación periódica de los datos de sus registros de usuario con el RGIAJ.

Para facilitar la implementación del software cliente consumidor de estos servicios web, la Dirección General de Ordenación del Juego pone a disposición de los operadores un entorno de pruebas, con unos juegos de ensayo que se describen en el presente documento, así como el formulario para solicitar el alta en el servicio.



1.1 Control de cambios del documento

Versión	Fecha	Descripción
0.1	SEP 2011	Versión inicial publicada como papeles de trabajo.
0.2	OCT 2011	Se especifica las políticas de firma de los mensajes Se incluyen juegos de ensayo del entorno de pruebas y formulario de solicitud de alta en el servicio.
0.3	NOV 2011	Corrección de errores en la dirección de correo de soporte que es dgoj.soporteoperadores@minhap.es y otros errores de redacción. Se publica el formulario de solicitud de alta en el servicio en un documento independiente.
0.4	DIC 2011	Se mejora la respuesta de la operación VerificarCambiosRGIAJ para contemplar la fecha en que fue introducido el cambio. Se adjunta juego de ensayo y método de test para verificar el funcionamiento de la funcionalidad VerificarCambiosRGIAJ.
0.5	DIC. 2ª 2011	Se incluye una aclaración sobre el formato de la fecha de nacimiento (pág. 8). Aclaración sobre la no obligatoriedad del segundo apellido cuando se identifica con NIE (pág. 8). Aclaración sobre el formato aceptado en los parámetros Nombre, Primer y Segundo Apellido (pág. 8 y 9). Se incluye la dirección de descarga de la parte pública del certificado de la DGOJ (pag. 11) Se corrigen errores en las respuestas de los mensajes de ejemplo en los puntos: 4.2.3.11 4.2.4.4 4.2.5.4 Aclaración sobre el resultado de los casos de prueba en VerificarCambiosRGIAJ (punto 2.b



		página 42)
0.6	FEB. 2012	<p>Se incluye un nuevo tipo de error Fault en el caso de que haya NIF's repetidos en la misma petición.</p> <p>Se restringe el juego de caracteres admitidos en los campos nombre, primer apellido y segundo apellido.</p> <p>Se cambian las cuentas de correo de soporte.</p>
0.7	MAR. 2012	<p>Se hacen recomendaciones sobre el uso de las distintas operaciones del servicio, incidiendo en la operación de VerificarCambiosRGIAJ (punto 2.1)</p> <p>Aclaración sobre el uso de los parámetros de entrada a los servicios (punto 2.2.1).</p> <p>Se detalla la especificación del timestamp de los mensajes (punto 2.4.3).</p> <p>Se aclara la política de firma en relación con los mensajes soap fault (no son firmados) y se incluyen ejemplos de este tipo de mensajes.</p> <p>Aclaración respecto del juego de caracteres permitidos en el nombre y los apellidos (punto 2.2.1).</p> <p>Se modifican los datos de algunos de los casos de prueba.</p> <p>Se establece la obligatoriedad de realizar las peticiones al servicio en el entorno de producción desde unas IP's concretas (máximo 4)</p>
0.8	ABR. 2012	<p>Aclaración gráfica uso adecuado de los diferentes métodos web services (punto 2.1)</p> <p>Aclaración sobre la separación del string y el código de los mensajes soap fault.</p> <p>Se incluye una nueva funcionalidad que permite realizar verificar la correcta configuración y acceso al entorno de producción. Se especifica el juego de pruebas para este entorno.(puntos 3.1 y 4.3)</p>
0.9	ABR. 2ª 2012	<p>Corrección errores redacción uso adecuado de los diferentes métodos web services (punto 2.1)</p>
1.0	JUN 2012	<p>Ampliación de la información sobre el uso adecuado de los diferentes métodos web services</p>



		<p>(punto 2.1)</p> <p>Corrección tabla de código de resultados y error (punto 4.1).</p> <p>Corrección del mensaje de error de "Identidad del usuario incorrecta." añadiendo el literal "-Titular no identificado" en los casos del punto 4.3, cuando el DNI no existe en los datos de prueba.</p> <p>Especificación de las longitudes a 40 caracteres de los campos nombre, apellido1 y apellido 2</p> <p>Especificación del error a devolver en el caso de utilizar la operación VerificarCambiosRGIAJ en el entorno de producción en modo pruebas.</p> <p>Nota sobre las IPs utilizadas en el acceso al sistema en el entorno de producción (punto 3.1)</p>
1.1	SEPT 2012	<p>Aclaración sobre los casos en que VerificarCambiosRGIAJ tiene en cuenta a los jugadores consultados (punto 2.1)</p> <p>Detalle de la diferencia entre respuestas de verificación de identidad incorrectas e imposibilidad de realizar la verificación (punto 4.1)</p> <p>Especificación de un nuevo tipo de error en el caso COD004 (punto 4.1)</p>
1.2	FEB 2013	<p>Aclaración sobre valores de fecha de nacimiento válidos (punto 2.2)</p>
1.3	MAY 2013	<p>Se establecen nuevo formato en las uri's del servicio en producción y pre-producción al objeto de facilitar la implantación progresiva de nuevas versiones en los entornos de producción de los operadores (punto 3.1)</p> <p>Se anuncia la nueva versión del software 2.1 que implica un cambio del entorno tecnológico de la DGOJ, así como de una serie de mejoras, por lo que es necesario realizar pruebas en preproducción por parte de los operadores para verificar la compatibilidad de sus sistemas con la nueva tecnología de la DGOJ, así como de las mejoras funcionales:</p> <p>Se detallan los nuevos códigos de error tipo COD9... (una de las mejoras funcionales) y en los ejemplos de casos de prueba se actualizan los</p>



		<p>códigos y mensajes de respuesta para estos casos (punto 4.1).</p> <p>Se incluyen casos de prueba de la operación de verificarCambiosRGIAJ en el entorno de producción hasta que el operador pueda operar con datos reales (página 53).</p> <p>Se emula el comportamiento real del COD005 cuando se está trabajando en el entorno de producción con casos de prueba (página 58 y ejemplo página 70).</p>
1.4	JULIO 2013	<p>Se incluye un nuevo campo, de uso opcional, (Numsoporte) para el caso de identificación de un NIE, de forma que se soluciona la problemática actual de los casos en que la D.G. de la Policía detecta que hay más de una filiación para dicho NIE. Se modifica la estructura de datos de "Jugador" en el WSDL y se incorporan juegos de prueba para el entorno de Preproducción (punto 2.2).</p> <p>Se incluye una nueva operación en el WS que permite al operador dar de baja a un NIE/NIF. De esta forma, el operador ya no será informado de los cambios en el estado del RGIAJ que dicho NIF/NIE pueda tener a partir del momento de solicitar dicha baja. Se incluyen casos de prueba para el entorno de Preproducción y de Producción. (punto 2.1).</p> <p>Se incluyen casos de prueba en Preproducción y Producción de la operación de VerificarCambiosRGIAJ para comprobar los datos una vez se hayan dado de baja un NIF/NIE que se informaba en dicha operación.</p>
1.5	Octubre 2013	<p>Nuevos códigos y textos de error del SVDI para el caso de que el DNI no se pueda identificar – COD004- (punto 4.1).</p> <p>Se modifican también los mensajes de respuesta de los ejemplos del punto 4.2 del documento en que se dan nuevos valores de respuesta a los textos de porque no se ha podido indentificar al jugador y se añaden nuevos ejemplos con los nuevos motivos.</p> <p>Clarificación sobre los casos de NIEs con 10</p>



		digitos (siempre comenzando por X0 y posteriormente seguido de 7 dígitos más el dígito de control), que se deben a una actuación histórica de la Policía (punto 2.2).
1.6	DIC 2013	Nueva funcionalidad para chequear el estado del servicio (apartado 5)
1.7	SEP 2014	Se añade la validación de la longitud del nombre y apellidos. La longitud no debe superar los 40 caracteres. Se introduce un nuevo código de error, COD907, que se retorna cuando la comprobación de la longitud falla.
1.8	ABRIL 2015	Se actualizan las instrucciones para la validación de la firma de los mensajes de respuesta devueltos por el servidor. Se modifica el endpoint del servicio, donde antes era <code>cnjuego.gob.es</code> pasa a ser <code>ordenacionjuego.gob.es</code> .



1.2 Calendario de implantaciones de nuevos endpoints

Versión documento	versión software	PREPRODUCCION		PRODUCCION	
		fecha implantación	fecha de repliegue (1)	fecha implantación	fecha de repliegue (1)
1.7	2.2	23/09/2013	27/04/2015	21/10/2013	11/05/2015
1.8 (2)	2.2	05/03/2015		06/04/2015	

(1) A partir de la fecha de repliegue se dejará de dar soporte en los entornos y versiones indicadas.

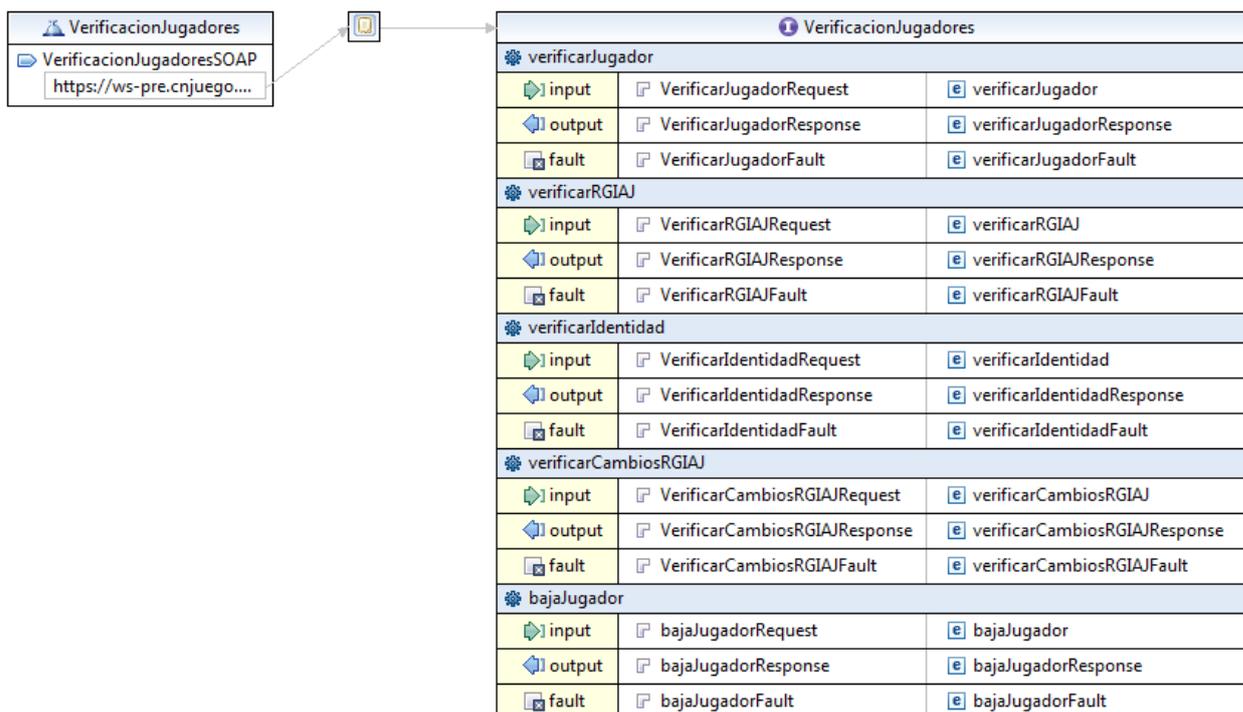
(2) Este cambio corresponde con el cambio del dominio del endpoint de cnjuego a ordenacionjuego y se dejará de admitir el protocolo ssl3 para establecer la conexión a partir de la fecha de repliegue



2 Especificación funcional

2.1 Servicio web y operaciones

A continuación se define la especificación funcional del servicio:



El servicio web de verificación de jugadores dispondrá de cinco operaciones:

1. “VerificarIdentidad”: el operador puede utilizar esta operación para comprobar que el DNI/NIE, nombre y fecha de nacimiento del jugador son acordes a los datos de los que dispone la Administración.

Esta operación no implica el establecimiento de una relación contractual entre el jugador y el operador, ya que se asume que éste ha realizado la consulta en la fase inicial de verificación de los datos de los residentes (artículo 26.3 RD. 1613/2011) durante la apertura de la cuenta de juego, donde aún no se ha activado la cuenta. En consecuencia, cualquier cambio que se produzca en la situación del jugador respecto al RGIAJ no será informado al operador en el servicio VerificarCambiosRGIAJ.



2. “VerificarRGIAJ”: el operador puede utilizar esta operación para comprobar si los jugadores están inscritos en el RGIAJ.

Aunque los parámetros de entrada incluyen nombre, apellidos y fecha de nacimiento, la operación sólo utiliza el DNI / NIE para acceder al RGIAJ.

Al utilizar esta operación, el sistema asume que se establece una relación contractual entre el jugador y el operador, pues el uso típico de esta operación es la verificación de que el jugador no está incurso en las prohibiciones subjetivas, previo a la activación de la cuenta (art. 27.2 R.D. / 1613/2011). En consecuencia, cualquier cambio que se produzca en la situación del jugador respecto al RGIAJ será informado al operador en el servicio VerificarCambiosRGIAJ.

3. “VerificarJugador”: el operador puede utilizar esta operación para realizar verificaciones sobre sus jugadores para determinar si están inscritos en el RGIAJ y también verificar los datos de identidad.

Esta operación esta orientada para aquellos operadores que están interesados, por la forma en que realizan el alta de sus usuarios en su plataforma de juego, en realizar simultáneamente ambas comprobaciones. Dado que internamente son independientes y que la verificación del RGIAJ sólo precisa del DNI / NIE, es posible que se devuelva respuesta respecto al RGIAJ, aunque el servicio de verificación de identidad devuelva una respuesta negativa o no esté disponible. En este caso, si la respuesta de RGIAJ es que el jugador está inscrito, no es necesario que el operador reintente la consulta una vez corregidos los datos de identidad o recuperado el servicio, ya que el jugador en todo caso está afectado por las prohibiciones subjetivas y no puede jugar.

Independientemente de la respuesta de verificación de identidad, cualquier cambio que se produzca en la situación del jugador respecto al RGIAJ será informado al operador en el servicio VerificarCambiosRGIAJ.

4. “VerificarCambiosRGIAJ”: el operador puede utilizar esta operación para recuperar los jugadores respecto a los que el operador realizó en su día alguna comprobación de RGIAJ tanto en el método “VerificarRGIAJ” como en “VerificarJugador” (y en este caso independientemente del resultado exitoso o no de la verificación de la identidad del jugador) y cuyo estado en el RGIAJ ha sido modificado. Esta operación no es útil cuando cualquiera de las operaciones anteriores de como resultado un Error de formato de los datos de la petición, de forma que en esos casos no se asocia el jugador con el operador.

Esta operación facilita los dos objetivos recogidos en el artículo. 27.2 del R.D. 1613/2011:



- a) verificación periódica de los datos de los registros de usuario del operador con los inscritos en el Registro General de Interdicciones de Acceso al Juego con el objeto de impedir que realicen alguna actividad de juego.
- b) verificación de que el jugador no esta incurso en las prohibiciones subjetivas previa al pago de premios (art. 27.2 R.D. 1613/2011).

Debe tenerse en cuenta que al objeto de coordinar esta función con las inscripciones realizadas en el RGIAJ, estas se activan en la siguiente hora a su grabación en la base de datos de la DGOJ. Por ejemplo, todas las inscripciones realizadas entre las 11:00 y las 11:59, tendrán hora de inscripción a las 12:00. A partir de ese momento serán informadas en "VerificarCambiosRGIAJ".

Por tanto, es suficiente con que el operador consulte la operación "VerificarCambiosRGIAJ" una vez cada hora para obtener la relación de cambios en el RGIAJ producidos en los jugadores sobre los que alguna vez han consultado su estado en el RGIAJ (clientes, y por tanto que en su día fueron dados de alta por no estar inscritos en RGIAJ, y que posteriormente han sido inscritos o posibles clientes que no pudieron ser dados de alta por estar inscritos en el RGIAJ y ahora se han dado de baja). Aplicada esta relación sobre su base de datos, podrá pagar premios excluyendo a aquellos clientes que resulten estar inscritos en el RGIAJ en esta lista.

Los servidores de la DGOJ tienen ajustados sus relojes internos mediante un servidor NTP sincronizado con el Real Observatorio de la Armada, organismo español responsable del mantenimiento y difusión oficial de la escala "Tiempo Universal Coordinado" (UTC(ROA)). Se recomienda que los servidores de los operadores se sincronicen igualmente con el organismo equivalente en sus respectivos países. Por otro lado se recomienda no realizar la consulta a VerificarCambiosRGIAJ a las horas en punto, si no cinco minutos más tarde para asegurar que el proceso de publicación en dicho servicio está terminado.

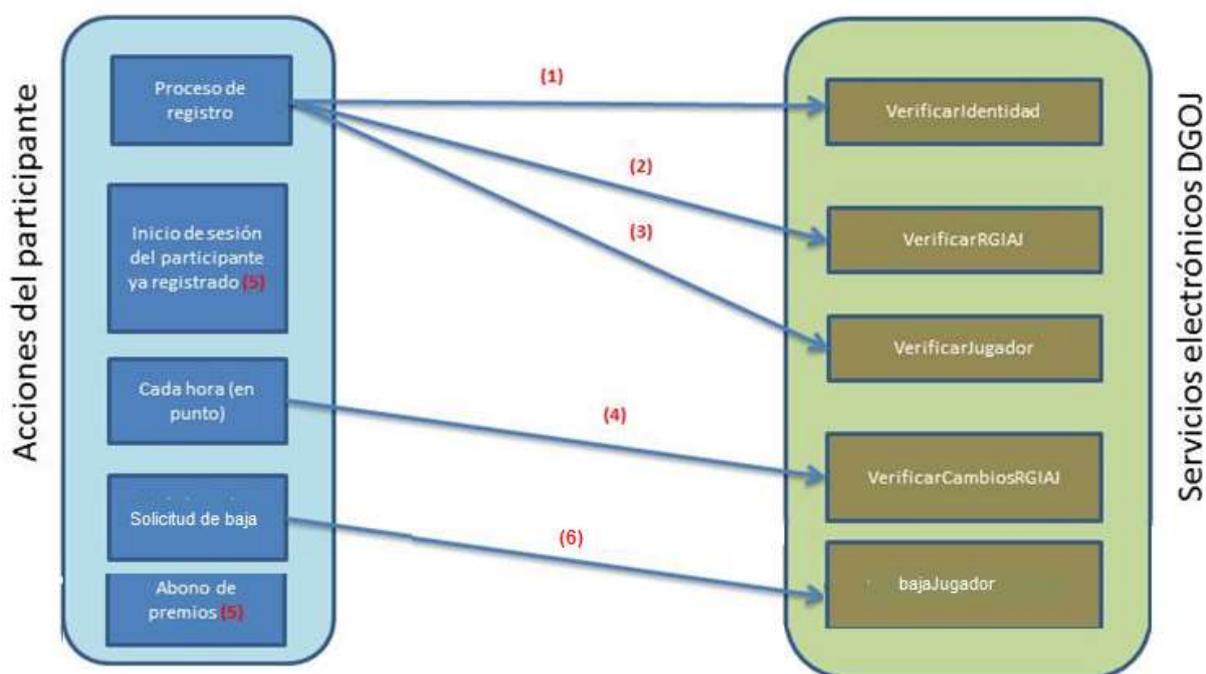
5. "bajaJugador": el operador puede utilizar esta operación para dar de baja a un jugador por el que realizó en su día alguna comprobación de RGIAJ, tanto con el método "VerificarRGIAJ" como en "VerificarJugador". De este modo no se informará del estado de dicho jugador en la operación de VerificarCambiosRGIAJ, por considerarse que al utilizar esta operación, queda rota la relación contractual entre el jugador y el operador.

Debe tenerse en cuenta que, al objeto de coordinar esta función con las inscripciones realizadas en el RGIAJ, informadas también en la operación de verificarCambiosRGIAJ, estas bajas se activan en la siguiente hora a su grabación en la base de datos de la DGOJ. Por ejemplo, todas las inscripciones realizadas entre las 11:00 y las 11:59, tendrán hora de inscripción a las 12:00. A partir de ese momento serán informadas en "VerificarCambiosRGIAJ".



Por tanto, sigue siendo suficiente con que el operador consulte la operación “VerificarCambiosRGIAJ” una vez cada hora para obtener la relación de cambios en el RGIAJ producidos en los jugadores sobre los que alguna vez han consultado su estado en el RGIAJ y de los que no ha solicitado su baja.

Un diagrama explicativo se encuentra en la siguiente figura



(1) En el caso de que el operador utilice los servicios de identificación de participantes proporcionados por la DGOJ, se hará uso de este servicio en la fase inicial del proceso de registro con el objeto de comprobar la identidad del jugador y en el caso de una identificación incorrecta no se acceda a fases posteriores del proceso de registro donde se realicen otras comprobaciones (estado en el Registro General de Interdicciones de Acceso al Juego –RGIAJ-) o se soliciten otros datos adicionales necesarios para la constitución efectiva de la cuenta de usuario (tarjetas de crédito, domicilio, ...). Desde el punto de vista de la DGOJ no se produce una asociación del jugador y del operador.

(2) A utilizar en la fase final del proceso de registro, tras el cual se podrá (si procede) activar la cuenta de usuario. La utilización de este método implica que la DGOJ asociará al jugador al operador que realiza la consulta, por lo que en el futuro y sobre el jugador



consultado, se notificarán los cambios de su estado en el RGIAJ vía VerificarCambiosRGIAJ.

(3) A utilizar cuando el proceso de registro del jugador se realiza en una sola fase, comprobando en un solo paso la identidad y su estado en RGIAJ, tras el cual se podrá (si procede) activar la cuenta de usuario. La utilización de este método implica que la DGOJ asociará al jugador al operador que realiza la consulta, por lo que en el futuro y sobre el jugador consultado, se notificarán los cambios de su estado en RGIAJ vía VerificarCambiosRGIAJ.

Se recuerda que, salvo la excepción prevista en el apartado posterior dedicado a las operaciones a realizar en el caso de que los servicios no estén disponibles, la utilización de los servicios electrónicos proporcionados por la DGOJ únicamente podrán ser utilizados como consecuencia de una acción iniciada por el participante.

4) Dado que los cambios de estado en RGIAJ se producen a las horas en punto, los datos obtenidos en la consulta del método serán válidos durante toda la hora siguiente y se deberán usar para impedir realizar cualquier actividad de juego en los términos recogidos en la normativa vigente.

(5) Ambas acciones deben tener en cuenta la situación de inscripción del jugador en el RGIAJ. Para ello se utilizará la información obtenida a través del servicio VerificarCambiosRGIAJ, con la que el operador deberán actualizar su base de datos de clientes y actuar en consecuencia.

(6) A utilizar cuando el operador quiera dar de baja a un jugador por el que previamente pregunto por su estado en RGIAJ, bien porque no llegó a ser nunca cliente o porque ha dejado de serlo, y no se le informe más de su estado o cambios de estado en el RGIAJ.

Actuaciones a realizar en el caso de que los servicios anteriores no estén disponibles:

1.- Indisponibilidad del servicio de verificación de identidad (servicio verificarIdentidad o VerificarJugador): el operador podrá utilizar los mecanismos de comprobación de identidad alternativos que tenga habilitados. En el caso de que el operador utilice habitualmente los servicios de comprobación de identidad proporcionados por la DGOJ, una vez restablecido el servicio –para verificar su restablecimiento puede realizar comprobaciones periódicas-, se podrá comprobar en los servicios de verificación de identidad aquellos usuarios que no pudieron ser verificados durante el periodo de caída del sistema.

2.- Indisponibilidad del servicio de verificación de presencia en RGIAJ (métodos VerificarRGIAJ y VerificarJugador la DGOJ establecerá un plan de contingencia para proporcionar esta información a través de un procedimiento alternativo.



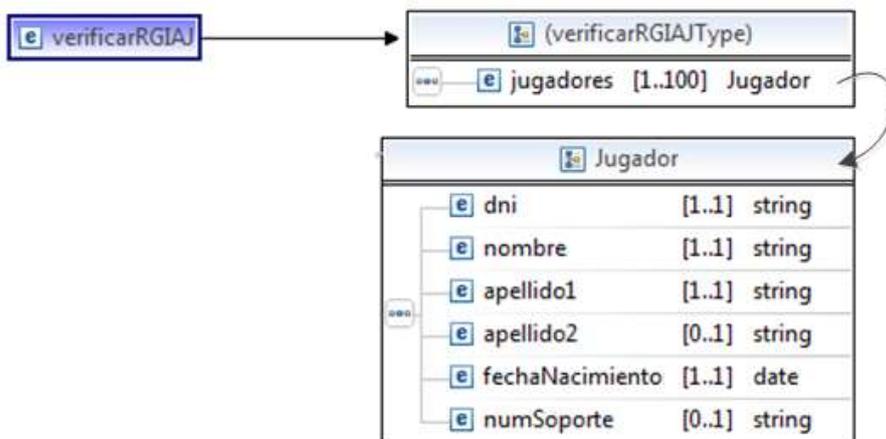
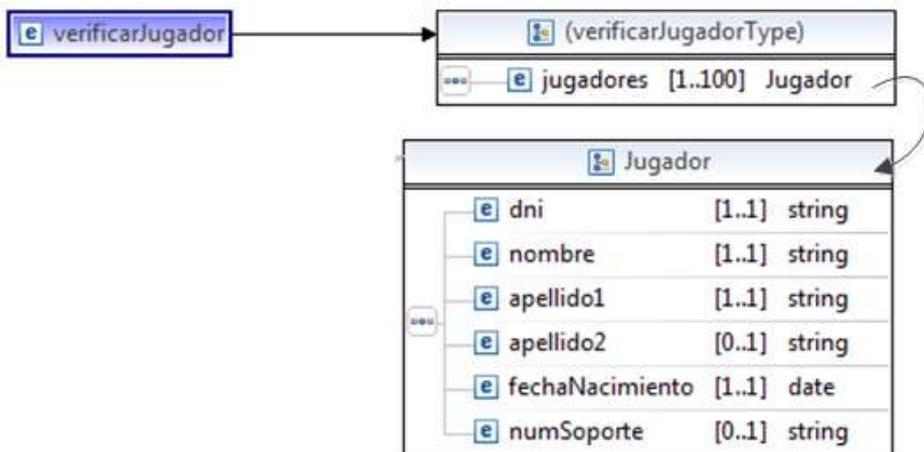
3.- VerificarCambiosRGIAJ: en el caso de no estar disponible el servicio, se reintentará su descarga en un momento posterior y se operará con la información previamente descargada en la consulta anterior.

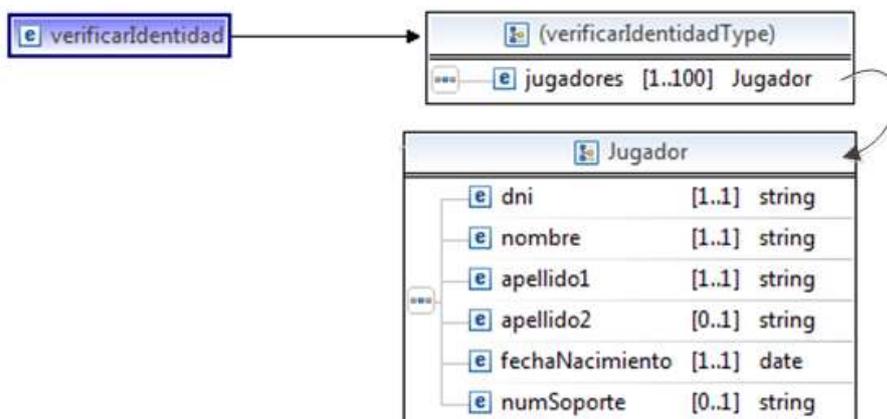
4.- bajaJugador: en el caso de no estar disponible el servicio se reintentará en un momento posterior.

2.2 Parámetros de entrada

2.2.1 VerificarJugador, VerificarRGIAJ y VerificarIdentidad

En las tres primeras operaciones, los parámetros de entrada de la operación coinciden, aunque hay que señalar que VerificarRGIAJ sólo utiliza DNI/NIE y aunque es necesario especificar nombre y los apellidos, estos no se comprueban:





Se trata de una o más ocurrencias del tipo Jugador, que está compuesto de:

- DNI/NIE del jugador. (obligatorio) Se entiende por DNI/NIE el Documento Nacional de Identidad, incluyendo el número de identificación junto con la letra asociada al mismo (NIF/NIE), por lo que el formato será uno de los siguientes:
 - Número de 8 dígitos seguido de una letra (dígito de control). (NIF). En caso de no ser un número con 8 dígitos, se completará el número con un 0 a la izquierda.
 - Letra (X,Y,Z) seguida de número de 7 cifras y otra letra (dígito de control), haciendo un total de 9 caracteres (NIE). Para el caso de los NIE es necesario aclarar una excepción debido a una actuación histórica de la policía. Antes de decidir que el formato del NIE tuviera 9 caracteres anteceditos de una letra (X,Y ó Z) se expedieron algunos documentos con 10 dígitos y una X al principio, y dado el número de expediciones realizadas, el primer dígito después de la X fue siempre un 0. El Web Service solo acepta los NIEs de 9 dígitos, por lo que cuando se detecte un NIE de 10 dígitos que empiece por X y a continuación un 0 y además la longitud total del NIE, incluyendo el dígito de control, sea de 10 caracteres, debe suprimirse el primer cero tras la X y realizar la consulta con el NIE resultante (formado por 9 caracteres). El resto de NIEs de 10 dígitos se deben descartar al tener un formato erróneo.
- Nombre o nombre compuesto del jugador, tal y como aparece en su Documento Nacional de Identidad con una longitud máxima de 40 caracteres. (obligatorio, en VerificarRGIAJ no se comprueba).
- Primer apellido del jugador, tal y como aparece en su Documento Nacional de Identidad con una longitud máxima de 40 caracteres. (obligatorio en VerificarRGIAJ no se comprueba).



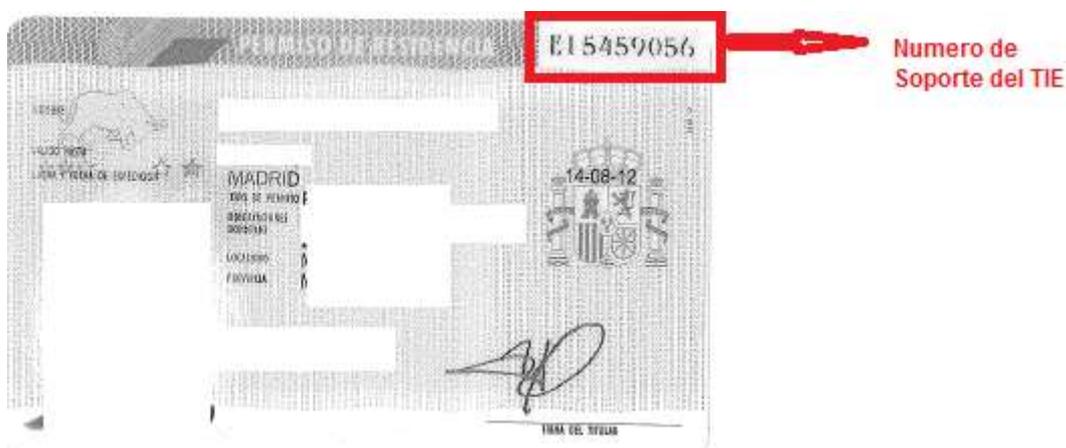
- Segundo apellido del jugador, tal y como aparece en su Documento Nacional de Identidad con una longitud máxima de 40 caracteres. (para el caso de NIE este campo no será obligatorio. En VerificarRGIAJ no se comprueba).
- Fecha de nacimiento del jugador. Ejemplo: <fechaNacimiento>2011-09-21</fechaNacimiento>. El sistema acepta el formato Date de W3C, que incluye la zona horaria, si bien se desechará esa zona horaria porque no es necesaria para el cometido del servicio. Fechas de nacimiento anteriores 01/01/1900 darán como resultado una respuesta de formato no válido.
- Número de soporte del NIE. En caso de que la identificación se haga por NIE, se puede opcionalmente enviar el "Número de soporte" asociado a la tarjeta de identificación de Extranjeros (TIE) o al Certificado Europeo, para solventar los casos en que la respuesta del WS de la D.G. de la Policía son del tipo "SE HA ENCONTRADO MAS DE UN REGISTRO CON ESE IDENTIFICADOR".

Solo se admitirá valor en este campo si el identificador es un NIE; en caso de NIF se devolverá error de formato (COD906 : Error de formato-Con NIF no se puede enviar número de Soporte).

El formato del "Número de soporte" debe ser uno de los siguientes:

- La letra "E" seguida de 8 dígitos numéricos para los TIE (longitud total obligatoriamente de 9 posiciones)
- La letra "C" seguida de 8 dígitos numéricos para los Certificados Europeos (longitud total obligatoriamente de 9 posiciones)

Ejemplo de TIE:





El número de soporte sería: E15459056

Ejemplo de certificado Europeo

Numero de Soporte del
Certificado Europeo

El número de soporte sería: C02161074.

NOTA: Como la longitud del campo debe ser de 9 posiciones, se completan con ceros por la izquierda la parte numérica.



En relación con los caracteres admitidos en los campos nombre, primer y segundo apellido, se establecerán una serie de caracteres que no serán admitidos en dichos campos, por no tener sentido su utilización en identificación de personas. Inicialmente se han fijados los siguientes:

1234567890\$&¿?¡|()@#~+*{}<>%/\^

Esta lista podría actualizarse en versiones posteriores del servicio. En caso de que los campos indicados contengan uno de estos caracteres, se devolverá el mensaje de error COD902. Como se puede ver, se han prohibido aquellos caracteres que se pueden obtener a través de un uso convencional del teclado y cuya presencia en el nombre y/o apellidos no tiene sentido. Por todo ello, el juego de caracteres que se pueden obtener mediante alguna combinación de teclas (por ejemplo AltGr+'combinación de teclas') estarían permitidos, aunque su uso en el nombre y/o apellidos de una persona no sea lógico. Además, probablemente, la utilización de dichos caracteres tendría como resultado un fallo en la identificación del jugador.

Es necesario recalcar que el sistema no es sensible a mayúsculas y minúsculas. Así mismo tampoco es sensible a las vocales acentuadas con la tilde utilizada en español u otros caracteres de acentuación de vocales a excepción de la diéresis con la u. De esta forma se puede contemplar la siguiente tabla:

Carácter recibido	Carácter con el que trabaja el sistema
áàâäÁÀÄÂ	a
éèêÉÈÊË	e
ïîïÏ	i
óòôÓÒÔ	o
úùÚÛÜ	U
Üü	Ü

De igual forma, el sistema ignorará en los parámetros anteriormente mencionados, cualquier carácter 'espacio en blanco' o guión '-' situado al principio o al final del



parámetro correspondiente. En el caso de nombres o apellidos compuestos (con espacios en blanco o guiones en su interior), el sistema sustituirá de forma automática cualquier aparición de dos o más espacios en blanco consecutivos por uno solo.

Por motivos de integridad técnica, se ha limitado el número máximo de ocurrencias de verificación de jugadores que se pueden realizar en una misma petición, siendo este número igual a 100. Si dentro de un mensaje existe algún NIF/NIE repetido, se rechazará todo el mensaje con un fault de código ERR-003.

El tiempo de respuesta para mensajes con varios NIF/NIE, al no depender de un repositorio local, es variable y en algún caso elevado, dependiendo de diversas circunstancias todas ellas ajenas a la DGOJ.

2.2.2 *Parámetros VerificarCambiosRGIAJ*

Esta operación no requiere parámetros de entrada, ya que el Operador queda autenticado e identificado con su certificado digital.



2.2.3 *Parámetros bajaJugador*

Esta operación tiene como parámetros de entrada una ocurrencia del tipo Baja que está compuesta de:

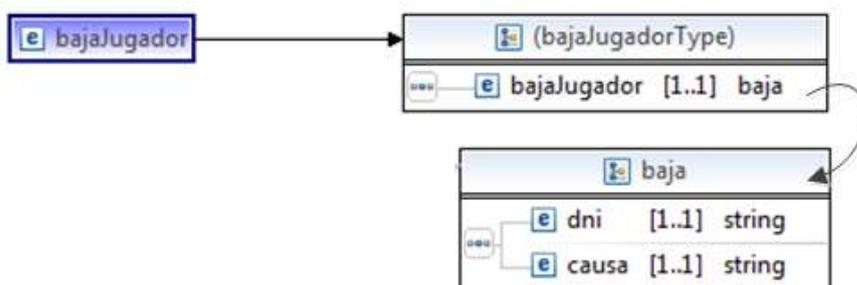
- DNI/NIE del jugador. (obligatorio) Se entiende por DNI/NIE el Documento Nacional de Identidad, incluyendo el número de identificación junto con la letra asociada al mismo (NIF/NIE), por lo que el formato será uno de los siguientes:
 - Número de 8 dígitos seguido de una letra (dígito de control). (NIF). En caso de no ser un número con 8 dígitos, se completará el número con un 0 a la izquierda.
 - Letra (X,Y,Z) seguida de número de 7 cifras y otra letra (dígito de control), haciendo un total de 9 caracteres (NIE). Para el caso de los NIE es necesario aclarar una excepción debido a una actuación histórica de la policía. Antes de decidir que el formato del NIE tuviera 9 caracteres anteceditos de una letra (X,Y ó Z) se expidieron algunos documentos



con 10 dígitos y una X al principio, y dado el número de expediciones realizadas, el primer dígito después de la X fue siempre un 0.

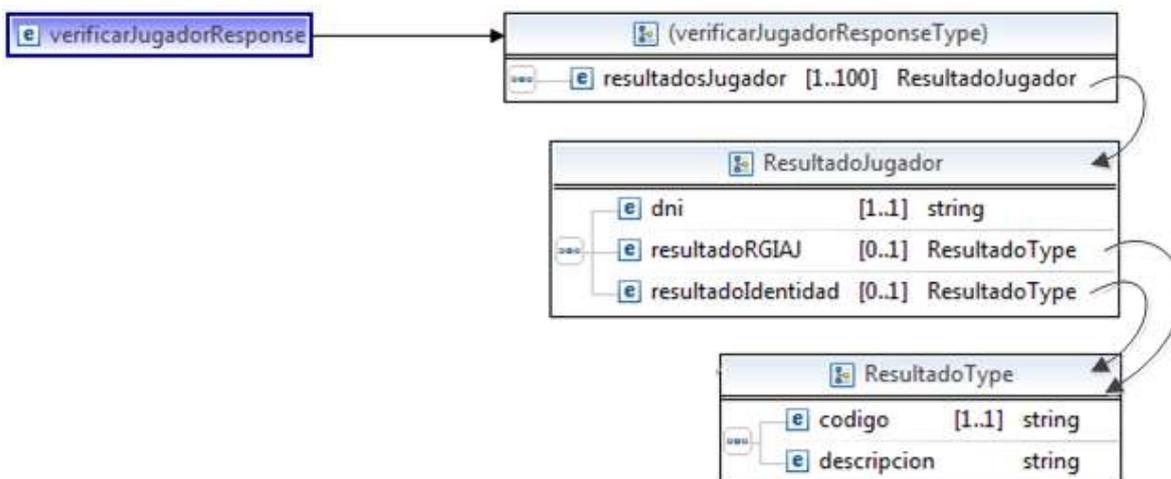
El Web Service solo acepta los NIEs de 9 dígitos, por lo que cuando se detecte un NIE de 10 dígitos que empiece por X y a continuación un 0 y además la longitud total del NIE, incluyendo el dígito de control, sea de 10 caracteres, debe suprimirse el primer cero tras la X y realizar la consulta con el NIE resultante (formado por 9 caracteres). El resto de NIEs de 10 dígitos se deben descartar al tener un formato erróneo.

- Causa de la Baja. (obligatorio) Puede tener solo los siguientes valores:
 - 'A' : si la causa de la incidencia es un jugador activo en el operador y que se desea dar de baja
 - 'R': si la causa es un jugador que aunque se preguntó su estado en el RGIAJ nunca se llegó a activar en el operador.



2.3 Respuestas del servicio

2.3.1 Respuesta para VerificarJugador

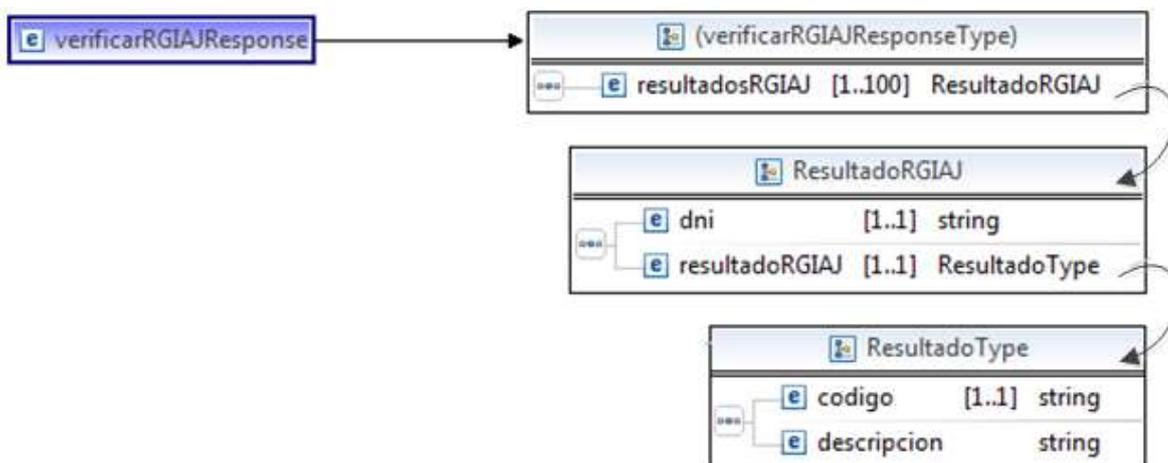




Se devolverán tantas repeticiones ResultadoJugador como repeticiones de tipo Jugador se han enviado en la petición. El tipo Resultado contiene los siguientes datos:

1. DNI verificado, correspondiente al DNI/NIE enviado en la petición.
2. Resultado de verificación en RGIAJ (código y descripción).
3. Resultado de verificación de identidad (código y descripción).

2.3.2 Respuesta para VerificarRGIAJ

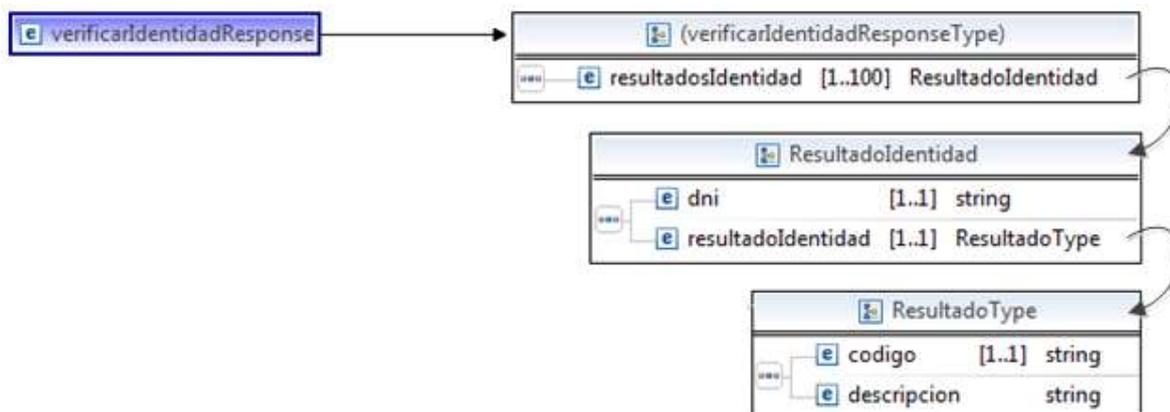


Se devolverán tantas repeticiones ResultadoRGIAJ como repeticiones de tipo Jugador se han enviado en la petición. El tipo Resultado contiene los siguientes datos:

1. DNI verificado, correspondiente al DNI/NIE enviado en la petición.
2. Resultado de verificación en RGIAJ (código y descripción).



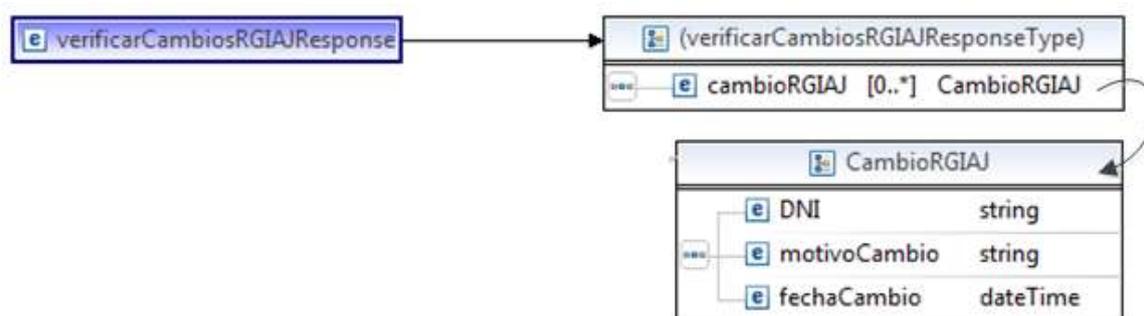
2.3.3 Respuesta para VerificarIdentidad



Se devolverán tantas repeticiones ResultadoIdentidad como repeticiones de tipo Jugador se han enviado en la petición. El tipo Resultado contiene los siguientes datos:

1. DNI verificado, correspondiente al DNI/NIE enviado en la petición.
2. Resultado de verificación de identidad (código y descripción).

2.3.4 Respuesta para VerificarCambiosRGIAJ



La operación devuelve una lista de DNIs/NIEs, motivo del cambio (“Alta” o “Baja”) y fecha del cambio, que se corresponde con los cambios de inscripción en el RGIAJ de jugadores cuya inscripción hubiera consultado previamente el operador mediante el servicio VerificarRGIAJ.

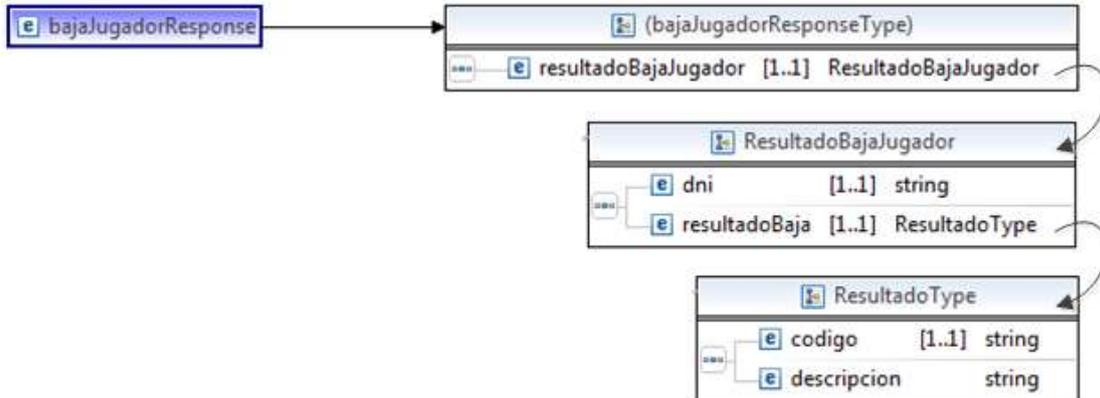
El tipo ResultadoCambiosRGIAJ contiene los siguientes datos:

1. DNI/NIE.



2. Motivo del cambio: A-Alta, o B-Baja.
3. fechaCambio, fecha y hora en que se produce el cambio en el registro.

2.3.5 Respuesta para bajaJugador



La operación devuelve una ocurrencia del tipo ResultadoBajaJugador

El tipo ResultadoBajaJugador contiene los siguientes datos:

1. DNI , correspondiente al DNI/NIE enviado en la petición.
2. Resultado de la operación de baja (código y descripción).

2.4 Autenticación del Operador

Los mensajes que se enviarán a la Dirección General de Ordenación del Juego para la verificación de jugadores se firmarán con el certificado de sello del Operador de Juego, usando para ello el estándar Web Services Security 1.0 (en adelante WS-Security), según la especificación del esquema <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd>.

De igual forma los mensajes de respuesta que devuelva el servicio irán firmados con el mismo estándar con el certificado de sello de la DGOJ, salvo los mensajes correspondientes a los soap fault.

Por motivos de seguridad, el certificado utilizado por el servicio será distinto al certificado utilizado para encriptar la comunicación por HTTPS.

Los operadores podrán validar la firma digital de los mensajes de respuesta utilizando el *token* (elemento **<BinarySecurityToken>**) que viaje en las cabeceras del mensaje. Ese *token* corresponderá a la parte pública del certificado que el servidor utilizará para la firma del mensaje. De esta manera, las cabeceras de los mensajes SOAP de respuesta serán similares a las cabeceras de los mensajes de entrada: contendrán un token que corresponderá a la parte pública del certificado utilizado para la firma, un timestamp y la firma digital del mensaje.



Se muestra a continuación un ejemplo de petición al servicio de verificación, sobre la que se describirán los elementos que la componen y que serán necesarios para el correcto funcionamiento del servicio.

Se debe prestar especial atención a la dirección del servicio donde se debe encaminar la petición. En el entorno de preproducción, accesible a través de la dirección:

- Endpoint: <https://ws-pre.ordenacionjuego.gob.es/VerificacionJugadores-2.2/VerificacionJugadores>
- WSDL: <https://ws-pre.ordenacionjuego.gob.es/VerificacionJugadores-2.2/VerificacionJugadores?WSDL>

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" S:mustUnderstand="1">
      (1) <wsse:BinarySecurityToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
        EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary"
        ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"
        wsu:Id="XWSSGID-1317050485906-773090739">MII I9zCCB9+gAwIBAgI JALsAAAAAAECMA0GCSqGSIb3DQEjBQUAMIHGMQswCQYDVQQG
        EwJFUFzEu
        MCwGCSqGSIb3DQEJARYfYWNfY2FtZXJmaXJtYV9jY0BjYW1lcmZpcmlhLmNvbTFFMEEGA1UEBXM6
        .....
        ...
        ...
        +Ubmq8yaWe9F/4QyOOSmO6UgDyge6fSvZaw1U1QSlzt2apSEhwsUOIQKBJNq7HqJ8SVnp9Ierlty
        LdvOZfDckdDwjNsXneiH61YEHA==</wsse:BinarySecurityToken>
      <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:SignedInfo>
          <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
            <InclusiveNamespaces xmlns="http://www.w3.org/2001/10/xml-exc-c14n#"
              PrefixList="wsse S"/>
          </ds:CanonicalizationMethod>
          <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
          (2) <ds:Reference URI="#XWSSGID-1317050486921978042889">
            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <ds:DigestValue>uVmWY1Vfjnd5NxTYceLHhkmQx0Y=</ds:DigestValue>
          </ds:Reference>
        </ds:SignedInfo>
      </ds:Signature>
    </wsse:Security>
  </S:Header>
  <S:Body>
  </S:Body>
</S:Envelope>
```



```
(3) <ds:Reference URI="#XWSSGID-13170504869211268263594">
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>ovHJ8CbnjXObbQupmJq6f++T+1A=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>YYoxAiuLxhBJwPIpNgJSv+06s868zyh3Yxmzn2siMkj+r6T7Q7PixBYKsieQ
dWWSMBFBgldlc3mM
nafVJE3K9w17gFrk0e+HE/jKkEyIHxVsuuweJ/s0tAAm3+CG/k3GktiNuhZrB3WC10XDw56ViU00
f5x13/Qoljmg291g5Lk=</ds:SignatureValue>
<ds:KeyInfo>
<wsse:SecurityTokenReference xmlns:wssu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
wssu:Id="XWSSGID-1317050486890-634088959">
<wsse:Reference URI="#XWSSGID-1317050485906-773090739"
ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-
profile-1.0#X509v3"/>
</wsse:SecurityTokenReference>
</ds:KeyInfo>
</ds:Signature>
(3) <wsu:Timestamp xmlns:wssu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd" wssu:Id="XWSSGID-
13170504869211268263594">
<wsu:Created>2011-09-26T15:21:26Z</wsu:Created>
<wsu:Expires>2011-09-26T15:21:31Z</wsu:Expires>
</wsu:Timestamp>
</wsse:Security>
</S:Header>
(2) <S:Body xmlns:wssu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" wssu:Id="XWSSGID-1317050486921978042889">
<ns2:verificarJugador xmlns:ns2="https://<ESPECIFICAR LA DIRECCIÓN DE ACCESO AL
SERVICIO/">
<jugadores>
<dni>74660112L</dni>
<nombre>Daniel</nombre>
<apellido1>López</apellido1>
<apellido2>Gómez</apellido2>
<fechaNacimiento>2100-10-02T00:00:00</fechaNacimiento>
<numSoporte></numSoporte>
</jugadores>
</ns2:verificarJugador>
```



```
</S:Body>  
</S:Envelope>
```

Los siguientes aspectos acerca de la petición, marcados en rojo sobre el ejemplo, deberán tenerse especialmente en cuenta:

1. En la petición se incluirá el certificado (su parte pública) con el que se firma la petición. Para verificar que la petición corresponde al operador dado de alta en el sistema, la parte pública del certificado a utilizar en este proceso deberá comunicarse a la Dirección General de Ordenación del Juego. Para el alta en el entorno de pruebas, se utilizará el formulario de solicitud de alta en servicio que indicado en el apartado 4.4 y que deberá remitirse a la dirección de correo dgoj.sopORTEoperadores@minhap.es indicando en el asunto del mensaje "ALTA WS-VERIFICACION". Para el alta en el entorno de producción, esos datos serán comunicados a la dirección General de Ordenación del Juego en el proceso de concesión de licencias.

El certificado estará codificado en Base64, mediante el tipo BinarySecurityToken que se incluye en la cabecera de seguridad del estándar. Nótese en el ejemplo que para mayor claridad se ha abreviado el contenido del certificado.

El Operador será responsable de comunicar a la DGOJ cualquier situación de revocación que se produzca en el certificado utilizado.

2. Se realizará firma del cuerpo del mensaje SOAP (etiqueta "body"), y la firma será XMLDSIG, concretamente con el algoritmo <http://www.w3.org/2000/09/xmlsig#rsa-sha1>. Se especifica la referencia al cuerpo del mensaje (body) mediante el correspondiente tag referente cuya URI coincide con el tag identificador del cuerpo del mensaje (wsu:Id). Quedan exceptuados de la firma los mensajes soap fault

```
<ds:Reference URI="#XWSSGID-1317050486921978042889">
```

```
<S:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-  
wssecurity-utility-1.0.xsd" wsu:Id="XWSSGID-1317050486921978042889">  
<ns2:verificarJugador xmlns:ns2="http://ws.juego.es/VerificacionJugadores/">  
<jugadores>  
<dni>74660112L</dni>
```



```
<nombre>Daniel</nombre>  
<apellido1>López</apellido1>  
<apellido2>Gómez</apellido2>  
<fechaNacimiento>2100-10-02T00:00:00</fechaNacimiento>  
</jugadores>  
</ns2:verificarJugador>  
</S:Body>  
</S:Envelope>
```

3. Se incorporará timestamp de la petición y referencia al mismo incluyendo el hash:

```
<wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="XWSSGID-13170504869211268263594">  
<wsu:Created>2011-09-26T15:21:26Z</wsu:Created>  
<wsu:Expires>2011-09-26T15:21:31Z</wsu:Expires>  
</wsu:Timestamp>
```

```
<ds:Reference URI="#XWSSGID-13170504869211268263594">  
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>  
<ds:DigestValue>ovHJ8CbnjXObbQupmJq6f++T+1A=</ds:DigestValue>  
</ds:Reference>
```

Los elementos Created y Expires deben ir de acuerdo al tipo dateTime con el formato UTC, tal y como se recomienda por parte de W3C (http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss)

Por motivos de seguridad sólo se admiten timestamp con tiempos de validez inferiores a 300 segundos, es decir, el intervalo entre <wsu:Created> y <wsu:Expires> debe ser inferior a 300 segundos.

Al objeto de facilitar las pruebas, en el entorno de pre-producción el timestamp será opcional, aunque si se especifica, se validará, En el entorno de producción, una vez implantado el servicio, será obligatorio.



3 Descripción de los entornos

Los servicios se invocarán a través de uri's que se indican a continuación

3.1 Producción

Podrán existir un máximo de dos versiones del servicio para permitir a los operadores la implantación en sus sistemas de producción de los clientes de una versión del servicio de forma progresiva. La DGOJ acordará con los operadores el calendario de los cambios de versión, estableciendo cual es el endpoint y wsdl del servicio con soporte en cada momento y el plazo de cese del servicio de la versión antigua.

A partir de la versión 2.1 del servicio, estas uri's irán calificadas con el número de versión correspondiente al servicio, coexistiendo con las uri's de la versión anterior durante un plazo establecido por la DGOJ para facilitar la implantación a los operadores.

Endpoint: <https://ws.ordenacionjuego.gob.es/VerificacionJugadores-v.r/VerificacionJugadores>

WSDL: <https://ws.ordenacionjuego.gob.es/VerificacionJugadores-v.r/VerificacionJugadores?WSDL>

Donde v es un dígito que indica el número de versión y r el número de reléase dentro de la versión.

La versión del software correspondiente a esta versión del documento de especificación (1.8) es la 2.2, por lo que las uri's serán:

Endpoint: <https://ws.ordenacionjuego.gob.es/VerificacionJugadores-2.2/VerificacionJugadores>

WSDL: <https://ws.ordenacionjuego.gob.es/VerificacionJugadores-2.2/VerificacionJugadores?WSDL>

Actualmente esta es la única versión en producción del servicio.

La numeración de las versiones del documento no coincide con las del software.

Para cada operador, el entorno no se activará hasta el momento de concesión de una licencia singular, no pudiendo solicitarse el alta en el servicio hasta ese momento, mediante el formulario indicado en el apartado 4.4



No obstante, la DGOJ podrá autorizar la realización de pruebas de integración en el entorno de producción a los operadores que hayan solicitado al menos una licencia singular en los términos recogidos en el apartado 4.3

El acceso al WS de producción sólo estará habilitado para las direcciones IPs comunicadas por los operadores en el correspondiente [formulario](#) de alta en el entorno de producción. Estas IPs serán las de las servidores o equipos de establezcan la comunicación con la DGOJ, debiendo ser por tanto direcciones individuales y nunca en un número superior a cuatro salvo petición justificada documentalmente. En el caso de que la DGOJ decida interrumpir el servicio prestado a un operador, se utilizarán las IPs comunicadas para prohibir el tráfico, por lo que es responsabilidad del operador el uso de IPs únicas y no compartidas con otros operadores para el acceso a los servicios de la DGOJ

3.2 Pruebas

Existe un único punto de acceso a la versión más reciente del servicio. Los cambios de versión se anunciarán a los operadores con suficiente antelación, dejando de dar servicio la versión anterior.

Endpoint: <https://ws-pre.ordenacionjuego.gob.es/VerificacionJugadores-v.r/VerificacionJugadores>
WSDL: <https://ws-pre.ordenacionjuego.gob.es/VerificacionJugadores-v.r/VerificacionJugadores?WSDL>

Donde v es un dígito que indica el número de versión y r el número de reléase dentro de la versión. Para esta versión del documento el software tienes la versión 2.2.

Los operadores de juego que deseen realizar pruebas con el servicio de verificación de jugadores, deberán solicitar el alta en el entorno de pruebas mediante el formulario que se indica en el apartado 4.4.

Las solicitudes se enviarán a la cuenta de correo dgoj.sopORTEoperadores@minhap.es indicando en el asunto del mensaje

“ALTA WS-VerificaciónJugadores”.



4 Anexos

4.1 Códigos de resultado y error

La siguiente tabla muestra los códigos que contendrán los tipos de resultado de las distintas operaciones de verificación.

Códigos de Resultado	
<i>Consulta de RGIAJ en los métodos habilitados</i>	
COD001	El usuario está inscrito en el RGIAJ
COD002	El usuario no está inscrito en el RGIAJ
COD006	No se ha podido verificar el estado en RGIAJ del usuario
<i>Consulta de identidad en los métodos habilitados</i>	
COD003	La identidad del usuario es correcta
COD004	<p>La identidad del usuario es incorrecta. <i>Motivo</i>¹</p> <p>¹ Texto aclaratorio del posible causa del problema de identificación. En la fecha de publicación del documento existen los siguientes <i>Motivos</i>:</p> <ol style="list-style-type: none">1. <i>Titular no identificado</i>2. <i>Titular no identificado positivamente - Ciudadano Nacionalizado Español</i>3. <i>El documento solicitado puede estar en trámite de renovación, retenido, retirado por autoridad judicial o anulado por duplicidad. Si necesita aclaración, póngase en contacto con la Unidad de Documentación de Españoles y Archivo.</i>4. <i>Se ha encontrado mas de un registro con ese identificador.</i> <p>Este error normalmente aparece en consultas de verificación de identidad usando NIE y se debe a que el NIE consultado tiene más de una filiación en el Registro Central de Extranjeros. Las causas de los cambios de filiación son variados, ya que afectan a cualquiera de los datos de filiación (nombre, apellidos, nombres de padres, país de nacimiento, nacionalidad, fecha de nacimiento, cambio de nacionalidad, ..). Cuando un ciudadano con NIE cambia cualquiera de esos datos, circunstancia que parece que se da con cierta frecuencia, la consulta a ese NIE determinado ofrece el error mencionado.</p> <ol style="list-style-type: none">5. <i>DNI del titular anulado</i>6. <i>Documento retirado judicialmente, retenido por existencia de duplicados o anulado por la unidad de documentación de españoles</i>
COD005	No se ha podido verificar la identidad del usuario
<i>Operación de Baja en los métodos habilitados</i>	
COD010	Baja realizada correctamente
COD021	Baja no realizada, el DNI no está activado para ese operador
COD022	Baja no realizada, el DNI ya está de baja para ese operador



COD023	Causa de la baja errónea
<i>Otros</i>	
COD901	Formato de datos incorrecto –DNI no válido
COD902	La petición tiene caracteres no válidos en el *Nombre/apellido1/apellido2 *Se especifica en que campo se ha detectado los caracteres inválidos.
COD903	Formato de datos incorrecto - Es obligatorio el *Nombre/apellido1/apellido2 *Se especifica que campo obligatorio no lleva valor
COD904	Formato de datos incorrecto - Fecha de nacimiento anterior a 1900
COD905	Formato de datos incorrecto - Número de soporte erróneo.
COD906	Formato de datos incorrecto - Con NIF no se puede enviar número de Soporte
COD907	Longitud del nombre o apellidos superior a 40 caracteres

La siguiente tabla muestra los códigos de error que se mostrarán en los tipos Fault de las operaciones de verificación.

Códigos de Error	
ERR001	Error técnico
ERR002	Operador de Juego inválido
ERR003	La petición tiene identificadores de jugador duplicados, no es válida

Los mensajes soap fault correspondientes a este tipo de error no irán firmados.

Desde la versión 0.7 se ha separado el código del mensaje del texto. Este último se especifica en el tag <faultstring> y el código en el tag <verificarJugadorFault> dentro del nodo <detail>.

Es necesario diferenciar los casos de identificación incorrecta de identidad de los casos de imposibilidad de realizar la operación de identificación de identidad.

En el primer caso se cubren las peticiones cuyo resultado sea COD004, COD901, COD902, COD903 , COD904, COD905, COD906 ó COD907. Cuando se obtenga cualquiera de estos resultados, no tiene sentido que el operador vuelva a preguntar por los mismos datos, ya que la respuesta será idéntica. Será necesario que se modifiquen los datos de la petición original para lograr una identificación correcta.



En el caso de que incluso enviando datos distintos se obtengan tres respuestas de identificación incorrecta, el operador deberá desistir de identificar al jugador utilizando los medios que la DGOJ pone a su disposición, comenzando un ciclo de verificación documental del jugador.

En el caso de imposibilidad de realizar la operación de identificación de identidad se deberá actuar de forma distinta dependiendo del error obtenido.

Si se debe a la caída del servicio de la DGOJ (ausencia de respuesta o debido a que una petición de identificación no devuelve la correspondiente respuesta en el plazo estimado - 30 segundos para cada uno de las peticiones de identificación individuales), debido a una imposibilidad técnica para resolver la petición (respuesta encuadrada en el código COD005) o a un error técnico (ERR001), el operador podrá volver a intentar la identificación pasado un tiempo que permita a la DGOJ restablecer el servicio (se estima en 30 minutos).

En el caso de inclusión de jugadores duplicados en una misma petición (ERR003), el operador podrá detectar el problema y reenviar la petición corregida de forma inmediata.

4.2 Juegos de ensayo en el entorno de pruebas

Endpoint: <https://ws-pre.ordenacionjuego.gob.es/VerificacionJugadores-v.r/VerificacionJugadores>
WSDL: <https://ws-pre.ordenacionjuego.gob.es/VerificacionJugadores-v.r/VerificacionJugadores?WSDL>

Donde v es un dígito que indica el número de versión y r el número de reléase dentro de la versión.

Las pruebas del servicio de verificación de jugadores están basadas en un Registro de Interdicciones de prueba, con datos simulados, cuyo contenido está controlado por la DGOJ y el Servicio de Verificación de Datos Identidad (SVDI) ofrecido por la Secretaría de Estado de Administraciones Públicas cuyo contenido puede variar en el periodo de pruebas por necesidades de la propia SEAP o de otros usuarios del sistema.

Por tanto, el objetivo de las pruebas es verificar la conectividad del sistema del operador con el servicio, sus mecanismos de autenticación, el tráfico seguro de mensajes y la utilización de los mensajes intercambiados, no tanto el contenido de los datos y de las respuestas dadas por el servicio.

Así mismo, no deberán realizarse pruebas de stress, con gran volumen de datos, contra el entorno de pruebas, sin comunicación previa a la DGOJ.

Para la resolución de incidencias o dudas de carácter técnico se deberá enviar un correo electrónico a la dirección de correo dgoj.sopORTEoperadores@minhap.es indicando en el asunto del mensaje



“PRUEBAS WS-VerificaciónJugadores”.

4.2.1 Descripción de los casos de Prueba

Este apartado recoge una relación de Casos de Prueba para las operaciones *verificarJugador*, *verificarIdentidad*, *verificarRGIAJ*, *verificarCambiosRGIAJ* y *bajaJugador* del servicio web de verificación de jugadores (*VerificacionJugadores*) que ofrecerá la DGOJ a los distintos operadores de juego.

WSDL: VerificacionJugadores.wsdl.

NOTA: El sistema no es sensible a mayúsculas y minúsculas, ni a las vocales acentuadas con la tilde utilizada en español o cualquier. Así mismo tampoco es sensible a las vocales acentuadas con la tilde utilizada en español u otros caracteres de acentuación de vocales a excepción de la diéresis con la u. Así mismo se recuerda el tratamiento de los espacios en blanco (pag. 8 y 9)

Operación: verificarJugador

Para cada uno de los casos de prueba no se especifica todo el mensaje SOAP, solamente los datos del nodo de <jugadores> para el mensaje Request y los datos del nodo de <resultadosJugador> para el mensaje Response.

Request (jugadores):

- Parámetro: **dni** (string): DNI/NIE del jugador.
- Parámetro: **nombre** (string) : Nombre del Jugador
- Parámetro: **apellido1** (string) : Primer apellido del jugador
- Parámetro: **apellido2** (string) : Segundo apellido del jugador
- Parámetro: **fechaNacimiento** (date) : Fecha de nacimiento del jugador
- Parámetro: **numSoporte** (string) : Número de Soporte del NIE

Response (resultadoJugador):

- Valor del **dni** (String) : DNI/NIE del jugador.
- resultadoRGIAJ:
 - **codigo**: (string) : código de retorno
 - **descripción** (string) : descripción del código de retorno



- **resultadoidentidad:**
 - **codigo:** (string) : código de retorno
 - **descripción** (string) : descripción del código de retorno

Por tanto para cada Caso de Prueba se especifican los siguientes datos:

Request:

```
<jugadores>  
  <dni>10000320N</dni>  
  <nombre>MARIA ENCINA</nombre>  
  <apellido1>FELIZ</apellido1>  
  <apellido2>VARELA</apellido2>  
  <fechaNacimiento>1944-05-18</fechaNacimiento>  
  <numSoporte></numSoporte>  
</jugadores>
```

Response:

```
<resultadosJugador>  
  <dni>10000320N</dni>  
  <resultadoRGIAJ>  
    <codigo>COD001</codigo>  
    <descripcion> El usuario está inscrito en el RGIAJ </descripcion>  
  </resultadoRGIAJ>  
  <resultadoidentidad>  
    <codigo>COD003</codigo>  
    <descripcion>La identidad del usuario es correcta</descripcion>  
  </resultadoidentidad>  
</resultadosJugador>
```

Operación: verificarIdentidad

Para cada uno de los casos de prueba no se especifica todo el mensaje SOAP , solamente los datos del nodo de <jugadores> para el mensaje Request y los datos del nodo de <resultadoidentidad> para el mensaje Response.

Request (jugadores):

- Parámetro: **dni** (string): DNI/NIE del jugador.



- Parámetro: **nombre** (string) : Nombre del Jugador
- Parámetro: **apellido1** (string) : Primer apellido del jugador
- Parámetro: **apellido2** (string) : Segundo apellido del jugador
- Parámetro: **fechaNacimiento** (date) : Fecha de nacimiento del jugador
- Parámetro: **numSoporte** (string) : Número de Soporte del NIE

Response (resultadosIdentidad):

- Valor del **dni** (string): DNI/NIE del jugador.
- **resultadoidentidad**:
 - **codigo**: (string) : código de retorno
 - **descripción** (string) : descripción del código de retorno

Por tanto para cada Caso de Prueba se especifican los siguientes datos:

Request:

```
<jugadores>
  <dni>10000320N</dni>
  <nombre>MARIA ENCINA</nombre>
  <apellido1>FELIZ</apellido1>
  <apellido2>VARELA</apellido2>
  <fechaNacimiento>1944-05-18</fechaNacimiento>
  <numSoporte></numSoporte>
</jugadores>
```

Response:

```
<resultadosJugador>
  <dni>10000320N</dni>
  <resultadoidentidad>
    <codigo>COD003</codigo>
    <descripcion>La identidad del usuario es correcta</descripcion>
  </resultadoidentidad>
</resultadosJugador>
```



Operación: verificarRGIAJ

Para cada uno de los casos de prueba no se especifica todo el mensaje SOAP , solamente los datos del nodo de <jugadores> para el mensaje Request y los datos del nodo de <resultadosRGIAJ> para el mensaje Response.

Request (jugadores):

- Parámetro: **dni** (string): DNI/NIE del jugador.
- Parámetro: **nombre** (string) : Nombre del Jugador
- Parámetro: **apellido1** (string) : Primer apellido del jugador
- Parámetro: **apellido2** (string) : Segundo apellido del jugador
- Parámetro: **fechaNacimiento** (date) : Fecha de nacimiento del jugador
- Parámetro: **numSoporte** (string) : Número de Soporte del NIE

Response (resultadosRGIAJ):

- Valor del **dni** (string) : DNI/NIE del jugador.
- resultadoRGIAJ:
 - **codigo**: (string) : código de retorno
 - **descripción** (string) : descripción del código de retorno

Por tanto para cada Caso de Prueba se especifican los siguientes datos:

Request:

```
<jugadores>
  <dni>10000320N</dni>
  <nombre>MARIA ENCINA</nombre>
  <apellido1>FELIZ</apellido1>
  <apellido2>VARELA</apellido2>
  <fechaNacimiento>1944-05-18</fechaNacimiento>
  <numSoporte></numSoporte>
</jugadores>
```

Response:

```
<resultadosJugador>
  <dni>10000320N</dni>
  <resultadoRGIAJ>
    <codigo>COD001</codigo>
```



```
<descripcion> El usuario está inscrito en el RGIAJ </descripcion>
</resultadoRGIAJ>
</resultadosJugador
```

Operación: verificarCambiosRGIAJ

Para cada uno de los casos de prueba no se especifica todo el mensaje SOAP. El mensaje Request no tiene parámetros de entrada. Se especifica sólo los datos del nodo de `< cambioRGIAJ >` para el mensaje Response.

Request:

No hay parámetros de entrada

Response (cambioRGIAJ):

- Valor del **dni** (string) : DNI/NIE del jugador.
- Valor del **motivoCambio** (string): “A” (alta) ó “B” (baja)
- Valor **fechaCambio** (datetime): fecha del cambio en el RGIAJ

Por tanto para cada Caso de Prueba se especifican los siguientes datos:

Response:

```
<cambioRGIAJ>
  <DNI>00000058N</DNI>
  <motivoCambio>A</motivoCambio>
  <fechaCambio>2011-11-01T14:07:13.257+01:00</fechaCambio>
</cambioRGIAJ>
```

Operación: bajaJugador

Para cada uno de los casos de prueba no se especifica todo el mensaje SOAP , solamente los datos del nodo de `<bajaJugador>` para el mensaje Request y los datos del nodo de `<resultadoBaja>` para el mensaje Response.



Request (bajaJugador):

- Parámetro: **dni** (string): DNI/NIE del jugador.
- Parámetro: **causa**(string) : Causa de la baja

Response (resultadoBajaJugador):

- Valor del **dni** (string) : DNI/NIE del jugador.
- resultadoBaja:
 - **codigo**: (string) : código de retorno
 - **descripción** (string) : descripción del código de retorno

Por tanto para cada Caso de Prueba se especifican los siguientes datos:

Request:

```
<bajaJugador>  
<dni>10000320N</dni>  
<causa>A</causa>  
</bajaJugador>
```

Response:

```
<resultadoBajaJugador>  
  <dni>10000320N</dni>  
  <resultadoBaja>  
    <codigo>COD010</codigo>  
    <descripcion> Baja realizada correctamente</descripcion>  
  </resultadoBaja>  
</resultadoBajaJugador>
```

4.2.2 Relación de casos de error genéricos

4.2.2.1 Datos Obligatorios no enviados

En el caso de que alguno o algunos de los datos, establecidos como obligatorios en el servicio, no venga se devuelve el error de esquema correspondiente en el *soapfault* del servicio.

4.2.2.2 Autenticación del Operador

En el caso de que el Operador no sea correctamente autenticado, se devolverá el



mensaje indicando que el Operador es inválido en el *soapfault* del servicio.

4.2.2.3 DNI o NIE repetidos

En el caso de que se envíe un lote de jugadores en el que se repitan alguno de los DNIs o de los NIEs se devolverá el mensaje indicando que la petición tiene identificadores de jugadores duplicados en el *soapfault* del servicio.

4.2.2.4 Validación de formato del DNI/NIE

En el caso de que el formato del DNI o el NIE no sea el correcto se devuelve el código “COD901- Formato de datos incorrecto –DNI no válido” en todos los resultados del Jugador.

4.2.2.5 Validación de letra del DNI/NIE

En el caso de que la letra del DNI o el NIE no sea la correcta se devuelve el código “COD901- Formato de datos incorrecto –DNI no válido” en todos los resultados del Jugador.

4.2.2.6 Validación de Apellido2 obligatorio si es un NIF

En el caso de que se pregunte por el NIF y el segundo Apellido sea nulo se devuelve el código “COD903- Formato de datos incorrecto – Es obligatorio el *Nombre/Apellido1/Apellido2” (se especifica que campo obligatorio va a nulos) en todos los resultados del Jugador.

4.2.2.7 Validación de fecha de nacimiento posterior al 01/01/1900

En el caso de que se pregunte por una fecha de nacimiento anterior al 01/01/1900 se devuelve el código “COD904- Formato de datos incorrecto – Fecha de nacimiento anterior a 1900” en todos los resultados del Jugador.

4.2.2.8 Chequeo de caracteres no válidos

En el caso de que el nombre o los apellidos contengan alguno de los caracteres considerados como no válidos, se devuelve el código “COD902-La petición tiene caracteres no validos en el *Nombre/Apellido1/Apellido2” (se especifica en que campo se han detectado los caracteres inválidos) en todos los resultados del Jugador.

4.2.2.9 El servidor del RGIAJ no está disponible

En el caso de que el servidor del RGIAJ no esté disponible se devuelve el mensaje de error técnico en el *soapfault* para toda la petición, por tanto en el caso de la operación de *verificarJugador* no se realizará la identificación del jugador.



4.2.2.10 El servidor del SVDI no está disponible

En el caso de que el servidor del SVDI no esté disponible se devuelve el código “COD005- No se ha podido verificar la identidad del usuario” en el <resultadoidentidad> del Jugador. Si estamos en la operación de *verificarJugador* se realizará la llamada al RGIAJ devolviéndose la respuesta correspondiente.

4.2.2.11 Validación de formato del Número de soporte

En el caso en que el Número de Soporte no tenga el formato correcto, se devuelve el código “COD905- Formato de datos incorrecto - Número de soporte erróneo., ” en todos los resultados del Jugador.

4.2.2.12 Validación de número de soporte con NIF

En el caso de que siendo un NF se envíe el número de soporte, se devuelve el código “COD906 - Formato de datos incorrecto - Con NIF no se puede enviar número de Soporte” en todos los resultados del Jugador.

4.2.3 Casos de Prueba de la Operación de verificarJugador

4.2.3.1 Verificar un jugador No OK-EN RGIAJ

En este caso de Prueba el jugador no está identificado correctamente, porque *apellido1* debería ser “FELIZ” en vez de “FELIX”, y está inscrito en el RGIAJ.

Request:

```
<jugadores>
  <dni>10000320N</dni>
  <nombre>MARIA ENCINA</nombre>
  <apellido1>FELIX</apellido1>
  <apellido2>VARELA</apellido2>
  <fechaNacimiento>1944-05-18</fechaNacimiento>
  <numSoporte></numSoporte>
</jugadores>
```

Response:

```
<resultadosJugador>
  <dni>10000320N </dni>
  <resultadoRGIAJ>
```



```
<codigo>COD001</codigo>
<descripcion> El usuario está inscrito en el RGIAJ </descripcion>
</resultadoRGIAJ>
<resultadoidentidad>
  <codigo>COD004</codigo>
  <descripcion>La identidad del usuario es incorrecta - TITULAR NO
  IDENTIFICADO</descripcion>
</resultadoidentidad>
</resultadosJugador>
```

4.2.3.2 Verificar un jugador OK- EN RGIAJ

En este caso de Prueba el jugador está identificado correctamente y está inscrito en el RGIAJ.

Request:

```
<jugadores>
  <dni>10000320N</dni>
  <nombre>MARIA ENCINA</nombre>
  <apellido1>FELIZ</apellido1>
  <apellido2>VARELA</apellido2>
  <fechaNacimiento>1944-05-18</fechaNacimiento>
  <numSoporte></numSoporte>
</jugadores>
```

Response:

```
<resultadosJugador>
  <dni>10000320N</dni>
  <resultadoRGIAJ>
    <codigo>COD001</codigo>
    <descripcion> El usuario está inscrito en el RGIAJ </descripcion>
  </resultadoRGIAJ>
  <resultadoidentidad>
    <codigo>COD003</codigo>
```



```
<descripcion>La identidad del usuario es correcta</descripcion>  
</resultadoidentidad>  
</resultadosJugador>
```

4.2.3.3 Verificar dos jugadores OK- NO RGIAJ

En este caso de Prueba se envía dos jugadores y ambos están identificados correctamente y no están inscritos en el RGIAJ.

Request:

```
<jugadores>  
  <dni>20000717D</dni>  
  <nombre>Ricardo</nombre>  
  <apellido1>García</apellido1>  
  <apellido2>García</apellido2>  
  <fechaNacimiento>1967-11-03</fechaNacimiento>  
  <numSoporte></numSoporte>  
</jugadores>  
<jugadores>  
  <dni>30000024C</dni>  
  <nombre>Francisco Javier</nombre>  
  <apellido1>García</apellido1>  
  <apellido2>Reyes</apellido2>  
  <fechaNacimiento>1945-12-21</fechaNacimiento>  
  <numSoporte></numSoporte>  
</jugadores>
```

Response:

```
<resultadosJugador>  
  <dni>20000717D</dni>  
  <resultadoRGIAJ>  
    <codigo>COD002</codigo>  
    <descripcion>El usuario no está inscrito en el RGIAJ</descripcion>  
  </resultadoRGIAJ>  
  <resultadoidentidad>  
    <codigo>COD003</codigo>  
    <descripcion>La identidad del usuario es correcta</descripcion>  
  </resultadoidentidad>
```



```
</resultadosJugador>
<resultadosJugador>
  <dni>30000024C</dni>
  <resultadoRGIAJ>
    <codigo>COD002</codigo>
    <descripcion>El usuario no está inscrito en el RGIAJ</descripcion>
  </resultadoRGIAJ>
  <resultadoidentidad>
    <codigo>COD003</codigo>
    <descripcion>La identidad del usuario es correcta</descripcion>
  </resultadoidentidad>
</resultadosJugador>
```

4.2.3.4 Verificar dos jugador, uno OK-NO RGIAJ y otro con ERROR FORMATO

En este caso de Prueba se envían dos jugadores. El primer jugador está correctamente identificado y no está inscrito en el RGIAJ. El segundo jugador no está identificado correctamente ya que se envía una letra errónea del *DNI/NIE* (el DNI/NIE correcto es 30000260A).

Request:

```
<jugadores>
  <dni>30000067V</dni>
  <nombre>Juan</nombre>
  <apellido1>García</apellido1>
  <apellido2>Losada</apellido2>
  <fechaNacimiento>1945-06-14</fechaNacimiento>
  <numSoporte></numSoporte>
</jugadores>
<jugadores>
  <dni>30000260B</dni>
  <nombre>Juan</nombre>
  <apellido1>García</apellido1>
  <apellido2>Soto</apellido2>
  <fechaNacimiento>1938-01-02</fechaNacimiento>
</jugadores>
```



Response:

```
<resultadosJugador>
  <dni>30000067V</dni>
  <resultadoRGIAJ>
    <codigo>COD002</codigo>
    <descripcion>El usuario no está inscrito en el RGIAJ</descripcion>
  </resultadoRGIAJ>
  <resultadoIdentidad>
    <codigo>COD003</codigo>
    <descripcion>La identidad del usuario es correcta</descripcion>
  </resultadoIdentidad>
</resultadosJugador>
<resultadosJugador>
  <dni>30000260B</dni>
  <resultadoRGIAJ>
    <codigo>COD901</codigo>
    <descripcion> Formato de datos incorrecto-DNI no válido </descripcion>
  </resultadoRGIAJ>
  <resultadoIdentidad>
    <codigo>COD901</codigo>
    <descripcion> Formato de datos incorrecto- DNI no válido </descripcion>
  </resultadoIdentidad>
</resultadosJugador>
```

4.2.3.5 Verificar dos jugador OK- EN RGIAJ los dos

En este caso de Prueba se envían dos jugadores. Los dos están identificados correctamente y están inscritos en el RGIAJ.

Request:

```
<jugadores>
  <dni>30000529L</dni>
  <nombre>Jose</nombre>
  <apellido1>García</apellido1>
  <apellido2>Plata</apellido2>
```



```
<fechaNacimiento>1944-10-04</fechaNacimiento>  
<numSoporte></numSoporte>  
</jugadores>  
<jugadores>  
  <dni>30000831E</dni>  
  <nombre>Pedro</nombre>  
  <apellido1>García</apellido1>  
  <apellido2>Giraldo</apellido2>  
  <fechaNacimiento>1946-02-20</fechaNacimiento>  
</jugadores>
```

Response:

```
<resultadosJugador>  
  <dni>30000529L</dni>  
  <resultadoRGIAJ>  
    <codigo>COD001</codigo>  
    <descripcion>El usuario está inscrito en el RGIAJ</descripcion>  
  </resultadoRGIAJ>  
  <resultadoldentidad>  
    <codigo>COD003</codigo>  
    <descripcion>La identidad del usuario es correcta</descripcion>  
  </resultadoldentidad>  
</resultadosJugador>  
<resultadosJugador>  
  <dni>30000831E</dni>  
  <resultadoRGIAJ>  
    <codigo>COD001</codigo>  
    <descripcion>El usuario está inscrito en el RGIAJ</descripcion>  
  </resultadoRGIAJ>  
  <resultadoldentidad>  
    <codigo>COD003</codigo>  
    <descripcion>La identidad del usuario es correcta</descripcion>  
  </resultadoldentidad>  
</resultadosJugador>
```



4.2.3.6 Verificar un jugador OK- NO RGIAJ

En este caso de Prueba se envía un jugador identificado correctamente y que no está inscrito en el RGIAJ.

Request:

```
<jugadores>
  <dni>65000010D</dni>
  <nombre>ÑÉBÌMA</nombre>
  <apellido1>L.ÜÇÄR</apellido1>
  <apellido2>ÇÖBËRĬÄ</apellido2>
  <fechaNacimiento>1969-08-31</fechaNacimiento>
</jugadores>
```

Response:

```
<resultadosJugador>
  <dni>65000010D</dni>
  <resultadoRGIAJ>
    <codigo>COD002</codigo>
    <descripcion>El usuario no está inscrito en el RGIAJ</descripcion>
  </resultadoRGIAJ>
  <resultadoIdentidad>
    <codigo>COD003</codigo>
    <descripcion>La identidad del usuario es correcta</descripcion>
  </resultadoIdentidad>
</resultadosJugador>
```

4.2.3.7 Verificar un jugador OK- NO RGIAJ (pero menor)

En este caso de Prueba el jugador está identificado correctamente y no está inscrito en el RGIAJ, y es menor de edad.

Request:

```
<jugadores>
  <dni>Y0336380L</dni>
```



```
<nombre>GUSTAVO KAUA</nombre>  
<apellido1>GARCIA</apellido1>  
<fechaNacimiento>2003-09-25</fechaNacimiento>  
<numSoporte></numSoporte>  
</jugadores>
```

Response:

```
<resultadosJugador>  
  <dni>Y0336380L</dni>  
  <resultadoRGIAJ>  
    <codigo>COD002</codigo>  
    <descripcion>El usuario no está inscrito en el RGIAJ</descripcion>  
  </resultadoRGIAJ>  
  <resultadoidentidad>  
    <codigo>COD003</codigo>  
    <descripcion>La identidad del usuario es correcta</descripcion>  
  </resultadoidentidad>  
</resultadosJugador>
```

4.2.3.8 Verificar un jugador con Nif Anulado- NO RGIAJ

En este caso de Prueba el jugador no está identificado correctamente por estar su Nif Anulado y no está inscrito en el RGIAJ.

Request:

```
<jugadores>  
  <dni>20000128H</dni>  
  <nombre>MARIA</nombre>  
  <apellido1>RUIZ</apellido1>  
  <apellido2>ALONSO</apellido2>  
  <fechaNacimiento>1944-05-18</fechaNacimiento>  
  <numSoporte></numSoporte>  
</jugadores>
```

Response:



```
<resultadosJugador>
  <dni>20000128H</dni>
  <resultadoRGIAJ>
    <codigo>COD002</codigo>
    <descripcion> El usuario no está inscrito en el RGIAJ </descripcion>
  </resultadoRGIAJ>
  <resultadoidentidad>
    <codigo>COD004</codigo>
    <descripcion>La identidad del usuario es incorrecta - DNI del Titular
    Anulado</descripcion>
  </resultadoidentidad>
</resultadosJugador>
```

4.2.3.9 Verificar un jugador con Nif Retenido- NO RGIAJ

En este caso de Prueba el jugador no está identificado correctamente por estar su Nif Retenido y no está inscrito en el RGIAJ.

Request:

```
<jugadores>
  <dni>00001823Y</dni>
  <nombre>JAIME</nombre>
  <apellido1>SANZ</apellido1>
  <apellido2>MOLINA</apellido2>
  <fechaNacimiento>1960-01-20</fechaNacimiento>
  <numSoporte></numSoporte>
</jugadores>
```

Response:

```
<resultadosJugador>
  <dni>00001823Y</dni>
  <resultadoRGIAJ>
    <codigo>COD002</codigo>
    <descripcion> El usuario no está inscrito en el RGIAJ </descripcion>
  </resultadoRGIAJ>
  <resultadoidentidad>
    <codigo>COD004</codigo>
    <descripcion> La identidad del usuario es incorrecta - Documento retirado
```



judicialmente, retenido por existencia de duplicados o anulado por la unidad de documentación de españoles.

```
</descripcion>
</resultadoidentidad>
</resultadosJugador>
```

4.2.3.10 Verificar un jugador al que le faltan datos obligatorios

En este caso de Prueba no se envía el *nombre* (debería ir "MARIUS").

Request:

```
<jugadores>
  <dni>X2975617S</dni>
  <apellido1>GARCIA</apellido1>
  <fechaNacimiento>1969-04-19</fechaNacimiento>
</jugadores>
```

Response:

Error de Esquema en el soapfault del servicio:

```
<env:Fault xmlns:env='http://schemas.xmlsoap.org/soap/envelope/'>
  <faultcode>env:Server</faultcode>
  <faultstring>
    org.xml.sax.SAXException: cvc-complex-type.2.4.a: Invalid content was
    found starting with element &apos;apellido1&apos;. One of
    &apos;{nombre}&apos; is expected.
  </faultstring>
</env:Fault>
```

4.2.3.11 Verificar un jugador OK- EN RGIAJ

En este caso de Prueba el jugador está identificado correctamente y está inscrito en el RGIAJ.

Request:



```
<jugadores>
  <dni>X2975617S</dni>
  <nombre>MARIUS</nombre>
  <apellido1>GARCIA</apellido1>
  <fechaNacimiento>1969-04-19</fechaNacimiento>
  <numSoporte></numSoporte>
</jugadores>
```

Response:

```
<resultadosJugador>
  <dni>X2975617S</dni>
  <resultadoRGIAJ>
    <codigo>COD001</codigo>
    <descripcion>El usuario está inscrito en el RGIAJ</descripcion>
  </resultadoRGIAJ>
  <resultadoidentidad>
    <codigo>COD003</codigo>
    <descripcion>La identidad del usuario es correcta</descripcion>
  </resultadoidentidad>
</resultadosJugador>
```

4.2.3.12 Verificar un jugador sin enviar fecha de nacimiento

En este caso de Prueba se envía un jugador sin su fecha de nacimiento (que es "1966-01-23").

Request:

```
<jugadores>
  <dni>X6246189X</dni>
  <nombre>DRISS</nombre>
  <apellido1>GARCIA</apellido1>
  <apellido2></apellido2>
</jugadores>
```

Response:



Error de Esquema en el soapfault del servicio

```
<env:Fault xmlns:env='http://schemas.xmlsoap.org/soap/envelope/'>
  <faultcode>
    env:Server
  </faultcode>
  <faultstring>
    org.xml.sax.SAXException: cvc-complex-type.2.4.b: The content of element
    &apos;jugadores&apos; is not complete. One of &apos;{nombre}&apos; is
    expected.
  </faultstring>
</env:Fault>
```

4.2.3.13 Verificar un jugador que NO OK – NO EN RGIAJ

En este caso de Prueba se envía un jugador, que no esta correctamente identificado.

Request:

```
<jugadores>
  <dni>00000074M</dni>
  <nombre>MARIA</nombre>
  <apellido1>GARCIA</apellido1>
  <apellido2>PEREZ</apellido2>
  <fechaNacimiento>1990-09-01</fechaNacimiento>
</jugadores>
```

Response:

```
<resultadosJugador>
  <dni>00000074M </dni>
  <resultadoRGIAJ>
    <codigo>COD002</codigo>
    <descripcion>El usuario no está inscrito en el RGIAJ</descripcion>
  </resultadoRGIAJ>
  <resultadoIdentidad>
```



```
<codigo>COD004</codigo>
  <descripcion>La identidad del usuario es incorrecta - Documento retirado
  judicialmente, retenido por existencia de duplicados o anulado por la unidad de
  documentación de españoles. </descripcion>
</resultadoIdentidad>
</resultadosJugador>
```

4.2.3.14 Verificar el mismo DNI dos veces en la misma petición

En este caso de Prueba se envía dos veces el mismo jugador en una única petición por lo que dará el error de DNI duplicado.

Request:

```
<jugadores>
  <dni>30000529L</dni>
  <nombre>Jose</nombre>
  <apellido1>García</apellido1>
  <apellido2>Plata</apellido2>
  <fechaNacimiento>1944-10-04</fechaNacimiento>
</jugadores>
<jugadores>
  <dni>30000529L</dni>
  <nombre>Jose</nombre>
  <apellido1>García</apellido1>
  <apellido2>Plata</apellido2>
  <fechaNacimiento>1944-10-04</fechaNacimiento>
</jugadores>
```

Response:

```
<env:Fault xmlns:env='http://schemas.xmlsoap.org/soap/envelope/'>
  <faultcode> env:Server</faultcode>
  <faultstring>La petición tiene identificadores de jugador duplicados, no es
  válida</faultstring>
  <detail><ns2:verificarJugadorFault
  xmlns:ns2='http://ws.ordenacionjuego.gob.es/VerificacionJugadores/'>
  <verificarJugadorFault>ERR003 </verificarJugadorFault></ns2:verificarJugadorFault>
  </detail>
</env:Fault>
```



4.2.3.15 Verificar un jugador con un NIE y número de soporte

En estos casos de Prueba se envía un jugador con NIE y número de soporte.

Request:

```
<jugadores>
  <dni>X6246189X</dni>
  <nombre>Driss</nombre>
  <apellido1>Garcia</apellido1>
  <apellido2></apellido2>
  <fechaNacimiento>1966-01-23</fechaNacimiento>
  <numSoporte>E11343669</numSoporte>
</jugadores>
```

Response:

```
<resultadosJugador>
  <dni>X6246189X</dni>
  <resultadoRGIAJ>
    <codigo>COD002</codigo>
    <descripcion>El usuario no está inscrito en el RGIAJ</descripcion>
  </resultadoRGIAJ>
  <resultadoidentidad>
    <codigo>COD003</codigo>
    <descripcion>La identidad del usuario es correcta</descripcion>
  </resultadoidentidad>
</resultadosJugador>
```

Request:

```
<jugadores>
  <dni>X0021895E</dni>
  <nombre>Eduardo</nombre>
  <apellido1>Garcia</apellido1>
  <apellido2></apellido2>
  <fechaNacimiento>1923-08-01</fechaNacimiento>
  <numSoporte>E00000012</numSoporte>
```



</jugadores>

Response:

```
<resultadosJugador>
  <dni>X0021895E</dni>
  <resultadoRGIAJ>
    <codigo>COD002</codigo>
    <descripcion>El usuario no está inscrito en el RGIAJ</descripcion>
  </resultadoRGIAJ>
  <resultadoidentidad>
    <codigo>COD003</codigo>
    <descripcion>La identidad del usuario es correcta</descripcion>
  </resultadoidentidad>
</resultadosJugador>
```



Si el número de soporte que se envía no es el correcto devolverá que el Titular no está identificado correctamente:

Request:

```
<jugadores>  
  <dni>X0021895E</dni>  
  <nombre>Eduardo</nombre>  
  <apellido1>Garcia</apellido1>  
  <apellido2></apellido2>  
  <fechaNacimiento>1923-08-01</fechaNacimiento>  
  <numSoporte>E00000011</numSoporte>  
</jugadores>
```

Response:

```
<resultadosJugador>  
  <dni>X0021895E</dni>  
  <resultadoRGIAJ>  
    <codigo>COD002</codigo>  
    <descripcion>El usuario no está inscrito en el RGIAJ</descripcion>  
  </resultadoRGIAJ>  
  <resultadoidentidad>  
    <codigo>COD004</codigo>  
    <descripcion> La identidad del usuario es incorrecta - TITULAR NO  
    IDENTIFICADO </descripcion>  
  </resultadoidentidad>  
</resultadosJugador>
```



4.2.3.16 Verificar un jugador con un NIE y número de soporte con formato erróneo

En este caso de Prueba se envía un jugador con NIE y Número de soporte, pero el formato de número de Soporte es erróneo. Se devuelve el error de formato correspondiente.

Request:

```
<jugadores>
  <dni>X6246189X</dni>
  <nombre>Driss</nombre>
  <apellido1>Garcia</apellido1>
  <apellido2></apellido2>
  <fechaNacimiento>1966-01-23</fechaNacimiento>
  <numSoporte>X11343669</numSoporte>
</jugadores>
```

Response:

```
<resultadosJugador>
  <dni>X6246189X</dni>
  <resultadoRGIAJ>
    <codigo>COD905</codigo>
    <descripcion> Formato de datos incorrecto - Número de soporte
    erróneo.</descripcion>
  </resultadoRGIAJ>
  <resultadoidentidad>
    <codigo>COD905</codigo>
    <descripcion> Formato de datos incorrecto- Número de soporte
    erróneo.</descripcion>
  </resultadoidentidad>
</resultadosJugador>
```



4.2.3.17 Verificar un jugador con un DNI enviando número de soporte

En este caso de Prueba se envía un jugador con DNI y Número de soporte. Como no está permitido devuelve el error de formato correspondiente.

Request:

```
<jugadores>
  <dni>00000074M</dni>
  <nombre>MARIA</nombre>
  <apellido1>GARCIA</apellido1>
  <apellido2>PEREZ</apellido2>
  <fechaNacimiento>1990-09-01</fechaNacimiento>
  <numSoporte>E22353243</numSoporte>
</jugadores>
```

Response:

```
<resultadosJugador>
  <dni>00000074M </dni>
  <resultadoRGIAJ>
    <codigo>COD906</codigo>
    <descripcion> Formato de datos incorrecto - Con NIF no se puede enviar
    número de Soporte </descripcion>
  </resultadoRGIAJ>
  <resultadoldentidad>
    <codigo>COD906</codigo>
    <descripcion> Formato de datos incorrecto - Con NIF no se puede enviar
    número de Soporte </descripcion>
  </resultadoldentidad>
</resultadosJugador>
```



4.2.4 Casos de Prueba de la Operación de verificarIdentidad

4.2.4.1 Verificar una Identidad OK

En este caso de Prueba el jugador está identificado correctamente, aunque es menor de edad.

Request:

```
<jugadores>
  <dni>Y0336380L</dni>
  <nombre>GUSTAVO KAUA</nombre>
  <apellido1>GARCIA</apellido1>
  <fechaNacimiento>2003-09-25</fechaNacimiento>
</jugadores>
```

Response:

```
<resultadosIdentidad>
  <dni>Y0336380L</dni>
  <resultadoIdentidad>
    <codigo>COD003</codigo>
    <descripcion>La identidad del usuario es correcta</descripcion>
  </resultadoIdentidad>
</resultadosIdentidad>
```

4.2.4.2 Verificar una identidad No OK

En este caso de Prueba el jugador no está identificado correctamente, porque *fechaNacimiento* debería ser "1944-05-18" en vez de "1940-05-18"

Request:



```
<jugadores>
  <dni>10000320N</dni>
  <nombre>MARIA ENCINA</nombre>
  <apellido1>FELIZ</apellido1>
  <apellido2>VARELA</apellido2>
  <fechaNacimiento>1940-05-18</fechaNacimiento>
</jugadores>
```

Response:

```
<resultadosIdentidad>
  <dni>10000320N </dni>
  <resultadoIdentidad>
    <codigo>COD004</codigo>
    <descripcion>La identidad del usuario es incorrecta - TITULAR NO
IDENTIFICADO</descripcion>
  </resultadoIdentidad>
</resultadosIdentidad>
```

4.2.4.3 Verificar una identidad con ERROR-FORMATO

En este caso de Prueba se envía un jugador no identificado correctamente porque falta la letra del DNI/NIE (lo correcto sería "65000010D").

Request:

```
<jugadores>
  <dni>65000010</dni>
  <nombre>ÑÉBÌMA</nombre>
  <apellido1>L.ÛÇÄR</apellido1>
  <apellido2>ÇÖBËRÏÄ</apellido2>
  <fechaNacimiento>1969-08-31</fechaNacimiento>
</jugadores>
```

Response:



```
<resultadosIdentidad>
  <dni>65000010</dni>
  <resultadoIdentidad>
    <codigo>COD901</codigo>
    <descripcion> Formato de datos incorrecto – DNI no válido </descripcion>
  </resultadoIdentidad>
</resultadosIdentidad>
```

4.2.4.4 Verificar una identidad que no esta correctamente identificada

En este caso de Prueba se envía un jugador, que no está correctamente identificado.

Request:

```
<jugadores>
  <dni>00000016Q</dni>
  <nombre>MARIO</nombre>
  <apellido1>LOPEZ</apellido1>
  <apellido2>GARCIA</apellido2>
  <fechaNacimiento>1991-09-01</fechaNacimiento>
</jugadores>
```

Response:

```
<resultadosIdentidad>
  <dni>00000016Q</dni>
  <resultadoIdentidad>
    <codigo>COD004</codigo>
    <descripcion> La identidad del usuario es incorrecta - Documento retirado
judicialmente, retenido por existencia de duplicados o anulado por la unidad de
documentación de españoles. </descripcion>
  </resultadoIdentidad>
</resultadosIdentidad>
```

4.2.4.5 Verificar identidad de Nif Anulado

En este caso de Prueba el jugador no está identificado correctamente por estar su Nif Anulado.



Request:

```
<jugadores>
  <dni>20000128H</dni>
  <nombre>MARIA</nombre>
  <apellido1>RUIZ</apellido1>
  <apellido2>ALONSO</apellido2>
  <fechaNacimiento>1944-05-18</fechaNacimiento>
  <numSoporte></numSoporte>
</jugadores>
```

Response:

```
<resultadosIdentidad>
  <dni>20000128H</dni>
  <resultadoIdentidad>
    <codigo>COD004</codigo>
    <descripcion>La identidad del usuario es incorrecta - DNI del Titular
    Anulado</descripcion>
  </resultadoIdentidad>
</resultadosIdentidad>
```

4.2.4.6 Verificar un jugador con Nif Retenido

En este caso de Prueba el jugador no está identificado correctamente por estar su Nif Retenido.

Request:

```
<jugadores>
  <dni>00001823Y</dni>
  <nombre>JAIME</nombre>
  <apellido1>SANZ</apellido1>
  <apellido2>MOLINA</apellido2>
  <fechaNacimiento>1960-01-20</fechaNacimiento>
  <numSoporte></numSoporte>
</jugadores>
```



Response:

```
<resultadosIdentidad>
  <dni>00001823Y</dni>
  <resultadoIdentidad>
    <codigo>COD004</codigo>
    <descripcion> La identidad del usuario es incorrecta - Documento retirado
judicialmente, retenido por existencia de duplicados o anulado por la unidad de
documentación de españoles.
  </descripcion>
  </resultadoIdentidad>
</resultadosIdentidad>
```

4.2.4.7 Verificar un jugador con un NIE y número de soporte

En estos casos de Prueba se envía un jugador con NIE y número de soporte.

Request:

```
<jugadores>
  <dni>X6246189X</dni>
  <nombre>Driss</nombre>
  <apellido1>Garcia</apellido1>
  <apellido2></apellido2>
  <fechaNacimiento>1966-01-23</fechaNacimiento>
  <numSoporte>E11343669</numSoporte>
</jugadores>
```

Response:

```
<resultadosIdentidad>
  <dni>X6246189X</dni>
  <resultadoIdentidad>
    <codigo>COD003</codigo>
    <descripcion>La identidad del usuario es correcta</descripcion>
  </resultadoIdentidad>
```



</resultadosIdentidad>

Request:

<jugadores>

<dni>X0021895E</dni>

<nombre>Eduardo</nombre>

<apellido1>Garcia</apellido1>

<apellido2></apellido2>

<fechaNacimiento>1923-08-01</fechaNacimiento>

<numSoporte>E00000012</numSoporte>

</jugadores>

Response:

<resultadosIdentidad>

<dni>X0021895E</dni>

<resultadoIdentidad>

<codigo>COD003</codigo>

<descripcion>La identidad del usuario es correcta</descripcion>

</resultadoIdentidad>

</resultadosIdentidad>

Si el número de soporte que se envía no es el correcto devolverá que el Titular no está identificado correctamente:

Request:

<jugadores>

<dni>X6246189X</dni>

<nombre>Driss</nombre>

<apellido1>Garcia</apellido1>

<apellido2></apellido2>

<fechaNacimiento>1966-01-23</fechaNacimiento>



```
<numSoporte>E11343665</numSoporte>  
</jugadores>
```

Response:

```
<resultadosIdentidad>  
  <dni> X6246189X</dni>  
  <resultadoidentidad>  
    <codigo>COD004</codigo>  
    <descripcion> La identidad del usuario es incorrecta - TITULAR NO  
    IDENTIFICADO </descripcion>  
  </resultadoidentidad>  
</resultadosIdentidad>
```

4.2.4.8 Verificar un jugador con un NIE y número de soporte con formato erróneo

En este caso de Prueba se envía un jugador con NIE y Número de soporte, pero el formato de número de Soporte es erróneo. Se devuelve el error de formato correspondiente.

Request:

```
<jugadores>  
  <dni>X6246189X</dni>  
  <nombre>Driss</nombre>  
  <apellido1>Garcia</apellido1>  
  <apellido2></apellido2>  
  <fechaNacimiento>1966-01-23</fechaNacimiento>  
  <numSoporte>X11343665</numSoporte>  
</jugadores>
```

Response:

```
<resultadosIdentidad>  
  <dni>X6246189X</dni>  
  <resultadoidentidad>  
    <codigo>COD905</codigo>
```



```
<descripcion> Formato de datos incorrecto- Número de soporte  
erróneo.</descripcion>  
</resultadoidentidad>  
<resultadosidentidad>
```

4.2.4.9 Verificar un jugador con un DNI enviando número de soporte

En este caso de Prueba se envía un jugador con DNI y Número de soporte. Como no está permitido devuelve el error de formato correspondiente.

Request:

```
<jugadores>  
  <dni>00000074M</dni>  
  <nombre>MARIA</nombre>  
  <apellido1>GARCIA</apellido1>  
  <apellido2>PEREZ</apellido2>  
  <fechaNacimiento>1990-09-01</fechaNacimiento>  
  <numSoporte>E22353243</numSoporte>  
</jugadores>
```

Response:

```
<resultadosidentidad>  
  <dni>00000074M </dni>  
  <resultadoidentidad>  
    <codigo>COD906</codigo>  
    <descripcion> Formato de datos incorrecto - Con NIF no se puede enviar  
    número de Soporte </descripcion>  
  </resultadoidentidad>  
</resultadosidentidad>
```



4.2.5 Casos de Prueba de la Operación de verificarRGIAJ

4.2.5.1 Verificar en RGIAJ un jugador que está inscrito

En este caso de Prueba se envía un jugador que está inscrito en el RGIAJ.

Request:

```
<jugadores>
  <dni>X2975617S</dni>
  <nombre>MARIUS</nombre>
  <apellido1>GARCIA</apellido1>
  <fechaNacimiento>1969-04-19</fechaNacimiento>
  <numSoporte></numSoporte>
</jugadores>
```

Response:

```
<resultadosRGIAJ>
  <dni>X2975617S</dni>
  <resultadoRGIAJ>
    <codigo>COD001</codigo>
    <descripcion>El usuario está inscrito en el RGIAJ</descripcion>
  </resultadoRGIAJ>
</resultadosRGIAJ>
```

4.2.5.2 Verificar en RGIAJ un jugador que no está inscrito

En este caso de Prueba el jugador no está no está inscrito en el RGIAJ.

Request:



```
<jugadores>
  <dni>30000067V</dni>
  <nombre>Juan</nombre>
  <apellido1>García</apellido1>
  <apellido2>Losada</apellido2>
  <fechaNacimiento>1945-06-14</fechaNacimiento>
  <numSoporte></numSoporte>
</jugadores>
```

```
<resultadosRGIAJ>
  <dni>30000067V </dni>
  <resultadoRGIAJ>
    <codigo>COD002</codigo>
    <descripcion>El usuario no está inscrito en el RGIAJ</descripcion>
  </resultadoRGIAJ>
</resultadosRGIAJ>
```

4.2.5.3 Verificar un jugador que está inscrito en RGIAJ pero sin enviar todos los datos

En este caso de Prueba se envía un jugador que está inscrito en el RGIAJ, pero solo se envía el DNI/NIE, por lo que da error al no estar todos los datos obligatorios.

Request:

```
<jugadores>
  <dni>30000529L</dni>
</jugadores>
```

Response:

Error de Esquema en el soapfault del servicio

```
<env:Fault xmlns:env='http://schemas.xmlsoap.org/soap/envelope/'>
  <faultcode>
    env:Server
```



```
</faultcode>
<faultstring>
  org.xml.sax.SAXException: cvc-complex-type.2.4.b: The content of element
  &apos;jugadores&apos; is not complete. One of &apos;{nombre}&apos; is
  expected.
</faultstring>
</env:Fault>
```

Si se envía todos los datos, ya devuelve que está inscrito en el RGIAJ, por lo que el Operador no debe darle de alta.

Request:

```
<jugadores>
  <dni>30000529L</dni>
  <nombre>Jose</nombre>
  <apellido1>García</apellido1>
  <apellido2>Plata</apellido2>
  <fechaNacimiento>1944-10-04</fechaNacimiento>
</jugadores>
```

Response:

```
<resultadosRGIAJ>
  <dni>30000529L</dni>
  <resultadoRGIAJ>
    <codigo>COD001</codigo>
    <descripcion>El usuario está inscrito en el RGIAJ</descripcion>
  </resultadoRGIAJ>
</resultadosRGIAJ>
```

4.2.5.4 Verificar en RGIAJ un jugador que no está inscrito

En este caso de Prueba el jugador no está no está inscrito en el RGIAJ. **Request:**

```
<jugadores>
  <dni>20000717D</dni>
  <nombre>Ricardo</nombre>
  <apellido1>García</apellido1>
```



```
<apellido2>García</apellido2>  
<fechaNacimiento>1967-11-03</fechaNacimiento>  
<numSoporte></numSoporte>  
</jugadores>
```

Response:

```
<resultadosRGIAJ>  
<dni>20000717D</dni>  
<resultadoRGIAJ>  
<codigo>COD002</codigo>  
<descripcion>El usuario no está inscrito en el RGIAJ</descripcion>  
</resultadoRGIAJ>  
</resultadosRGIAJ>
```

4.2.5.5 Verificar un RGIAJ con ERROR-FORMATO

En este caso de Prueba se envía un jugador no identificado correctamente porque falta la letra del DNI/NIE (lo correcto sería “65000010D”).

Request:

```
<jugadores>  
<dni>65000010</dni>  
<nombre>ÑÉBÌMA</nombre>  
<apellido1>L.ÜÇÄR</apellido1>  
<apellido2>ÇÖBËRİÀ</apellido2>  
<fechaNacimiento>1969-08-31</fechaNacimiento>  
</jugadores>
```

Response:

```
<resultadosRGIAJ>  
<dni>65000010</dni>  
<resultadoRGIAJ>  
<codigo>COD901</codigo>  
<descripcion> Formato de datos incorrecto – DNI no válido </descripcion>  
</resultadoRGIAJ>  
</resultadosRGIAJ>
```



4.2.5.6 Verificar un jugador con un NIE y número de soporte con formato erróneo

En este caso de Prueba se envía un jugador con NIE y Número de soporte, pero el formato de número de Soporte es erróneo. Se devuelve el error de formato correspondiente.

Request:

```
<jugadores>
  <dni>X6246189X</dni>
  <nombre>Driss</nombre>
  <apellido1>Garcia</apellido1>
  <apellido2></apellido2>
  <fechaNacimiento>1966-01-23</fechaNacimiento>
  <numSoporte>X11343665</numSoporte>
</jugadores>
```

Response:

```
<resultadosRGIAJ>
  <dni>X6246189X</dni>
  <resultadoRGIAJ>
    <codigo>COD905</codigo>
    <descripcion> Formato de datos incorrecto - Número de soporte
    erróneo.</descripcion>
  </resultadoRGIAJ>
</resultadosRGIAJ>
```



4.2.5.7 Verificar un jugador con un DNI enviando número de soporte

En este caso de Prueba se envía un jugador con DNI y Número de soporte. Como no está permitido devuelve el error de formato correspondiente.

Request:

```
<jugadores>  
  <dni>00000074M</dni>  
  <nombre>MARIA</nombre>  
  <apellido1>GARCIA</apellido1>  
  <apellido2>PEREZ</apellido2>  
  <fechaNacimiento>1990-09-01</fechaNacimiento>  
  <numSoporte>E22353243</numSoporte>  
</jugadores>
```

Response:

```
<resultadosRGIAJ>  
  <dni>00000074M </dni>  
  <resultadoRGIAJ>  
    <codigo>COD906</codigo>  
    <descripcion> Formato de datos incorrecto - Con NIF no se puede enviar  
    número de Soporte </descripcion>  
  </resultadoRGIAJ>  
</resultadosRGIAJ>
```



4.2.6 Casos de Prueba de la Operación de verificarCambiosRGIAJ

Con el objetivo de comprobar el funcionamiento de la operación *verificarCambiosRGIAJ* se pone a disposición una serie de juegos de prueba y un procedimiento para utilizarlo.

En la operación *VerificarCambiosRGIAJ* del entorno de producción se devolverá un listado con los DNIs , cuya situación en el RGIAJ haya sido preguntada por el operador, y que haya cambiado dicha situación desde entonces.

El contenido de esta tabla se mantendrá indefinidamente , ya que será utilizada para consultar antes de realizar un pago si un jugador cuya cuenta estaba debidamente activada, ha sido posteriormente inscrito en el RGIAJ, y en consecuencia no se puede realizar el pago.

El campo *fechaCambio* podrá ser utilizado por el operador para discriminar los cambios efectuados en la tabla desde su última consulta a esta operación y realizar el tratamiento correspondiente en su sistema de información. Así mismo, es preciso recalcar que sucesivos cambios de estado para un mismo DNI no generarán varias entradas en la tabla, si no que utilizará la entrada ya existente, actualizando el contenido de los campos *motivoCambio* y *fechaCambio*.

A continuación se muestra la lista de DNI's con las que se podrán realizar las pruebas de la operación de *verificarCambiosRGIAJ*.

- **Lista de DNI's:**

<u>DNI</u>	<u>ESTADO</u>	<u>FECHA</u>
00000076F	A	01/11/2011T14:07:13
00585858W	A	02/11/2011T03:08:10
60111111J	A	10/11/2011T01:04:09
X0000021K	A	01/11/2011T23:02:01
Y0909090Y	A	12/11/2011T22:02:01
31313131B	B	01/11/2011T17:01:05
X0405060F	B	04/11/2011T06:06:01
43214321N	B	15/11/2011T19:03:01
Y9080701Q	B	13/11/2011T12:08:01
00000284P	B	20/11/2011T09:10:02



Nota: las fechas del ejemplo variarán en función del momento de la prueba de cada operador.

Con objeto de conseguir evaluar correctamente el funcionamiento de la operación que nos ocupa, el proceso a seguir será el siguiente:

1. En primer lugar se comprobará que antes de validar la situación en el RGIAJ de alguno de los DNI's de la lista, la operación no devuelve ningún dato.
2. A continuación se irán verificando jugadores (DNI's) con las operaciones de *VerificarJugador* o *VerificarRGIAJ* y se irá comprobando que al llamar a la operación de *VerificarCambiosRGIAJ* se devuelven la información indicada anteriormente de dichos DNI's.
 - a. Es importante verificar que en ningún caso se devuelven cambios RGIAJ para DNIs por los que no se ha utilizado las operaciones de verificación indicadas.
 - b. Es importante recalcar que las respuestas para estos DNIs de pruebas en relación con los servicios de identidad no son vinculantes para la prueba de esta operación, por lo que no han de ser tenido en cuenta el resultado de las validaciones de identidad realizadas durante este proceso (nota sólo válida para el entorno de pruebas). Por ello, en el proceso de verificación del funcionamiento de esta operación puede usarse cualquier valor, dentro de los formatos aceptados dentro de cada tipo de dato, en los campos nombre, apellidos o fecha de nacimiento.
 - c. Una vez se hayan verificado toda la lista de DNIs del juego de pruebas, esta operación siempre devolverá la lista completa. En el caso de querer desasignar un DNI concreto de un operador, se enviará un correo electrónico a dgoj.soproteoperadores@minhap.es indicando la operación solicitada.

4.2.6.1 Verificar cambios en RGIAJ sin verificar jugadores

En este caso de prueba el operador aún no ha verificado ninguno de los DNI's indicados, el servicio devuelve un mensaje vacío.

Response:

```
<env:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="element-10-1317739360466-4168018">  
<ns2:verificarCambiosRGIAJResponse  
xmlns:ns2="http://ws.juego.es/VerificacionJugadores/"></env:Body>
```



4.2.6.2 Verificar cambios en RGIAJ al ir verificando jugadores

En este caso de prueba, en primer lugar un operador debe haber verificado uno de los DNI's indicados, a través de la operación de *verificarJugador* o *verificarRGIAJ*. A partir de la siguiente hora en punto, al llamar a la operación de *verificarCambiosRGIAJ* se recibe la información del cambio en el Registro General de Interdictos al Juego del DNI anteriormente verificado.

Ejemplo:

Se verifica el DNI 00000058N y se llama a la operación de *verificarCambiosRGIAJ*:

Response:

```
<cambioRGIAJ>
  <DNI>00000058N</DNI>
  <motivoCambio>A</motivoCambio>
  <fechaCambio>2011-11-01T14:07:13.257+01:00</fechaCambio>
</cambioRGIAJ>
```

A continuación se podrá verificar otro DNI de la lista o varios de forma simultánea. Posteriormente se volverá a llamar a la VerificaciónCambiosRGIAJ, esperando a la siguiente hora en punto. Se devuelven los dos DNI's ya verificados.

Se verifica el DNI 31313131B y se llama a la operación de *verificarCambiosRGIAJ*:

Response:

```
<cambioRGIAJ>
  <DNI>00000058N</DNI>
  <motivoCambio>A</motivoCambio>
  <fechaCambio>2011-11-01T14:07:13.257+01:00</fechaCambio>
</cambioRGIAJ>
<cambioRGIAJ>
  <DNI>31313131B </DNI>
  <motivoCambio>B</motivoCambio>
  <fechaCambio>01/11/2011T17:01:05.257+01:00</fechaCambio>
```



</cambioRGIAJ>

De esta forma se podrán ir verificando el resto de jugadores (de uno en uno o en grupos), comprobando el resultado mediante la llamada a la operación de *verificarCambiosRGIAJ* y verificando que únicamente se devuelven DNIS por los que previamente se ha preguntado. Una vez todos los jugadores de la lista estén verificados la respuesta será siempre la misma (la lista completa).

4.2.7 Casos de Prueba de la Operación de bajaJugador

4.2.7.1 Baja un jugador dado de alta por el Operador

En este caso de Prueba el jugador está previamente dado de alta para el operador, ya que se verificó su estado en RGIAJ a través de la operación de *verificarJugador* o de *verificarRGIAJ*, obteniendo una respuesta sin errores, es decir COD001 o COD002. Por tanto al realizar la petición de baja, esta se realizará correctamente.

Request:

```
<bajaJugador>  
<dni>10000320N</dni>  
<causa>A</causa>  
</bajaJugador>
```

Response:

```
<resultadoBajaJugador>  
  <dni>10000320N</dni>  
  <resultadoBaja>  
    <codigo>COD010</codigo>  
    <descripcion> Baja realizada correctamente</descripcion>  
  </resultadoBaja>  
</resultadoBajaJugador>
```

4.2.7.2 Baja de un jugador no dado de alta previamente por el Operador

En este caso de Prueba el jugador no está previamente dado de alta para el operador, ya que no se verificó su estado en RGIAJ a través de la operación de *verificarJugador* o de *verificarRGIAJ* o si se verificó dio una respuesta de error de formato o de error de sistema. Por tanto no se realiza baja, informando de ello en el resultado de la petición.



Request:

```
<bajaJugador>  
  <dni>99999030K</dni>  
  <causa>R</causa>  
</bajaJugador>
```

Response:

```
<resultadoBajaJugador>  
  <dni>99999030K</dni>  
  <resultadoBaja>  
    <codigo>COD021</codigo>  
    <descripcion> Baja no realizada, el DNI no está activado para este  
    Operador </descripcion>  
  </resultadoBaja>  
</resultadoBajaJugador>
```

4.2.7.3 Baja de un jugador ya dado de baja previamente por el Operador

En este caso de Prueba el jugador ya fue dado de baja previamente para el operador. Por tanto no se realiza baja, informando de ello en el resultado de la petición.

Request:

```
<bajaJugador>  
  <dni>10000320N</dni>  
  <causa>A</causa>  
</bajaJugador>
```

Response:

```
<resultadoBajaJugador>  
  <dni>10000320N</dni>  
  <resultadoBaja>  
    <codigo>COD022</codigo>
```



```
<descripcion> Baja no realizada, el DNI ya está dado de baja para este
Operador </descripcion>
</resultadoBaja>
</resultadoBajaJugador>
```

4.2.7.4 Causa de la baja errónea

En este caso de Prueba se envía una causa de baja no valida o vacia. Por tanto no se realiza baja, informando de ello en el resultado de la petición.

Request:

```
<bajaJugador>
  <dni>10000949C</dni>
  <causa>Z</causa>
</bajaJugador>
```

O el valor de causa vacio

```
<bajaJugador>
  <dni>10000949C</dni>
  <causa></causa>
</bajaJugador>
```

Response:

```
<resultadoBajaJugador>
  <dni>10000949C</dni>
  <resultadoBaja>
    <codigo>COD023</codigo>
    <descripcion>>Formato de datos incorrecto - Baja no realizada, causa de la
baja errónea </descripcion>
  </resultadoBaja>
</resultadoBajaJugador>
```



4.2.7.5 Baja de un jugador enviando un DNI/NIE erróneo

En este caso de Prueba se solicita la baja de un jugador no identificado correctamente ya que se envía una letra errónea del *DNI/NIE* (el DNI/NIE correcto es 30000260A).

Por tanto no se realiza baja, informando de ello en el resultado de la petición.

Request:

```
<bajaJugador>  
  <dni>30000260B</dni>  
  <causa>A</causa>  
</bajaJugador>
```

Response:

```
<resultadoBajaJugador>  
  <dni>30000260B</dni>  
  <resultadoBaja>  
    <codigo>COD901</codigo>  
    <descripcion> Formato de datos incorrecto - DNI no válido </descripcion>  
  </resultadoBaja>  
</resultadoBajaJugador>
```

4.2.7.6 Baja de un jugador donde no se envía el DNI/NIE

En este caso de Prueba se solicita la baja de un jugador pero sin enviar el DNI/NIE que es obligatorio. Por tanto no se realiza baja, informando de ello por fault del servicio.

Request:

```
<bajaJugador>  
  <causa>A</causa>  
</bajaJugador>
```

Response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
<soap:Body>
```



```
<soap:Fault>  
<faultcode>soap:Client</faultcode>  
<faultstring>Unmarshalling Error: cvc-complex-type.2.4.a: Invalid content was found starting  
with element 'causa'. One of '{dni}' is expected. </faultstring>  
</soap:Fault>  
</soap:Body>  
</soap:Envelope>
```

4.2.7.7 Baja de un jugador donde no se envía la causa

En este caso de Prueba se solicita la baja de un jugador pero sin enviar la causa que es obligatorio. Por tanto no se realiza baja, informando de ello por fault del servicio.

Request:

```
<bajaJugador>  
  <dni>10000949C</dni>  
</bajaJugador>
```

Response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
<soap:Body>  
<soap:Fault>  
<faultcode>soap:Client</faultcode>  
<faultstring>Unmarshalling Error: cvc-complex-type.2.4.b: The content of element  
'bajaJugador' is not complete. One of '{causa}' is expected. </faultstring>  
</soap:Fault>  
</soap:Body>  
</soap:Envelope>
```

4.2.7.8 Verificar cambios en RGIAJ después de una baja

En este caso de prueba, se da de baja de uno de los DNI's indicados en el caso de prueba de VerificarCambiosRGIAJ y que se haya previamente verificado a través de la operación de *verificarJugador* o *verificarRGIAJ*. A partir de la siguiente hora en punto, al llamar a la operación de *verificarCambiosRGIAJ* se comprobará que dicho DNI ya no aparece en la lista de respuesta.



Ejemplo:

Anteriormente se verificó tanto el DNI 00000058N y el DNI 31313131B , y al llamar a la operación de *verificarCambiosRGIAJ*, se comprobó que aparecían ambos en la lista

Response:

```
<cambioRGIAJ>
  <DNI>00000058N</DNI>
  <motivoCambio>A</motivoCambio>
  <fechaCambio>2011-11-01T14:07:13.257+01:00</fechaCambio>
</cambioRGIAJ>
<cambioRGIAJ>
  <DNI>31313131B </DNI>
  <motivoCambio>B</motivoCambio>
  <fechaCambio>01/11/2011T17:01:05.257+01:00</fechaCambio>
</cambioRGIAJ>
```

En este caso de prueba se da de baja el DNI 00000058N a través de la operación de *bajaJugadores* y posteriormente se volverá a llamar a la *VerificarCambiosRGIAJ*, esperando a la siguiente hora en punto. Ya no aparece este DNI's en la lista de resultados.

Response:

```
<cambioRGIAJ>
  <DNI>31313131B </DNI>
  <motivoCambio>B</motivoCambio>
  <fechaCambio>01/11/2011T17:01:05.257+01:00</fechaCambio>
</cambioRGIAJ>
```



4.3 Juegos de ensayo para verificar el entorno de producción

Endpoint: <https://ws.ordenacionjuego.gob.es/VerificacionJugadores-v.r/VerificacionJugadores>

WSDL: <https://ws.ordenacionjuego.gob.es/VerificacionJugadores-v.r/VerificacionJugadores?WSDL>

Donde v es un dígito que indica el número de versión y r el número de reléase dentro de la versión.

A solicitud de los operadores que hayan solicitado licencia singular y una vez analizadas las pruebas realizadas en el entorno de pre-producción, la DGOJ podrá autorizar la realización de pruebas de integración contra el entorno de producción, sin utilizar datos reales, con el juego de ensayo descrito en este apartado.

Este entorno de test en producción incluye todas las operaciones disponibles en el entorno de producción con casos reales.

Previamente, las empresas deberán haber solicitado el alta en el entorno de producción en las condiciones indicadas en el apartado 4.4. Una vez autorizado el acceso, las empresas deberán comunicar el inicio y el final de estas pruebas, a la dirección de correo dgj.soproteoperadores@minhap.es indicando en el asunto del mensaje “TEST WS-VerificaciónJugadores”.

La DGOJ autorizará la fecha y hora de inicio y fin de las pruebas en el entorno de producción, una vez verificado que la empresa ha solicitado licencia singular y que las pruebas en el entorno de pre-producción han sido satisfactorias.

4.3.1 Relación de casos de error genéricos

4.3.1.1 Datos Obligatorios a null

En el caso de que alguno o algunos de los datos establecidos como obligatorios en el servicio, venga a nulos se devuelve el error de esquema correspondiente en el *soapfault* del servicio.

4.3.1.2 Autenticación del Operador

En el caso de que el Operador no sea correctamente autenticado, se devolverá el mensaje indicando que el Operador es inválido en el *soapfault* del servicio.



4.3.1.3 Validación de formato del DNI/NIE

En el caso de que el formato del DNI o el NIE no sea el correcto se devuelve el código “COD901- Formato de datos incorrecto –DNI no válido” en todos los resultados del Jugador.

4.3.1.4 Validación de letra del DNI/NIE

En el caso de que la letra del DNI o el NIE no sea la correcta se devuelve el código “COD901- Formato de datos incorrecto –DNI no válido” en todos los resultados del Jugador.

4.3.1.5 Validación de Apellido2 obligatorio si es un NIF

En el caso de que se pregunte por el NIF y el segundo Apellido sea nulo se devuelve el código “COD903- Formato de datos incorrecto – Es obligatorio el *Nombre/Apellido1/Apellido2” (se especifica que campo obligatorio va a nulos) en todos los resultados del Jugador.

4.3.1.6 Validación de fecha de nacimiento posterior al 01/01/1900

En el caso de que se pregunte por una fecha de nacimiento anterior al 01/01/1900 se devuelve el código “COD904- Formato de datos incorrecto – Fecha de nacimiento anterior a 1900” en todos los resultados del Jugador.

4.3.1.7 Validación de caracteres no permitidos

En el caso de que el nombre o los apellidos contengan alguno de los caracteres considerados como no válidos (como por ejemplo números, \$,% ,etc...), se devuelve el código “COD902-La petición tiene caracteres no validos en el *Nombre/Apellido1/Apellido2” (se especifica en que campo se han detectado los caracteres inválidos) en todos los resultados del Jugador.de datos incorrecto” en todos los resultados del Jugador.

4.3.1.8 DNI/NIE repetidos

En el caso de que se envíe un lote de jugadores en el que se repitan alguno de los DNIs o de los NIEs se devolverá el mensaje indicando que la petición tiene identificadores de jugadores duplicados en el soapfault del servicio.

COD005

Con el objetivo de emular un posible comportamiento del WS en el entorno de producción accediendo a datos reales devolviendo COD005 durante un determinado espacio de tiempo, y luego recuperando el servicio, en los casos de prueba en el entorno de producción, cuando el resultado de la VerificarIdentidad devuelto COD005, a partir del tercer intento de consulta se devolverá COD003.



Validación de formato del Número de soporte

En el caso en que el Número de Soporte no tenga el formato correcto, se devuelve el código “COD905- Formato de datos incorrecto- Número de soporte erróneo.” en todos los resultados del Jugador.

Validación de número de soporte con NIF

En el caso de que siendo un NF se envíe el número de soporte, se devuelve el código “COD906- Formato de datos incorrecto-Con NIF no se puede enviar número de Soporte” en todos los resultados del Jugador.

4.3.2 Casos de Prueba de la Operación de verificar Jugador

4.3.2.1 Verificar un jugador No OK-NO RGIAJ

En este caso de Prueba el jugador no está identificado correctamente y no está inscrito en el RGIAJ. Por tanto el Operador no debe permitirle darse de alta por no estar correctamente identificado.

Request:

```
<jugadores>  
  <dni>00000000T</dni>  
  <nombre>MARIA</nombre>  
  <apellido1>GONZALEZ</apellido1>  
  <apellido2>LOPEZ</apellido2>  
  <fechaNacimiento>1960-04-10</fechaNacimiento>  
  <numSoporte></numSoporte>  
</jugadores>
```

Response:

```
<resultadosJugador>  
  <dni>00000000T</dni>  
  <resultadoRGIAJ>  
    <codigo>COD002</codigo>
```



```
<descripcion>El usuario no está inscrito en el RGIAJ</descripcion>
</resultadoRGIAJ>
<resultadoIdentidad>
  <codigo>COD004</codigo>
  <descripcion>La identidad del usuario es incorrecta - TITULAR NO
IDENTIFICADO </descripcion>
</resultadoIdentidad>
</resultadosJugador>
```

4.3.2.2 Verificar un jugador OK- EN RGIAJ

En este caso de Prueba el jugador está identificado correctamente y está inscrito en el RGIAJ. Por tanto el Operador no debe permitirle darse de alta por estar inscrito en el RGIAJ.

Request:

```
<jugadores>
  <dni>00000034B</dni>
  <nombre>EMILIO JOSE</nombre>
  <apellido1>GARCIA-RUIZ</apellido1>
  <apellido2>GONZALEZ</apellido2>
  <fechaNacimiento>1953-08-01</fechaNacimiento>
  <numSoporte></numSoporte>
</jugadores>
```

Response:

```
<resultadosJugador>
  <dni>00000034B</dni>
  <resultadoRGIAJ>
    <codigo>COD001</codigo>
    <descripcion> El usuario está inscrito en el RGIAJ </descripcion>
  </resultadoRGIAJ>
  <resultadoIdentidad>
```



```
<codigo>COD003</codigo>
<descripcion>La identidad del usuario es correcta</descripcion>
</resultadoidentidad>
</resultadosJugador>
```

4.3.2.3 Verificar dos jugadores OK- NO RGIAJ

En este caso de Prueba se envía dos jugadores y ambos están identificados correctamente y no están inscritos en el RGIAJ. Por tanto pueden ser dados de alta por el Operador.

Request:

```
<jugadores>
  <dni>00000060Z</dni>
  <nombre>Santiago</nombre>
  <apellido1>Martín</apellido1>
  <apellido2>Güel</apellido2>
  <fechaNacimiento>1982-02-23</fechaNacimiento>
  <numSoporte></numSoporte>
</jugadores>
<jugadores>
  <dni>X0000040V</dni>
  <nombre>Pedro</nombre>
  <apellido1>Araña</apellido1>
  <apellido2></apellido2>
  <fechaNacimiento>1977-09-09</fechaNacimiento>
  <numSoporte></numSoporte>
</jugadores>
```

Response:

```
<resultadosJugador>
  <dni>00000060Z</dni>
```



```
<resultadoRGIAJ>
  <codigo>COD002</codigo>
  <descripcion>El usuario no está inscrito en el RGIAJ</descripcion>
</resultadoRGIAJ>
<resultadoidentidad>
  <codigo>COD003</codigo>
  <descripcion>La identidad del usuario es correcta</descripcion>
</resultadoidentidad>
</resultadosJugador>
<resultadosJugador>
  <dni>X0000040V</dni>
  <resultadoRGIAJ>
    <codigo>COD002</codigo>
    <descripcion>El usuario no está inscrito en el RGIAJ</descripcion>
  </resultadoRGIAJ>
  <resultadoidentidad>
    <codigo>COD003</codigo>
    <descripcion>La identidad del usuario es correcta</descripcion>
  </resultadoidentidad>
</resultadosJugador>
```

4.3.2.4 Verificar dos jugador, uno OK-NO RGIAJ y otro con ERROR FORMATO

En este caso de Prueba se envían dos jugadores. El primer jugador está correctamente identificado y no está inscrito en el RGIAJ, por tanto puede ser dado de alta por el Operador de Juego. El segundo jugador no está identificado correctamente ya que se envía una letra errónea del *DNI* (el Dni correcto es 00000018H). Por tanto el Operador no debe permitirle darse de alta por no estar identificado.

Request:

```
<jugadores>
  <dni>Y9999999G</dni>
  <nombre>Esperanza</nombre>
  <apellido1>San Juan</apellido1>
```



```
<apellido2>Mateo</apellido2>
<fechaNacimiento>1985-06-05</fechaNacimiento>
</jugadores>
<jugadores>
  <dni>00000018K</dni>
  <nombre>Carlos</nombre>
  <apellido1>Guerrero</apellido1>
  <apellido2>Grau</apellido2>
  <fechaNacimiento>1996-05-11</fechaNacimiento>
</jugadores>
```

Response:

```
<resultadosJugador>
  <dni>Y99999999G</dni>
  <resultadoRGIAJ>
    <codigo>COD002</codigo>
    <descripcion>El usuario no está inscrito en el RGIAJ</descripcion>
  </resultadoRGIAJ>
  <resultadoidentidad>
    <codigo>COD003</codigo>
    <descripcion>La identidad del usuario es correcta</descripcion>
  </resultadoidentidad>
</resultadosJugador>
<resultadosJugador>
  <dni>00000018K</dni>
  <resultadoRGIAJ>
    <codigo>COD901</codigo>
    <descripcion> Formato de datos incorrecto– DNI no válido </descripcion>
  </resultadoRGIAJ>
  <resultadoidentidad>
    <codigo>COD901</codigo>
    <descripcion> Formato de datos incorrecto – DNI no válido </descripcion>
  </resultadoidentidad>
</resultadosJugador>
```



4.3.2.5 Verificar dos jugador OK- EN RGIAJ los dos

En este caso de Prueba se envían dos jugadores. Los dos están identificados correctamente y están inscritos en el RGIAJ. Por tanto el Operador no debe permitirles darse de alta por estar en el RGIAJ.

Request:

```
<jugadores>
  <dni>00000024R</dni>
  <nombre>Juan Manuel</nombre>
  <apellido1>Terras</apellido1>
  <apellido2>Colon</apellido2>
  <fechaNacimiento>1962-09-20</fechaNacimiento>
</jugadores>
<jugadores>
  <dni>00000057B</dni>
  <nombre>Maria Fernanda</nombre>
  <apellido1>Velas</apellido1>
  <apellido2>Leon</apellido2>
  <fechaNacimiento>1973-04-06</fechaNacimiento>
</jugadores>
```

Response:

```
<resultadosJugador>
  <dni>00000024R</dni>
  <resultadoRGIAJ>
    <codigo>COD001</codigo>
    <descripcion>El usuario está inscrito en el RGIAJ</descripcion>
  </resultadoRGIAJ>
  <resultadoIdentidad>
    <codigo>COD003</codigo>
    <descripcion>La identidad del usuario es correcta</descripcion>
```



```
</resultadoidentidad>
</resultadosJugador>
<resultadosJugador>
  <dni>00000057B</dni>
  <resultadoRGIAJ>
    <codigo>COD001</codigo>
    <descripcion>El usuario está inscrito en el RGIAJ</descripcion>
  </resultadoRGIAJ>
  <resultadoidentidad>
    <codigo>COD003</codigo>
    <descripcion>La identidad del usuario es correcta</descripcion>
  </resultadoidentidad>
</resultadosJugador>
```

4.3.2.6 Verificar un jugador OK- NO RGIAJ

En este caso de Prueba se envía un jugador identificado correctamente y que no está inscrito en el RGIAJ. Por tanto puede ser dado de alta por el Operador.

Request:

```
<jugadores>
  <dni>X0000066C</dni>
  <nombre>Susana</nombre>
  <apellido1>Ayala</apellido1>
  <apellido2>Gracia</apellido2>
  <fechaNacimiento>1971-09-24</fechaNacimiento>
  <numSoporte></numSoporte>
</jugadores>
```

Response:

```
<resultadosJugador>
  <dni>X0000066C</dni>
```



```
<resultadoRGIAJ>
  <codigo>COD002</codigo>
  <descripcion>El usuario no está inscrito en el RGIAJ</descripcion>
</resultadoRGIAJ>
<resultadoidentidad>
  <codigo>COD003</codigo>
  <descripcion>La identidad del usuario es correcta</descripcion>
</resultadoidentidad>
</resultadosJugador>
```

4.3.2.7 Verificar un jugador No OK- EN RGIAJ

En este caso de Prueba se envía un jugador no identificado correctamente y que está inscrito en el RGIAJ. Por tanto puede ser dado de alta por el Operador.

Request:

```
<jugadores>
  <dni>X0000019L</dni>
  <nombre>Hassan</nombre>
  <apellido1>Almud</apellido1>
  <apellido2></apellido2>
  <fechaNacimiento>1960-01-01</fechaNacimiento>
</jugadores>
```

Response:

```
<resultadosJugador>
  <dni> X0000019L</dni>
  <resultadoRGIAJ>
    <codigo>COD001</codigo>
    <descripcion>El usuario está inscrito en el RGIAJ</descripcion>
  </resultadoRGIAJ>
  <resultadoidentidad>
    <codigo>COD004</codigo>
```



```
<descripcion>La identidad del usuario es incorrecta - TITULAR NO  
IDENTIFICADO POSITIVAMENTE - CIUDADANO NACIONALIZADO  
ESPAÑOL  
</descripcion>  
</resultadoIdentidad>  
</resultadosJugador>
```

4.3.2.8 Verificar un jugador OK- NO RGIAJ (pero menor)

En este caso de Prueba el jugador está identificado correctamente y no está inscrito en el RGIAJ. Pero en este caso el jugador es menor de edad por lo que el Operador de Juego no debe permitirle darse de alta.

Request:

```
<jugadores>  
  <dni>00000018H</dni>  
  <nombre>Carlos</nombre>  
  <apellido1>Guerrero</apellido1>  
  <apellido2>Grau</apellido2>  
  <fechaNacimiento>1996-05-11</fechaNacimiento>  
  <numSoporte></numSoporte>  
</jugadores>
```

Response:

```
<resultadosJugador>  
  <dni>00000018H</dni>  
  <resultadoRGIAJ>  
    <codigo>COD002</codigo>  
    <descripcion>El usuario no está inscrito en el RGIAJ</descripcion>  
  </resultadoRGIAJ>  
</resultadoIdentidad>
```



```
<codigo>COD003</codigo>  
<descripcion>La identidad del usuario es correcta</descripcion>  
</resultadoidentidad>  
</resultadosJugador>
```

4.3.2.9 Verificar un jugador al que le faltan datos obligatorios

En este caso de Prueba no se envía el *nombre* (debería ir "LORENA). Por tanto el Operador de Juego no debe permitirle darse de alta.

Si se envía el nombre , la identificación es correcta y esta inscrito en el RGIAJ.

Request:

```
<jugadores>  
  <dni>00000001R</dni>  
  <apellido1>D´con</apellido1>  
  <apellido2>Fernandez Ruiz</apellido2>  
  <fechaNacimiento>1955-05-18</fechaNacimiento>  
</jugadores>
```

Response:

Error de Esquema en el soapfault del servicio:

```
<env:Fault xmlns:env='http://schemas.xmlsoap.org/soap/envelope/'>  
  <faultcode>env:Server</faultcode>  
  <faultstring>  
    org.xml.sax.SAXException: cvc-complex-type.2.4.a: Invalid content was  
    found starting with element &apos;apellido1&apos;. One of  
    &apos;{nombre}&apos; is expected.  
  </faultstring>  
</env:Fault>
```

4.3.2.10 Verificar un jugador OK- EN RGIAJ

En este caso de Prueba el jugador está identificado correctamente y está inscrito en el RGIAJ. Por tanto el Operador no debe permitirle darse de alta por estar inscrito en



el RGIAJ.

Request:

```
<jugadores>  
  <dni>X0000052Y</dni>  
  <nombre>OSCAR</nombre>  
  <apellido1>L•ÜÇÄR</apellido1>  
  <apellido2></apellido2>  
  <fechaNacimiento>1964-03-03</fechaNacimiento>  
</jugadores>
```

Response:

```
<resultadosJugador>  
  <dni>X0000052Y</dni>  
  <resultadoRGIAJ>  
    <codigo>COD001</codigo>  
    <descripcion>El usuario está inscrito en el RGIAJ</descripcion>  
  </resultadoRGIAJ>  
  <resultadoIdentidad>  
    <codigo>COD003</codigo>  
    <descripcion>La identidad del usuario es correcta</descripcion>  
  </resultadoIdentidad>  
</resultadosJugador>
```



4.3.2.11 Verificar un jugador sin enviar fecha de nacimiento

En este caso de Prueba se envía un jugador sin su fecha de nacimiento (que es "1992-08-20"). Por tanto el Operador no debe permitirle darse de alta por no estar correctamente identificado.

Si se envía la fecha, dará que el jugador no está identificado y que no está inscrito en el RGIAJ,

Request:

```
<jugadores>
  <dni>00000047R</dni>
  <nombre>XAVIER</nombre>
  <apellido1>Cañas</apellido1>
  <apellido2>Gonzalez</apellido2>
</jugadores>
```

Response:

Error de Esquema en el soapfault del servicio

```
<env:Fault xmlns:env='http://schemas.xmlsoap.org/soap/envelope/'>
  <faultcode>
    env:Server
  </faultcode>
  <faultstring>
    org.xml.sax.SAXException: cvc-complex-type.2.4.b: The content of element
    &apos;jugadores&apos; is not complete. One of &apos;{nombre}&apos; is
    expected.
  </faultstring>
</env:Fault>
```



4.3.2.12 Verificar un jugador que NO EXISTE el DNI

En este caso de Prueba se envía un jugador que no está dentro de los DNI de prueba . por lo que se devolverá que no está identificado ni en el RGIAJ. Por tanto el Operador no debe permitirle darse de alta por no estar identificado.

Request:

```
<jugadores>
  <dni>00000074M</dni>
  <nombre>MARIA</nombre>
  <apellido1>GARCIA</apellido1>
  <apellido2>PEREZ</apellido2>
  <fechaNacimiento>1990-09-01/fechaNacimiento>
</jugadores>
```

Response:

```
<resultadosJugador>
  <dni>00000074M </dni>
  <resultadoRGIAJ>
    <codigo>COD002</codigo>
    <descripcion>El usuario no está inscrito en el RGIAJ</descripcion>
  </resultadoRGIAJ>
  <resultadoIdentidad>
    <codigo>COD004</codigo>
    <descripcion> La identidad del usuario es incorrecta - TITULAR NO IDENTIFICADO
  </descripcion>
  </resultadoIdentidad>
</resultadosJugador>
```



4.3.2.13 Verificar un jugador que da error de Servicio (COD005)

En este caso de Prueba se envía un jugador, en el que la respuesta será que no se ha podido verificar la identidad del usuario (COD005). Esta respuesta se obtendrá las dos primera veces que se haga una petición de este DNI en un mismo día .La tercera petición en adelante de ese DNI en el mismo día devolverá que se ha podido identificar correctamente (COD003) , para así simular el comportamiento que se dará con datos reales.

Request:

```
<jugadores>
  <dni>00000015S</dni>
  <nombre>JAIME</nombre>
  <apellido1>TOLEDO</apellido1>
  <apellido2>HERNANDEZ</apellido2>
  <fechaNacimiento>1992-08-17</fechaNacimiento>
</jugadores>
```

Response:

Las dos primeras peticiones en un mismo día:

```
<resultadosJugador>
  <dni>00000015S </dni>
  <resultadoRGIAJ>
    <codigo>COD002</codigo>
    <descripcion>El usuario no está inscrito en el RGIAJ</descripcion>
  </resultadoRGIAJ>
  <resultadoldentidad>
    <codigo>COD005</codigo>
    <descripcion No se ha podido verificar la identidad del usuario
  </descripcion>
  </resultadoldentidad>
</resultadosJugador>
```



La tercera petición en adelante en un mismo día:

```
<resultadosJugador>
  <dni>00000015S </dni>
  <resultadoRGIAJ>
    <codigo>COD002</codigo>
    <descripcion>El usuario no está inscrito en el RGIAJ</descripcion>
  </resultadoRGIAJ>
  <resultadoIdentidad>
    <codigo>COD003</codigo>
    <descripcion La identidad del usuario es correcta</descripcion>
  </resultadoIdentidad>
</resultadosJugador>
```

4.3.2.14 Verificar el mismo DNI dos veces en una misma petición

En este caso de Prueba se envía dos veces el mismo jugador en una única petición por lo que dará el error de DNI duplicado.

Request:

```
<jugadores>
  <dni>X0000066C</dni>
  <nombre>Susana</nombre>
  <apellido1>Ayala</apellido1>
  <apellido2>Gracia</apellido2>
  <fechaNacimiento>1971-09-24</fechaNacimiento>
</jugadores>
<jugadores>
  <dni>X0000066C</dni>
  <nombre>Susana</nombre>
  <apellido1>Ayala</apellido1>
  <apellido2>Gracia</apellido2>
  <fechaNacimiento>1971-09-24</fechaNacimiento>
</jugadores>
```



```
Response:<env:Fault xmlns:env='http://schemas.xmlsoap.org/soap/envelope/'>  
<faultcode> env:Server</faultcode>  
<faultstring>La petición tiene identificadores de jugador duplicados, no es  
válida</faultstring>  
<detail><ns2:verificarJugadorFault  
xmlns:ns2='http://ws.ordenacionjuego.gob.es/VerificacionJugadores/'>  
<verificarJugadorFault>ERR003 </verificarJugadorFault></ns2:verificarJugadorFault>  
</detail>  
</env:Fault>
```

4.3.3 Casos de Prueba de la Operación de verificarIdentidad

4.3.3.1 Verificar una Identidad OK pero menor de edad

En este caso de Prueba el jugador está identificado correctamente. Pero en este caso el jugador es menor de edad por lo que el Operador de Juego no debe permitirle darse de alta.

Request:

```
<jugadores>  
  <dni>00000018H</dni>  
  <nombre>Carlos</nombre>  
  <apellido1>Guerrero</apellido1>  
  <apellido2>Grau</apellido2>  
  <fechaNacimiento>1996-05-11</fechaNacimiento>  
</jugadores>
```

Response:

```
<resultadosIdentidad>  
  <dni>00000018H</dni>  
  <resultadoidentidad>  
    <codigo>COD003</codigo>  
    <descripcion>La identidad del usuario es correcta</descripcion>  
  </resultadoidentidad>
```



</resultadosIdentidad>

4.3.3.2 Verificar una Identidad OK

En este caso de Prueba el jugador está identificado correctamente.

Request:

```
<jugadores>
  <dni>Y9999999G</dni>
  <nombre>Esperanza</nombre>
  <apellido1>San Juan</apellido1>
  <apellido2>Mateo</apellido2>
  <fechaNacimiento>1985-06-05</fechaNacimiento>
  <numSoporte></numSoporte>
</jugadores>
```

Response:

```
<resultadosIdentidad>
  <dni>Y9999999G</dni>
  <resultadoidentidad>
    <codigo>COD003</codigo>
    <descripcion>La identidad del usuario es correcta</descripcion>
  </resultadoidentidad>
</resultadosIdentidad>
```

4.3.3.3 Verificar una identidad No OK

En este caso de Prueba el jugador no está identificado correctamente, porque *fechaNacimiento* debería ser “1953-08-01” en vez de “1953-08-02” Por tanto el Operador no debe permitirle darse de alta por no estar correctamente identificado.

Si se envía la fecha correcta se devuelve que la entidad del usuario es correcta.



Request:

```
<jugadores>
  <dni>00000034B</dni>
  <nombre>EMILIO JOSE</nombre>
  <apellido1>GARCIA-RUIZ</apellido1>
  <apellido2>GONZALEZ</apellido2>
  <fechaNacimiento>1953-08-02</fechaNacimiento>
</jugadores>
```

Response:

```
<resultadosIdentidad>
  <dni>00000034B</dni>
  <resultadoIdentidad>
    <codigo>COD004</codigo>
    <<descripcion>La identidad del usuario es incorrecta - TITULAR NO
IDENTIFICADO </descripcion>
  </resultadoIdentidad>
</resultadosIdentidad>
```

4.3.3.4 Verificar una identidad No OK

En este caso de Prueba el jugador no está identificado correctamente. Por tanto el Operador no debe permitirle darse de alta por no estar correctamente identificado.

Request:

```
<jugadores>
  <dni>00000048W</dni>
  <nombre>Laura</nombre>
  <apellido1>Dorado</apellido1>
  <apellido2>Laçon</apellido2>
```



```
<fechaNacimiento>1968-04-27</fechaNacimiento>  
</jugadores>
```

Response:

```
<resultadosIdentidad>  
  <dni>00000048W</dni>  
  <resultadoidentidad>  
    <codigo>COD004</codigo>  
    <<descripcion>La identidad del usuario es incorrecta - El documento  
    solicitado puede estar en trámite de renovación, retenido, retirado por autoridad  
    judicial o anulado por duplicidad. Si necesita aclaración, póngase en contacto con  
    la Unidad de Documentación de Españoles </descripcion>  
  </resultadoidentidad>  
</resultadosIdentidad>
```

4.3.3.5 Verificar una identidad con ERROR-FORMATO

En este caso de Prueba se envía un jugador no identificado correctamente porque falta la letra del DNI (lo correcto sería "00000060Z"). Por tanto el Operador no debe permitirle darse de alta por no estar identificado.

Si se envía el DNI correcto se devuelve que la entidad del usuario es correcta.

Request:

```
<jugadores>  
  <dni>00000060</dni>  
  <nombre>Santiago</nombre>  
  <apellido1>Martín</apellido1>  
  <apellido2>Güel</apellido2>  
  <fechaNacimiento>1982-02-23</fechaNacimiento>  
</jugadores>
```

Response:



```
<resultadosIdentidad>
  <dni>00000060</dni>
  <resultadoIdentidad>
    <codigo>COD901</codigo>
    <descripcion> Formato de datos incorrecto– DNI no válido </descripcion>
  </resultadoIdentidad>
</resultadosIdentidad>
```

4.3.3.6 Verificar una identidad con caracteres no permitidos

En este caso de Prueba se envía un jugador no identificado correctamente porque se envían caracteres no permitidos en el nombre (números). Por tanto el Operador no debe permitirle darse de alta por no estar identificado.

Si se envía el nombre correcto (“Pedro”) se devuelve que la entidad del usuario es correcta.

Request:

```
<jugadores>
  <dni>X0000040V</dni>
  <nombre>Pedro 123</nombre>
  <apellido1>Araña</apellido1>
  <apellido2></apellido2>
  <fechaNacimiento>1977-09-09</fechaNacimiento>
</jugadores>
```

Response:

```
<resultadosIdentidad>
  <dni> X0000040V</dni>
  <resultadoIdentidad>
    <codigo>COD902</codigo>
    <descripcion> La petición tiene caracteres no válidos </descripcion>
  </resultadoIdentidad>
</resultadosIdentidad>
```



4.3.3.7 Verificar una identidad que NO EXISTE

En este caso de Prueba se envía un jugador que no está dentro de los DNI de prueba . por lo que se devolverá que no está identificado. Por tanto el Operador no debe permitirle darse de alta por no estar identificado.

Request:

```
<jugadores>
  <dni>00000016Q</dni>
  <nombre>MARIO</nombre>
  <apellido1>LOPEZ</apellido1>
  <apellido2>GARCIA</apellido2>
  <fechaNacimiento>1991-09-01</fechaNacimiento>
</jugadores>
```

Response:

```
<resultadosIdentidad>
  <dni>00000016Q </dni>
  <resultadoIdentidad>
    <codigo>COD004</codigo>
    <descripcion> La identidad del usuario es incorrecta - TITULAR NO
IDENTIFICADO
  </descripcion>
  </resultadoIdentidad>
</resultadosIdentidad>
```

4.3.3.8 Verificar una identidad que da error de Servicio (COD005)

En este caso de Prueba se envía un jugador, en el que la respuesta será que no se ha podido verificar la identidad del usuario (COD005). Esta respuesta se obtendrá las dos primera veces que se haga una petición de este DNI en un mismo día .La tercera petición en adelante de ese DNI en el mismo día devolverá que se ha podido



identificar correctamente (COD003) , para así simular el comportamiento que se dará con datos reales.

Request:

```
<jugadores>
  <dni>00000015S</dni>
  <nombre>JAIME</nombre>
  <apellido1>TOLEDO</apellido1>
  <apellido2>HERNANDEZ</apellido2>
  <fechaNacimiento>1992-08-17</fechaNacimiento>
</jugadores>
```

Response:

Las dos primeras peticiones en un mismo día:

```
<resultadosIdentidad>
  <dni>00000015S </dni>
  <resultadoIdentidad>
    <codigo>COD005</codigo>
    <descripcion No se ha podido verificar la identidad del usuario
  </descripcion>
  </resultadoIdentidad>
</resultadosIdentidad>
```

La tercera petición en adelante en un mismo día:

```
<resultadosIdentidad>
  <dni>00000015S </dni>
  <resultadoIdentidad>
    <codigo>COD003</codigo>
    <descripción> La identidad del usuario es correcta </descripción>
  </resultadoIdentidad>
</resultadosIdentidad>
```



4.3.4 Casos de Prueba de la Operación de verificarRGIAJ

4.3.4.1 Verificar en RGIAJ un jugador que está inscrito

En este caso de Prueba se envía un jugador que está inscrito en el RGIAJ. Por tanto el Operador no debe permitirle darse de alta por estar en el RGIAJ.

Request:

```
<jugadores>
  <dni>00000034B</dni>
  <nombre>EMILIO JOSE</nombre>
  <apellido1>GARCIA-RUIZ</apellido1>
  <apellido2>GONZALEZ</apellido2>
  <fechaNacimiento>1953-08-01</fechaNacimiento>
</jugadores>
```

Response:

```
<resultadosRGIAJ>
  <dni>00000034B </dni>
  <resultadoRGIAJ>
    <codigo>COD001</codigo>
    <descripcion>El usuario está inscrito en el RGIAJ</descripcion>
  </resultadoRGIAJ>
</resultadosRGIAJ>
```

4.3.4.2 Verificar en RGIAJ un jugador que no está inscrito

En este caso de Prueba el jugador no está no está inscrito en el RGIAJ.

Request:



```
<jugadores>
  <dni>00000000T</dni>
  <nombre>MARIA</nombre>
  <apellido1>GONZALEZ</apellido1>
  <apellido2>LOPEZ</apellido2>
  <fechaNacimiento>1960-04-10</fechaNacimiento>
  <numSoporte></numSoporte>
</jugadores>
```

```
<resultadosRGIAJ>
  <dni>00000000T </dni>
  <resultadoRGIAJ>
    <codigo>COD002</codigo>
    <descripcion>El usuario no está inscrito en el RGIAJ</descripcion>
  </resultadoRGIAJ>
</resultadosRGIAJ>
```

4.3.4.3 Verificar un jugador que está inscrito en RGIAJ pero sin enviar todos los datos

En este caso de Prueba se envía un jugador que está inscrito en el RGIAJ, pero solo se envía el DNI, por lo que da error al no estar todos los datos.

Request:

```
<jugadores>
  <dni>00000048W</dni>
</jugadores>
```

Response:

Error de Esquema en el soapfault del servicio:

```
<env:Fault xmlns:env='http://schemas.xmlsoap.org/soap/envelope/'>
  <faultcode>env:Server</faultcode>
```



```
<faultstring>  
    org.xml.sax.SAXException: cvc-complex-type.2.4.a: Invalid content was  
    found starting with element &apos;apellido1&apos;. One of  
    &apos;{nombre}&apos; is expected.  
</faultstring>  
</env:Fault>
```

Si se envía todos los datos, ya devuelve que está inscrito en el RGIAJ, por lo que el Operador no debe darle de alta.

Request:

```
<jugadores>  
    <dni>00000048W</dni>  
    <nombre>Laura</nombre>  
    <apellido1>Dorado</apellido1>  
    <apellido2>Laçon</apellido2>  
    <fechaNacimiento>1968-04-27</fechaNacimiento>  
</jugadores>
```

Response:

```
<resultadosRGIAJ>  
    <dni>00000048W</dni>  
    <resultadoRGIAJ>  
        <codigo>COD001</codigo>  
        <descripcion>El usuario está inscrito en el RGIAJ</descripcion>  
    </resultadoRGIAJ>  
</resultadosRGIAJ>
```

4.3.4.4 Verificar en RGIAJ un jugador que no está inscrito

En este caso de Prueba el jugador no está no está inscrito en el RGIAJ.

Request:



```
<jugadores>
  <dni>X0000066C</dni>
  <nombre>Susana</nombre>
  <apellido1>Ayala</apellido1>
  <apellido2>Gracia</apellido2>
  <fechaNacimiento>1971-09-24</fechaNacimiento>
</jugadores>
```

Response:

```
<resultadosRGIAJ>
  <dni> X0000066C</dni>
  <resultadoRGIAJ>
    <codigo>COD002</codigo>
    <descripcion>El usuario no está inscrito en el RGIAJ</descripcion>
  </resultadoRGIAJ>
</resultadosRGIAJ>
```

4.3.4.5 Verificar un RGIAJ con ERROR-FORMATO

En este caso de Prueba se envía un jugador al que le falta la letra del DNI (lo correcto sería "00000060Z"). Por tanto el Operador no debe permitirle darse de alta por dar error de formato. Si se envía el DNI correcto se devuelve que no está inscrito en el RGIAJ.

Request:

```
<jugadores>
  <dni>00000060</dni>
  <nombre>Santiago</nombre>
  <apellido1>Martín</apellido1>
  <apellido2>Güel</apellido2>
  <fechaNacimiento>1982-02-23</fechaNacimiento>
</jugadores>
```



Response:

```
<resultadosIdentidad>  
  <dni>00000060</dni>  
  <resultadoidentidad>  
    <codigo>COD901</codigo>  
    <descripcion> Formato de datos incorrecto– DNI no válido </descripcion>  
  </resultadoidentidad>  
</resultadosIdentidad>
```

4.3.5 Casos de Prueba de la Operación de verificarCambiosRGIAJ

Con el objetivo de comprobar el funcionamiento de la operación *verificarCambiosRGIAJ* se pone a disposición una serie de juegos de prueba y un procedimiento para utilizarlo.

En la operación *VerificarCambiosRGIAJ* del entorno de producción, cuando el operador ya esté operando con datos reales, se devolverá un listado con los DNIs , cuya situación en el RGIAJ haya sido preguntada por el operador, y que haya cambiado dicha situación desde entonces.

El contenido de esta tabla se mantendrá indefinidamente , ya que será utilizada para consultar antes de realizar un pago si un jugador cuya cuenta estaba debidamente activada, ha sido posteriormente inscrito en el RGIAJ, y en consecuencia no se puede realizar el pago.

El campo *fechaCambio* podrá ser utilizado por el operador para discriminar los cambios efectuados en la tabla desde su última consulta a esta operación y realizar el tratamiento correspondiente en su sistema de información. Así mismo, es preciso recalcar que sucesivos cambios de estado para un mismo DNI no generarán varias entradas en la tabla, si no que utilizará la entrada ya existente, actualizando el contenido de los campos *motivoCambio* y *fechaCambio*.

A continuación se muestra la lista de DNI's con las que se podrán realizar las pruebas en el entorno de producción de la operación de *verificarCambiosRGIAJ*, hasta que el operador empiece a operar con datos reales.



• **Lista de DNI's:**

<u>DNI</u>	<u>ESTADO</u>	<u>FECHA</u>
00000002W	A	01/05/2013T11:00:00
00000003A	A	01/05/2013T11:00:00
X0000004G	A	01/05/2013T11:00:00
X0000005M	A	01/05/2013T11:00:00
X0000002W	B	01/05/2013T11:00:00
X0000003A	B	01/05/2013T11:00:00
00000004G	B	01/05/2013T11:00:00
00000005M	B	01/05/2013T11:00:00

Nota: las fechas del ejemplo variarán en función del momento de la prueba de cada operador.

Con objeto de conseguir evaluar correctamente el funcionamiento de la operación que nos ocupa, el proceso a seguir será el siguiente:

1. En primer lugar se comprobará que antes de validar la situación en el RGIAJ de alguno de los DNI's de la lista, la operación no devuelve ningún dato.
2. A continuación se irán verificando jugadores (DNI's) con las operaciones de *VerificarJugador* o *VerificarRGIAJ* y se irá comprobando que al llamar a la operación de *VerificarCambiosRGIAJ* se devuelven la información indicada anteriormente de dichos DNI's.
 - a. Es importante verificar que en ningún caso se devuelven cambios RGIAJ para DNIs por los que no se ha utilizado las operaciones de verificación indicadas.
 - b. Es importante recalcar que las respuestas para estos DNIs de pruebas en relación con los servicios de identidad no son vinculantes para la prueba de esta operación, por lo que no han de ser tenido en cuenta el resultado de las validaciones de identidad realizadas durante este proceso (nota sólo válida para estos datos de prueba). Por ello, en el proceso de verificación del funcionamiento de esta operación puede usarse cualquier valor, dentro de los formatos aceptados dentro de cada tipo de dato, en los campos nombre, apellidos o fecha de nacimiento.
 - c. Una vez se hayan verificado toda la lista de DNIs del juego de pruebas, esta operación siempre devolverá la lista completa. En el caso de querer desasignar un DNI concreto de un operador, se enviará un correo electrónico a dgoj.soporteoperadores@minhap.es indicando la operación solicitada.



4.3.5.1 Verificar cambios en RGIAJ sin verificar jugadores

En este caso de prueba el operador aún no ha verificado ninguno de los DNI's indicados, el servicio devuelve un mensaje vacío.

Response:

```
<soap:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="Id-1204052348">
<ns2:verificarCambiosRGIAJResponse
xmlns:ns2="http://ws.ordenacionjuego.gob.es/VerificacionJugadores"/></soap:Body>
```

4.3.5.2 Verificar cambios en RGIAJ al ir verificando jugadores

En este caso de prueba, en primer lugar un operador debe haber verificado uno de los DNI's indicados, a través de la operación de *verificarJugador* o *verificarRGIAJ*. A partir de la siguiente hora en punto, al llamar a la operación de *verificarCambiosRGIAJ* se recibe la información del cambio en el Registro General de Interdictos al Juego del DNI anteriormente verificado.

Ejemplo:

Se verifica el DNI 00000002W y se llama a la operación de *verificarCambiosRGIAJ*:

Response:

```
<cambioRGIAJ>
  <DNI>00000002W</DNI>
  <motivoCambio>A</motivoCambio>
  <fechaCambio>2013-05-01T14:07:13.257+01:00</fechaCambio>
</cambioRGIAJ>
```

A continuación se podrá verificar otro DNI de la lista o varios de forma simultánea. Posteriormente se volverá a llamar a la VerificaciónCambiosRGIAJ, esperando a la siguiente hora en punto. Se devuelven los dos DNI's ya verificados.

Se verifica el DNI X0000003A y se llama a la operación de *verificarCambiosRGIAJ*:



Response:

<cambioRGIAJ>

<DNI>00000002W </DNI>

<motivoCambio>A</motivoCambio>

<fechaCambio>2013-05-01T14:00:00.00+01:00</fechaCambio>

</cambioRGIAJ>

<cambioRGIAJ>

<DNI>X0000003A </DNI>

<motivoCambio>B</motivoCambio>

<fechaCambio>2013-05-01T15:00:00.00+01:00</fechaCambio>

</cambioRGIAJ>

De esta forma se podrán ir verificando el resto de jugadores (de uno en uno o en grupos), comprobando el resultado mediante la llamada a la operación de *verificarCambiosRGIAJ* y verificando que únicamente se devuelven DNIs por los que previamente se ha preguntado. Una vez todos los jugadores de la lista estén verificados la respuesta será siempre la misma (la lista completa).



4.3.6 Casos de Prueba de la Operación de bajaJugador

4.3.6.1 Baja un jugador dado de alta por el Operador

En este caso de Prueba el jugador está previamente dado de alta para el operador, ya que se verificó su estado en RGIAJ a través de la operación de *verificarJugador* o de *verificarRGIAJ*, obteniendo una respuesta sin errores, es decir COD001 o COD002. Por tanto al realizar la petición de baja, esta se realizará correctamente.

Request:

```
<bajaJugador>  
<dni>00000034B</dni>  
<causa>A</causa>  
</bajaJugador>
```

Response:

```
<resultadoBajaJugador>  
  <dni>00000034B</dni>  
  <resultadoBaja>  
    <codigo>COD010</codigo>  
    <descripcion> Baja realizada correctamente</descripcion>  
  </resultadoBaja>  
</resultadoBajaJugador>
```

4.3.6.2 Baja de un jugador no dado de alta previamente por el Operador

En este caso de Prueba el jugador no está previamente dado de alta para el operador, ya que no se verificó su estado en RGIAJ a través de la operación de *verificarJugador* o de *verificarRGIAJ* o si se verificó dio una respuesta de error de formato o de error de sistema. Por tanto no se realiza baja, informando de ello en el resultado de la petición.

Request:

```
<bajaJugador>  
  <dni>99999030K</dni>  
  <causa>R</causa>  
</bajaJugador>
```



Response:

```
<resultadoBajaJugador>  
  <dni>99999030K</dni>  
  <resultadoBaja>  
    <codigo>COD021</codigo>  
    <descripcion> Baja no realizada, el DNI no está activado para este  
Operador </descripcion>  
  </resultadoBaja>  
</resultadoBajaJugador>
```

4.3.6.3 Baja de un jugador ya dado de baja previamente por el Operador

En este caso de Prueba el jugador ya fue dado de baja previamente para el operador. Por tanto no se realiza baja, informando de ello en el resultado de la petición.

Request:

```
<bajaJugador>  
  <dni>00000034B</dni>  
  <causa>A</causa>  
</bajaJugador>
```

Response:

```
<resultadoBajaJugador>  
  <dni>00000034B</dni>  
  <resultadoBaja>  
    <codigo>COD022</codigo>  
    <descripcion> Baja no realizada, el DNI ya está dado de baja para este  
Operador</descripcion>  
  </resultadoBaja>  
</resultadoBajaJugador>
```



4.3.6.4 Causa de la baja errónea

En este caso de Prueba se envía una causa de baja no valida o vacia. Por tanto no se realiza baja, informando de ello en el resultado de la petición.

Request:

```
<bajaJugador>  
  <dni>X0000066C<</dni>  
  <causa>Z</causa>  
</bajaJugador>
```

O mandando el campo causa pero vacio.

```
<bajaJugador>  
  <dni>X0000066C<</dni>  
  <causa></causa>  
</bajaJugador>
```

Response:

```
<resultadoBajaJugador>  
  <dni>X0000066C<</dni>  
  <resultadoBaja>  
  <codigo>COD023</codigo>  
  <descripcion>Formato de datos incorrecto - Baja no realizada, causa de la baja  
  errónea</descripcion>  
</resultadoBaja>  
</resultadoBajaJugador>
```

4.3.6.5 Baja de un jugador enviando un DNI/NIE erróneo

En este caso de Prueba se solicita la baja de un jugador no identificado correctamente ya que se envía una letra errónea del *DNI/NIE* (el DNI/NIE correcto es 30000260A).

Por tanto no se realiza baja, informando de ello en el resultado de la petición.



Request:

```
<bajaJugador>  
  <dni>30000260B</dni>  
  <c ausa>A</causa>  
</bajaJugador>
```

Response:

```
<resultadoBajaJugador>  
  <dni>30000260B</dni>  
  <resultadoBaja>  
    <codigo>COD901</codigo>  
    <descripcion> Formato de datos incorrecto - DNI no válido </descripcion>  
  </resultadoBaja>  
</resultadoBajaJugador>
```

4.3.6.6 Baja de un jugador donde no se envía el DNI/NIE

En este caso de Prueba se solicita la baja de un jugador pero sin enviar el DNI/NIE que es obligatorio. Por tanto no se realiza baja, informando de ello por fault del servicio.

Request:

```
<bajaJugador>  
  <causa>A</causa>  
</bajaJugador>
```

Response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>  
    <soap:Fault>  
      <faultcode>soap:Client</faultcode>  
      <faultstring>Unmarshalling Error: cvc-complex-type.2.4.a: Invalid content was found starting  
with element 'causa'. One of '{dni}' is expected. </faultstring>  
    </soap:Fault>
```



```
</soap:Body>  
</soap:Envelope>
```

4.3.6.7 Baja de un jugador donde no se envía la causa

En este caso de Prueba se solicita la baja de un jugador pero sin enviar la causa que es obligatorio. Por tanto no se realiza baja, informando de ello por fault del servicio.

Request:

```
<bajaJugador>  
    <dni>X0000066C<</dni>  
</bajaJugador>
```

Response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
<soap:Body>  
<soap:Fault>  
<faultcode>soap:Client</faultcode>  
<faultstring>Unmarshalling Error: cvc-complex-type.2.4.b: The content of element  
'bajaJugador' is not complete. One of '{causa}' is expected. </faultstring>  
</soap:Fault>  
</soap:Body>  
</soap:Envelope>
```

4.3.6.8 Verificar cambios en RGIAJ después de una baja

En este caso de prueba, se da de baja de uno de los DNI's indicados en el caso de prueba de VerificarCambiosRGIAJ y que se haya previamente verificado a través de la operación de *verificarJugador* o *verificarRGIAJ*. A partir de la siguiente hora en punto, al llamar a la operación de *verificarCambiosRGIAJ* se comprobará que dicho DNI ya no aparece en la lista de respuesta.

Ejemplo:

Anteriormente se verificó tanto el DNI 00000002W y el DNI X0000003A, y al llamar a la operación de *verificarCambiosRGIAJ*, se comprobó que aparecían ambos en la lista

Response:



```
<cambioRGIAJ>
  <DNI>00000002W </DNI>
  <motivoCambio>A</motivoCambio>
  <fechaCambio>2013-05-01T14:00:00.00+01:00</fechaCambio>
</cambioRGIAJ>
<cambioRGIAJ>
  <DNI>X0000003A </DNI>
  <motivoCambio>B</motivoCambio>
  <fechaCambio>2013-05-01T15:00:00.00+01:00</fechaCambio>
</cambioRGIAJ>
```

En este caso de prueba se da de baja el DNI 00000002W a través de la operación de *bajaJugadores* y posteriormente se volverá a llamar a la *VerificarCambiosRGIAJ*, esperando a la siguiente hora en punto. Ya no aparece este DNI's en la lista de resultados.

Response:

```
<cambioRGIAJ>
  <DNI>X0000003A </DNI>
  <motivoCambio>B</motivoCambio>
  <fechaCambio>2013-05-01T15:00:00.00+01:00</fechaCambio>
</cambioRGIAJ>
```

4.4 Formulario de solicitud de alta en el servicio.

Los operadores de juego que deseen realizar pruebas con el servicio de verificación de jugadores, deberán solicitar el alta en el entorno de pruebas mediante el formulario publicado en la sección de "Documentos de trabajo", epígrafe "Servicios Web" del portal del Ministerio de Hacienda y Administraciones Públicas.

Las solicitudes se enviarán a la cuenta de correo dgoj.soporteoperadores@minhap.es indicando en el asunto del mensaje "ALTA WS-VerificaciónJugadores".

Una vez comunicada por la DGOJ el alta en el entorno de pruebas, los operadores utilizarán el juego de ensayo del apartado 4.2 para realizarlas.



La solicitud de alta en el entorno de producción se realizará mediante el mismo formulario y procedimiento, marcando el recuadro de “producción”. De igual forma la solicitud se enviara a la a la cuenta de correo dgoj.soproteoperadores@minhap.es indicando en el asunto del mensaje “ALTA WS-VerificaciónJugadores

La empresa solicitante deberá haber solicitado previamente al menos una licencia singular. El acceso a los datos reales estará condicionado a la concesión de la licencia.

Hasta entonces sólo se podrán realizar pruebas de integración, con los datos de prueba contenidos en el apartado 4.3, previa comunicación a la DGOJ a través de la dirección de dgoj.soproteoperadores@minhap.es indicando en el asunto del mensaje “TEST WS-VerificaciónJugadores”.



5 Test del servicio

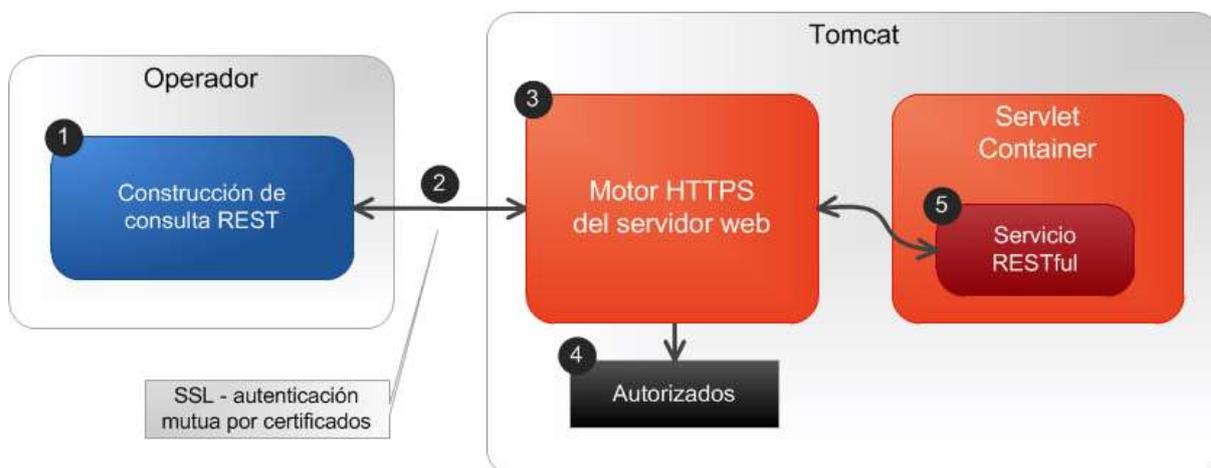
5.1 Objetivos

Proporcionar a los operadores del juego que actualmente estén utilizando el servicio de verificación de jugadores la posibilidad de comprobar el estado del servicio mediante un nuevo servicio de TEST independiente del servicio de verificación de jugadores.

5.2 Especificación funcional

El sistema ofrece un servicio de TEST para que los operadores puedan consultar el estado de las operaciones verificarJugador y verificarRGIAJ del servicio de verificación de jugadores. El estado del servicio se actualiza periódicamente, en intervalos regulares que fija la DGOJ. Este estado queda almacenado en una caché que es la que consultan los operadores a través de este sistema. En la actualidad el intervalo de tiempo definido para actualizar el estado del servicio es de 5 minutos.

5.3 Descripción del sistema



5.4 Conexión segura

- El operador se conectará mediante canal SSL. Todos los accesos serán auditados por parte de la DGOJ.



- Las peticiones deben utilizar el canal SSL (HTTPS) de forma que deben construirse con el certificado del operador para que el servidor Web establezca correctamente la comunicación; el servidor a su vez también se identifica con un certificado. Los operadores deberán utilizar el mismo certificado que usan para el servicio de verificación de jugadores.

5.5 Peticiones de consulta desde un explorador.

La consulta al servicio puede realizarse a través de una aplicación web que facilita la DGOJ o directamente mediante una conexión https realizando peticiones GET.

5.5.1 Aplicación web

<https://sede.ordenacionjuego.gob.es/testoperadores/test/html/>

5.5.2 Mediane peticiones GET con protocolo HTTPS

El operador debe construir una petición para el recurso; esta petición es simplemente una petición GET al recurso con protocolo HTTPS, es decir una petición similar a la siguiente:

https://sede.ordenacionjuego.gob.es/testoperadores/test?opcion=VERIFICAR_JUGADOR

https://sede.ordenacionjuego.gob.es/testoperadores/test?opcion=VERIFICAR_RGIAJ



5.6 Respuestas del servicio de TEST (TESTOPERADORES).

5.6.1 En formato WEB.



Pulsando en los enlaces Estado de VerificarJugador y Estado de Verificar RGIAJ se obtendría información sobre cada una de las dos operaciones principales del servicio web de verificación.

Los distintos estados en los que puede estar cada uno de los dos servicios anteriores:

- Correcto: el servicio se presta con normalidad
- Error: existe algún problema interno que impide prestar el servicio
- Inestable: existe algún problema interno que provoca que algunas peticiones sean atendidas con normalidad pero otras no

Si el certificado utilizado para identificarse es incorrecto, se obtendrá la siguiente pantalla.





5.6.2 Con peticiones GET con protocolo HTTPS

El servicio recibe la opción de búsqueda ([opcion=VERIFICAR_JUGADOR](#) o [opcion=VERIFICAR_RGIAJ](#)) construida por el operador, comprueba la identidad del operador y si el operador es válido realiza la búsqueda en la cache de test. Construye una respuesta en formato JSON. Las posibles respuestas del sistema son las siguientes:

```
{"opcion":"VERIFICAR_JUGADOR","estado":"Correcto","fecha-hora-control":"2013-10-29 16:37:25 CET"}
```

```
{"opcion":"VERIFICAR_JUGADOR","estado":"Inestable","fecha-hora-control":"2013-11-18 10:59:07 CET"}
```

```
{"opcion":"VERIFICAR_JUGADOR","estado":"Error","fecha-hora-control":"2013-11-18 10:59:07 CET"}
```

```
{"opcion":"VERIFICAR_RGIAJ","estado":"Correcto","fecha-hora-control":"2013-10-29 16:37:53 CET"}
```

```
{"opcion":"VERIFICAR_RGIAJ","estado":"Error","fecha-hora-control":"2013-11-18 10:59:07 CET"}
```

```
{"codigo":"ERR002","descripcion":"Certificado no válido"}
```

Los códigos de error en la respuesta.

Código de respuesta	Significado
{"Code":"COD007"}	Opción de consulta no valida
{"Code":"COD008"}	Opción no permitida
{"Code":"ERR001"}	Error interno del servicio
{"Code":"ERR002"}	Certificado no valido (no esta autorizado)
{"Code":"COD006"}	No se encuentra el certificado.



5.7 Ejemplo para utilización del servicio TESTOPERADORES desde un cliente Java.

El desarrollo del cliente para consultar el servicio testoperadores es responsabilidad de los operadores de juego.

No obstante, la DGOJ facilita un API, con sus dependencias, para que el operador desarrolle un cliente en tecnología java disponible en la página web de la Dirección General: <https://www.ordenacionjuego.es/>

La DGOJ no da soporte sobre dicho software ni asume ninguna responsabilidad sobre el funcionamiento del mismo. Así mismo no se proporcionará soporte sobre el contenido del mismo ni se tiene la responsabilidad de solucionar los problemas que se puedan detectar en su funcionamiento.

Este cliente está diseñado para ser integrado dentro de un desarrollo en Java.

El jar proporcionado (API) provee de funcionalidad para realizar la llamada SSL al servicio de testoperadores.

Debemos colocar el jar proporcionado en el CLASSPATH de la aplicación.

Para realizar la llamada al servicio testoperadores se realiza de la siguiente manera.

Se importan las clases TestOperadoresService y ConnectionParameters.

```
import es.dgoj.services.ConnectionParameters;  
import es.dgoj.services.TestOperadoresService;
```

Creamos una instancia de dichas clase e invocamos el método searchRestService con los parámetros adecuados.

```
ConnectionParameters conParams = new  
    ConnectionParameters(hostName, port, option,  
        keyStorePath, keyStorePass,  
        trustStorePath, trustStorePass);  
  
TestOperadoresService TestService = new TestOperadoresService();  
String response = TestService.searchRestService(conParams);
```

A continuación describimos los parámetros.

hostName: Dirección (ip o nombre DNS) del servidor donde está instalada la aplicación Web TESTOPERADORES.



port: Puerto SSL habilitado en el servidor 443.

opcion: Opcion VERIFICAR_JUGADOR o VERIFICAR_RGIAJ.

keyStorePath: Ruta del keyStore.

keyStorePass: Password del keyStore.

trustStorePath: Ruta del trustStore.

trustStorePass: Password del trustStore

5.8 Requisitos.

- Es necesario disponer de un certificado para el cliente dentro de un almacén de claves de tipo JKS. El certificado tiene que ser el mismo que el que se utiliza para acceder al servicio web de Verificación de Jugadores.
- Es necesario disponer de un almacén de certificados de confianza de tipo JKS, en el que guardaremos la parte pública del certificado del servidor de o el certificado de la CA que lo firmó.

5.9 Ejemplos.

Ejemplo llamada a un cliente que utiliza la API. ([dgojtest-api-1.0.jar](#))

```
java -jar testoperadores.jar sede.ordenacionjuego.gob.es 443 VERIFICAR_JUGADOR .\CNJ_keystore.jks clavekeystore .\CNJ_trustore.jks clavetrustore
```

Ejemplo codigo fuente de utilizacion de la API.

```
import es.dgoj.services.ConnectionParameters;
import es.dgoj.services.TestOperadoresService;

public class Cliente {

    public static void main(String[] args) throws Exception {
        String hostName = args[0];
        int port = Integer.valueOf(args[1]).intValue();
        String option = args[2];
        String keyStorePath = args[3];
        String keyStorePass = args[4];
        String trustStorePath = args[5];
        String trustStorePass = args[6];

        ConnectionParameters conParams = new ConnectionParameters(hostName, port, option,
            keyStorePath, keyStorePass, trustStorePath, trustStorePass);
        TestOperadoresService TestService = new TestOperadoresService();
        String response = TestService.searchRestService(conParams);
    }
}
```



```
        System.out.println("\n" + response + "\n");  
    }  
}
```

5.10 Dependencias

Las librerías a incorporar al proyecto son las siguientes

- log4j-1.2.16.jar
- commons-codec-1.6.jar
- commons-logging-1.1.1.jar
- httpclient-4.2.jar
- httpcore-4.2.jar

Estas librerías pueden ser obtenidas por el operador a través de la página web de la Dirección General de Ordenación del Juego a través de la siguiente dirección [aquí](#)