



MINISTERIO
DE DERECHOS SOCIALES, CONSUMO
Y AGENDA 2030

**Servicio Web de
Verificación de Jugadores**

SECRETARÍA GENERAL DE
CONSUMO Y JUEGO

DIRECCIÓN GENERAL DE
ORDENACIÓN DEL JUEGO

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

Versión 1.15.2

Septiembre de 2024



1	Objetivos	11
1.1	Control de cambios del documento	12
1.2	Calendario de implantaciones de nuevos endpoints	17
2	Especificación funcional	18
2.1	Servicio web y operaciones	18
2.2	Parámetros de entrada	25
2.2.1	VerificarJugador, VerificarRGIAJ y VerificarIdentidad	25
2.2.2	Parámetros VerificarCambiosRGIAJ	31
2.2.3	Parámetros VerificarCambiosDefunciones	31
2.2.4	Parámetros bajaJugador	31
2.2.5	Parámetros inicioRevisionPeriodica	32
2.2.6	Parámetros finRevisionPeriodica	33
2.2.7	Parámetros revisionJugador	33
2.2.8	Parámetros confirmarRevisionJugador	34
2.3	Respuestas del servicio	35
2.3.1	Respuesta para VerificarJugador	35
2.3.2	Respuesta para VerificarRGIAJ	35
2.3.3	Respuesta para VerificarIdentidad	36
2.3.4	Respuesta para VerificarCambiosRGIAJ	36
2.3.5	Respuesta para bajaJugador	37
2.3.6	Respuesta para verificarCambiosDefunciones	37
2.3.7	Respuesta para inicioRevisionPeriodica	38
2.3.8	Respuesta para finRevisionPeriodica	38
2.3.9	Respuesta para revisionJugador	39
2.3.10	Respuesta para confirmarRevisionJugador	40
2.4	Autenticación del Operador	40



3	Descripción de los entornos	46
3.1	Producción	46
3.2	Pruebas.....	47
4	Anexos	48
4.1	Códigos de resultado y error	48
4.2	Juegos de ensayo en el entorno de pruebas	52
4.2.1	Descripción de los casos de Prueba	52
4.2.2	Relación de casos de error genéricos.....	63
4.2.2.1	Datos Obligatorios no enviados	63
4.2.2.2	Autenticación del Operador.....	63
4.2.2.3	DNI o NIE repetidos.....	63
4.2.2.4	Validación de formato del DNI/NIE	64
4.2.2.5	Validación de letra del DNI/NIE	64
4.2.2.6	Validación de Apellido2 obligatorio si es un NIF	64
4.2.2.7	Validación de fecha de nacimiento posterior al 01/01/1900.....	64
4.2.2.8	Chequeo de caracteres no válidos	64
4.2.2.9	El servidor del RGIAJ no está disponible.....	64
4.2.2.10	El servidor del SVDI no está disponible	64
4.2.2.11	Validación de formato del Número de soporte.....	65
4.2.2.12	Validación de número de soporte con NIF	65
4.2.3	Casos de Prueba de la Operación de verificarJugador.....	65
4.2.3.1	Verificar un jugador No OK (Nombre incorrecto)-EN RGIAJ.....	65
4.2.3.2	Verificar un jugador No OK (Primer apellido incorrecto)-EN RGIAJ	66
4.2.3.3	Verificar un jugador No OK (Segundo apellido incorrecto)-EN RGIAJ.	66
4.2.3.4	Verificar un jugador No OK (Fecha nacimiento incorrecta)-EN RGIAJ	67
4.2.3.5	Verificar un jugador OK- EN RGIAJ.....	68



4.2.3.6	Verificar dos jugadores OK- NO RGIAJ.....	69
4.2.3.7	Verificar dos jugador, uno OK-NO RGIAJ y otro con ERROR FORMATO70	
4.2.3.8	Verificar dos jugadores OK- EN RGIAJ los dos	71
4.2.3.9	Verificar un jugador OK- NO RGIAJ	73
4.2.3.10	Verificar un jugador OK- NO RGIAJ (pero menor).....	73
4.2.3.11	Verificar un jugador con Nif Anulado- NO RGIAJ	75
4.2.3.12	Verificar un jugador con Nif Retenido- NO RGIAJ	75
4.2.3.13	Verificar un jugador al que le faltan datos obligatorios	76
4.2.3.14	Verificar un jugador OK- EN RGIAJ.....	77
4.2.3.15	Verificar un jugador sin enviar fecha de nacimiento	78
4.2.3.16	Verificar un jugador que NO OK – NO EN RGIAJ	79
4.2.3.17	Verificar el mismo DNI dos veces en la misma petición.....	79
4.2.3.18	Verificar un jugador con un NIE y número de soporte	81
4.2.3.19	Verificar un jugador con un NIE y número de soporte con formato erroneo	83
4.2.3.20	Verificar un jugador con un DNI enviando número de soporte	84
4.2.3.21	Verificar un jugador que se encuentra en el registro de defunciones ..	85
4.2.3.22	Verificar un jugador cuya identidad no es correcta y además es menor de edad.	86
4.2.4	Casos de Prueba de la Operación de verificarIdentidad.....	87
4.2.4.1	Verificar una Identidad OK.....	87
4.2.4.2	Verificar una identidad No OK (Nombre del titular incorrecto)	87
4.2.4.3	Verificar una identidad No OK (Primer apellido incorrecto)	88
4.2.4.4	Verificar una identidad No OK (Segundo apellido incorrecto).....	89



4.2.4.5	Verificar una identidad No OK (Fecha de nacimiento incorrecta)	89
4.2.4.6	Verificar una identidad con ERROR-FORMATO	90
4.2.4.7	Verificar una identidad que no está correctamente identificada.....	91
4.2.4.8	Verificar identidad de Nif Anulado.....	92
4.2.4.9	Verificar un jugador con Nif Retenido	92
4.2.4.10	Verificar un jugador con un NIE y número de soporte	93
4.2.4.11	Verificar un jugador con un NIE y número de soporte con formato erroneo	95
4.2.4.12	Verificar un jugador con un DNI enviando número de soporte	96
4.2.4.13	Verificar identidad de un jugador que se encuentra en el Registro Civil como fallecido.....	96
	Enviar la siguiente petición:	96
4.2.4.14	Verificar identidad de un jugador. La identidad no es correcta y además es menor de edad.....	97
4.2.5	Casos de Prueba de la Operación de verificarRGIAJ	99
4.2.5.1	Verificar en RGIAJ un jugador que está inscrito	99
4.2.5.2	Verificar en RGIAJ un jugador que no está inscrito	99
4.2.5.3	Verificar un jugador que está inscrito en RGIAJ pero sin enviar todos los datos	100
4.2.5.4	Verificar en RGIAJ un jugador que no está inscrito	101
4.2.5.5	Verificar un RGIAJ con ERROR-FORMATO	102
4.2.5.6	Verificar un jugador con un NIE y número de soporte con formato erróneo	103
4.2.5.7	Verificar un jugador con un DNI enviando número de soporte	104
4.2.6	Casos de Prueba de la Operación de verificarCambiosRGIAJ	105



4.2.6.1	Verificar cambios en RGIAJ sin haber verificado ningún jugador	106
4.2.6.2	Verificar cambios en RGIAJ al ir verificando jugadores	107
4.2.7	Casos de Prueba de la Operación de verificarCambiosDefunciones.....	108
4.2.7.1	Verificar cambios en defunciones sin verificar jugadores	109
4.2.7.2	Verificar cambios en defunciones después de verificar Jugadores del juego de pruebas.....	110
4.2.8	Casos de Prueba de la Operación de bajaJugador	112
4.2.8.1	Baja un jugador dado de alta por el Operador.....	112
4.2.8.2	Baja de un jugador no dado de alta previamente por el Operador	112
4.2.8.3	Baja de un jugador ya dado de baja previamente por el Operador....	113
4.2.8.4	Causa de la baja errónea.....	113
4.2.8.5	Baja de un jugador enviando un DNI/NIE erróneo.....	114
4.2.8.6	Baja de un jugador donde no se envía el DNI/NIE	115
4.2.8.7	Baja de un jugador donde no se envía la causa	115
4.2.8.8	Verificar cambios en RGIAJ después de una baja.....	116
4.2.9	Casos de Prueba de la Operación de inicioRevisionPeriodica	117
4.2.9.1	Inicio de una revisión periódica por el Operador.....	117
4.2.9.2	Inicio de una revisión periódica por el Operador - ERROR	118
4.2.10	Casos de Prueba de la Operación de finRevisionPeriodica.....	118
4.2.10.1	Fin de una revisión periódica por el Operador.....	118
4.2.10.2	Fin de una revisión periódica por el Operador - ERROR.....	119
4.2.11	Casos de Prueba de la Operación de revisionJugador.....	120
4.2.11.1	Revisión de un jugador por el Operador	120
4.2.11.2	Revisión de un jugador por el Operador - ERROR.....	121



4.2.12	Casos de Prueba de la Operación de confirmarRevisiónJugador.....	121
4.2.12.1	Confirmación de la revisión de un jugador por el Operador.....	121
4.2.12.2	Confirmación de la revisión de un jugador por el Operador - ERROR122	
4.3	Juegos de ensayo para verificar el entorno de producción	123
4.3.1	Relación de casos de error genéricos.....	123
4.3.1.1	Datos Obligatorios a null.....	123
4.3.1.2	Autenticación del Operador.....	123
4.3.1.3	Validación de formato del DNI/NIE	124
4.3.1.4	Validación de letra del DNI/NIE	124
4.3.1.5	Validación de Apellido2 obligatorio si es un NIF	124
4.3.1.6	Validación de fecha de nacimiento posterior al 01/01/1900.....	124
4.3.1.7	Validación de caracteres no permitidos	124
4.3.1.8	DNI/NIE repetidos.....	124
4.3.2	Casos de Prueba de la Operación de verificarJugador.....	125
4.3.2.1	Verificar un jugador No OK-NO RGIAJ.....	125
4.3.2.2	Verificar un jugador OK- EN RGIAJ.....	126
4.3.2.3	Verificar dos jugadores OK- NO RGIAJ.....	127
4.3.2.4	Verificar dos jugador, uno OK-NO RGIAJ y otro con ERROR FORMATO128	
4.3.2.5	Verificar dos jugadores OK- EN RGIAJ los dos.....	130
4.3.2.6	Verificar un jugador OK- NO RGIAJ	131
4.3.2.7	Verificar un jugador No OK- EN RGIAJ	132
4.3.2.8	Verificar un jugador OK- NO RGIAJ (pero menor).....	133
4.3.2.9	Verificar un jugador al que le faltan datos obligatorios	134
4.3.2.10	Verificar un jugador OK- EN RGIAJ.....	134



4.3.2.11	Verificar un jugador sin enviar fecha de nacimiento	136
4.3.2.12	Verificar un jugador que NO EXISTE el DNI.....	137
4.3.2.13	Verificar un jugador que da error de Servicio (COD005)	138
4.3.2.14	Verificar el mismo DNI dos veces en una misma petición	139
4.3.2.15	Verificar un jugador que se encuentra incluido en el Registro Civil como fallecido	140
4.3.2.16	Verificar un jugador cuya identidad no es correcta y además es menor de edad.	141
4.3.3	Casos de Prueba de la Operación de verificarIdentidad.....	142
4.3.3.1	Verificar una Identidad OK pero menor de edad.....	142
4.3.3.2	Verificar una Identidad OK.....	143
4.3.3.3	Verificar una identidad No OK (Nombre incorrecto).....	143
4.3.3.4	Verificar una identidad No OK (Primer apellido incorrecto)	144
4.3.3.5	Verificar una identidad No OK (Segundo apellido incorrecto).....	145
4.3.3.6	Verificar una identidad No OK (Fecha de nacimiento incorrecta)	146
4.3.3.7	Verificar una identidad No OK	146
4.3.3.8	Verificar una identidad con ERROR-FORMATO	147
4.3.3.9	Verificar una identidad con caracteres no permitidos	148
4.3.3.10	Verificar una identidad que NO EXISTE	149
4.3.3.11	Verificar una identidad que da error de Servicio (COD005).....	149
4.3.3.12	Verificar identidad de un jugador que se encuentra incluido en el Registro Civil como fallecido.....	151
4.3.3.13	Verificar identidad de un jugador. La identidad no es correcta y además es menor de edad.....	151
4.3.4	Casos de Prueba de la Operación de verificarRGIAJ	152



4.3.4.1	Verificar en RGIAJ un jugador que está inscrito	152
4.3.4.2	Verificar en RGIAJ un jugador que no está inscrito	153
4.3.4.3	Verificar un jugador que está inscrito en RGIAJ pero sin enviar todos los datos	153
4.3.4.4	Verificar en RGIAJ un jugador que no está inscrito	155
4.3.4.5	Verificar un RGIAJ con ERROR-FORMATO	155
4.3.5	Casos de Prueba de la Operación de verificarCambiosRGIAJ	156
4.3.5.1	Verificarcambios en RGIAJ sin verificar jugadores	158
4.3.5.2	Verificarcambios en RGIAJ al ir verificando jugadores	158
4.3.6	Casos de Prueba de la Operación de verificarCambiosDefunciones.....	160
4.3.7	Casos de Prueba de la Operación de bajaJugador	162
4.3.7.1	Baja un jugador dado de alta por el Operador	162
4.3.7.2	Baja de un jugador no dado de alta previamente por el Operador	162
4.3.7.3	Baja de un jugador ya dado de baja previamente por el Operador....	163
4.3.7.4	Causa de la baja errónea.....	164
4.3.7.5	Baja de un jugador enviando un DNI/NIE erróneo	164
4.3.7.6	Baja de un jugador donde no se envía el DNI/NIE	165
4.3.7.7	Baja de un jugador donde no se envía la causa	166
4.3.7.8	Verificar cambios en RGIAJ después de una baja.....	166
4.3.8	Casos de Prueba de la Operación de inicioRevisionPeriodica	167
4.3.8.1	Inicio de una revisión periódica por el Operador.....	167
4.3.8.2	Inicio de una revisión periódica por el Operador - ERROR	168
4.3.9	Casos de Prueba de la Operación de finRevisionPeriodica.....	169
4.3.9.1	Fin de una revisión periódica por el Operador	169



4.3.9.2	Fin de una revisión periódica por el Operador - ERROR.....	169
4.3.10	Casos de Prueba de la Operación de revisionJugador.....	170
4.3.10.1	Petición Revisiónjugador, correcta	170
4.3.10.2	Petición RevisiónJugador - ERROR	171
4.3.11	Casos de Prueba de la Operación de confirmarRevisionJugador.....	172
4.3.11.1	Confirmación de la revisión de un jugador por el Operador.....	172
4.3.11.2	Confirmación de la revisión de un jugador por el Operador - ERROR.....	172
4.4	Formulario de solicitud de alta en el servicio.	173
5	Test del servicio	174
5.1	Objetivos	174
5.2	Especificación funcional.....	174
5.3	Descripción del sistema	174
5.4	Conexión segura	174
5.5	Peticiones de consulta desde un explorador.....	175
5.5.1	Aplicación web	175
5.5.2	Mediane peticiones GET con protocolo HTTPS.....	175
5.6	Respuestas del servicio de TEST (TESTOPERADORES).	176
5.6.1	En formato WEB.	176
5.6.2	Con peticiones GET con protocolo HTTPS.....	177
5.7	Ejemplo para utilización del servicio TESTOPERADORES desde un cliente Java.	178
5.8	Requisitos.	179
5.9	Ejemplos.	179
5.10	Dependencias	180



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 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@ordenacionjuego.gob.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	Se incluye un nuevo tipo de error Fault en el caso de que haya NIF's repetidos en la misma petición.



		<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 (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 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 dígitos (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).</p>
1.6	DIC 2013	<p>Nueva funcionalidad para chequear el estado del servicio (apartado 5)</p>
1.7	SEP 2014	<p>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.</p>
1.8	ABRIL 2015	<p>Se actualizan las instrucciones para la validación de la firma de los mensajes de respuesta devueltos por el servidor.</p> <p>Se modifica el endpoint del servicio, donde antes era cnjuego.gob.es pasa a ser ordenacionjuego.gob.es.</p>



1.9	NOV 2016	Añadidos nuevos mensajes de motivo en respuesta del titular no identificado en la verificación de jugador: <i>nombre incorrecto, 1º apellido incorrecto, 2º apellido incorrecto y fecha de nacimiento incorrecta.</i>
1.10	AGO 2017	Modificados el nombre, primer y segundo apellido de los casos de prueba del DNI 65000010D por cambios en los datos de prueba del servicio de la Policía.
1.11	OCT 2018	<p>Se añaden nuevos códigos de respuesta de verificarJugador y verificarIdentidad para señalar jugadores incorrectamente identificados pero cuyo número de DNI/NIE corresponde a un menor de edad, así como a jugadores incluidos en el Registro Civil como fallecidos</p> <p>Se añade la nueva operación de verificarCambiosDefunciones.</p> <p>Se añaden nuevos casos de prueba para la operación de verificarCambiosDefunciones y para los nuevos códigos de verificarJugador y verificarIdentidad (jugador incorrectamente identificado pero cuyo número de DNI/NIE corresponde a un menor de edad y jugador incluido en el Registro Civil como fallecido).</p>
1.12	Abril 2020	Corrección de erratas o cambio en las respuestas de la policía en los casos de prueba de producción.
1.13	Febrero 2021	Cambio en los datos de fecha de nacimiento para obtener una respuesta de identificación correcta de la policía en los casos de prueba de preproducción para el DNI= 10000320N
1.14	Noviembre 2021	Cambio usuario de pruebas con DNI = Y0336380L para los casos de prueba en los que se espera que sea menor de edad. Nuevo usuario con DNI = 20000128H
1.15	Enero 2024	Implementación de nuevas operaciones de revisión periódica y revisión de jugadores.
1.15.1	Febrero 2024	Aclaración sobre los casos de prueba del apartado 4.3.10 Operación de revisionJugador
1.15.2	Septiembre 2024	Se modifica la operación verificarCambiosDefunciones para que no informe sobre los jugadores dados de baja por el operador mediante la operación bajaJugador.



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)
1.8 (2)	2.2	05/03/2015	31/01/2017	06/04/2015	06/04/2017
1.9 (3)	2.3	21/11/2016	02/01/2019	12/12/2016	16/03/2019
1.11 (4)	2.4	15/11/2018	29/03/2024	14/01/2019	05/04/2024
1.15 (5)	2.5	16/01/2024		05/02/2024	

- (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
- (3) Este cambio corresponde a los nuevos motivos de titular no identificado devueltos en el resultado de la respuesta SOAP del servicio.
- (4) Este cambio corresponde a la nueva operación de VerificarConsultaDefunciones y los nuevos códigos de retorno COD007 (defunciones) y COD104 (menores)
- (5) Nuevas operaciones inicioRevisionPeriodica, finRevisionPeriodica, revisionJugador y confirmarRevisionJugador



2 Especificación funcional

2.1 Servicio web y operaciones

A continuación, se define la especificación funcional del servicio:

VerificacionJugadores VerificacionJugadoresSOAP https://ws-pre.cnjuego...	VerificacionJugadores																																																																																																																								
<table border="1"> <thead> <tr> <th colspan="3">* verificarJugador</th> </tr> </thead> <tbody> <tr> <td>input</td> <td>VerificarJugadorRequest</td> <td>verificarJugador</td> </tr> <tr> <td>output</td> <td>VerificarJugadorResponse</td> <td>verificarJugadorResponse</td> </tr> <tr> <td>fault</td> <td>VerificarJugadorFault</td> <td>verificarJugadorFault</td> </tr> <tr> <th colspan="3">* verificarRGIAJ</th> </tr> <tr> <td>input</td> <td>VerificarRGIAJRequest</td> <td>verificarRGIAJ</td> </tr> <tr> <td>output</td> <td>VerificarRGIAJResponse</td> <td>verificarRGIAJResponse</td> </tr> <tr> <td>fault</td> <td>VerificarRGIAJFault</td> <td>verificarRGIAJFault</td> </tr> <tr> <th colspan="3">* verificarIdentidad</th> </tr> <tr> <td>input</td> <td>VerificarIdentidadRequest</td> <td>verificarIdentidad</td> </tr> <tr> <td>output</td> <td>VerificarIdentidadResponse</td> <td>verificarIdentidadResponse</td> </tr> <tr> <td>fault</td> <td>VerificarIdentidadFault</td> <td>verificarIdentidadFault</td> </tr> <tr> <th colspan="3">* verificarCambiosRGIAJ</th> </tr> <tr> <td>input</td> <td>VerificarCambiosRGIAJRequest</td> <td>verificarCambiosRGIAJ</td> </tr> <tr> <td>output</td> <td>VerificarCambiosRGIAJResponse</td> <td>verificarCambiosRGIAJResponse</td> </tr> <tr> <td>fault</td> <td>VerificarCambiosRGIAJFault</td> <td>verificarCambiosRGIAJFault</td> </tr> <tr> <th colspan="3">* verificarCambiosDefunciones</th> </tr> <tr> <td>input</td> <td>VerificarCambiosDefuncionesRequest</td> <td>verificarCambiosDefunciones</td> </tr> <tr> <td>output</td> <td>VerificarCambiosDefuncionesResponse</td> <td>verificarCambiosDefuncionesResponse</td> </tr> <tr> <td>fault</td> <td>VerificarCambiosDefuncionesFault</td> <td>verificarCambiosDefuncionesFault</td> </tr> <tr> <th colspan="3">* bajaJugador</th> </tr> <tr> <td>input</td> <td>bajaJugadorRequest</td> <td>bajaJugador</td> </tr> <tr> <td>output</td> <td>bajaJugadorResponse</td> <td>bajaJugadorResponse</td> </tr> <tr> <td>fault</td> <td>bajaJugadorFault</td> <td>bajaJugadorFault</td> </tr> <tr> <th colspan="3">* inicioRevisionPeriodica</th> </tr> <tr> <td>input</td> <td>inicioRevisionPeriodicaRequest</td> <td>inicioRevisionPeriodica</td> </tr> <tr> <td>output</td> <td>inicioRevisionPeriodicaResponse</td> <td>inicioRevisionPeriodicaResponse</td> </tr> <tr> <td>fault</td> <td>inicioRevisionPeriodicaFault</td> <td>inicioRevisionPeriodicaFault</td> </tr> <tr> <th colspan="3">* finRevisionPeriodica</th> </tr> <tr> <td>input</td> <td>finRevisionPeriodicaRequest</td> <td>finRevisionPeriodica</td> </tr> <tr> <td>output</td> <td>finRevisionPeriodicaResponse</td> <td>finRevisionPeriodicaResponse</td> </tr> <tr> <td>fault</td> <td>finRevisionPeriodicaFault</td> <td>finRevisionPeriodicaFault</td> </tr> <tr> <th colspan="3">* revisionJugador</th> </tr> <tr> <td>input</td> <td>revisionJugadorRequest</td> <td>revisionJugador</td> </tr> <tr> <td>output</td> <td>revisionJugadorResponse</td> <td>revisionJugadorResponse</td> </tr> <tr> <td>fault</td> <td>revisionJugadorFault</td> <td>revisionJugadorFault</td> </tr> <tr> <th colspan="3">* confirmarRevisionJugador</th> </tr> <tr> <td>input</td> <td>confirmarRevisionJugadorRequest</td> <td>confirmarRevisionJugador</td> </tr> <tr> <td>output</td> <td>confirmarRevisionJugadorResponse</td> <td>confirmarRevisionJugadorResponse</td> </tr> <tr> <td>fault</td> <td>confirmarRevisionJugadorFault</td> <td>confirmarRevisionJugadorFault</td> </tr> </tbody> </table>		* verificarJugador			input	VerificarJugadorRequest	verificarJugador	output	VerificarJugadorResponse	verificarJugadorResponse	fault	VerificarJugadorFault	verificarJugadorFault	* verificarRGIAJ			input	VerificarRGIAJRequest	verificarRGIAJ	output	VerificarRGIAJResponse	verificarRGIAJResponse	fault	VerificarRGIAJFault	verificarRGIAJFault	* verificarIdentidad			input	VerificarIdentidadRequest	verificarIdentidad	output	VerificarIdentidadResponse	verificarIdentidadResponse	fault	VerificarIdentidadFault	verificarIdentidadFault	* verificarCambiosRGIAJ			input	VerificarCambiosRGIAJRequest	verificarCambiosRGIAJ	output	VerificarCambiosRGIAJResponse	verificarCambiosRGIAJResponse	fault	VerificarCambiosRGIAJFault	verificarCambiosRGIAJFault	* verificarCambiosDefunciones			input	VerificarCambiosDefuncionesRequest	verificarCambiosDefunciones	output	VerificarCambiosDefuncionesResponse	verificarCambiosDefuncionesResponse	fault	VerificarCambiosDefuncionesFault	verificarCambiosDefuncionesFault	* bajaJugador			input	bajaJugadorRequest	bajaJugador	output	bajaJugadorResponse	bajaJugadorResponse	fault	bajaJugadorFault	bajaJugadorFault	* inicioRevisionPeriodica			input	inicioRevisionPeriodicaRequest	inicioRevisionPeriodica	output	inicioRevisionPeriodicaResponse	inicioRevisionPeriodicaResponse	fault	inicioRevisionPeriodicaFault	inicioRevisionPeriodicaFault	* finRevisionPeriodica			input	finRevisionPeriodicaRequest	finRevisionPeriodica	output	finRevisionPeriodicaResponse	finRevisionPeriodicaResponse	fault	finRevisionPeriodicaFault	finRevisionPeriodicaFault	* revisionJugador			input	revisionJugadorRequest	revisionJugador	output	revisionJugadorResponse	revisionJugadorResponse	fault	revisionJugadorFault	revisionJugadorFault	* confirmarRevisionJugador			input	confirmarRevisionJugadorRequest	confirmarRevisionJugador	output	confirmarRevisionJugadorResponse	confirmarRevisionJugadorResponse	fault	confirmarRevisionJugadorFault	confirmarRevisionJugadorFault
* verificarJugador																																																																																																																									
input	VerificarJugadorRequest	verificarJugador																																																																																																																							
output	VerificarJugadorResponse	verificarJugadorResponse																																																																																																																							
fault	VerificarJugadorFault	verificarJugadorFault																																																																																																																							
* verificarRGIAJ																																																																																																																									
input	VerificarRGIAJRequest	verificarRGIAJ																																																																																																																							
output	VerificarRGIAJResponse	verificarRGIAJResponse																																																																																																																							
fault	VerificarRGIAJFault	verificarRGIAJFault																																																																																																																							
* verificarIdentidad																																																																																																																									
input	VerificarIdentidadRequest	verificarIdentidad																																																																																																																							
output	VerificarIdentidadResponse	verificarIdentidadResponse																																																																																																																							
fault	VerificarIdentidadFault	verificarIdentidadFault																																																																																																																							
* verificarCambiosRGIAJ																																																																																																																									
input	VerificarCambiosRGIAJRequest	verificarCambiosRGIAJ																																																																																																																							
output	VerificarCambiosRGIAJResponse	verificarCambiosRGIAJResponse																																																																																																																							
fault	VerificarCambiosRGIAJFault	verificarCambiosRGIAJFault																																																																																																																							
* verificarCambiosDefunciones																																																																																																																									
input	VerificarCambiosDefuncionesRequest	verificarCambiosDefunciones																																																																																																																							
output	VerificarCambiosDefuncionesResponse	verificarCambiosDefuncionesResponse																																																																																																																							
fault	VerificarCambiosDefuncionesFault	verificarCambiosDefuncionesFault																																																																																																																							
* bajaJugador																																																																																																																									
input	bajaJugadorRequest	bajaJugador																																																																																																																							
output	bajaJugadorResponse	bajaJugadorResponse																																																																																																																							
fault	bajaJugadorFault	bajaJugadorFault																																																																																																																							
* inicioRevisionPeriodica																																																																																																																									
input	inicioRevisionPeriodicaRequest	inicioRevisionPeriodica																																																																																																																							
output	inicioRevisionPeriodicaResponse	inicioRevisionPeriodicaResponse																																																																																																																							
fault	inicioRevisionPeriodicaFault	inicioRevisionPeriodicaFault																																																																																																																							
* finRevisionPeriodica																																																																																																																									
input	finRevisionPeriodicaRequest	finRevisionPeriodica																																																																																																																							
output	finRevisionPeriodicaResponse	finRevisionPeriodicaResponse																																																																																																																							
fault	finRevisionPeriodicaFault	finRevisionPeriodicaFault																																																																																																																							
* revisionJugador																																																																																																																									
input	revisionJugadorRequest	revisionJugador																																																																																																																							
output	revisionJugadorResponse	revisionJugadorResponse																																																																																																																							
fault	revisionJugadorFault	revisionJugadorFault																																																																																																																							
* confirmarRevisionJugador																																																																																																																									
input	confirmarRevisionJugadorRequest	confirmarRevisionJugador																																																																																																																							
output	confirmarRevisionJugadorResponse	confirmarRevisionJugadorResponse																																																																																																																							
fault	confirmarRevisionJugadorFault	confirmarRevisionJugadorFault																																																																																																																							

El servicio web de verificación de jugadores dispondrá de diez 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 está 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. "VerificarCambiosDefunciones": el operador utilizará esta operación para recuperar los jugadores cuya identidad verificó en su día de forma correcta (mediante las operaciones "VerificarIdentidad" o "VerificarJugador" obteniendo un resultado exitoso en la verificación de la identidad del Jugador), que aún no hayan sido dados de baja por el operador y que posteriormente a esta verificación, el documento de



identificación correspondiente ha sido incluido en el Registro Civil dependiente del Ministerio de Justicia como fallecido.

Es suficiente con que el operador consulte la operación “VerificarCambiosDefunciones” una vez al día para obtener la relación de nuevos jugadores encontrados en el registro de defunciones.

6. “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.

En el proceso de baja de un jugador se comprueba además si éste ha sido previamente registrado como fallecido en el registro de jugadores fallecidos. En el caso que ya estuviera registrado como fallecido, se actualizaría el registro de jugadores fallecidos para marcar el estado del jugador como de baja por el operador y que de esta forma no se muestre el mismo en el proceso “verificarCambioDefunciones”.

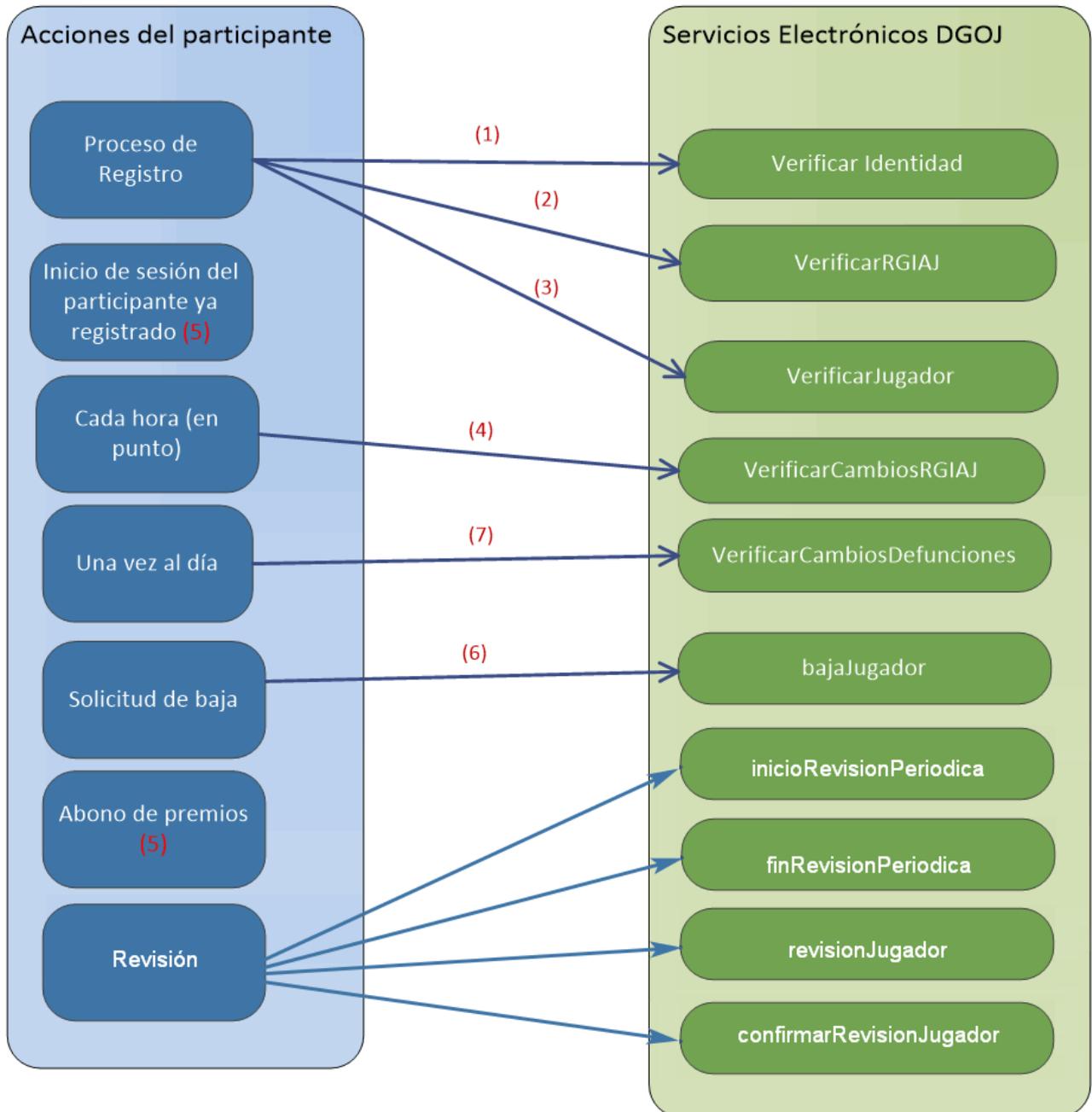
7. “InicioRevisionPeriodica”: Operación que proporciona al operador una ventana temporal y una franja horaria para llevar a cabo las revisiones periódicas de los datos de identidad y de RGIAJ que pueden hacer sobre los jugadores. El operador facilita una fecha de inicio deseada y la operación devuelve la fecha de inicio, fecha fin y la franja horaria en la que realizar la revisión periódica.
8. “FinRevisionPeriodica”: Operación que permite al Operador confirmar que ha realizado correctamente la revisión periódica de los datos de identidad y de RGIAJ sobre los jugadores. El operador confirma la fecha de inicio, fecha fin y la franja horaria en la que ha realizado la revisión periódica.
9. “RevisionJugador”: Operación que permite a los operadores obtener un listado de jugadores que tienen operaciones pendientes de realizar. Dichas operaciones se identificarán por un código. Del mismo modo, la operación facilitará en código de



confirmación que debe utilizarse como parámetro de entrada de la operación "ConfirmarRevisiónJugador".

10. "ConfirmarRevisiónJugador": Operación que permite confirmar que los operadores han llevado a cabo la revisión de las operaciones sobre los jugadores. El código de confirmación se proporciona en la respuesta de la operación "RevisiónJugador".

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 el 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 el 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 deberá 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.

(7) Los operadores deberán utilizar esta operación una vez al día para verificar qué jugadores, cuya identidad comprobaron correctamente han pasado a estar incluidos como fallecidos en el Registro Civil. Los jugadores que fueron dados de baja por los operadores no se muestran en este listado.

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.- VerificarCambiosDefunciones: 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.

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

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

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

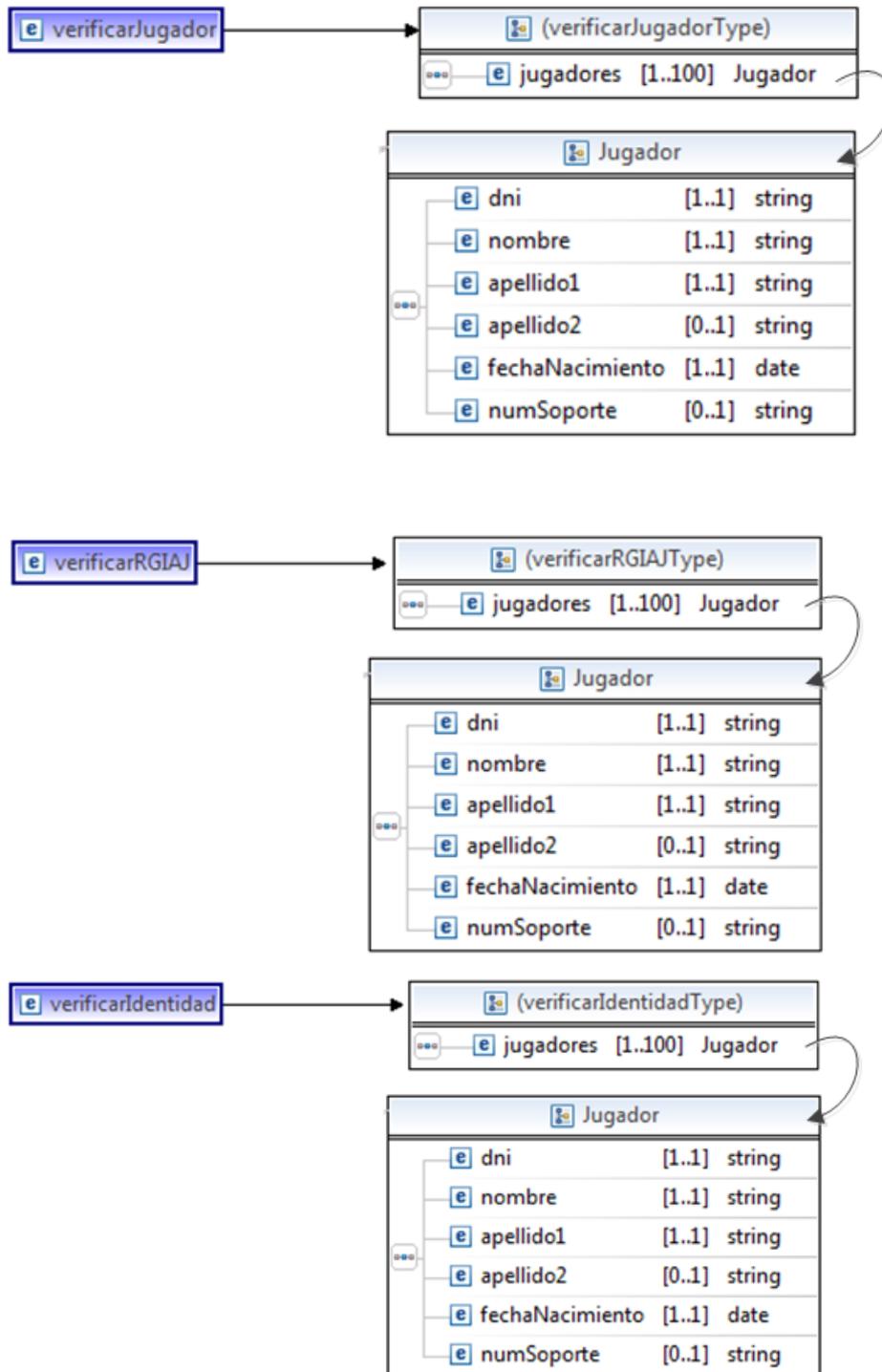
8.- RevisionJugador: en el caso de no estar disponible el servicio se reintentará en un momento posterior.

9.- ConfirmarRevisionJugador: 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 precedidos 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). Para el caso de NIE este campo no será obligatorio.
 - 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”.

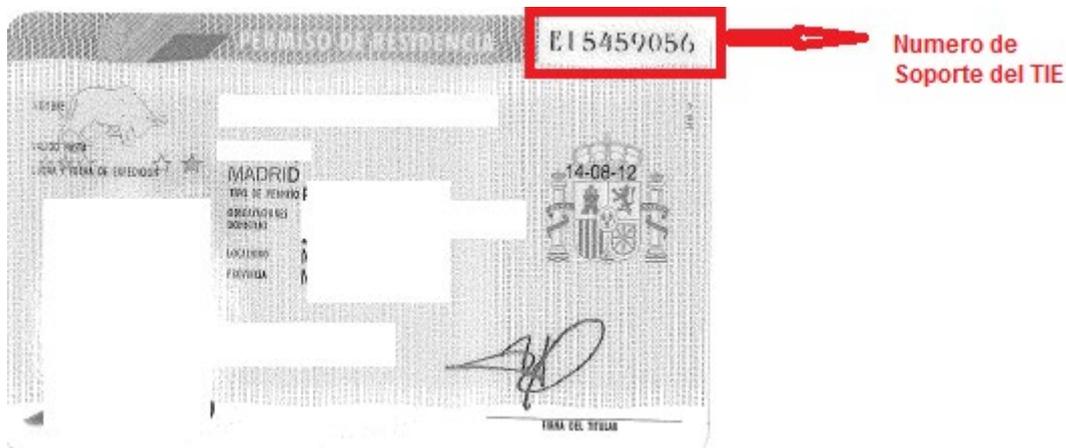


Solo se admitirá valor en este campo si el identificador es un NIE; en caso de NIF se devolverá error de formato (COD906: Error de formato-Con NIF no se puede enviar número de Soporte).

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



The image shows a Spanish Union Citizen Registration Certificate (Certificado de Registro de Ciudadano de la Unión) issued by the Local Police Station of La Línea de la Concepción. The document is light green and features the Spanish coat of arms at the top center. A red box highlights the support number '2161074' in the top right corner, with a red arrow pointing to it from the text 'Numero de Soporte del Certificado Europeo'. The certificate includes a warning: 'AVISO: DOCUMENTO NO VÁLIDO PARA ACREDITAR LA IDENTIDAD NI LA NACIONALIDAD DEL PORTADOR'. It certifies that the holder is registered in the Central Register of Foreigners. The certificate is signed by the Chief of the Local Police Station of La Línea de la Concepción. The bottom of the document also contains the same warning: 'AVISO: DOCUMENTO NO VÁLIDO PARA ACREDITAR LA IDENTIDAD'.

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
áàâãÄÅÄ	a
éèêÉÈËË	e
íîïÏ	i
óòôÓÒÕ	o
úùÚÙ	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.

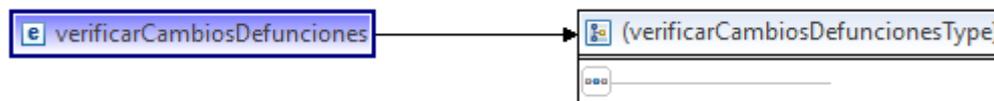
2.2.2 *Parámetros VerificarCambiosRGIAJ*

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



2.2.3 *Parámetros VerificarCambiosDefunciones*

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



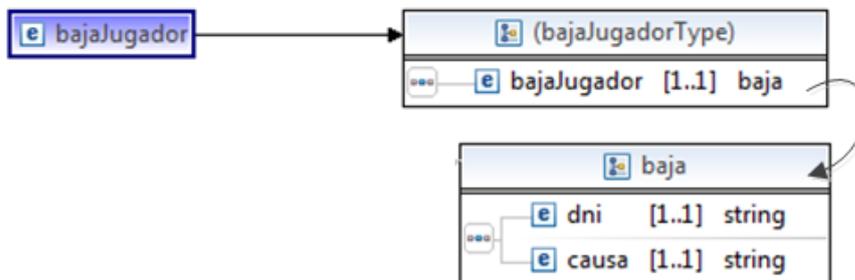
2.2.4 *Parámetros bajaJugador*

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.



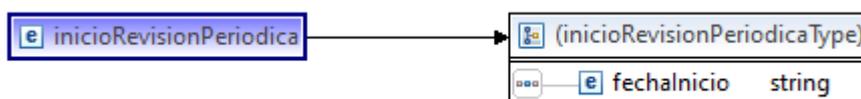
- Letra (X, Y, Z) seguida de número de 7 cifras y otra letra (dígito de control), haciendo un total de 9 caracteres (NIE).
Para el caso de los NIE es necesario aclarar una excepción debido a una actuación histórica de la policía.
Antes de decidir que el formato del NIE tuviera 9 caracteres precedidos 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.



2.2.5 Parámetros inicioRevisionPeriodica

Esta operación tiene como parámetros de entrada una ocurrencia del tipo String que indicará la fecha de inicio de la revisión propuesta por el operador:

- Fecha inicio (fechalnicio): String de 10 caracteres. Formato: “dd/mm/aaaa”

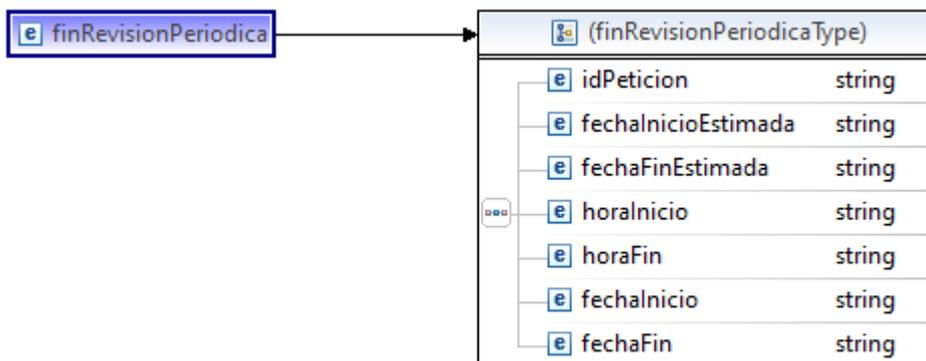




2.2.6 *Parámetros finRevisionPeriodica*

Esta operación tiene como parámetros de entrada unas ocurrencias del tipo String que indican una fecha de inicio estimada (facilitada por la operación de inicio), una fecha de fin estimada (facilitada por la operación de inicio), una hora de inicio, una hora de fin, una fecha de inicio y una fecha de fin. Además, será necesario que el operador proporcione el id de petición devuelto por la operación de inicio:

- Id petición (idPetición): String formado por el identificador de la operación “inicioRevisionPeriodica”.
- Fecha inicio estimada (fechaInicioEstimada): String de 10 caracteres. Proporcionado en la respuesta de la operación “inicioRevisionPeriodica”. Formato: “dd/mm/aaaa”
- Fecha fin estimada (fechaFinEstimada): String de 10 caracteres. Proporcionado en la respuesta de la operación “inicioRevisionPeriodica”. Formato: “dd/mm/aaaa”
- Hora inicio (horaInicio): String de 5 caracteres. Formato: “hh:mm”
- Hora fin (horaFin): String de 5 caracteres. Formato: “hh:mm”
- Fecha inicio (fechaInicio): String de 10 caracteres. Formato: “dd/mm/aaaa”
- Fecha fin (fechaFin): String de 10 caracteres. Formato: “dd/mm/aaaa”



2.2.7 *Parámetros revisionJugador*

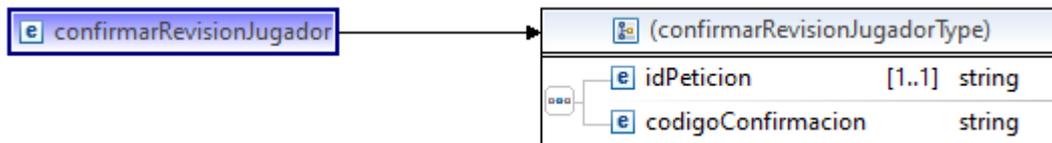
Esta operación no requiere parámetros de entrada.



2.2.8 *Parámetros confirmarRevisionJugador*

Esta operación tiene como parámetros de entrada una ocurrencia del tipo String que indica el código por el que se realizara la confirmación del jugador. Además, será necesario que el operador proporcione el id de petición devuelto por la operación de “revisionJugador”:

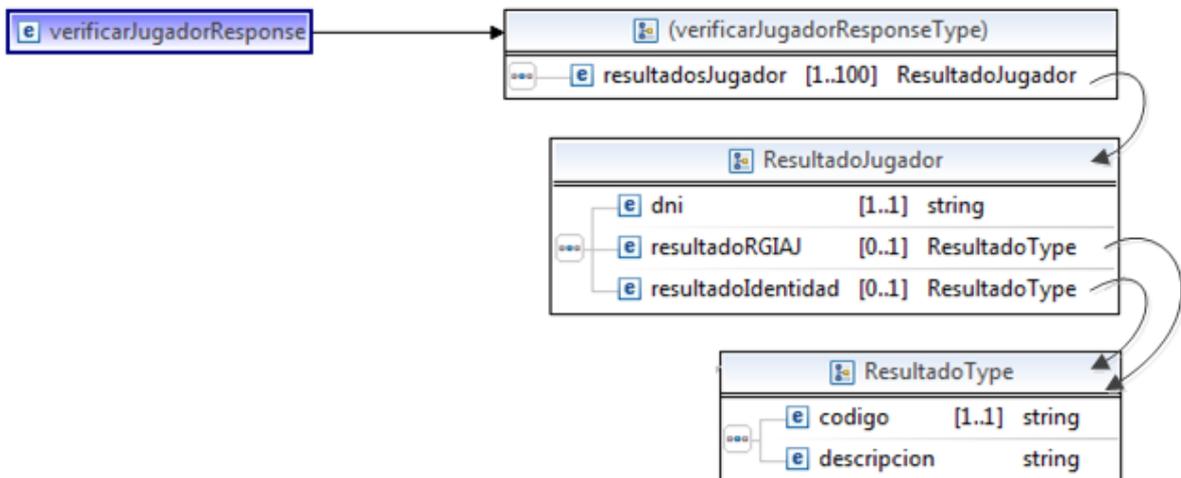
- Id petición (idPetición): String formado por el identificador de la operación “revisionJugador”.
- Código confirmación (codigoConfirmacion): String formado por el código de confirmación facilitado en la operación de “revisionJugador”.





2.3 Respuestas del servicio

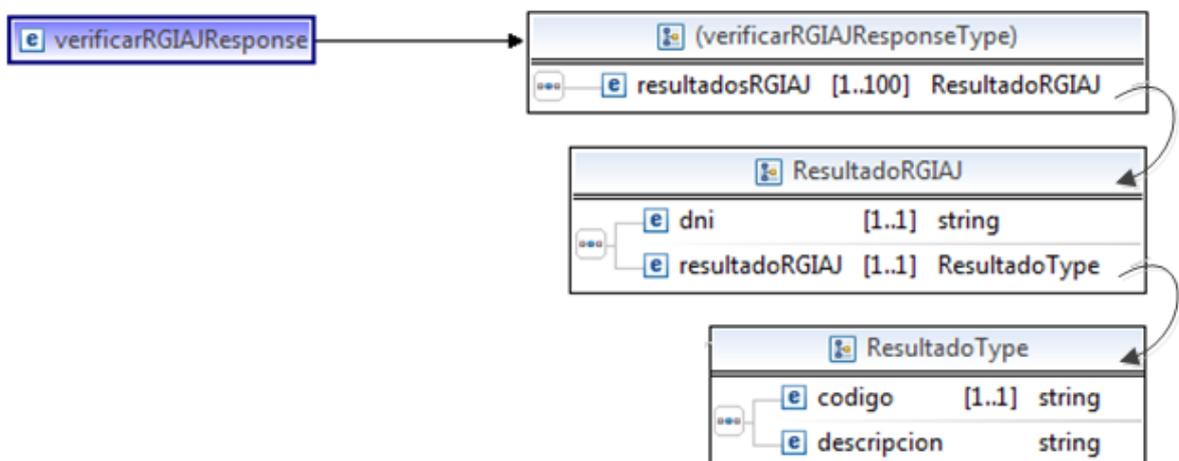
2.3.1 Respuesta para VerificarJugador



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

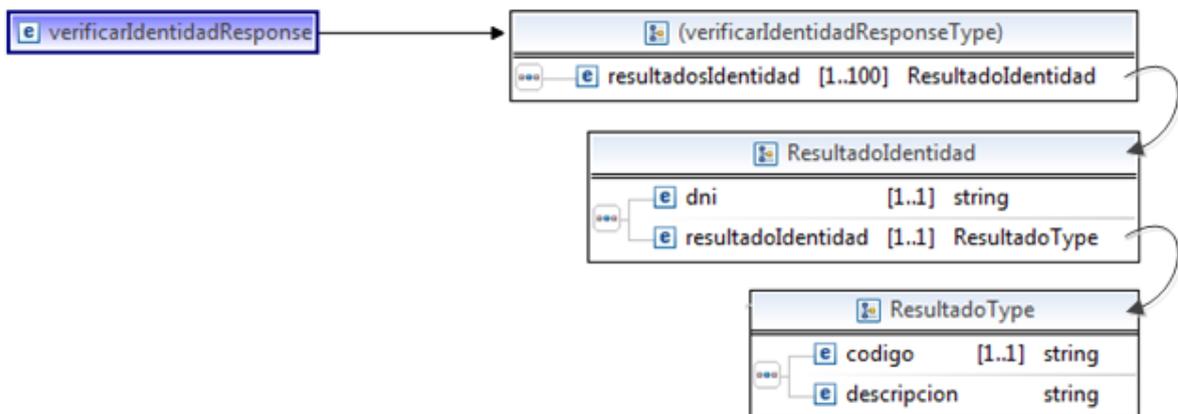




Se devolverán tantas repeticiones ResultadoRGIAJ como repeticiones de tipo Jugador se han enviado en la petición. El tipo Resultado contiene los siguientes datos:

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

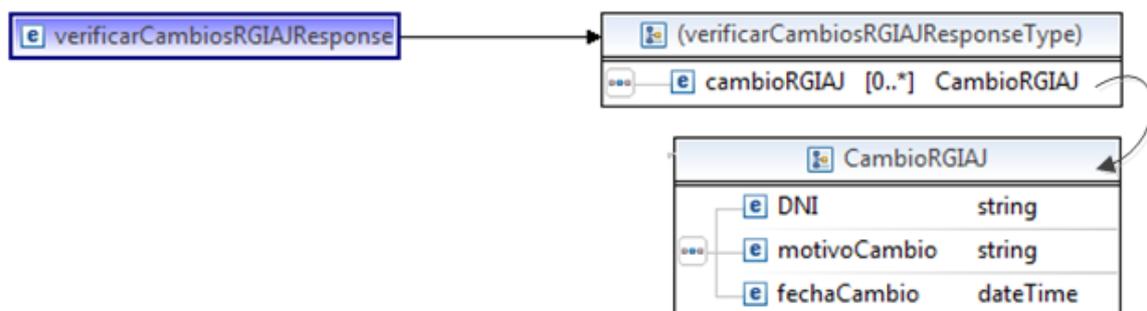
2.3.3 Respuesta para VerificarIdentidad



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

2.3.4 Respuesta para VerificarCambiosRGIAJ



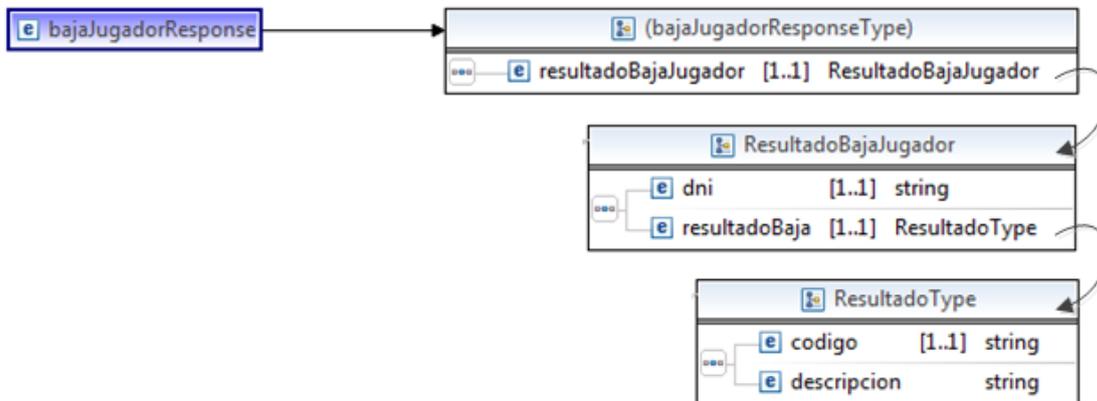


La operación devuelve una lista de DNIs/NIEs, motivo del cambio (“Alta” o “Baja”) y fecha del cambio, que se corresponde con los cambios de inscripción en el RGIAJ de jugadores cuya inscripción hubiera consultado previamente el operador mediante el servicio VerificarRGIAJ.

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:

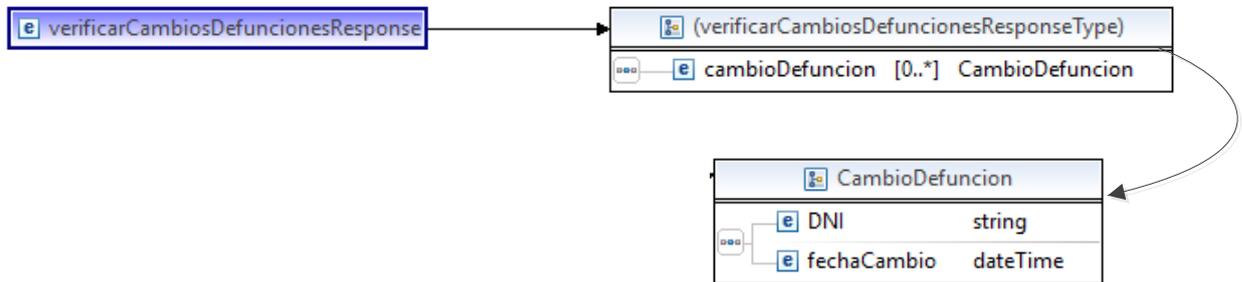
1. DNI/NIE, correspondiente al DNI/NIE enviado en la petición.
2. Resultado de la operación de baja (código y descripción).

2.3.6 Respuesta para verificarCambiosDefunciones

La operación devuelve una lista de DNIs/NIEs y fecha del cambio, que se corresponde con los jugadores cuya identidad el operador hubiera verificado correctamente mediante el servicio VerificarIdentidad ó VerificarJugador, que sigan activos para los operadores, es decir, que no hayan sido dados de baja por los mismos y que se encuentran identificados como fallecidos en el Registro Civil dependiente del Ministerio de Justicia.

El tipo CambioDefuncion contiene los siguientes datos:

1. DNI/NIE.
2. fechaCambio, fecha y hora en que la DGOJ detectó que el jugador con este DNI/NIE se encuentra identificado como fallecido en el Registro Civil.

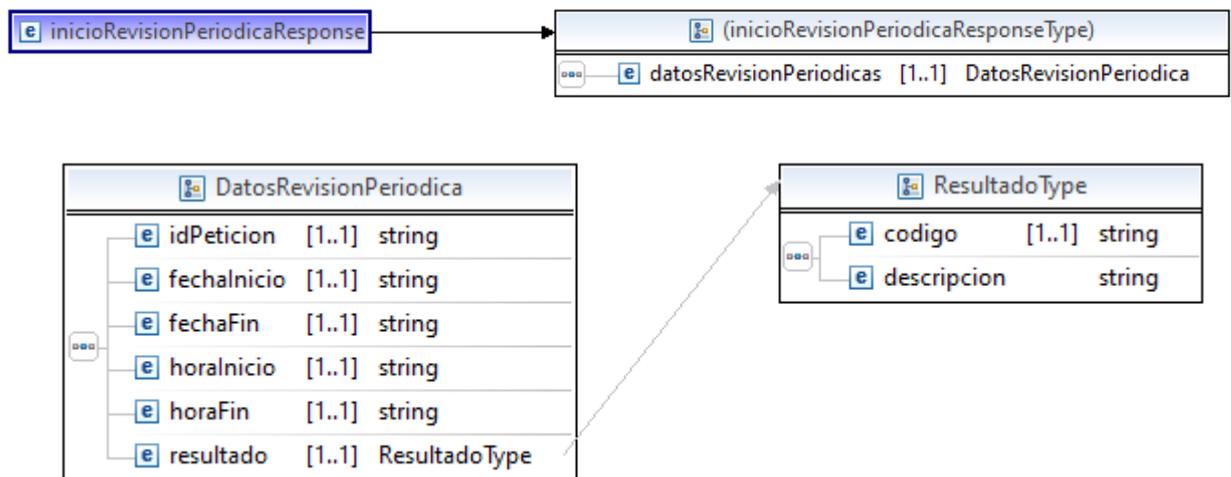


2.3.7 Respuesta para inicioRevisionPeriodica

Esta operación devuelve una lista de fechas junto las horas que empiezas y terminan cada revisión.

El tipo DatosRevisionPeriodica contiene los siguientes datos:

1. fechaInicio, fecha en la que se inicia la revisión.
2. fechaFin, fecha en la que se finaliza la revisión.
3. horaInicio, hora en la que empieza la revisión en la fecha de inicio.
4. horaFin, hora en la que finaliza la revisión en la fecha de fin.
5. idPetición, identificador de la petición.
6. Resultado de la operación de baja (código y descripción).



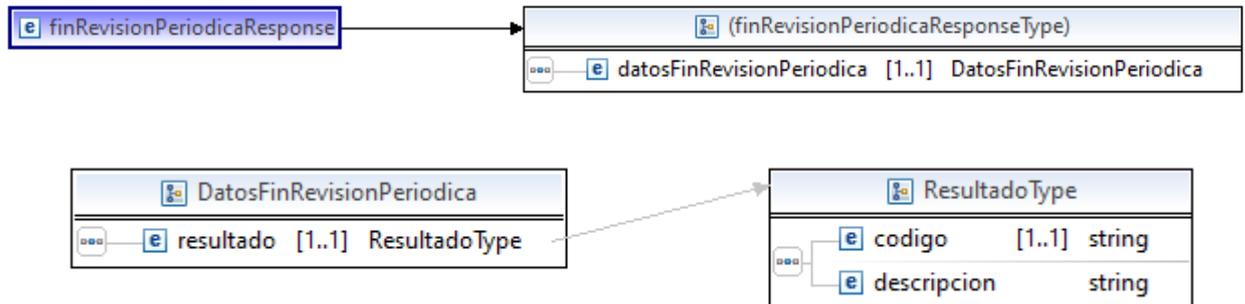
2.3.8 Respuesta para finRevisionPeriodica

Esta operación devuelve el resultado de confirmar la operación de revisión periódica por parte del operador, mediante la composición de un mensaje de operación realizada correctamente o de error.



El tipo DatosFinRevisionPeriodica contiene los siguientes datos:

1. Resultado de la operación (código y descripción).



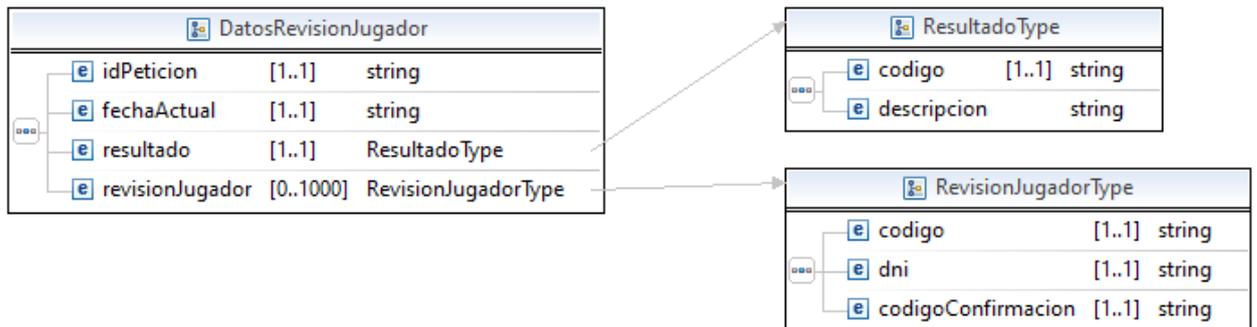
2.3.9 Respuesta para revisionJugador

Esta operación devuelve todos los datos registrados de las revisiones de los jugadores, que hay en el sistema, junto a la fecha en la que se ha realizado la operación y el resultado de esta.

El tipo DatosRevisionJugador contiene los siguientes datos:

1. fechaActual, La fecha actual en la que se ha realizado la operación.
2. idPeticion, El identificador de la petición.
3. Resultado de la operación (código y descripción).
4. RevisionJugador, Lista de las revisiones de los jugadores:
 - a. codigo, Código único, de hasta 12 caracteres, que indica la operación que deberá realizar el operador para revisar el jugador indicado.
 - b. dni, Identificador (DNI o NIE) del jugador
 - c. codigoConfirmacion, Código único asociado a la operación de solicitud de revisión.



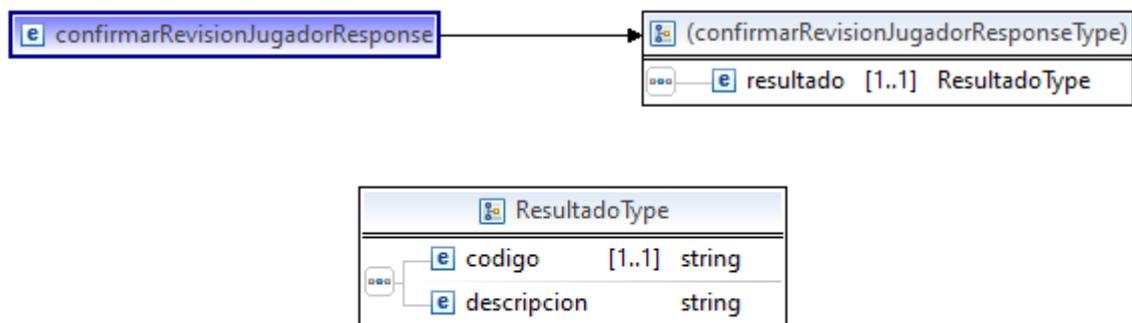


Actualmente están definidos dos códigos para revisión de los jugadores en el servicio:

- IDENT-CHK: El operador deberá volver a invocar la operación VerificarIdentidad sobre el jugador indicado en el campo DNI
- RGIAJ-CHK: El operador deberá volver a invocar la operación VerificarRGIAJ sobre el jugador indicado en el campo DNI

2.3.10 Respuesta para confirmarRevisiónJugador

Esta operación devuelve el resultado de confirmar la operación de revisión del jugador, para un código de confirmación determinado, actualizando la fecha en la que se ha realizado la confirmación.



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-2.5/VerificacionJugadores>
- WSDL: <https://ws-pre.ordenacionjuego.gob.es/VerificacionJugadores-2.5/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:wssu="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" Value="MI I I 9zCCB9+gAwIBAgI JALsAAAAAAAECA0GCSqGS Ib3DQE B B Q U A M I H g M Q s w C Q Y D V Q Q G
E w J F U z E u
wssu:Id="XWSSGID-1317050485906-773090739">MI I I 9zCCB9+gAwIBAgI JALsAAAAAAAECA0GCSqGS Ib3DQE B B Q U A M I H g M Q s w C Q Y D V Q Q G
E w J F U z E u
MCwGCSqGS Ib3DQE J A R Y f Y W N f Y 2 F t Z X J m a X J t Y V 9 j Y 0 B j Y W 1 l c m Z p c m 1 h L m N v b T F D M E E G A 1 U E B x M 6
.....
... .
```



```
...
+Ubmq8yaWe9F/4QyOOSmO6UgDyge6fSvZaw1U1QSlzt2apSEhwsUOIQKBJNq7HqJ8SVnp9Ierlty
LdvOZfDckdDwjNsXneiH6lYEHA==</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>
(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/Qo1jmg291g5Lk=</ds:SignatureValue>
<ds:KeyInfo>
<wsse:SecurityTokenReference xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
wsu: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: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>
</wsse:Security>
</S:Header>
```



```
(2)<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="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@ordenacionjuego.gob.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/xmldsig#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 (1.11) es la 2.5, por lo que las url's serán:

Endpoint: <https://ws.ordenacionjuego.gob.es/VerificacionJugadores-2.5/VerificacionJugadores>

WSDL: <https://ws.ordenacionjuego.gob.es/VerificacionJugadores-2.5/VerificacionJugadores?WSDL>

Actualmente la única versión en producción del servicio es la 2.5 que coexistirá con la versión 2.4 durante un tiempo (ver cuadro [Calendario de implantaciones de nuevos endpoints](#))

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 los 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 tiene la versión 2.5

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@ordenacionjuego.gob.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"><i>Titular no identificado</i><i>Titular no identificado positivamente - Ciudadano Nacionalizado Español</i><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><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"><i>DNI del titular anulado</i><i>Nombre del titular incorrecto.</i><i>Primer apellido del titular incorrecto.</i><i>Segundo apellido del titular incorrecto.</i><i>Fecha de nacimiento del titular incorrecta.</i>
COD007	El usuario fué identificado en el Registro Civil como fallecido en fechas anteriores a recibir la petición.



COD104	La identidad no es correcta y además el jugador es menor de edad
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
<i>Operaciones de revisión de jugadores</i>	
COD908	El formato de la fecha de inicio es inválido.
COD909	Fecha de inicio inválida.
COD910	La fecha de inicio es anterior a la fecha actual.
COD911	Los parámetros de entrada son inválidos.
COD912	El código de confirmación no existe.
COD913	No hay jugadores para verificar
COD914	Id de petición inválido
COD915	El código de confirmación no pertenece al Id de petición



COD916	Operación ya confirmada previamente
COD917	Los parámetros de entrada (X) no existen en la petición de inicioRevisionPeriodica con id Y
COD918	La fecha de inicio (X) es posterior a la fecha fin (Y).
COD919	La hora de inicio (X) es posterior a la hora fin (Y).
COD000	Operación realizada correctamente.



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, COD104, 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.soproteoperadores@ordenacionjuego.gob.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*, *verificarCambiosDefunciones* 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. 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. 26 y 27)

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>2003-05-18</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>2003-05-18</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>2003-05-18</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: verificarCambiosDefunciones

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 *< cambioDefuncion >* para el mensaje Response.

Request:

No hay parámetros de entrada

Response (cambioDefuncion):

- Valor del **dni** (string) : DNI/NIE del jugador.
- Valor **fechaCambio** (datetime): fecha de detección por la DGOJ en registro defunciones



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

Response:

```
<cambioDefuncion>  
  
    <DNI>00000025W</DNI>  
  
    <fechaCambio>2018-08-10T12:12:12.000+02:00</fechaCambio>  
  
</cambioDefuncion>
```

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>
```

Operación: inicioRevisionPeriodica

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

Request (inicioRevisionPeriodica):

- Parámetro: **fechaInicio** (string): fecha de inicio propuesta por el operador.

Response (inicioRevisionPeriodicaResponse):

- Valor del **fechaInicio** (string): fecha de inicio propuesta por el operador.
- Valor del **fechaFin** (string): fecha fin calculada por el Sistema.
- Valor del **horalInicio** (string): fecha fin calculada por el Sistema.
- Valor del **horaFin** (string): fecha fin calculada por el Sistema.
- resultado:
 - **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:

```
<inicioRevisionPeriodica>  
  <fechaInicio>10/12/2023</fechaInicio>  
</inicioRevisionPeriodica>
```

Response:



```
<inicioRevisionPeriodicaResponse>  
  <fechaInicio>10/12/2023</fechaInicio>  
  <fechaFin>13/12/2023</fechaFin>  
  <horaInicio>00:00</horaInicio>  
  <horaFin>08:00</horaFin>  
  <resultado>  
    <codigo>COD000</codigo>  
    <descripcion>Operación realizada correctamente</descripcion>  
  </resultado>  
</inicioRevisionPeriodicaResponse>
```

Operación: *finRevisionPeriodica*

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

Request (*finRevisionPeriodica*):

- Parámetro: **fechaInicioEstimada** (string): fecha de inicio propuesta por el operador.
- Parámetro: **fechaFinEstimada** (string): fecha de inicio propuesta por el operador.
- Parámetro: **fechaInicio** (string): fecha de inicio real en la que el Operador ha realizado las operaciones de revisión periódica.
- Parámetro: **fechaFin** (string): fecha fin real en la que el Operador ha realizado las operaciones de revisión periódica.
- Parámetro: **horaInicio** (string): hora de inicio propuesta por el operador.
- Parámetro: **horaFin** (string): hora de inicio propuesta por el operador.

Response (*finRevisionPeriodicaResponse*):

- resultado:
 - **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:

```
<finRevisionPeriodica>  
  <fechaInicioEstimada>10/12/2023</fechaInicioEstimada>  
  <fechaFinEstimada>13/12/2023</fechaFinEstimada>  
  <fechaInicio>10/12/2023</fechaInicio>  
  <fechaFin>13/12/2023</fechaFin>  
  <horalInicio>00:00</horalInicio>  
  <horaFin>08:00</horaFin>  
</finRevisionPeriodica>
```

Response:

```
<finRevisionPeriodicaResponse>  
  <resultado>  
    <codigo>COD000</codigo>  
    <descripcion>Operación realizada correctamente</descripcion>  
  </resultado>  
</finRevisionPeriodicaResponse>
```

Operación: revisionJugador

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

Request (revisionJugador): N/A

Response (revisionJugador Response):

- Valor del **dni** (string): identificador (DNI o NIE) del jugador.
- Valor del **codigo** (string): código relacionado con las operaciones a realizar sobre dicho jugador.
- Valor del **fechaActual** (string): fecha actual del sistema.
- Valor del **codigoConfirmacion** (string): código único asociado a la operación de solicitud de revisión.
- resultado:
 - **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:

```
<revisionJugador>  
</revisionJugador>
```

Response:

```
<revisionJugadorResponse>  
  <datosRevisionJugador>  
    <dni>00000018H</dni>  
    <codigo>COD-REV-01</codigo>  
    <codigoConfirmacion>COD-CONF-0000001</codigoConfirmacion>  
  </datosRevisionJugador>  
  <datosRevisionJugador>  
    <dni>00000018H</dni>  
    <codigo>COD-REV-02</codigo>  
    <codigoConfirmacion>COD-CONF-0000002</codigoConfirmacion>  
  </datosRevisionJugador>  
  <fechaActual>10/12/2023</fechaActual>  
  <resultado>  
    <codigo>COD000</codigo>  
    <descripcion>Operación realizada correctamente</descripcion>  
  </resultado>  
</revisionJugadorResponse>
```

Operación: confirmarRevisionJugador

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

Request (revisionJugador): N/A

Response (revisionJugador Response):

- Valor del **codigoConfirmacion** (string): código de confirmación facilitado en la respuesta de la operación "revisionJugador".



- resultado:
 - **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:

```
<confirmarRevisionJugador>  
  <codigoConfirmacion>COD-CONF-0000002</codigoConfirmacion>  
</confirmarRevisionJugador>
```

Response:

```
<revisionJugadorResponse>  
  <resultado>  
    <codigo>COD000</codigo>  
    <descripcion>Operación realizada correctamente</descripcion>  
  </resultado>  
</revisionJugadorResponse>
```

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 qué 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 (Nombre incorrecto)-EN RGIAJ

En este caso de Prueba el jugador no está identificado correctamente, porque *nombre* Debería ser "MARIA ENCINA" en vez de "MARIA", y está inscrito en el RGIAJ.

Request:

```
<jugadores>
  <dni>10000320N</dni>
  <nombre>MARIA</nombre>
  <apellido1>FELIZ</apellido1>
  <apellido2>VARELA</apellido2>
  <fechaNacimiento>2003-05-18</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 - Nombre del titular
```



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

4.2.3.2 Verificar un jugador No OK (Primer apellido incorrecto)-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>2003-05-18</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 - Primer apellido del  
titular incorrecto</descripcion>  
  </resultadoidentidad>  
</resultadosJugador>
```

4.2.3.3 Verificar un jugador No OK (Segundo apellido incorrecto)-EN RGIAJ

En esta caso de Prueba el jugador no está identificado correctamente, porque *apellido2* debería ser “VARELA” en vez de “BARELA”, y está inscrito en el RGIAJ.



Request:

```
<jugadores>  
  <dni>10000320N</dni>  
  <nombre>MARIA ENCINA</nombre>  
  <apellido1>FELIZ</apellido1>  
  <apellido2>BARELA</apellido2>  
  <fechaNacimiento>2003-05-18</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 - Segundo apellido del  
titular incorrecto</descripcion>  
  </resultadoidentidad>  
</resultadosJugador>
```

4.2.3.4 Verificar un jugador No OK (Fecha nacimiento incorrecta)-EN RGIAJ

En esta caso de Prueba el jugador no está identificado correctamente, porque *fechaNacimiento* debería ser "18/05/2003" en vez de "17/05/2003", y está inscrito en el RGIAJ.

Request:

```
<jugadores>  
  <dni>10000320N</dni>  
  <nombre>MARIA ENCINA</nombre>  
  <apellido1>FELIZ</apellido1>  
  <apellido2>VARELA</apellido2>
```



```
<fechaNacimiento>2003-05-17</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 - Fecha de nacimiento  
del titular incorrecta</descripcion>  
  </resultadoidentidad>  
</resultadosJugador>
```

4.2.3.5 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>2003-05-18</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>
```

4.2.3.6 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>
```

4.2.3.7 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>
```

4.2.3.8 Verificar dos jugadores 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>
  <resultadoIdentidad>
    <codigo>COD003</codigo>
    <descripcion>La identidad del usuario es correcta</descripcion>
  </resultadoIdentidad>
</resultadosJugador>
<resultadosJugador>
  <dni>30000831E</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>
```

4.2.3.9 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> ÇÖBĒRĪÄ</nombre>
  <apellido1> ÑÉBİMA </apellido1>
  <apellido2> L·LÜÇÄR</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>
```

4.2.3.10 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>20000128H</dni>  
  <nombre>LUIS</nombre>  
  <apellido1>GARCIA</apellido1>  
  <apellido2>ESCRIVA</apellido2>  
  <fechaNacimiento>2010-02-23</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>COD003</codigo>  
    <descripcion>La identidad del usuario es correcta</descripcion>  
  </resultadoidentidad>  
</resultadosJugador>
```



4.2.3.11 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>07253092L</dni>
  <nombre>MARIA</nombre>
  <apellido1>RUIZ</apellido1>
  <apellido2>ALONSO</apellido2>
  <fechaNacimiento>1944-05-18</fechaNacimiento>
  <numSoporte></numSoporte>
</jugadores>
```

Response:

```
<resultadosJugador>
  <dni> 07253092L</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>
```

4.2.3.12 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 - El documento  
    solicitado puede estar en trámite de renovación, retenido, retirado por autoridad  
    judicial o anulado por duplicidad.  
  </descripcion>  
  </resultadoidentidad>  
</resultadosJugador>
```

4.2.3.13 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:

```
<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: Se ha
      encontrado contenido no válido a partir del elemento 'apellido1'. Se
      esperaba uno de '{nombre}'. </faultstring>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

4.2.3.14 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>
```



</resultadoidentidad>
</resultadosJugador>

4.2.3.15 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

```
<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: El contenido del
      elemento 'jugadores' no está completo. Se esperaba uno de '{apellido2,
      fechaNacimiento}'. </faultstring>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```



4.2.3.16 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 - El documento solicitado
    puede estar en trámite de renovación, retenido, retirado por autoridad judicial o
    anulado por duplicidad</descripcion>
  </resultadoidentidad>
</resultadosJugador>
```

4.2.3.17 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:

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



4.2.3.18 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>
```



4.2.3.19 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>
```



4.2.3.20 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>
```



4.2.3.21 Verificar un jugador que se encuentra en el registro de defunciones

En este caso el jugador es identificado correctamente y posteriormente se detecta que está incluido en el Registro Civil como fallecido. El resultado de la verificación en el RGIAJ es indiferente para esta prueba.

Enviar la siguiente petición:

Request:

```
<jugadores>
  <dni>00000025W</dni>
  <nombre> Antonio</nombre>
  <apellido1>Gomez</apellido1>
  <apellido2>Bueno</apellido2>
  <fechaNacimiento> 06/04/1978</fechaNacimiento>
</jugadores>
```

Si es la primera vez que el operador está preguntando por este jugador se obtendría un COD003 para su identidad (identidad verificada correctamente):

Response:

```
<resultadosJugador>
  <dni>00000025W</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>
```

Y a partir de ese momento al ser un dato de prueba para defunciones la DGOJ lo detectaría en su registro de defunciones y devolvería un COD007 (jugador fallecido):

Response:

```
<resultadosJugador>
  <dni>00000025W</dni>
  <resultadoRGIAJ>
    <codigo>COD002</codigo>
    <descripcion>El usuario no está inscrito en el RGIAJ</descripcion>
```



```
</resultadoRGIAJ>
<resultadoidentidad>
  <codigo>COD007</codigo>
  <descripcion>El usuario fue identificado en el Registro Civil como fallecido
  con anterioridad a la recepción de la petición</descripcion>
</resultadoidentidad>
</resultadosJugador>
```

4.2.3.22 Verificar un jugador cuya identidad no es correcta y además es menor de edad.

En este caso de Prueba el jugador no está identificado correctamente, y es menor de edad. Para esta prueba es indiferente el resultado obtenido en verificación del RGIAJ.

Request:

```
<jugadores>
  <dni>20000128H</dni>
  <nombre>LUIS</nombre>
  <apellido1>GARCIA</apellido1>
  <apellido2>ESCRIVA</apellido2>
  <fechaNacimiento>2010-02-21</fechaNacimiento>
</jugadores>
```

Response:

```
<resultadosJugador>
  <dni>20000128H</dni>
  <resultadoRGIAJ>
    <codigo>COD002</codigo>
    <descripcion>El usuario no está inscrito en el RGIAJ</descripcion>
  </resultadoRGIAJ>
  <resultadoidentidad>
    <codigo>COD104</codigo>
    <descripcion>La identidad del usuario es incorrecta - Fecha de nacimiento
    del titular incorrecta - Identificado como menor</descripcion>
  </resultadoidentidad>
</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>20000128H</dni>  
  <nombre>LUIS</nombre>  
  <apellido1>GARCIA</apellido1>  
  <apellido2>ESCRIVA</apellido2>  
  <fechaNacimiento>2010-02-23</fechaNacimiento>  
</jugadores>
```

Response:

```
<resultadosIdentidad>  
  <dni>20000128H</dni>  
  <resultadoIdentidad>  
    <codigo>COD003</codigo>  
    <descripcion>La identidad del usuario es correcta</descripcion>  
  </resultadoIdentidad>  
</resultadosIdentidad>
```

4.2.4.2 Verificar una identidad No OK (Nombre del titular incorrecto)

En este caso de Prueba el jugador no está identificado correctamente, porque *nombre* debería ser "MARIA ENCINA" en vez de "ENCINA".

Request:



```
<jugadores>
  <dni>10000320N</dni>
  <nombre>ENCINA</nombre>
  <apellido1>FELIZ</apellido1>
  <apellido2>VARELA</apellido2>
  <fechaNacimiento>2003-05-18</fechaNacimiento>
</jugadores>
```

Response:

```
<resultadosIdentidad>
  <dni>10000320N </dni>
  <resultadoidentidad>
    <codigo>COD004</codigo>
    <descripcion>La identidad del usuario es incorrecta - Nombre del titular
incorrecto</descripcion>
  </resultadoidentidad>
</resultadosIdentidad>
```

4.2.4.3 Verificar una identidad No OK (Primer apellido incorrecto)

En este caso de Prueba el jugador no está identificado correctamente, porque *apellido1* debería ser "FELIZ" en vez de "MARTIN".

Request:

```
<jugadores>
  <dni>10000320N</dni>
  <nombre>MARIA ENCINA</nombre>
  <apellido1>MARTIN</apellido1>
  <apellido2>VARELA</apellido2>
  <fechaNacimiento>2003-05-18</fechaNacimiento>
</jugadores>
```

Response:

```
<resultadosIdentidad>
```



```
<dni>10000320N </dni>
<resultadoIdentidad>
  <codigo>COD004</codigo>
  <descripcion>La identidad del usuario es incorrecta – Primer apellido del
titular incorrecto</descripcion>
</resultadoIdentidad>
</resultadosIdentidad>
```

4.2.4.4 Verificar una identidad No OK (Segundo apellido incorrecto)

En este caso de Prueba el jugador no está identificado correctamente, porque *apellido2* debería ser “VARELA” en vez de “MARTIN”.

Request:

```
<jugadores>
  <dni>10000320N</dni>
  <nombre>MARIA ENCINA</nombre>
  <apellido1>FELIZ</apellido1>
  <apellido2>MARTIN</apellido2>
  <fechaNacimiento>2003-05-18</fechaNacimiento>
</jugadores>
```

Response:

```
<resultadosIdentidad>
  <dni>10000320N </dni>
  <resultadoIdentidad>
    <codigo>COD004</codigo>
    <descripcion>La identidad del usuario es incorrecta – Segundo apellido del
titular incorrecto</descripcion>
  </resultadoIdentidad>
</resultadosIdentidad>
```

4.2.4.5 Verificar una identidad No OK (Fecha de nacimiento incorrecta)

En este caso de Prueba el jugador no está identificado correctamente, porque *fechaNacimiento* debería ser “2003-05-18” en vez de “2004-05-18”.



Request:

```
<jugadores>
  <dni>10000320N</dni>
  <nombre>MARIA ENCINA</nombre>
  <apellido1>FELIZ</apellido1>
  <apellido2>VARELA</apellido2>
  <fechaNacimiento>2004-05-18</fechaNacimiento>
</jugadores>
```

Response:

```
<resultadosIdentidad>
  <dni>10000320N </dni>
  <resultadoIdentidad>
    <codigo>COD004</codigo>
    <descripcion>La identidad del usuario es incorrecta - Fecha de nacimiento
del titular incorrecta</descripcion>
  </resultadoIdentidad>
</resultadosIdentidad>
```

4.2.4.6 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>ÇÖBËRİÄ</nombre>
  <apellido1>ÑËBÌMA</apellido1>
  <apellido2>L·LÜÇÄR</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>
```

4.2.4.7 Verificar una identidad que no está 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 - El documento  
solicitado puede estar en trámite de renovación, retenido, retirado por autoridad  
judicial o anulado por duplicidad.</descripcion>  
  </resultadoIdentidad>  
</resultadosIdentidad>
```



4.2.4.8 Verificar identidad de Nif Anulado

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

Request:

```
<jugadores>  
  <dni>07253092L </dni>  
  <nombre>MARIA</nombre>  
  <apellido1>RUIZ</apellido1>  
  <apellido2>ALONSO</apellido2>  
  <fechaNacimiento>1944-05-18</fechaNacimiento>  
  <numSoporte></numSoporte>  
</jugadores>
```

Response:

```
<resultadosIdentidad>  
  <dni>07253092L </dni>  
  <resultadoidentidad>  
    <codigo>COD004</codigo>  
    <descripcion>La identidad del usuario es incorrecta - DNI del Titular  
    Anulado</descripcion>  
  </resultadoidentidad>  
</resultadosIdentidad>
```

4.2.4.9 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 - El documento solicitado puede estar en trámite de renovación, retenido, retirado por autoridad judicial o anulado por duplicidad.

</descripcion>

</resultadoidentidad>

</resultadosIdentidad>

4.2.4.10 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>
```

4.2.4.11 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>
```

4.2.4.12 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>
```

4.2.4.13 Verificar identidad de un jugador que se encuentra en el Registro Civil como fallecido

Enviar la siguiente petición:

Request:

```
<jugadores>
```



```
<dni>00575757K</dni>  
<nombre Maria Fernanda</nombre>  
<apellido1>Ruiz</apellido1>  
<apellido2>Montilla</apellido2>  
<fechaNacimiento>1959-02-27</fechaNacimiento>  
</jugadores>
```

Si es la primera vez que el operador está preguntando por la identidad de este jugador, se obtendría un COD003 (identidad verificada correctamente):

Response:

```
<resultadosIdentidad>  
<dni>00575757K</dni>  
<resultadoidentidad>  
<codigo>COD003</codigo>  
<descripcion>La identidad del usuario es correcta</descripcion>  
</resultadoidentidad>  
</resultadosIdentidad>
```

Y a partir de ese momento al ser un dato de prueba para defunciones la DGOJ lo detectaría en su registro de defunciones y devolvería un COD007 (jugador fallecido):

Response:

```
<resultadosIdentidad>  
<dni>00575757K </dni>  
<resultadoidentidad>  
<codigo>COD007</codigo>  
<descripcion>El usuario fue identificado en el Registro Civil como fallecido  
con anterioridad a la recepción de la petición</descripcion>  
</resultadoidentidad>  
</resultadosIdentidad>
```

4.2.4.14 Verificar identidad de un jugador. La identidad no es correcta y además es menor de edad.

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

Request:



```
<jugadores>  
  <dni>20000128H</dni>  
  <nombre>LUIS</nombre>  
  <apellido1>GARCIA</apellido1>  
  <apellido2>ESCRIVA</apellido2>  
  <fechaNacimiento>2010-02-21</fechaNacimiento>  
</jugadores>
```

Response:

```
<resultadosIdentidad>  
  <dni>20000128H</dni>  
  <resultadoidentidad>  
    <codigo>COD104</codigo>  
    <descripcion>La identidad del usuario es incorrecta - Fecha de nacimiento del titular  
    incorrecta - Identificado como menor</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

```
<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: Se ha encontrado contenido no válido a partir del elemento 'apellido1'. Se esperaba uno de '{nombre}'. </faultstring>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```



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 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> ÇÖBËRİÄ</nombre>  
  <apellido1> ÑÉBÌMA </apellido1>  
  <apellido2> L·LÜÇÄR</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 preproducción se devolverá un listado con los DNIs de prueba indicados mas abajo, cuya situación en el RGIAJ haya sido preguntada por el operador.

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

- **Lista de DNI's/NIE'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/NIE's de la lista, la operación no devuelve ningún dato.
2. A continuación se irán verificando los jugadores (DNI's/NIE's) de prueba 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/NIE'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/NIE concreto de un operador, se enviará un correo electrónico a dgoj.soproteoperadores@ordenacionjuego.gob.es indicando la operación solicitada.

4.2.6.1 Verificar cambios en RGIAJ sin haber verificado ningún jugador

En este caso de prueba, el operador aún no ha verificado ninguno de los DNI's/NIE'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/NIE'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/NIE 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/NIE'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 *verificarCambiosDefunciones*

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

En la operación *VerificarCambiosDefunciones* del entorno de pruebas se devolverá un listado con los DNIs, cuya identidad haya sido correctamente verificada por el operador y que después de dicha verificación se haya encontrado el DNI/NIE en el Registro Civil como fallecido, avisando así al operador para que adopte las medidas pertinentes.

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, se encuentra como difunto 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.

A continuación se muestra la lista de datos de Jugadores con las que se podrán realizar las pruebas de la operación *verificarCambiosDefunciones*.

- **Lista de Jugadores para las pruebas:**

<u>DNI</u>	<u>Nombre</u>	<u>Apellido1</u>	<u>Apellido2</u>	<u>Fecha Nacimiento</u>
00000025W	Antonio	Gomez	Bueno	06/04/1978
00575757K	Maria Fernanda	Ruiz	Montilla	27/02/1959
X0000024R	Boris	Karlev		14/12/1960

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 verificar la identidad de alguno de los jugadores de la lista, la operación no devuelve ningún dato.



2. A continuación se irán verificando jugadores de la lista con las operaciones de *VerificarJugador* o *VerificarIdentidad*. Se deben enviar los datos exactamente tal y como se indica en el listado anterior. Se obtendrá para la identidad, un COD003 de identidad correcta. A partir de ahí la DGOJ simulará que ha encontrado este DNI/NIE en el Registro Civil como fallecido.
3. Posteriormente se llamará a la operación *VerificarCambiosDefunciones*, y el operador recibirá éste DNI/NIE en la lista de resultados de difuntos.
4. Si se volviera a preguntar por la identidad de uno de estos DNIs (con *VerificarJugador* ó *VerificarIdentidad*) obtendríamos ahora un COD007 informando de que el jugador consta en el Registro Civil como fallecido.

Es importante verificar que en ningún caso se devuelven en esta operación DNIs/NIEs de los que nunca se ha verificado su identidad de forma correcta.

Una vez se hayan verificado toda la lista de DNIs/NIEs del juego de pruebas, esta operación siempre devolverá la lista completa. En el caso de querer realizar la prueba desde el principio, con algún DNI/NIE concreto de la lista, se enviará un correo electrónico a dgoj.soproteoperadores@ordenacionjuego.gob.es indicando la operación solicitada.

4.2.7.1 Verificar cambios en defunciones sin verificar jugadores

En este caso de prueba el operador aún no ha verificado la identidad de ninguno de los DNI's/NIE's indicados en el punto anterior para las pruebas, el servicio devuelve un mensaje vacío. Todavía no se encuentra ninguno de los DNIs/NIEs en el Registro Civil como fallecido:

Response:

```
<soap:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="Id-2122782205">
  <ns2:verificarCambiosDefuncionesResponse
    xmlns:ns2="http://ws.cnjuego.gob.es/VerificacionJugadores"/>
</soap:Body>
```

Si el operador ya hubiera verificado la identidad de alguno de estos DNIs/NIEs, el servicio devolvería estos DNI's/NIE's en la respuesta.



4.2.7.2 Verificar cambios en defunciones después de verificar Jugadores del juego de pruebas

En este caso de prueba, en primer lugar un operador debe haber verificado uno de los jugadores de las pruebas (indicados en la lista del punto 4.2.7), a través de la operación de *verificarJugador o verificarIdentidad*. Se simularía entonces que este jugador a pasado a ser difunto y a partir de ese momento, al llamar a la operación *verificarCambiosDefunciones* se recibe ese DNI en el listado por haber pasado a formar parte de la lista de fallecidos en el Registro Civil.

Ejemplo:

Se verifica la identidad del DNI 00000025W:

Request:

```
<verificarIdentidad>  
  <jugadores>  
    <dni>00000025W</dni>  
    <nombre>Antonio</nombre>  
    <apellido1>Gomez</apellido1>  
    <apellido2>Bueno</apellido2>  
    <fechaNacimiento>1978-04-06</fechaNacimiento>  
  </jugadores>  
</verificarIdentidad>
```

Se obtiene:

Response:

```
<verificarIdentidadResponse>  
  <resultadosIdentidad>  
    <dni>00000025W</dni>  
    <resultadoidentidad>  
      <codigo>COD003</codigo>  
      <descripcion>La identidad del usuario es correcta</descripcion>  
    </resultadoidentidad>  
  </resultadosIdentidad>  
</verificarIdentidadResponse>
```



Y se llama a la operación de *verificarCambiosDefunciones*, obteniendo ya el DNI/NIE cuya identidad hemos verificado antes, por estar este DNI incluido en el Registro Civil como fallecido:

Response:

```
<verificarCambiosDefuncionesResponse>  
  <cambioDefuncion>  
    <DNI>00000025W</DNI>  
    <fechaCambio>2018-08-17T10:04:44.000+02:00</fechaCambio>  
  </cambioDefuncion>  
</verificarCambiosDefuncionesResponse>
```

A continuación se podrá verificar otro DNI/NIE de la lista o varios de forma simultánea. Posteriormente se volverá a llamar a la VerificaciónCambiosDefunciones. Se devuelven los DNIs/NIEs ya verificados.

Se verifica el DNI 00575757K y se llama a la operación de *verificarCambiosDefunciones*:

Response:

```
<verificarCambiosDefuncionesResponse>  
  <cambioDefuncion>  
    <DNI>00000025W</DNI>  
    <fechaCambio>2018-08-17T10:04:44.000+02:00</fechaCambio>  
  </cambioDefuncion>  
  <cambioDefuncion>  
    <DNI>00575757K</DNI>  
    <fechaCambio>2018-08-17T10:38:57.000+02:00</fechaCambio>  
  </cambioDefuncion>  
</verificarCambiosDefuncionesResponse>
```

De esta forma se podrán ir verificando los tres jugadores de prueba, comprobando el resultado mediante la llamada a la operación de *verificarCambiosDefunciones* y verificando que únicamente se devuelven DNIs/NIEs 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).



4.2.8 Casos de Prueba de la Operación de bajaJugador

4.2.8.1 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, ésta 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>
```

4.2.8.2 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>
```

4.2.8.3 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>
```

4.2.8.4 Causa de la baja errónea

En este caso de Prueba se envía una causa de baja no valida o vacía. 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 vacío

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

Response:

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

4.2.8.5 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>
```

4.2.8.6 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: Se ha encontrado contenido no  
válido a partir del elemento 'causa'. Se esperaba uno de '{dni}'. </faultstring>  
    </soap:Fault>  
  </soap:Body>  
</soap:Envelope>
```

4.2.8.7 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: El contenido del elemento  
'bajaJugador' no está completo. Se esperaba uno de '{causa}'. </faultstring>  
    </soap:Fault>  
  </soap:Body>  
</soap:Envelope>
```

4.2.8.8 Verificar cambios en RGIAJ después de una baja

En este caso de prueba, se da de baja de uno de los DNI's/NIE'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>
```

4.2.9 Casos de Prueba de la Operación de inicioRevisionPeriodica

4.2.9.1 Inicio de una revisión periódica por el Operador

En este Caso de Prueba el operador propone una fecha de inicio en la que comenzar el proceso de revisión.

Request:

```
<inicioRevisionPeriodica>  
<fechaInicio>10/12/2023</fechaInicio>  
</inicioRevisionPeriodica>
```

Response:

```
<inicioRevisionPeriodicaResponse>  
<fechaInicio>10/12/2023</fechaInicio>  
<fechaFin>13/12/2023</fechaFin>  
<horaInicio>00:00</horaInicio>  
<horaFin>08:00</horaFin>  
<idPeticon>234342</idPeticon>  
<resultado>  
<codigo>COD000</codigo>
```



```
<descripcion>Operación realizada correctamente</descripcion>
</resultado>
</inicioRevisionPeriodicaResponse>
```

4.2.9.2 Inicio de una revisión periódica por el Operador - ERROR

En este Caso de Prueba el operador propone una fecha de inicio en la que comenzar el proceso de revisión, produciéndose un error en la operación por formato de fecha de inicio inválido.

Request:

```
<inicioRevisionPeriodica>
  <fechaInicio>120/1212/2023</fechaInicio>
</inicioRevisionPeriodica>
```

Response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Server</faultcode>
      <faultstring>El formato de la fecha de inicio es inválido.</faultstring>
      <detail>
        <ns2:inicioRevisionPeriodicaFault
xmlns:ns2="http://ws.cnjuego.gob.es/VerificacionJugadores/">
          <inicioRevisionPeriodicaFault>COD908</inicioRevisionPeriodicaFault>
        </ns2:inicioRevisionPeriodicaFault>
      </detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

4.2.10 Casos de Prueba de la Operación de finRevisionPeriodica

4.2.10.1 Fin de una revisión periódica por el Operador

En este Caso de Prueba el operador facilita la fecha de inicio y fin (estimada en la operación de inicioRevisionPeriodica), la fecha de inicio y fin en la que se ha realizado la operación, así como la franja horaria y el identificador de la petición de inicioRevisionPeriodica.



Request:

```
<finRevisionPeriodica>  
  <fechaInicioEstimada>10/12/2023</fechaInicioEstimada>  
  <fechaFinEstimada>13/12/2023</fechaFinEstimada>  
  <fechaInicio>10/12/2023</fechaInicio>  
  <fechaFin>13/12/2023</fechaFin>  
  <horalInicio>00:00</horalInicio>  
  <horaFin>08:00</horaFin>  
  <idPetición>234342</idPetición> <!-- el identificador de la petición devuelto al llamar  
a inicioRevisiónPeriodica -->  
</finRevisionPeriodica>
```

Response:

```
<finRevisionPeriodicaResponse>  
  <resultado>  
    <codigo>COD000</codigo>  
    <descripcion>Operación realizada correctamente</descripcion>  
  </resultado>  
</finRevisionPeriodicaResponse>
```

4.2.10.2 Fin de una revisión periódica por el Operador - ERROR

En este Caso de Prueba el operador facilita la fecha de inicio y fin (estimada en la operación de inicioRevisionPeriodica), la fecha de inicio y fin en la que se ha realizado la operación, así como la franja horaria y el identificador de la petición de inicioRevisionPeriodica. Introducimos un id de petición inválido. Se produce un error al validar el identificador idPetición:

Request:

```
<finRevisionPeriodica>  
  <fechaInicioEstimada>10/12/2023</fechaInicioEstimada>  
  <fechaFinEstimada>13/12/2023</fechaFinEstimada>  
  <fechaInicio>10/12/2023</fechaInicio>  
  <fechaFin>13/12/2023</fechaFin>  
  <horalInicio>00:00</horalInicio>  
  <horaFin>08:00</horaFin>  
  <idPetición>234340</idPetición> <!-- el identificador inválido de la petición devuelto  
al llamar a inicioRevisiónPeriodica -->  
</finRevisionPeriodica>
```



Response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Server</faultcode>
      <faultstring>Id de petición inválido</faultstring>
      <detail>
        <ns2:finRevisionPeriodicaFault
xmlns:ns2="http://ws.cnjuego.gob.es/VerificacionJugadores/">
          <finRevisionPeriodicaFault>COD914</finRevisionPeriodicaFault>
        </ns2:finRevisionPeriodicaFault>
      </detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

4.2.11 Casos de Prueba de la Operación de revisionJugador

4.2.11.1 Revisión de un jugador por el Operador

En este caso de Prueba el operador solicita información sobre los jugadores que tienen pendiente una serie de acciones.

Request:

```
<revisionJugador>
</revisionJugador>
```

Response:

```
<revisionJugadorResponse>
  <datosRevisionJugador>
    <dni>00000018H</dni>
    <codigo>COD-REV-01</codigo>
    <codigoConfirmacion>COD-CONF-0000001</codigoConfirmacion>
  </datosRevisionJugador>
  <datosRevisionJugador>
    <dni>00000001R</dni>
    <codigo>COD-REV-02</codigo>
    <codigoConfirmacion>COD-CONF-0000002</codigoConfirmacion>
  </datosRevisionJugador>
  <idPeticon>234343</idPeticon>
```



```
<fechaActual>10/12/2023</fechaActual>  
<resultado>  
  <codigo>COD000</codigo>  
  <descripcion>Operación realizada correctamente</descripcion>  
</resultado>  
</revisiónJugadorResponse>
```

NOTA: Previamente a la ejecución de la petición, debe verificarse que los jugadores con DNI 00000018H y 00000001R están dados de alta como jugadores del operador.

4.2.11.2 Revisión de un jugador por el Operador - ERROR

En este caso de Prueba el operador solicita información sobre los jugadores que tienen pendiente una serie de acciones, pero se produce un error interno en la petición.

Request:

```
<revisiónJugador>  
</revisiónJugador>
```

Response:

```
<revisiónJugadorResponse>  
  <resultado>  
    <codigo>ERR001</codigo>  
    <descripcion> Error de aplicación, consulte a la Dirección General de  
    Ordenación del Juego.</descripcion>  
  </resultado>  
</revisiónJugadorResponse>
```

4.2.12 Casos de Prueba de la Operación de confirmarRevisiónJugador

4.2.12.1 Confirmación de la revisión de un jugador por el Operador

En este caso de Prueba el operador confirma la revisión de las acciones pendientes asociadas al código de confirmación facilitado como parámetro de entrada. Del mismo modo, proporciona el identificador de la petición relacionada "revisiónJugador".

Request:

```
<confirmarRevisiónJugador>  
  <codigoConfirmación>COD-CONF-0000002</codigoConfirmación>
```



<idPetición>234343</idPetición> <!-- el identificador de la petición devuelto al llamar a revisionJugador -->

</confirmarRevisionJugador>

Response:

<revisionJugadorResponse>

<resultado>

<codigo>COD000</codigo>

<descripcion>Operación realizada correctamente</descripcion>

</resultado>

</revisionJugadorResponse>

4.2.12.2 Confirmación de la revisión de un jugador por el Operador - ERROR

En este caso de Prueba el operador confirma la revisión de las acciones pendientes asociadas al código de confirmación facilitado como parámetro de entrada. Del mismo modo, proporciona el identificador de la petición relacionada "revisionJugador". Se produce un error en la petición porque el código de la operación revisada no existe:

Request:

<confirmarRevisionJugador>

<codigoConfirmacion>COD-CONF-00sdf09</codigoConfirmacion>

<idPetición>234343</idPetición> <!-- el identificador de la petición devuelto al llamar a revisionJugador -->

</confirmarRevisionJugador>

Response:

<confirmarRevisionJugadorResponse>

<resultado>

<codigo>COD915</codigo>

<descripcion>El código de confirmación no pertenece al Id de petición</descripcion>

</resultado>

</confirmarRevisionJugadorResponse>



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 dgoj.sopORTEoperadores@ordenacionjuego.gob.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 válidos 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 verificarJugador

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>2010-05-11</fechaNacimiento>  
</jugadores>
```

Response:

```
<resultadosJugador>  
  <dni>Y9999999G</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 jugadores 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>2010-05-11</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 está 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:

```
<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: Se ha encontrado contenido no
válido a partir del elemento 'apellido1'. Se esperaba uno de '{nombre}'. </faultstring>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

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

```
<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: El contenido del
      elemento 'jugadores' no está completo. Se esperaba uno de '{apellido2,
      fechaNacimiento}'. </faultstring>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```



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>99887766P</dni>  
  <nombre>MARIA</nombre>  
  <apellido1>GARCIA</apellido1>  
  <apellido2>PEREZ</apellido2>  
  <fechaNacimiento>1990-09-01/fechaNacimiento>  
</jugadores>
```

Response:

```
<resultadosJugador>  
  <dni>99887766P</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 primeras 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>
  <resultadoidentidad>
    <codigo>COD005</codigo>
    <descripcion> No se ha podido verificar la identidad del usuario
  </descripcion>
  </resultadoidentidad>
</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>

4.3.2.15 Verificar un jugador que se encuentra incluido en el Registro Civil como fallecido

En este caso el jugador es identificado correctamente y posteriormente se detecta que está en el registro de defunciones.

Enviar la siguiente petición:

Request:

```
<jugadores>  
  <dni>00000022E</dni>  
  <nombre>Marta</nombre>  
  <apellido1>Terron</apellido1>  
  <apellido2>Costa</apellido2>  
  <fechaNacimiento>1972-09-09</fechaNacimiento>  
</jugadores>
```

Response:

```
<resultadosJugador>  
  <dni>00000022E </dni>  
  <resultadoRGIAJ>  
    <codigo>COD002</codigo>  
    <descripcion>El usuario no está inscrito en el RGIAJ</descripcion>  
  </resultadoRGIAJ>  
  <resultadoidentidad>  
    <codigo>COD007</codigo>  
    <descripcion> El usuario fue identificado en el Registro Civil como fallecido  
    con anterioridad a la recepción de la petición</descripcion>  
  </resultadoidentidad>  
</resultadosJugador>
```



4.3.2.16 Verificar un jugador cuya identidad no es correcta y además es menor de edad.

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

Request:

```
<jugadores>  
  <dni> X0000011B</dni>  
  <nombre>Roland</nombre>  
  <apellido1>Cienfuegos</apellido1>  
  <fechaNacimiento>1998-12-10</fechaNacimiento>  
</jugadores>
```

Response:

```
<resultadosJugador>  
  <dni> X0000011B </dni>  
  <resultadoRGIAJ>  
    <codigo>COD002</codigo>  
    <descripcion>El usuario no está inscrito en el RGIAJ</descripcion>  
  </resultadoRGIAJ>  
  <resultadoidentidad>  
    <codigo>COD104</codigo>  
    <descripcion>La identidad del usuario es incorrecta - Fecha de nacimiento del titular incorrecta - Identificado como menor</descripcion>  
  </resultadoidentidad>  
</resultadosJugador>
```



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>2010-05-11</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>Y99999999G</dni>  
  <nombre>Esperanza</nombre>  
  <apellido1>San Juan</apellido1>  
  <apellido2>Mateo</apellido2>  
  <fechaNacimiento>1985-06-05</fechaNacimiento>  
  <numSoporte></numSoporte>  
</jugadores>
```

Response:

```
<resultadosIdentidad>  
  <dni>Y99999999G</dni>  
  <resultadoIdentidad>  
    <codigo>COD003</codigo>  
    <descripcion>La identidad del usuario es correcta</descripcion>  
  </resultadoIdentidad>  
</resultadosIdentidad>
```

4.3.3.3 Verificar una identidad No OK (Nombre incorrecto)

En este caso de Prueba el jugador no está identificado correctamente, porque *nombre* debería ser “EMILIO JOSE” en vez de “EMILIO”. Por tanto el Operador no debe permitirle darse de alta por no estar correctamente identificado.



Request:

```
<jugadores>
  <dni>00000034B</dni>
  <nombre>EMILIO</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 – Nombre del titular incorrecta</descripcion>
  </resultadoIdentidad>
</resultadosIdentidad>
```

4.3.3.4 Verificar una identidad No OK (Primer apellido incorrecto)

En este caso de Prueba el jugador no está identificado correctamente, porque *apellido1* debería ser “GARCIA-RUIZ” en vez de “MARTIN”. Por tanto el Operador no debe permitirle darse de alta por no estar correctamente identificado.

Request:

```
<jugadores>
  <dni>00000034B</dni>
  <nombre>EMILIO JOSE</nombre>
  <apellido1>MARTIN</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 – Primer apellido del
    titular incorrecto</descripcion>
  </resultadoidentidad>
</resultadosIdentidad>
```

4.3.3.5 Verificar una identidad No OK (Segundo apellido incorrecto)

En este caso de Prueba el jugador no está identificado correctamente, porque *apellido2* debería ser “GONZALEZ” en vez de “MARTIN”. Por tanto el Operador no debe permitirle darse de alta por no estar correctamente identificado.

Request:

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

Response:

```
<resultadosIdentidad>
  <dni>00000034B</dni>
  <resultadoidentidad>
    <codigo>COD004</codigo>
    <<descripcion>La identidad del usuario es incorrecta – Segundo apellido
    del titular incorrecta</descripcion>
  </resultadoidentidad>
</resultadosIdentidad>
```



4.3.3.6 Verificar una identidad No OK (Fecha de nacimiento incorrecta)

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 – Fecha de nacimiento del titular incorrecta</descripcion>
  </resultadoidentidad>
</resultadosIdentidad>
```

4.3.3.7 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>
```

4.3.3.8 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>
```

4.3.3.9 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>
```



4.3.3.10 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>99887766P</dni>
  <nombre>MARIO</nombre>
  <apellido1>LOPEZ</apellido1>
  <apellido2>GARCIA</apellido2>
  <fechaNacimiento>1991-09-01</fechaNacimiento>
</jugadores>
```

Response:

```
<resultadosIdentidad>
  <dni>99887766P</dni>
  <resultadoIdentidad>
    <codigo>COD004</codigo>
    <descripcion> La identidad del usuario es incorrecta – Titular no
    identificado
  </descripcion>
  </resultadoIdentidad>
</resultadosIdentidad>
```

4.3.3.11 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 primeras 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>  
    <descripción> No se ha podido verificar la identidad del usuario  
  </descripción>  
  </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>
```



4.3.3.12 Verificar identidad de un jugador que se encuentra incluido en el Registro Civil como fallecido

En este caso de prueba se envía un jugador que la DGOJ ha detectado que está incluido en el Registro Civil como fallecido.

Enviar la siguiente petición:

Request:

```
<jugadores>
  <dni>00000022E</dni>
  <nombre>Marta</nombre>
  <apellido1>Terron</apellido1>
  <apellido2>Costa </apellido2>
  <fechaNacimiento>1972-09-09</fechaNacimiento>
</jugadores>
```

Response:

```
<resultadosIdentidad>
  <dni>00000022E </dni>
  <resultadoidentidad>
    <codigo>COD007</codigo>
    <descripcion> El usuario fue identificado en el Registro Civil como fallecido
    con anterioridad a la recepción de la petición</descripcion>
  </resultadoidentidad>
</resultadosIdentidad>
```

4.3.3.13 Verificar identidad de un jugador. La identidad no es correcta y además es menor de edad.

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

Request:

```
<jugadores>
  <dni> X0000011B</dni>
  <nombre>Roland</nombre>
  <apellido1>Cienfuegos</apellido1>
  <fechaNacimiento>1998-12-10</fechaNacimiento>
</jugadores>
```



Response:

```
<resultadosIdentidad>  
  <dni> X0000011B</dni>  
  <resultadoidentidad>  
    <codigo>COD104</codigo>  
    <descripcion>La identidad del usuario es incorrecta - Fecha de nacimiento del titular incorrecta - Identificado como menor</descripcion>  
  </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 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 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/NIE 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/NIE'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/NIE's de la lista, la operación no devuelve ningún dato.
2. A continuación se irán verificando jugadores (DNI's/NIE'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/NIE'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.soproteoperadores@ordenacionjuego.gob.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/NIE'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/NIE'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/NIE'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 verificarCambiosDefunciones

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

En la operación *VerificarCambiosDefunciones* del entorno de producción, cuando el operador ya esté operando con datos reales, se devolverá un listado con los DNIs, cuya identidad haya sido correctamente verificada por el operador y que después de dicha verificación se haya encontrado el DNI como fallecido en el Registro Civil..

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, se encuentra como difunto 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.

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

- **Lista de Jugadores para las pruebas:**

<u>DNI</u>	<u>Nombre</u>	<u>Apellido1</u>	<u>Apellido2</u>	<u>Fecha Nacimiento</u>
00000025W	Antonio	Gomez	Bueno	06/04/1978
00575757K	Maria Fernanda	Ruiz	Montilla	27/02/1959
X0000024R	Boris	Karlev		14/12/1960

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 verificar la identidad de alguno de los jugadores de la lista, la operación no devuelve ningún dato.
2. A continuación se irán verificando jugadores de la lista con las operaciones de *VerificarJugador* o *VerificarIdentidad*. Se deben enviar los datos exactamente tal y como se indica en el listado anterior. Se obtendrá para la identidad, un COD003 de identidad correcta. A partir de ahí la DGOJ simulará que ha encontrado este DNI/NIE como fallecido en el Registro Civil.



3. Posteriormente se llamará a la operación VerificarCambiosDefunciones, y el operador recibirá éste DNI/NIE en la lista de resultados de difuntos.
4. Si se volviera a preguntar por la identidad de uno de estos DNIs/NIEs (con VerificarJugador ó VerificarIdentidad) obtendríamos ahora un COD007 informando de que el jugador consta como fallecido en el Registro de Defunciones.

Es importante verificar que en ningún caso se devuelven en esta operación DNIs/NIEs de los que nunca se ha verificado su identidad de forma correcta.

Una vez se hayan verificado toda la lista de DNIs/NIEs del juego de pruebas, esta operación siempre devolverá la lista completa. En el caso de querer realizar la prueba desde el principio, con algún DNI/NIE concreto de la lista, se enviará un correo electrónico a dgoj.soproteoperadores@ordenacionjuego.gob.es indicando la operación solicitada.



4.3.7 Casos de Prueba de la Operación de bajaJugador

4.3.7.1 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, ésta 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>
```

4.3.7.2 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>
```

4.3.7.3 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>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>
```



4.3.7.4 Causa de la baja errónea

En este caso de Prueba se envía una causa de baja no válida o vacía. 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 vacío.

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

Response:

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

4.3.7.5 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>  
  <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>
```

4.3.7.6 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: Se ha encontrado contenido no  
válido a partir del elemento 'causa'. Se esperaba uno de '{dni}'. </faultstring>  
    </soap:Fault>  
  </soap:Body>
```



</soap:Envelope>

4.3.7.7 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>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: El contenido del elemento
' bajaJugador' no está completo. Se esperaba uno de '{causa}'. </faultstring>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

4.3.7.8 Verificar cambios en RGIAJ después de una baja

En este caso de prueba, se da de baja de uno de los DNI's/NIE'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>
```

4.3.8 Casos de Prueba de la Operación de inicioRevisionPeriodica

4.3.8.1 Inicio de una revisión periódica por el Operador

En este Caso de Prueba el operador propone una fecha de inicio en la que comenzar el proceso de revisión.

Request:

```
<inicioRevisionPeriodica>
<fechaInicio>10/12/2023</fechaInicio>
</inicioRevisionPeriodica>
```

Response:

```
<inicioRevisionPeriodicaResponse>
<fechaInicio>10/12/2023</fechaInicio>
```



```
<fechaFin>13/12/2023</fechaFin>
<horalInicio>00:00</horalInicio>
<horaFin>08:00</horaFin>
<idPetición>234342</idPetición>
<resultado>
  <codigo>COD000</codigo>
  <descripcion>Operación realizada correctamente</descripcion>
</resultado>
</inicioRevisionPeriodicaResponse>
```

4.3.8.2 Inicio de una revisión periódica por el Operador - ERROR

En este Caso de Prueba el operador propone una fecha de inicio en la que comenzar el proceso de revisión, produciéndose un error en la operación por formato de fecha de inicio inválido.

Request:

```
<inicioRevisionPeriodica>
  <fechaInicio>120/1212/2023</fechaInicio>
</inicioRevisionPeriodica>
```

Response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Server</faultcode>
      <faultstring>El formato de la fecha de inicio es inválido.</faultstring>
      <detail>
        <ns2:inicioRevisionPeriodicaFault
xmlns:ns2="http://ws.cnjuego.gob.es/VerificacionJugadores/">
          <inicioRevisionPeriodicaFault>COD908</inicioRevisionPeriodicaFault>
        </ns2:inicioRevisionPeriodicaFault>
      </detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```



4.3.9 Casos de Prueba de la Operación de finRevisionPeriodica

4.3.9.1 Fin de una revisión periódica por el Operador

En este Caso de Prueba el operador facilita la fecha de inicio y fin (estimada en la operación de inicioRevisionPeriodica), la fecha de inicio y fin en la que se ha realizado la operación, así como la franja horaria y el identificador de la petición de inicioRevisionPeriodica.

Request:

```
<finRevisionPeriodica>
  <fechaInicioEstimada>10/12/2023</fechaInicioEstimada>
  <fechaFinEstimada>13/12/2023</fechaFinEstimada>
  <fechaInicio>10/12/2023</fechaInicio>
  <fechaFin>13/12/2023</fechaFin>
  <horaInicio>00:00</horaInicio>
  <horaFin>08:00</horaFin>
  <idPeticon>234342</idPeticon> > <!-- el identificador de la petición devuelto al
  llamar a inicioRevisiónPeriodica -->
</finRevisionPeriodica>
```

Response:

```
<finRevisionPeriodicaResponse>
  <resultado>
    <codigo>COD000</codigo>
    <descripcion>Operación realizada correctamente</descripcion>
  </resultado>
</finRevisionPeriodicaResponse>
```

4.3.9.2 Fin de una revisión periódica por el Operador - ERROR

En este Caso de Prueba el operador facilita la fecha de inicio y fin (estimada en la operación de inicioRevisionPeriodica), la fecha de inicio y fin en la que se ha realizado la operación, así como la franja horaria y el identificador de la petición de inicioRevisionPeriodica. Introducimos un id de petición inválido. Se produce un error al validar el identificador idPeticon:

Request:

```
<finRevisionPeriodica>
  <fechaInicioEstimada>10/12/2023</fechaInicioEstimada>
  <fechaFinEstimada>13/12/2023</fechaFinEstimada>
```



```
<fechaInicio>10/12/2023</fechaInicio>
<fechaFin>13/12/2023</fechaFin>
<horaInicio>00:00</horaInicio>
<horaFin>08:00</horaFin>
<idPetición>234300</idPetición> > <!-- el identificador inválido de la petición
devuelto al llamar a inicioRevisiónPeriodica -->
</finRevisiónPeriodica>
```

Response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Server</faultcode>
      <faultstring>Id de petición inválido</faultstring>
      <detail>
        <ns2:finRevisiónPeriodicaFault
xmlns:ns2="http://ws.cnjuego.gob.es/VerificacionJugadores/">
          <finRevisiónPeriodicaFault>COD914</finRevisiónPeriodicaFault>
        </ns2:finRevisiónPeriodicaFault>
      </detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

4.3.10 Casos de Prueba de la Operación de revisiónJugador

4.3.10.1 Petición Revisiónjugador, correcta

En este caso de Prueba el operador solicita información sobre los jugadores que tienen pendiente una serie de acciones.

Request:

```
<revisiónJugador>
</revisiónJugador>
```

Response:

```
<revisiónJugadorResponse>
  <datosRevisiónJugador>
    <dni>00000060Z</dni>
    <codigo>IDENT-CHK</codigo>
    <codigoConfirmación> DGOJP43200000000060199953</codigoConfirmación>
```



```
</datosRevisionJugador>
<datosRevisionJugador>
  <dni>X0000040V</dni>
  <codigo>RGIAJ-CHK</codigo>
  <codigoConfirmacion> DGOJP43200000000060199955</codigoConfirmacion>
</datosRevisionJugador>
<idPeticion>234343</idPeticion>
<fechaActual>10/12/2023</fechaActual>
<resultado>
  <codigo>COD000</codigo>
  <descripcion>Operación realizada correctamente</descripcion>
</resultado>
</revisionJugadorResponse>
```

NOTA: Previamente a la ejecución de la petición, debe verificarse que los jugadores con DNI 00000060Z y X0000040V están dados de alta como jugadores del operador. Si no los han verificado en pruebas anteriores, háganlo y contacten con el buzón de soporte para que se les generen los casos de prueba de esta operación.

4.3.10.2 Petición RevisiónJugador - ERROR

En este caso de Prueba el operador solicita información sobre los jugadores que tienen pendiente una serie de acciones, pero se produce un error interno en la petición.

Request:

```
<revisionJugador>
</revisionJugador>
```

Response:

```
<revisionJugadorResponse>
  <resultado>
    <codigo>ERR001</codigo>
    <descripcion> Error de aplicación, consulte a la Dirección General de
    Ordenación del Juego.</descripcion>
  </resultado>
</revisionJugadorResponse>
```



4.3.11 Casos de Prueba de la Operación de confirmarRevisionJugador

4.3.11.1 Confirmación de la revisión de un jugador por el Operador

En este caso de Prueba el operador confirma la revisión de las acciones pendientes asociadas al código de confirmación facilitado como parámetro de entrada. Del mismo modo, proporciona el identificador de la petición relacionada "revisionJugador":

Request:

```
<confirmarRevisionJugador>  
    <codigoConfirmacion>DGOJP43200000000060199953</codigoConfirmacion>  
    <idPeticion>234343</idPeticion> <!-- el identificador de la petición devuelto al llamar  
a revisionJugador -->  
</confirmarRevisionJugador>
```

Response:

```
<confirmarRevisionJugadorResponse>  
    <resultado>  
        <codigo>COD915</codigo>  
        <descripcion>El código de confirmación no pertenece al Id de petición</descripcion>  
    </resultado>  
</confirmarRevisionJugadorResponse>
```

4.3.11.2 Confirmación de la revisión de un jugador por el Operador - ERROR

En este caso de Prueba el operador confirma la revisión de las acciones pendientes asociadas al código de confirmación facilitado como parámetro de entrada. Del mismo modo, proporciona el identificador de la petición relacionada "revisionJugador". Se produce un error en la petición:

Request:

```
<confirmarRevisionJugador>  
    <codigoConfirmacion>DGOJP43200000000060199955</codigoConfirmacion>  
    <idPeticion>234343</idPeticion> > <!-- el identificador de la petición devuelto al  
llamar a revisionJugador -->  
</confirmarRevisionJugador>
```

Response:

```
<confirmarRevisionJugadorResponse>  
    <resultado>  
        <codigo>COD915</codigo>  
        <descripcion>El código de confirmación no pertenece al Id de petición</descripcion>  
    </resultado>
```



</confirmarRevisionJugadorResponse>

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 “Supervisión y control”, epígrafe “Servicios web de verificación de jugadores” del portal de la Dirección General de Ordenación del Juego.

Las solicitudes se enviarán a la cuenta de correo dgoj.sopORTEoperadores@ordenacionjuego.gob.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 enviará a la a la cuenta de correo dgoj.sopORTEoperadores@ordenacionjuego.gob.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.sopORTEoperadores@ordenacionjuego.gob.es indicando en el asunto del mensaje “TEST WS-VerificaciónJugadores”.



5 Test del servicio

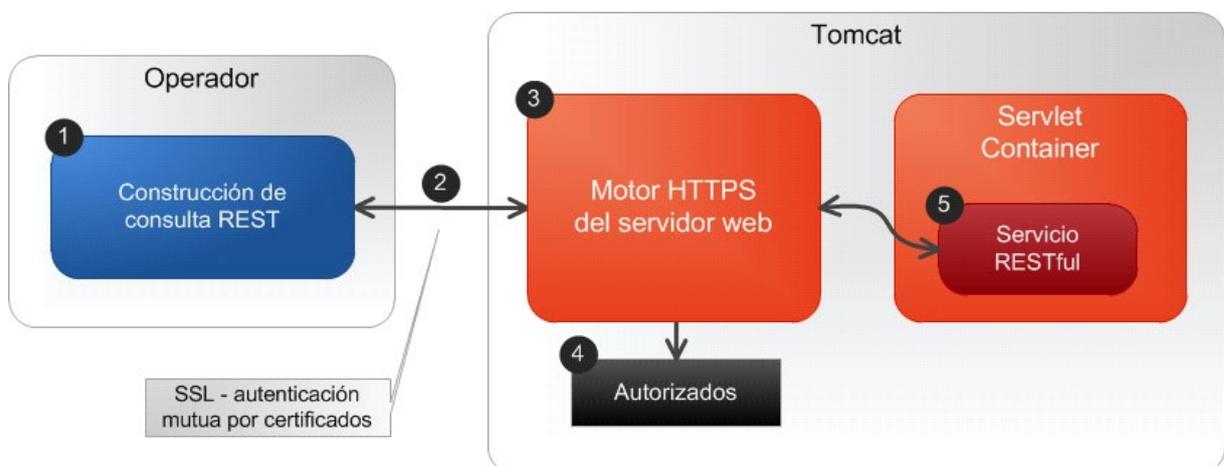
5.1 Objetivos

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 Mediante 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.

The screenshot shows a web browser window with the URL <https://sede.ordenacionjuego.gob.es/testoperadores/test/html/>. The page content includes a green checkmark icon and the text "Servicio VerificarJugador: **Correcto** (2013-11-21 12:40:22 CET)". To the right of this text is a "Limpiar" button. Below this, there is a section titled "Estado de los servicios" with two links: "Estado de VerificarJugador" and "Estado de VerificarRGIAJ".

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.

The screenshot shows a web browser window with the same URL as the previous screenshot. The page content includes a red 'X' icon and the text "ERR002: **Certificado no válido**". To the right of this text is a "Limpiar" button. Below this, there is a section titled "Estado de los servicios" with two links: "Estado de VerificarJugador" and "Estado de VerificarRGIAJ".



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>{"Code":"COD007"}</code>	Opción de consulta no valida
<code>{"Code":"COD008"}</code>	Opción no permitida
<code>{"Code":"ERR001"}</code>	Error interno del servicio
<code>{"Code":"ERR002"}</code>	Certificado no valido (no esta autorizado)
<code>{"Code":"COD006"}</code>	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 código fuente de utilización 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](#)