

URIs, funcionalidades y reglas de negocio

Versión	Autor	Fecha	Cambio
1.0	David Posada Mena, Álvaro Domínguez Núñez	28/11/17	Creación del primer documento
2.0	Álvaro Domínguez Núñez	30/11/17	Añadidas URIs de prevención (POST, PUT, DELETE)
3.0	David Posada Mena	03/12/17	Añadida aclaración de qué subgrupos no van a usar URIs hasta nuevo cambio.
4.0	Álvaro Domínguez Núñez	05/12/17	Simplificadas las URIs necesarias para cada subgrupo.
5.0	Miguel A. Baños Carretón	06/12/17	Añadida política de acceso a base de datos.

Política de acceso a base de datos

Con el objetivo de reducir la dependencia entre subsistemas en la medida de lo posible, se ha establecido que cualquier subsistema tiene permiso **DE LECTURA** sobre cualquier tabla. Esto aumentará la interacción con BBDD, pero hará que un fallo en un subsistema tenga menos impacto en el resto. Este permiso de lectura **no debe utilizarse sin tener en cuenta la lógica de negocio de la aplicación**, por lo que en los casos en los que la información deba ser filtrada por algún criterio, seguirá siendo necesario realizar una llamada a la API facilitada por el subsistema encargado de filtrar dicha información.

El permiso de escritura sobre tablas vendrá establecido en una nueva columna en la tabla de funcionalidades, **quedando prohibido la modificación en tablas sobre las que no se le haya concedido permiso**. De esta manera, los subsistemas son independientes a la hora de hacer consultas simples a BBDD. El uso de la API queda reservado a casos en los que un subsistema tiene que ofrecer funcionalidad de su competencia a otros subsistemas, esto incluye la obtención o persistencia de información de base de datos aplicando la lógica de negocio que corresponda.

URIs y funcionalidades

La siguiente tabla representa las funcionalidades de cada subsistema y las URI que generarán para consumo del resto de subsistemas, mediante datos en formato Json.

Estas URIs se irán ampliando en función de las solicitudes de cada grupo, por lo que un grupo que aparezca sin ninguna puede acabar teniendo alguna y viceversa.

Equipo	Funciones	URIs	Tablas con permiso de escritura
Aut	<ul style="list-style-type: none">Controlar el logado en el sistema de los diferentes usuarios.Proporcionar información de autenticación al resto de sistemas sobre quién está logado en el sistema y su rol.	<ul style="list-style-type: none">/login/session/{cookieID}: GET para obtener la información sobre la sesión de un usuario y su rol./login/login: GET para iniciar sesión en el sistema/login/logout: GET para cerrar sesión en el sistema.	<ul style="list-style-type: none">cookie
AdmV	<ul style="list-style-type: none">Crear, modificar, mostrar y eliminar votaciones y sus preguntas y asignarles un censo existente.Asegurarse de que los usuarios que hacen operaciones con las votaciones son los permitidos según el rol.Asegurarse de que los usuarios que participan en las votaciones son los permitidos según el censo.	No se les solicitará nada vía API hasta nuevo aviso.	<ul style="list-style-type: none">pollquestionquestion_option

AdmC	<ul style="list-style-type: none"> Crear, modificar, mostrar y eliminar los censos de una votación. Crear censos en función de la comunidad autónoma y el código postal. Crear un censo para un grupo de usuarios concreto. 	No se les solicitará nada vía API hasta nuevo aviso.	<ul style="list-style-type: none"> census user_account_per_census
CV	<ul style="list-style-type: none"> Asegurarse de que los usuarios que participan en las votaciones son los permitidos según el censo y el rol. Mostrar en una vista las preguntas que tengan las votaciones. Asegurarse de que se permita el voto dentro de la fecha especificada en la votación. 	No se les solicitará nada vía API hasta nuevo aviso.	No tienen permiso de escritura, dado que envían los votos a AlmV.
CT (GA)	<ul style="list-style-type: none"> Asegurarse de que los usuarios que participan en las votaciones son los permitidos según el censo y el rol. Pendiente de la integración con la gestión de registros. Mostrar en una vista las preguntas que tengan las votaciones. Asegurarse de que se permita el voto dentro de la fecha especificada en la votación. 	No se les solicitará nada vía API hasta nuevo aviso.	No tienen permiso de escritura, dado que envían los votos a AlmV.
CS (GA)	<ul style="list-style-type: none"> Asegurarse de que los usuarios que participan en las votaciones son los permitidos según el censo y el rol. Pendiente de la integración con la gestión de registros. Mostrar en una vista las preguntas que tengan las votaciones. Asegurarse de que se permita el voto dentro de la fecha especificada en la votación. 	No se les solicitará nada vía API hasta nuevo aviso.	No tienen permiso de escritura, dado que envían los votos a AlmV.
AlmV	<ul style="list-style-type: none"> Almacenamiento de los votos en la base de datos asignándolos a sus votaciones. Asegurarse de que el número de votos de una votación no supere el número de usuarios del censo. Cifrar los usuarios de los votos con el uso del token RSA. Asegurarse de que el votante utiliza únicamente un medio para votar. 	<ul style="list-style-type: none"> almvotes/vote: POST para añadir un voto a la base de datos (en caso de validar dicho voto). 	<ul style="list-style-type: none"> option_per_vote vote
RV	<ul style="list-style-type: none"> Asegurarse de que el usuario que pida información sobre el recuento de votos esté logado en el sistema. Obtener los resultados de una votación y proporcionárselos a VR. 	<ul style="list-style-type: none"> /recvotes/{votationID}: GET para obtener los datos de los votos de una votación. /recvotes/{voteID}: GET para obtener los datos de un voto dado. 	No tienen permiso de escritura, dado que tienen que desarrollar funcionalidad en base a consultas a la información existente.
VR	<ul style="list-style-type: none"> Mostrar los resultados de una votación en una vista. Asegurarse de que el usuario que pida información sobre los resultados esté logado en el sistema. Mostrar información estadística sobre los votos: Nº de votos, % de participación... Siempre que el usuario esté logado en el sistema. 	Renderiza los resultados y no se le solicita nada vía API hasta nuevo aviso.	No tienen permiso de escritura, dado que tienen que desarrollar funcionalidad en base a consultas a la información existente y a peticiones a la API de RV.
GP	<ul style="list-style-type: none"> Almacenar y añadir información referente a la programación del congreso. Pendiente de hacerlo vía Excel o vía Base de Datos. 	No se les solicitará nada vía API hasta nuevo aviso.	Pendiente la validación del modelo de datos requerido para darles permiso a las tablas correspondientes.

GR	<ul style="list-style-type: none"> • Generar un formulario para permitir el registro de los usuarios tanto en la página web como en el sistema de votación. Pendiente de confirmar la integración con slack y telegram. • Controlar el registro de los nuevos usuarios desde la página Wordpress. • Asegurarse del registro en la BD de los nuevos usuarios. • Permitir que un usuario sólo se registre en Wordpress y posteriormente en votaciones. 	<ul style="list-style-type: none"> • /greg/user: POST para añadir un nuevo usuario al sistema. 	<ul style="list-style-type: none"> • user • user_account <p>También tienen permiso de escritura sobre cualquier tabla de WordPress relacionada con la creación de usuarios.</p>
GGE	<ul style="list-style-type: none"> • Generar de forma estática el contenido del sistema Wordpress de la web del congreso. 	No se les solicitará nada vía API hasta nuevo aviso.	No necesitan persistir datos.
GVP	<ul style="list-style-type: none"> • Mostrar la información referente al programa del congreso. 	No se les solicitará nada vía API hasta nuevo aviso.	No necesitan persistir datos.
RRSS	<ul style="list-style-type: none"> • Permitir el acceso de los usuarios de la web del congreso a funcionalidades de diferentes redes sociales. • Mostrar un feed en la web del congreso de estas redes sociales. 	No se les solicitará nada vía API hasta nuevo aviso.	No necesitan persistir datos.

Reglas de negocio

Estas reglas de negocio han sido extraídas del modelo de la base de datos y de la tabla de funcionalidades de cada subsistema. Se irán ampliando en función de las solicitudes generadas por cada subgrupo y las identificaciones del equipo de integración.

Reglas de negocio:

1. Existen dos tipos de usuarios que pueden acceder al sistema:
 - a. Voter
 - b. Admin
2. Un usuario invitado podrá registrarse en el sistema. Para ello hará uso del sistema de registro de WordPress.
3. Un usuario con rol Voter puede:
 - a. Votar en una votación en caso de pertenecer al censo de la misma
 - b. Comprobar los resultados de una votación concreta
 - c. Hacer uso de todas las funcionalidades de WordPress
 - d. Consultar los resultados estadísticos de una votación concreta

4. Un usuario con rol Admin puede:
 - a. Realizar las mismas acciones que el usuario con rol Voter
 - b. Crear, modificar o eliminar una votación propia
 - c. Crear, modificar o eliminar un censo propio
5. Los votos deben tener un Token que identifique a su emisor. Este Token tiene que ser cifrado mediante RSA.
6. Debe existir un censo general de todos los usuarios en base a la comunidad autónoma de residencia. A partir de dicho censo generar se podrán generar otros más pequeños, ya sea filtrando por comunidad autónoma o eligiendo determinados usuarios en base a otros datos de interés.
7. Los cuestionarios tendrán cuatro modalidades de respuesta
 - a. Obligatoria y única
 - b. Obligatoria y múltiple
 - c. Opcional y única
 - d. Opcional y múltiple