

URIs, funcionalidades y reglas de negocio

Versión	Autor	Fecha
1.0	David Posada Mena, Álvaro Domínguez Núñez	28/11/17
2.0	Álvaro Domínguez Núñez	30/11/17
3.0	David Posada Mena	03/12/17

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

Los grupos marcados con GA deben ponerse en contacto con su grupo de apoyo o en su defecto con el equipo de integración para solucionar algunas inconsistencias de sus funcionalidades.

Equipo	Funciones	URIs
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.
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.	<ul style="list-style-type: none">/admvotes/votation/{votationID}: GET para obtener los datos de una votación, PUT para actualizarla, DELETE para borrarla./admvotes/votation: POST para crear la votación./admvotes/question/{votationID}: Post para crear una pregunta para una votación.
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.	<ul style="list-style-type: none">/admcenses/cense/{censeID}: GET para obtener los datos de un censo, PUT para actualizarlo, DELETE para borrarlo./admcenses/cense: POST para crear un censo.
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.	<ul style="list-style-type: none">/cabin/{votationID}: GET con los datos de las preguntas de una votación./cabin/{QuestionID}: GET con los datos de una pregunta dada.
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	<ul style="list-style-type: none">/cabin/{votationID}: GET con los datos de las preguntas de una votación./cabin/{QuestionID}: GET con los datos de una pregunta dada.

	<ul style="list-style-type: none"> 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. 	
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. 	<ul style="list-style-type: none"> /cabin/{votationID}: GET con los datos de las preguntas de una votación. /cabin/{QuestionID}: GET con los datos de una pregunta dada.
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.
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.
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 via API en un principio
GP (GA)	<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. 	Pendiente de discusión
GR (GA)	<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.
GGE	<ul style="list-style-type: none"> Generar de forma estática el contenido del sistema Wordpress de la web del congreso. 	No se le solicitará nada vía API en principio.
GVP (GA)	<ul style="list-style-type: none"> Mostrar la información referente al programa del congreso. 	Pendiente de discusión

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 le solicitará nada vía API en principio.
------	--	--

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:

- Existen dos tipos de usuarios que pueden acceder al sistema:
 - Voter
 - Admin
- Un usuario invitado podrá registrarse en el sistema. Para ello hará uso del sistema de registro de WordPress.
- Un usuario con rol Voter puede:
 - Votar en una votación en caso de pertenecer al censo de la misma
 - Comprobar los resultados de una votación concreta
 - Hacer uso de todas las funcionalidades de WordPress
 - Consultar los resultados estadísticos de una votación concreta
- Un usuario con rol Admin puede:
 - Realizar las mismas acciones que el usuario con rol Voter
 - Crear, modificar o eliminar una votación propia
 - Crear, modificar o eliminar un censo propio
- Los votos deben tener un Token que identifique a su emisor. Este Token tiene que ser cifrado mediante RSA.
- 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