

2017/
18

Documento Explicativo del Proyecto EGC

En este documento indica todas las ideas básicas para el desarrollo del proyecto de la asignatura Evolución y Gestión de la Configuración durante el curso 2017/2018



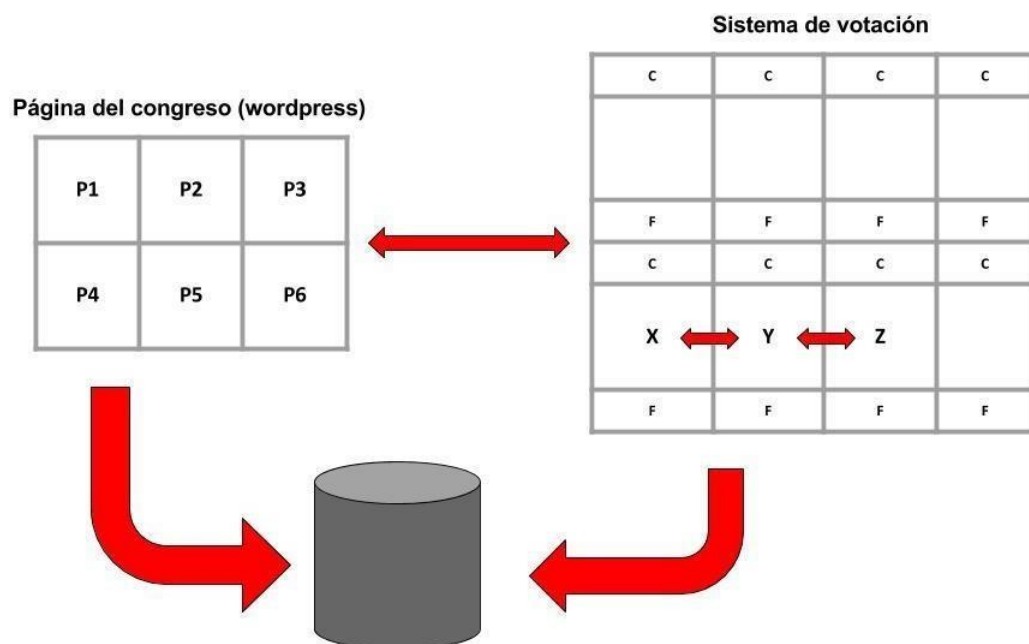
Índice de Contenido

1.	Diagrama de la arquitectura del sistema y explicación del mismo	3
2.	Tabla de relación entre subsistemas	5
3.	Tabla de funciones de subsistemas.....	6
4.	Directrices de gestión de cambios, código e incidencias	7
5.	Directrices de gestión de dudas y documentación	8
6.	Directrices de gestión de ramas.....	9
7.	URIs y puertos asignados a cada subsistema	9
8.	Planificación temporal y explicación de la misma.....	10

Control de versiones

Versión	Autor	Fecha	Contenido
1.0	Álvaro Domínguez Núñez, David Posada Mena	25/11/17	Primera versión del documento.
2.0	Álvaro Domínguez Núñez, David Posada Mena	28/11/17	Corregidas las URIs de partida
3.0	Álvaro Domínguez Núñez	07/12/17	Actualizados los milestones y añadido sistema de gestión de dudas y documentos.
4.0	Álvaro Domínguez Núñez	12/12/17	Añadida aclaración sobre los subgrupos en el Milestone 3.

1. Diagrama de la arquitectura del sistema y explicación del mismo



El sistema de votación se compone de una serie de subsistemas que se comunicarán entre sí ya que para realizar su trabajo en principio todos deben comunicarse con otros (uno o más). Se proporcionará una plantilla HTML (Cabecera MasterPage y Footer) quedando a cargo de los mismos la elaboración del contenido dinámico, respetando el estilo de la plantilla suministrada.

La página del congreso se complementará con plugins realizados en Wordpress (p1-p6) respetando el estilo del sistema Wordpress ya existente.

Los subsistemas (sistema de votación) se comunicarán “por debajo” con una base de datos MySQL. En cambio “por arriba” se comunicarán con los otros subsistemas mediante API Rest que generará los datos en formato Json.

Ambos sistemas (página del congreso y sistema de votación) se comunicarán con una base de datos para la que se facilitarán credenciales propias a cada subsistema. El SGBD es único, pero a nivel lógico habrá una base de datos para cada sistema (WP y Sistema de votación).

No obstante, se recomienda que los subsistemas sean capaces de comunicarse con una base de datos MySQL en local a fin de comprobar su correcto funcionamiento.

Los subsistemas que trabajarán en el sistema de votaciones se dividirán en dos grupos:

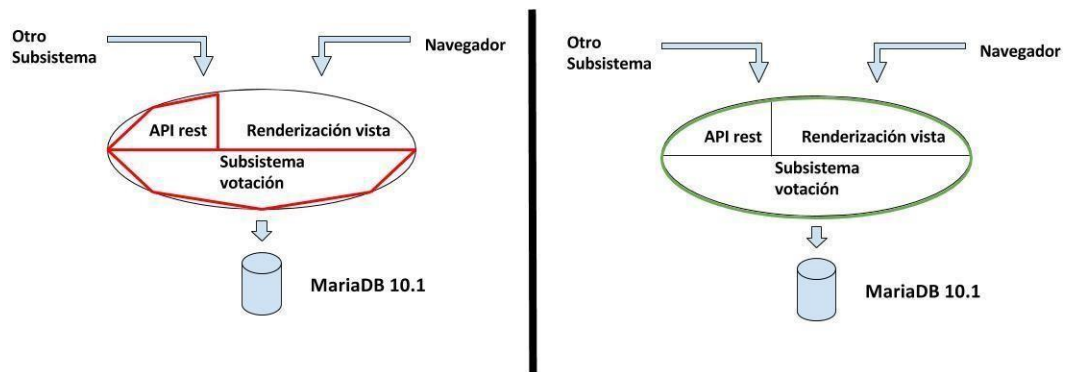
- Grupo 1 (Verde): Estos subsistemas renderizarán vistas mostrando el contenido que van a desarrollar durante el proyecto.
- Grupo 2 (Rojo): Estos subsistemas ofrecerán servicios en forma de API a partir de lo desarrollado por ellos. Gráficamente:

Subsistemas que renderizan vistas

#1	Login	Admin. Votaciones	Admin. Censos	Cabina Votaciones	Visual. Resultados

Subsistemas que generan Servicios

#2	Cabina Telegram	Cabina Slack	Almacen votos	Recuento votos



2. Tabla de relación entre subsistemas

Identificando a cada subsistema de la siguiente forma, en la tabla que aparece posteriormente se indican las relaciones entre unos y otros de ellos, marcando con una X si debe existir colaboración entre cada par. Dichas relaciones son opcionales para el desarrollo de las funciones de cada subsistema:

- Autenticación: Aut.
- Administración de votaciones: Adm.V.
- Administración de censos: Adm.C.
- Cabina de votaciones: C.V.
- Cabina de Telegram: C.T.
- Cabina de Slack: C.S.
- Almacenamiento de votos: Alm.V.
- Recuento de votos: R.V.
- Visualización de resultados: V.R.
- Deliberaciones: D.
- Gestión del Programa: G.P.
- Gestión del Registro: G.R.
- Gestión de Generación Estática del Sitio Web: G.G.E.
- Gestión de Visualización del Programa: G.V.P.
- Gestión de Integración con RRSS: RRSS
- Gestión de Comités: G.C.

[illegible]

3. Tabla de funciones de subsistemas

Las funciones de cada subsistema son las siguientes, sujetas a ampliación y modificación:

Equipo	Funciones
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.
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.
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.
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.
CT	<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.
CS	<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.
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.
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.
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.
GP	<ul style="list-style-type: none"> • Almacenar y añadir información referente a la programación del congreso.
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. • 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.
GGE	<ul style="list-style-type: none"> • Generar de forma estática el contenido del sistema Wordpress de la web del congreso.
GVP	<ul style="list-style-type: none"> • Mostrar la información referente al programa del congreso.
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.

Es imperativa la coordinación entre los distintos subsistemas en cuanto a cuestiones de seguridad (cifrado, autenticación etcétera).

4. Directrices de gestión de cambios, código e incidencias

Para llevar a cabo las tareas de gestión de cambios, código e incidencias, se hará uso de la plataforma GitHub, siguiendo las directrices expuestas en clase:

- Para gestionar el código, se harán commits y push en dichos repositorios siguiendo las pautas especificadas en las diapositivas de clase durante la explicación de la gestión del código fuente (http://1984.lsi.us.es/wiki-egc/index.php/Teor%C3%ADa_-_17/18). A partir de las versiones finales desarrolladas por cada grupo en cada iteración, los integrantes del equipo de Integración indicarán una semana antes de cada milestone si es necesario subir con antelación el avance funcional hecho por cada subgrupo al repositorio del Equipo de Arquitectura e Integración para disponer del código hecho.

Los commits en local se harán con cada incremento de subversión, mientras que en caso de ser requerido por el equipo de integración en su repositorio, se harán dentro de éste cada número de versión. El número de versionado y su nomenclatura se explican en el punto siguiente.

- Para la gestión de las incidencias, se usarán los Issues que GitHub permite gestionar en cada repositorio. El uso de estos Issues está basado en las diapositivas de clase de Gestión de Incidencias (http://1984.lsi.us.es/wiki-egc/index.php/Teor%C3%ADa_-_17/18), haciéndose de la siguiente manera:
 - Cada grupo es responsable de la gestión de sus Issues.
 - Cada Issue representará una tarea o problema encontrado por el grupo.
 - Cada Issue será situado en un proyecto interno al repositorio, siendo actualizado entre las fases TO DO, En progreso, En espera/Con problemas, En revisión y Hecho según corresponda a su avance.
 - Para cada avance o actualización de un Issue, se hará un comentario dentro de éste especificando el cambio hecho, indicando su motivo si procede.
 - Una vez terminado un Issue se debe cerrar con un comentario de cierre y cerrándolo en sí mismo tras colocarlo en la columna Hecho de su proyecto correspondiente.
 - Cada Issue será etiquetado en función de su temática, prioridad, estado y tipo. Los tipos son Cambio/Mejora y Bug, que sólo se incluirán en dichos casos. Las temáticas son libres para cada subgrupo. Las prioridades son Critical, High, Medium y Low, y los estados son New, Accepted (sólo para cambios), Started, Fixed, Verified, Duplicate (cambios y bugs) y Wontfix (cambios y bugs).
 - Las etiquetas de cada Issue deben ser actualizadas durante el avance del mismo, con su correspondiente actualización en el proyecto y su comentario pertinente.

- Para la gestión de cambios, se procederá de la siguiente manera:
 1. Cada grupo que desee plantear un cambio, deberá plantearlo antes dentro de su grupo.
 2. Si se rechaza termina el proceso, y si se acepta se traslada dicho cambio al Equipo de Integración.
 3. El Equipo de Integración valorará dicho cambio. Si lo rechaza el cambio no se aplicará, y si lo acepta se comunicará a los subgrupos de dicho cambio y se actualizarán los elementos de documentación pertinentes.

5. Directrices de gestión de dudas y documentación

El procedimiento a seguir para plantear y resolver dudas al equipo de integración es el siguiente:

1. Plantear la duda dentro del grupo de trabajo.
2. Contactar con el equipo de integración para indicarle la duda.
3. El equipo de integración creará el issue en su repositorio indicando la duda o incidencia.
4. El equipo de integración asignará a alguien para el seguimiento de dicha duda y su resolución junto con el equipo que la ha planteado.

Respecto al desarrollo de documentos, éstos seguirán el siguiente formato:

- Si se tratan de documentos cuyo contenido sea código, la nomenclatura del archivo subido será: NombreSubgrupo-versión. Las versiones se incrementarán cada vez que el subsistema reciba una nueva funcionalidad completa (por ejemplo el proceso de creación completo de una votación). Las subversiones se incrementarán cada vez que se añada una nueva parte a un proceso funcional completo (por ejemplo añadir el formulario de creación de una votación), y los números de parche se incrementarán cada vez que se solucione un bug dentro de dicha subversión.
- En el caso de documentación de otro tipo, se seguirá el esquema NombreSubgrupo-NombreDocumento, con el fin de aclarar el contenido de dicho archivo.

Independientemente del tipo de documento, a la hora de subirlo al repositorio del Equipo de Integración, deberá hacerse en la carpeta especificada por éste cuando se solicite el documento.

6. Directrices de gestión de ramas

Cada grupo debe tener en su repositorio una rama principal, a partir de la cual saldrán las siguientes ramas secundarias opcionales:

- Una de pruebas, en la que se introducirán todas las funcionalidades que estén diseñadas pero aún no se hayan probado ni validado, procediendo posteriormente a corregirlas en caso de fallo.
- Una de documentación, en la que se incluirán todos los documentos de texto del grupo.
- Ramas personales de cada miembro del grupo, según estime el coordinador de éste.

Es responsabilidad del coordinador del grupo tanto el decidir qué ramas opcionales usar, si expandirlas con más de ellas, y hacer merge de las mismas cuando se complete cada ciclo del milestone si lo considera oportuno.

7. URIs y puertos asignados a cada subsistema

Esta tabla presenta los puertos sobre los que tendrá que trabajar cada subsistema dentro del contexto del sistema de votaciones, así como el dominio principal de las URIs que van a tener que desarrollar.

Equipo	Puerto asignado	URIs
Autenticación	52000	/login
Administración de votaciones	52001	/admvotes
Administración de censos	52002	/admcenses
Cabina de votaciones	52003	/cabin
Cabina de telegram	52004	/cabint
Cabina de slack	52005	/cabins
Almacenamiento de votos	52006	/almvotes
Recuento de votos	52007	/recvotes
Visualización de resultados	52008	/visres
Gestión de registro	52009	/greg

8. Planificación temporal y explicación de la misma

Tipo de Equipo	Milestone			
	1: Ecosistemas preparados (23/11/17)	2: Sistema funcionando con incremento (14/12/17)	3: Taller de automatización (21/12/17)	4: Entrega y defensa de los trabajos (18/01/18)
Integración	Planificación de desarrollo del proyecto y arquitectura.	Base de datos operativa, reparto de URIs y puertos. Reparto de responsabilidades.	Integración entre al menos dos subsistemas diferentes.	Integración de todos los sistemas posibles.
Equipo de trabajo del sistema de votación	Entorno de trabajo propio listo.	Subsistema funcionando en local con al menos una funcionalidad del incremento funcional.	Sistema funcionando sin errores en un contenedor docker configurado para integrar con todas las funcionalidades posibles.	Sistema funcionando sin errores en un contenedor docker configurado para integrar con todas las funcionalidades posibles.
Equipo de trabajo del portal del congreso	Entorno de trabajo propio listo.	Plugin/Extensión/ Visualización funcionando en local.	Plugin/Extensión/Visualización completa en el wordpress del servidor.	Plugin/ Extensión/ Visualización completa en el wordpress del servidor.

Para cada milestone, distinguiremos entre los equipos que van a trabajar desarrollando los diferentes subsistemas del sistema de votación, el equipo de integración y los equipos que trabajarán con la web del congreso:

- En el caso del Equipo de Integración, la primera meta en el primer milestone consiste en crear una planificación del desarrollo del proyecto, junto con una descripción de la arquitectura y un esquema de la misma. A continuación en el segundo milestone deberá tener operativa la Base de Datos para que el resto de equipos pueda trabajar con ella y acceder a su contenido, además de indicarles a éstos las URIs que serán de su competencia y a las que deberán responder con un JSON al desarrollar la lógica de negocio de sus APIs. También identificará de forma clara las distintas responsabilidades de cada subgrupo, así como sus relaciones con el resto. Respecto al tercer milestone, el equipo se asegurará de integrar al menos dos subsistemas en el servidor pertinente para que funcionen adecuadamente, trabajo ampliable hasta el cuarto milestone en el que deberá integrar tantos subsistemas como sea posible.

- Respecto a los equipos que trabajen con el sistema de votación, para el primer milestone deberán tener elegida la tecnología y el entorno de trabajo a usar, siendo de elección libre, y disponer de los elementos necesarios para usarlo preparados en sus equipos. Para el segundo milestone, y siguiendo las directrices de las URIs entregadas por el equipo de integración, el correspondiente equipo deberá desarrollar de forma funcional en local al menos una de las funcionalidades asignadas a su subsistema. Esta tarea continuará desarrollándose en los siguientes milestones, llegando a incorporar tantas funcionalidades como sean posibles. En el tercer milestone, además de lo indicado, **se deberá incluir el subsistema en un contenedor de docker optimizado para su integración, siguiendo las especificaciones vistas en prácticas y las dadas por el Equipo de Integración.** Este cometido deberá seguirse en el último milestone, ya que al añadir nuevas funcionalidades será necesario que se actualice dicho contenedor.
- En cuanto a los equipos que trabajarán en la web del congreso, deberán tener listo en el primer milestone la tecnología a usar por ellos junto con el entorno en el que llevarán a cabo su proceso de desarrollo. Para el segundo milestone deberán tener su trabajo con al menos una de las funcionalidades trabajando correctamente en un entorno local, lo cual se expandirá al siguiente milestone, el tercero, en el que deberán asegurarse de la visualización completa de los elementos desarrollados en la página wordpress principal. Esto es extensible al último milestone, en el que al ir añadiendo más funcionalidades deberán también asegurarse de que están correctamente implantadas en la página final.