

# Certamen de Proyectos Libres UGR – OSL 2016

De un total de 35 proyectos presentados (<http://osl.ugr.es/2016/03/01/equipos-inscritos-en-el-certamen-de-proyectos-libres-de-la-ugr/>) solamente 7 han resultado finalista y han pasado a la ronda final (<http://osl.ugr.es/2016/05/10/finalistas-del-certamen-de-proyectos-libres-de-la-ugr/>) quedando descalificado el proyecto Node OS por carecer de licencia. La licencia, recordamos debe de estar en el raíz del repositorio dentro del fichero LICENCE

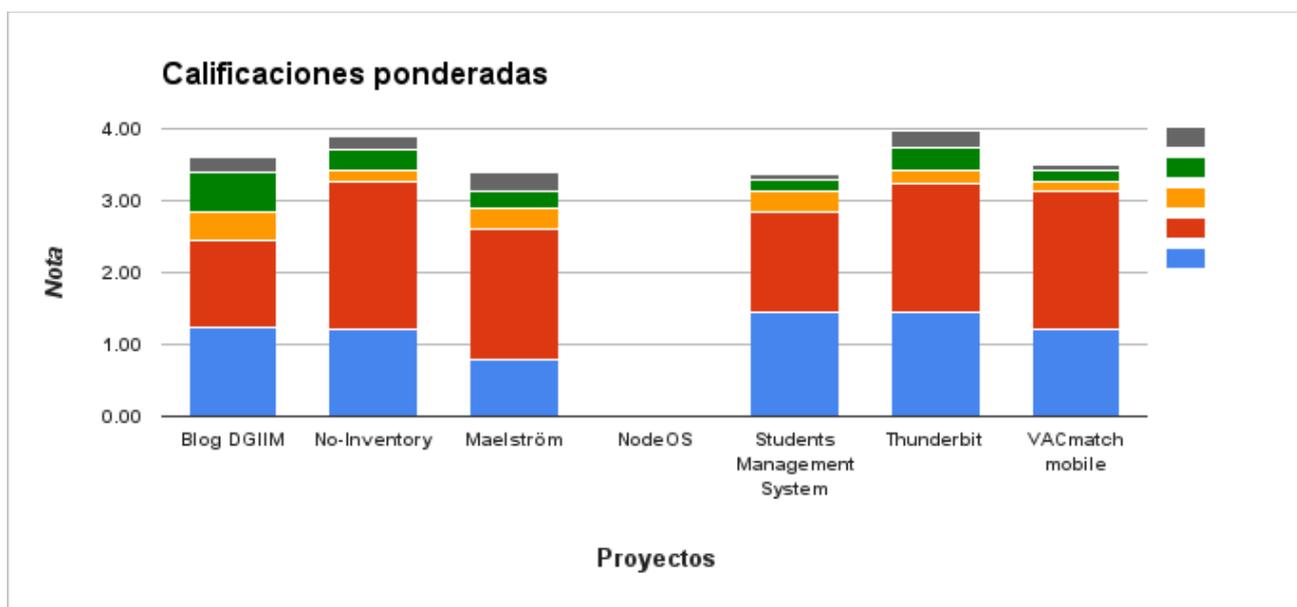
Los proyectos SMS y VACmatch mobile no se han podido probar por lo que la usabilidad no se ha puntuado.

Los proyectos No-Inventory y Maeströn sí se han podido llegar a ejecutar en local, sin embargo la ayuda ofrecida en los propios proyectos para llevar esta labor a cabo estaba incompleta.

Los proyectos, Blog DGIIM, ThunderBit y No-Inventory si disponían de una copia online.

Se ha valorado positivamente este aspecto, al igual que aquellos que se han presentado al hackaton.

Se recomienda para futuras ediciones que exista una copia online funcional de prueba para poder evaluar estos aspectos de los proyectos.



Items	Peso	Blog DGIIIM	No-Inventory	Maelström	NodeOS	Students Management System	Thunderbit	VACmatch mobile
Valor Social	0,3	1,25	1,20	0,80	0,00	1,45	1,45	1,20
Tecnologías usadas y calidad del desarrollo	0,4	1,20	2,07	1,80	0,00	1,40	1,80	1,93
Grado de aceptación por parte de la comunidad de software libre	0,1	0,40	0,15	0,28	0,00	0,28	0,18	0,12
Grado de finalización	0,1	0,55	0,28	0,26	0,00	0,15	0,30	0,18
Accesibilidad y Usabilidad	0,1	0,20	0,20	0,25	0,00	0,08	0,25	0,07
<b>Total</b>	<b>1,</b>	<b>3,60</b>	<b>3,90</b>	<b>3,39</b>	<b>0,00</b>	<b>3,37</b>	<b>3,98</b>	<b>3,50</b>

Se ha decidido otorgar el primer premio al proyecto ThunderBit, y el segundo premio al proyecto No-Inventory.

Como siempre **agradecer a todos los inscritos** su participación y la dedicación y el tiempo que han prestado a este certamen.

Así mismo **agradecemos al jurado**, Carmen López, Isabel Cabeza, Paloma de las Cuevas y María Hinojosa la buena disposición así como el tiempo que han prestado a las correcciones que nos consta han sido en profundidad.

En este documento se muestran los razonamientos que han llevado al jurado a tomar su decisión así como posibles mejoras para que los participantes puedan aplicarlas si lo desean a sus proyectos.

Si se desea se pueden consultar las bases del certamen aquí:

<http://osl.ugr.es/bases-de-los-premios-a-proyectos-libres-de-la-ugr/>

**Agradecemos también a los patrocinadores** [Delegación de la Rectora para la Universidad Digital](#), [Codeko.com](#), [OpenXXI](#), [Documedia](#), [EasyName](#), [ESRI](#) y Bitergia. su colaboración, esencial para que este certamen siga produciéndose.

Por ultimo, y no menos importante **agradecerle la Oficina de Software Libre de la Universidad de Granada** la organización de este certamen y su labor en la difusión y apoyo al Software Libre.

A continuación detallaremos los proyectos y la evaluación de cada uno.

## Blog DGIIM

“El blog del DGIIM es un sitio web colaborativo realizado con tecnologías libres y pensado para acoger contenidos libres de distinto tipo, como documentación y artículos técnicos, guías, entrevistas, relaciones de problemas y soluciones; todo relacionado con la informática y las matemáticas y sus disciplinas. Funciona sobre el generador de sitios estáticos Jekyll y el framework responsive Pure.css, y todo el contenido se escribe en formatos abiertos: Markdown y LaTeX. El blog está actualmente en funcionamiento y tiene algo de contenido pero queremos aumentar la variedad de publicaciones y realizar mejoras técnicas para facilitar las contribuciones.”

---

Con respecto a este proyecto se ha evaluado lo desarrollado entre el día 15 de Noviembre de 2015 hasta el 31 de Mayo de 2016

Que se ajusten a los criterios de evaluación hay pocos commits, siendo la mayoría pequeños ajustes (por ejemplo añadir un enlace, correcciones pequeñas..).

De los commits evaluables, que son pocos dentro del periodo del concurso, muchos son para el post y la mayoría apenas cambian una línea.

En cuanto a infraestructura han avanzado en la incorporación de la posibilidad de colaborar externamente con posts o correcciones.

Podemos decir entonces que el grueso de este proyecto se ha llevado a cabo antes del inicio del concurso.

Se usa Ruby y Jekyll para el despliegue, y HTML y CSS para la página web, el código es limpio y entendible.

Sobre el fichero principal que construye el blog “Rakefile” se han efectuado dos cambios que serian evaluables, el 22 de Diciembre de 2015 se quitó una línea y se modificó otra, y el 13 de febrero que se añadió un espacio en blanco.

Consideramos que es una iniciativa interesante y ayuda también a los compañeros, que además pueden colaborar.

Sin embargo para ser un proyecto en activo desde desde 2014, consideramos

que se ha creado poca comunidad, mas aun dada la naturaleza del proyecto, quizá podrían haberse puesto en contacto con estudiantes de doble grado de otras ciudades, por ejemplo, para que la red se extendiese.

Con respecto al contenido, un sólo post entraría dentro del contenido a evaluar, aunque se ve que tienen algunos en proyecto ya que hay muchos temas propuestos en los issues, con lo que reiteramos el consejo de contactar con otros estudiantes para ampliar la colaboración.

Como cosas nuevas, han metido la parte colaborativa de escritura de posts, además de la revisión, en la que también se puede colaborar.

La interfaz del blog es responsiva, y los posts son fáciles de leer, pero el menú es un poco pequeño, recomendamos mejorar este aspecto para hacerlo mas accesible.

Otra sugerencia es que incluyan el README en inglés, o la posibilidad de poner los posts en inglés o en inglés y español, para abrir la posibilidad de colaboración a estudiantes de otros países.

Si bien es un proyecto curioso quizá la tecnología utilizada no sea la mas adecuada para llevar a cabo un blog colaborativo, no obstante aunque puede que no sea un proyecto que tenga una gran cantidad de código es una buena idea que necesita terminar de arrancar y expandirse.

# No Inventory

“Se trata de realizar la infraestructura virtual necesaria para levantar una aplicación web que se encargue de controlar y automatizar la tarea de administrar y controlar la entrega y recogida de equipos informáticos por la Oficina de Software Libre así como el proceso de catalogarlos adecuadamente y generar informes.”

---

Aunque hay herramientas similares en las que podría haber colaborado en lugar de hacer un sistema de gestión de inventarios propio entendemos que no haya sido así al tratarse de un TFG.

El desarrollo es muy completo, limpio, con uso de varios lenguajes y tecnologías, además de hacer despliegue en la nube y de incorporar servicios externos a través de APIs.

La calidad del código aunque es buena, tiene algunos aspectos mejorables, por ejemplo hay procesos repetitivos

```
lista_tag1=manejadorClasificacion.database.tag1.find({"organizacion":organizacion}).sort([("CLAVE1", 1)])
lista_tag2=manejadorClasificacion.database.tag2.find({"organizacion":organizacion}).sort([("CLAVE2", 1)])
lista_tag3=manejadorClasificacion.database.tag3.find({"organizacion":organizacion}).sort([("CLAVE3", 1)])
#print "formulario"
#print lista_tag2[0]["VALOR2"]
self.fields['tag1'] = forms.ChoiceField(label='', choices=[(x["VALOR1"], x["VALOR1"]) for x in lista_tag1])
self.fields['tag2'] = forms.ChoiceField(label='', choices=[(x["VALOR2"], x["VALOR2"]) for x in lista_tag2])
self.fields['tag3'] = forms.ChoiceField(label='', choices=[(x["VALOR3"], x["VALOR3"]) for x in lista_tag3])
```

Que se podrían solucionar con un bucle, constantes iguales repetidas

```
TIPO = (('Publico', 'Publico'),('Privado', 'Privado'),)
```

Que pueden complicarle futuras actualizaciones.

En general recomendamos reutilizar el máximo código posible, y utilizar decorators para validaciones repetitivas y frecuentes de parámetros.

Se ha conseguido desplegar la aplicación en local, sin embargo no se ha encontrado documentación respecto a los usuarios por defecto, o cualquier otro dato sobre el funcionamiento de la aplicación, así que es complicado de testear, además da fallos al crear items.

Por otra parte se ha intentado descargar la APK desde el link provisto en el README pero aparece como eliminado de Dropbox de manera que no se ha podido probar.

Aunque el proyecto no ha creado comunidad se valora que se presentó al hackaton, en el momento de la corrección se ha visto que se ha actualizado el blog creando una presentación, aunque esta no entraría dentro de lo evaluable al estar fuera de plazo.

Por otra parte la demo que se encuentra desplegada en Heroku no funciona del todo bien, los desplegables no están rellenos, no se muestran mensajes de error o de tarea completada.

Seria recomendable por una parte mejorar la documentación para ayudar a la hora de constituir una comunidad en torno al proyecto.

Por otra parte de cara a usabilidad seria bueno también una correcta implementación de los mensajes de error.

# Maelström

“Desarrollo de un framework para videojuegos de estrategia económica MMO (Massively Multiplayer Online) con arquitectura cliente-servidor y videojuego aplicado. El subgénero sería “MMO-RTS” (MMO de estrategia en tiempo real). Será desarrollado con arquitectura de microservicios en NodeJS (entre otras tecnologías). Licencia GPL Affero”

---

Sería recomendable ubicar todos los ficheros que componen la aplicación en un mismo repositorio, dentro de carpetas diferentes por ejemplo.

La estructura usada podría tener sentido en el caso de una aplicación cliente - servidor para separar ambos, pero en este caso carece de valor.

○ si se pretende implementar esa estructura como microservicios hacerlo con microservicios que realmente vayan a tener mucha carga, lo cual, no es el caso.

En cuanto a la documentación del proyecto los README de los distintos repositorios solo indican cómo lanzar los distintos servicios pero el blog no tiene ningún post.

El hecho de que todo lo que hay implementado, los issues, o la poca documentación que hay (READMEs) esté en inglés ayuda a dar visibilidad al proyecto en la comunidad de SL, pero hace falta más información y más movimiento en el blog, redes sociales...

Con respecto al código éste está poco documentado, sería recomendable mejorar este aspecto si se desea crear comunidad, aunque es limpio y legible.

Se recomienda el uso de constantes en sentencias como estas, para evitar en la medida de lo posible comparaciones con cadenas

```
if (this.status.value !== "docked")
○
thisShip.setStatus("traveling", {
    remaining: time,
    destiny: destiny
});
if (this.status.value === "traveling") {
```

De esta manera no tendremos problemas con modificaciones futuras.  
De los cuatro módulos que consta el proyecto hay terminado uno.

También recomendamos mejorar los mensajes de error (por ejemplo obliga a que las clave cumplan unas condiciones que no indica)

Así mismo en la creación de barcos, no vendría mal informar de la velocidad de cada tipo de nave, ya que según ship.js es lo que se utiliza para calcular el tiempo que queda hasta llegar al puerto elegido.

Suponemos que la parte del mapa es para mostrar cómo va el viaje o los puertos visitados pero ahora mismo no hemos visto la funcionalidad.

La lógica base del juego es mejorable, en el momento en que compras puedes vender en el mismo puerto ya que se aplica un incremento del doble valor por escasez, vuelves a recomprar mas barato por que hay abundancia, y repites la operación.

Quizá sería bueno que tuviera en cuenta el puerto donde se desarrolla la operación.

Se valora que se presentó al hackaton.

# StudentsManagementSystem

“SMS es un sistema para centros docentes que agiliza y mejora la gestión de estudiantes, haciéndola más simple y eficiente. Las tecnologías con las que se desarrollará serán Google App Engine aunque construiremos una versión para hosts privados, usando frameworks como webapp2 y kit de css como UIKit para el diseño. Actualmente se encuentra en un 5% de desarrollo aproximadamente y esperamos tenerlo para la final lo suficientemente avanzado como para ponerlo en producción.”

---

No se ha conseguido echar a andar el sistema en local.

Obliga a tener una clave de root concreta y además al final lanza un error que no conseguimos solucionar ninguno de los jurados.

Por otra parte los test dan fallos incluso ejecutando *requirements\_bash.sh*

En cuanto a la aplicación la documentación es muy pobre, nos encontramos carpetas como “microservicio1”, “microservicio2” pero apenas se explica los casos de uso.

En cuanto al código hay cierta documentación, sin embargo no está muy bien organizado.

Con respecto al código:

En cada método hay código como este:

```
if v:
    print nombreMicroservicio
    print 'Llamando a /controlesAsistencia GET getAllControlesAsistencia()'

if v:
    print nombreMicroservicio
    print ' Llamando a /controlesAsistencia GET getAllControlesAsistencia()'
    print ' Salida: '
    print str(listaCAs)
```

Todo esto debería implementarse mediante decorators + reflection, no teniendo que añadir el mismo código junto con cadenas en cada función del API.

Las funciones demasiado largas, deberían de reducirse para eliminar complejidad.

Cuando se realizan llamadas a grandes listados de datos en el API no se dispone de sistema de paginación, la solución de añadir un límite muy grande no es elegante

```
def obtenerALLCA(self):
    ...

    Devuelve una lista completa con todos los controles de asistencia.
    ...

    listaCA = []
    listaCA = ControlAsistencia.devolver_todo().fetch(100000)

    return listaCA
```

No se respeta la notación Camel Case como en el ejemplo anterior, que debería ser obtenerAllCA.

Tampoco hay uniformidad en los nombres, en este caso se le llama CA y en otros se les llama ControlAsistencia. Debería usarse una nomenclatura uniforme.

En otros casos se mezcla nomenclatura camel case con separada por guiones:

```
def insertarResumenControlAsistencia(self, listaIdCA, fechaHora, idProfesor, idASignatura, idClase):
    ...

    nuevoRCA.lista_idCA=listaIdCA
    nuevoRCA.fecha_hora=fechaHora
    nuevoRCA.id_profesor=idProfesor
```

Para la creación de objetos se recomienda el uso de constructores

```
profesor = Profesor(id,nombre)
```

en vez de

```
profesor = Profesor()
profesor.idProfesor=int(idProfesor)
profesor.nombreProfesor=nombreProfesor
```

Vemos que todo la lógica de la API recae sobre una única clase Gestor que

tiene toda la funcionalidad completa. Se recomienda modularlo.

Aunque vemos que el código está bien comentado, la lógica, nomenclatura, estructura de ficheros y clases y en general todo el código es mejorable, consideramos que es una buena idea sobre la que es recomendable trabajar para mejorar los aspectos reseñados

No se ha podido evaluar la accesibilidad al no poder ver el proyecto funcionando, se valora que se presentasen al hackaton.

# ThunderBit

“Nuestro proyecto es una aplicación web para el intercambio de ficheros en instituciones, como alternativa a los clásicos directorios compartidos mediante protocolo FTP. Nuestra aplicación permitirá organizar los ficheros en una estructura plana mediante etiquetas, lo que facilitará su búsqueda. Además, ofrecerá facilidades para que sus usuarios se mantengan informados de la incorporación de nuevos ficheros que puedan interesarles, por ejemplo mediante la suscripción a notificaciones por etiqueta. Utilizaremos el popular framework para desarrollo web Play! de Typesafe, que integra Java, Scala, SBT, IVY, LESS y otras tecnologías. Haremos Integración Continua con drone.io y durante el concurso mantendremos corriendo una instancia de la aplicación en Heroku para que los usuarios potenciales puedan probarla sin tener que instalarla. Cuando tengamos una versión lista para usuarios finales la publicaremos en Bintray. El proyecto actualmente es un puñado de ideas y sticky notes en el scrumboard.”

---

Con respecto a la aplicación es una buena idea aunque vemos que aun faltan por desarrollar funcionalidades.

En la aplicación que hemos podido ver desplegada en Heroku comprobamos que una vez agregadas las etiquetas no es posible modificarlas, siendo el taggeo el punto fuerte de la aplicación sería importante mejorar este aspecto de cara a un futuro.

Así mismo si se elimina un fichero tras realizar la búsqueda por etiquetas y solamente haber aparecido ese fichero, efectivamente se elimina pero se queda la pantalla en blanco. Es lo mismo que si se busca una etiqueta que no existe.

Sugerimos añadir un mensaje tipo

"No hay ficheros con estos criterios de búsqueda".

Vemos también que el buscador sólo permite buscar por tags, no por nombre, y que esta búsqueda sólo permite que si escribes el nombre de un tag solo puedes seleccionarlo del desplegable. Quizá permitir seleccionarlo pulsando enter sería una buena mejora a este respecto

En cuanto a la lógica del programa vemos que al ir añadiendo tags a la

búsqueda no se muestran los tags de los resultados actuales sino de toda la BD con lo que es poco útil y va en contra de la funcionalidad que comenta en la presentación.

Actualmente no hemos encontrado utilidad a los ya que no vemos que los elementos guardados tengan asociados usuarios. Parece que el uso es público y el usuario sólo puede añadir y borrar ficheros. Quizá sea bueno incorporar esta funcionalidad mas adelante

Con respecto a la documentación de la aplicación es muy completa, hemos echado en falta, eso sí, colaboración por parte de la comunidad de SL, recomendamos la asistencia a hackatones o moverlo por redes sociales de cara a un futuro.

En cuanto al código, este esta documentado en su mayor parte, sin embargo hemos detectado que hay algunas en las que la documentación es nula, quizá por se código reutilizado

Vemos también que función de login se ha implementado de manera extraña

```
public String validate() {  
    String authenticationUsername = Play.application().configuration().getString("authentication.username");  
    String authenticationPassword = Play.application().configuration().getString("authentication.password");  
  
    if (name.equals(authenticationUsername) && password.equals(authenticationPassword)) {  
        return null;  
    } else {  
        return "invalidUsernameOrPassword";  
    }  
}
```

Recomendamos devolver un `true/false` o devolver `true` y si falla lanzar una excepción.

En vez de devolver `null` si es correcto y una cadena de texto si no lo es.

Es un proyecto sencillo e interesante aunque poco terminado, recomendamos trabajar los aspectos señalados y crear comunidad en torno al mismo.

Valoramos positivamente que exista una demo accesible, así como que se presentase a una conferencia de Software Libre en Cuba.

## VACmatch mobile

“VACmatch mobile es una aplicación para que árbitros de competiciones deportivas puedan cubrir las actas de los partidos directamente en su teléfono móvil, pudiendo actualizar resultados en tiempo real así como funcionar offline cuando no se dispone de cobertura. Así mismo se integrará dicha aplicación con otra también en desarrollo, que ofrecerá una API Rest y desde la cual las federaciones deportivas gestionarán totalmente sus competiciones. Las tecnologías principales son Reactjs y diversas herramientas Javascript como Reflux, Flow o Jest para testing entre otras, así como una base de datos PouchDB en el navegador y previsiblemente Apache Cordova para permitir su utilización en diversas plataformas. Actualmente la aplicación se encuentra en las fases iniciales de desarrollo, ya se ha realizado previamente un prototipo temporal para comprobar la usabilidad del sistema con usuarios reales y se está comenzando con el desarrollo de la aplicación definitiva.”

---

Aunque vemos que el código esta por lo general bien documentado y organizado, a la hora de realizar la instalación nos ha sido imposible.

Aconsejamos para futuros certámenes tener disponible una versión web.

En la propia documentación la parte de instalar Docker no esta correctamente indicada, el jurado ha utilizado el siguiente tutorial

“<https://docs.docker.com/engine/installation/linux/ubuntu/linux/>. ”

Por otra parte, vemos que es necesario gulp, recomendamos automatizar su instalación o indicar como se instala (sudo npm install gulp -g), para que el usuario no tenga que buscarlo

Consideramos que es una idea interesante y vemos que han entrado en contacto con algunas federaciones, según indican en su blog, pero sin embargo no hemos detectado que se haya conseguido atraer a la comunidad, para mejorar este aspecto recomendamos mejorar la documentación del proyecto y asistir a hackatones.

Este documento es publico y se permite sin restricción su uso y difusión, aunque no así su modificación.

Ha sido elaborado por el jurado del Certamen de Proyectos Libres de la UGR

**Isabel Cabezas** es ingeniera informática por la UGR desde 2005 y ha trabajado en diferentes empresas, diferentes ciudades y variopintas tecnologías. Geek hasta la médula y miembro activa desde hace años en diferentes comunidades y fundadora de algunas relacionadas con JavaScript o que incentivan la presencia de las mujeres en la tecnología, como BarcelonaJS, AdaJs y Ada Code Group. Actualmente trabaja en Madrid como evangelista técnico en Microsoft.

**Paloma de las Cuevas Delgado** es Ingeniera de Telecomunicaciones y actualmente se encuentra realizando un doctorado en el Departamento de Arquitectura y Tecnología de los Computadores de la Universidad de Granada. También ha sido investigadora del proyecto europeo MUSES, colaborando en el desarrollo de un sistema de seguridad móvil cuyo código está liberado en Github.

Ha colaborado en eventos organizados por la OSL de Granada, como el Campus UGR/Google para chicas, en el que se trata de acercar a las chicas de secundaria y bachiller las ingenierías, a través de la formación con herramientas de software libre como Arduino o Scratch.

**María Hinojosa Gutiérrez** dirigió durante varios años el departamento de informática de una empresa hasta que decidió emprender su propio proyecto y fundó en 2007 Codeko.com junto a su socio. Durante estos años ha participado en diversas actividades relacionadas con el software libre, entre ellas varias charlas en conjunción con la Oficina de Software Libre de la Universidad de Granada, ha sido miembro del jurado del Concurso Universitario de Software Libre y ha participado como experta en la Open Source World Conference.

Actualmente continua desarrollando su labor en Codeko.com, empresa que no ha dejado de crecer desde su fundación.

**Carmen Lopez Martinez** estudió Ingeniería Informática en la ETSIIT de la Universidad de Granada. Fue socia fundadora de [Gcubo](#) y desde el 2003 trabaja como desarrolladora en el Servicio de Informática de la Universidad de Granada.

Este certamen ha sido patrocinado por:

- Delegación de la Rectora para la Universidad Digital.
- Codeko.
- OpenXXI.
- Documedia.
- EasyName.
- ESRI.
- Bitergia.

A los que agradecemos su colaboración.