



TRABAJO DE FIN DE GRADO  
GRADO EN INGENIERÍA INFORMÁTICA

# Un portal de transparencia para datos libres

---

**Autor**

Germán Martínez Maldonado

**Tutor**

Juan Julián Merele Guervós



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE  
TELECOMUNICACIÓN

—  
Granada, 2 de junio de 2015



# Prefacio

## Un portal de transparencia para datos libres

Germán Martínez Maldonado

### Resumen

**Palabras clave:** software libre, transparencia, datos abiertos, sistema de control de versiones, aprovisionamiento, tests, integración continua, despliegue automático

Se pretenden desarrollar una plataforma para la transparencia que sea desplegable en un infraestructura física o virtual basándose en el trabajo realizado en el portal UGR Transparente, respaldada por los datos abiertos publicados en la OpenData UGR.

Este desarrollo tendrá como objetivo que el resultado sea una plataforma totalmente basada en el software libre que se pudiera adaptar con facilidad a otro organismo, teniendo en cuenta funcionalidades y requisitos obligatorios, además de aspectos de accesibilidad y escalabilidad.

La herramientas básicas a usar serán un sistema de control de versiones y una plataforma de desarrollo colaborativo que albergue el proyecto, además usando dicho sistema de control de versiones. También se quiere que la plataforma se pueda ir desarrollando y administrando a la vez, por lo que para convertir todo esto es un proceso más ágil e ininterrumpido se usarán otras herramientas que permitan lo siguiente:

- Realizar aprovisionamiento de las infraestructuras.
- Validación mediante tests unitarios.
- Comprobación de conflictos mediante integración continua.
- Actualizaciones mediante despliegue automático.



# A transparency portal for open data

Germán Martínez Maldonado

## Extended abstract

**Keywords:** free software, transparency, opendata, version control system, provisioning, tests, continuous integration, automated deployment

It is intended to develop a platform for transparency that is deployable on a physical or virtual infrastructure based on work done on the site UGR Transparente, backed by open data published in the Open Data UGR.

This development aims that the result is a platform completely based on free software that could be easily adapted to another organization, taking into consideration features and mandatory requirements, as well as issues of accessibility and scalability. It will develop in Node.js, which is a programming environment that operates at runtime based on the Google's V8 JavaScript engine; also will use Express for web application development and Jade to generate HTML files based on templates.

Being free software, this development is not limited to the working group that started the work, but rather is focused on that anyone can make their contribution to the project; therefore, for all this can be handled is necessary a version control system, in this case Git will be used, which is practically a standard in this area. That in addition a open collaborative development platform is used and recognized as GitHub that is integrated with the Git itself, we will provide ease in the development and distribution of work, because any who access the project repository can freely copy it for its open license.

Provisioning a machine, as the name suggests, is to provide to the machine all the resources needed for this performance, in our case we refer to all the software necessary for that the platform developed works properly in such infrastructure. As the entire project is hosted on GitHub, we can set up a utility like Ansible to download the project and then install on the machine all the necessary software so that the platform can function properly.

During the development of any software application, new features are introduced to the software gradually, so must be ensured that the new features do not compromise the overall stability of the project. For this purpose the unit tests are written, which could be considered as small programs within

the system that handles check by assertions and behavior patterns that all elements operate as they should. To verify that our unit tests are sufficient also need to pass a coverage test that tells us that all the functionality of our platform are properly validated by the corresponding unit test. There are many libraries that allow both actions, but for the facility of work between them, unit test will be passed with Mocha and coverage test will be with Istanbul.

Once we have the tests written, we don't need concern ourselves with run them manually, we have the option of running in a external platform every time we make changes and directly get the results, this is continuous integration. Continuous integration will be made with Travis CI and the operation is very simple: every time we make a change in GitHub, Travis CI download the latest version of the code, builds it and passes the tests we have written, to finish by returning the results of the tests that will make us know whether changes have produced a conflict in the system.

The last thing to keep in mind is about deploying the changes automatically on our server, can be a tedious procedure having to manually access to the infrastructure of our platform every time that we want to apply the new changes we have made in the application, so we'll use the tool Flightplan for automatic deployment; only with specify our server as the target we can set a series of tasks that will make that automatically for the updates we're making become effective on the main machine.

By using these tools we are getting an application in which development and management are closely related and automated, which makes the application much more reliable and easy recovery in case of problems.

---

Yo, **Germán Martínez Maldonado**, alumno de la titulación Grado en Ingeniería Informáticade la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI XXXXXXXXX, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Germán Martínez Maldonado

Granada, a 2 de junio de 2015





---

D. **Juan Julián Merelo Guervós**, Profesor del Área de XXXX del Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada.

**Informa:**

Que el presente trabajo, titulado *Un portal de transparencia para datos libres*, ha sido realizado bajo su supervisión por **Germán Martínez Maldonado**, y autorizo la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expide y firma el presente informe en Granada a 2 de junio de 2015.

**El tutor:**

**Juan Julián Merelo Guervós**



# Agradecimientos

Poner aquí agradecimientos...



## Capítulo 1

# Introducción

La transparencia en los organismos públicos es importante.  
Los desarrollos de software con despliegue continuo son interesantes.



## Capítulo 2

# Especificación de requisitos

### 2.1. Objetivos

El objetivo de este proyecto es el de obtener un portal de transparencia para la Universidad de Granada basado en el software libre que además permita una metodología de desarrollo en el que continuamente se puedan añadir nuevas funcionalidades a la vez que se van testeando, para después de una fácil integración culminar con un despliegue automático.

La aplicación final se usará en la Universidad de Granada, pero además el objetivo del desarrollo es que se pueda exportar para su uso en cualquier organización sin demasiada dificultad.

Un resumen de los principales objetivos a alcanzar son:

- **OBJ-1.** La plataforma enlazará los datos abiertos contenidos en la plataforma OpenData UGR (algo que ya hace, pero con problemas existentes que se deben solucionar).
- **OBJ-2.** El desarrollo de la plataforma seguirá una metodología de desarrollo que contará con tests unitarios para cada una de sus funcionalidades, test de cobertura que evalúen los test unitarios, integración continua ante cambios introducidos, despliegue automático de las actualizaciones que se vayan produciendo y aprovisionamiento software para la infraestructura.

### 2.2. Descripción de los usuarios finales

El usuario de la aplicación será cualquier persona que tenga interés por conocer datos internos de la Universidad de Granada fácilmente. Como no se quiere enfocar en un público objetivo, el portal tiene que ser fácil de utilizar tanto para personas con experiencia en la navegación de páginas web como para las que no la tengan.

## 2.3. Requisitos Funcionales

Los requisitos funcionales son las características que tiene que implementar el sistema para cubrir todas las necesidades de los distintos usuarios. Al usuario lo único que le interesa es ver una página web estática con la información que desea consultar, por lo que el único requisito imprescindible es que cuando pulse una categoría esta se despliegue y cuando se pulse un enlace a OpenData UGR este nos lleve al conjunto de datos correspondiente. Además, se habilitará un buscador para que también se pueda acceder a la información mediante la introducción de palabras clave. Dicho esto podemos distinguir los siguientes requisitos funcionales:

- **RF-1.** Acceso a la información:
  - **RF-1.1.** Al seleccionar una categoría esta se despliega y aparecen sus subcategorías.
  - **RF-1.2.** Cuando se pulsa sobre el enlace de un elemento, automáticamente se accede a la información contenida de ese elemento en OpenData UGR.
  - **RF-1.3.** Para obtener información sobre una tema en concreto se introducirá el termino clave relacionado en el buscador.
- **RF-2.** Presentación de la información:
  - **RF-2.1.** Si el elemento de información no es un archivo PDF este se podrá previsualizar desde el mismo portal de transparencia sin necesidad de acceder a OpenData UGR.
  - **RF-2.2.** Siempre se podrá descargar el archivo con la información contenido en OpenData UGR sobre cualquier elemento visible.

Esta es la parte que ya estaba implementada en mayor medida y sobre la que se va a iniciar el trabajo.

## 2.4. Requisitos no Funcionales

Los requisitos no funcionales son las características propias del desarrollo, pero que no tienen que estar relacionadas con su funcionalidad. En este caso nos referimos a todas las características que se requieren para que la aplicación siga un desarrollo ágil de despliegue continuo y administración automática:

- **RN-1.** Procesar la información que será mostrada en las distintas páginas del portal.



- **RN-2.** Implementar tests unitarios para validar que todas las funcionalidades programadas funcionan correctamente.
- **RN-3.** Realizar test de cobertura que compruebe la fiabilidad que proporcionan los tests unitarios.
- **RN-4.** Usar integración continua para asegurarse que los cambios introducidos no producen conflictos en la plataforma.
- **RN-5.** Usar despliegue automático para actualizar con los nuevos cambios la plataforma una vez estos han sido validados.
- **RN-6.** Elaborar una aprovisionamiento que permita que en una infraestructura por determinar se pueda instalar automáticamente la plataforma y todos los elementos necesarios.

## 2.5. Requisitos de Información

Los requisitos de información se refieren a la información que es necesaria almacenar en el sistema. La única información relevante que se va a almacenar son los datos descriptivos y de enlace de cada uno de los elementos del portal OpenData UGR que se van a mostrar en UGR Transparente.

- **RI-1.** Datos abiertos.
  - Información sobre cada uno de los elementos que se van a mostrar en el portal de transparencia como datos abiertos.
  - Contenido: nombre, categoría, conjunto de datos, enlace a OpenData UGR, enlace al recurso.