



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, 4 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 pretende 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 plataforma 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.

Las herramientas básicas a usar serán un sistema de control de versiones y un sistema de desarrollo colaborativo que albergue el proyecto, además use dicho sistema de control de versiones. También se quiere que la plataforma se pueda desarrollarr y administrar simultáneamente, 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. The platform which presents the data 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. Moreover, the platform that contains the data is based on CKAN, a open-source data portal platform developed by The Open Knowledge Foundation, a not-for-profit organisation that promotes creating and sharing knowledge freely.

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. In addition, it will also be used a recognized and open collaborative based development platform as is GitHub, which is also integrated with Git, providing ease in the development and distribution of work, because any who access the project repository can freely copy it for its open license.

An important point to facilitate system administration is the provisioning. Provisioning a machine, as the name suggests, is to provide to the machine all the resources needed for its performance, in our case we refer to all the software necessary for that the platform developed works properly in such infrastructure. In the case of transparency portal, 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. In contrast, such as the open data portal

is not a own development, their provisioning will consist of installing and customizing CKAN for the University of Granada.

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 4 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 4 de junio de 2015.

**El tutor:**

**Juan Julián Merelo Guervós**





# Agradecimientos

Poner aquí agradecimientos...



# Índice general

<b>1. Introducción y motivación</b>	<b>12</b>
<b>2. Planificación</b>	<b>13</b>
2.1. Fases y entregas . . . . .	13
2.1.1. Fases . . . . .	13
2.1.2. Lista de entregas . . . . .	13
2.2. Estructura de Descomposición del Trabajo . . . . .	14
2.3. Actividades y tiempo . . . . .	14
2.4. Dependencias entre tareas . . . . .	14
2.5. Asignación de recursos . . . . .	14
2.6. Diagrama de Gantt . . . . .	14
2.7. Análisis de costes . . . . .	14
<b>3. Especificación de requisitos</b>	<b>16</b>
3.1. Objetivos . . . . .	16
3.2. Descripción de los actores . . . . .	17
3.3. Requisitos . . . . .	17
3.3.1. Requisitos Funcionales . . . . .	17
3.3.2. Requisitos no Funcionales . . . . .	18
3.3.3. Requisitos de Información . . . . .	19
3.4. Modelos de casos de uso . . . . .	19



## Capítulo 1

# Introducción y motivación

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

## Capítulo 2

# Planificación

### 2.1. Fases y entregas

#### 2.1.1. Fases

Como este va a ser un proyecto que ya cuenta con un trabajo previo realizado, la fase inicial de gestión va a ser muy breve porque se parte de que ya se han realizado varias reuniones y el proyecto está parcialmente funcionando, por lo que no se parte de cero, es una ampliación de un proyecto inicial.

- **Fase 0:** Gestión del proyecto
- **Fase 1:** Análisis del entorno
- **Fase 2:** Desarrollo del Documento de Objetivos del Proyecto
- **Fase 3:** Desarrollo técnico

#### 2.1.2. Lista de entregas

Se harán una serie de breves informes sobre el contenido de cada una de las fases de planificación del proyecto.

- **Fase 0:** Gestión del proyecto
  - Descripción: Se realizarán reuniones para concretar los objetivos del proyecto y el desarrollo necesario para cumplirlo.
  - Tipo: informe.
- **Fase 1:** Análisis del entorno
  - Descripción: Se establecen los criterios de elección de las herramientas a utilizar.

- Tipo: informe.
- **Fase 2:** Desarrollo del Documento de Objetivos del Proyecto
  - Descripción: Se desarrolla un documento que recoja los objetivos a cumplir en el proyecto, así como diferentes aspectos del proyecto que habrá que tener en cuenta para su puesta en desarrollo.
  - Tipo: informe.
- **Fase 3:** Desarrollo técnico
  - Descripción: Implementación del software necesario para cumplir los objetivos, pasando por sus diferentes fases: análisis, diseño, implementación, pruebas. También se realizará una documentación explicativa sobre funcionamiento del mismo.
  - Tipo: informe.

## 2.2. Estructura de Descomposición del Trabajo

El diagrama de Estructura de Descomposición del Trabajo (figura 2.1) es una descomposición jerárquica de las diferentes fases y entregas en las que está planificado el proyecto.

## 2.3. Actividades y tiempo

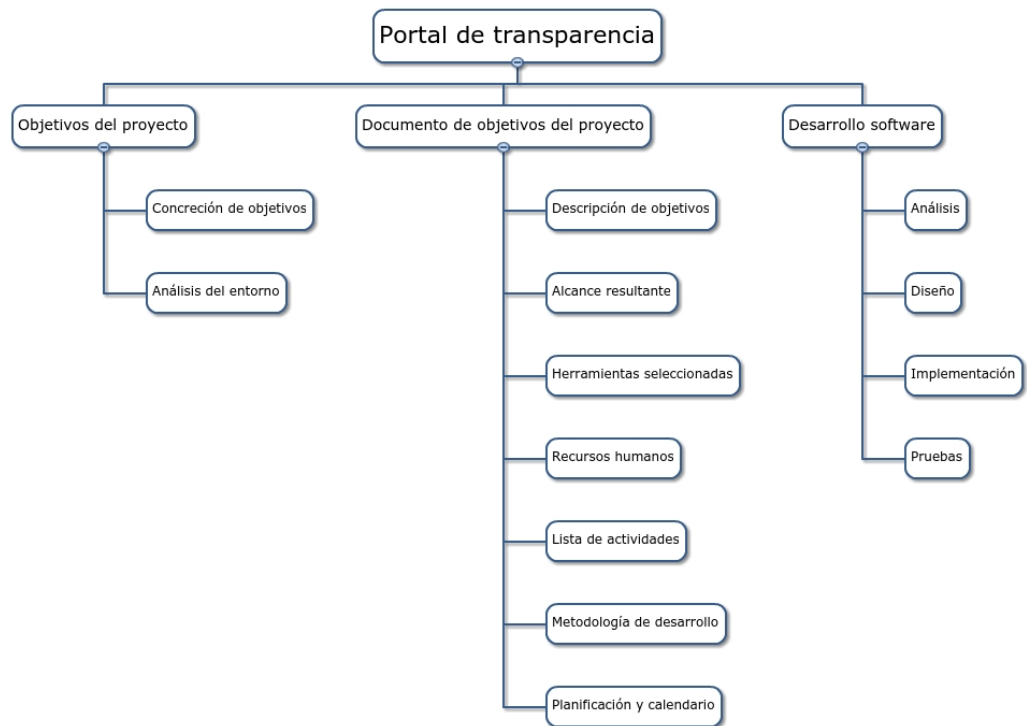
## 2.4. Dependencias entre tareas

## 2.5. Asignación de recursos

## 2.6. Diagrama de Gantt

## 2.7. Análisis de costes

Figura 2.1: Diagramas de casos de uso





## Capítulo 3

# Especificación de requisitos

### 3.1. 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. Este portal no almacenará los propios datos abiertos, si no que hará de presentación de los datos que estarán contenidos en otra plataforma destinada únicamente a dicho fin (OpenData UGR).

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 UGR Transparente 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 UGR Transparente seguirá una metodología de desarrollo continuo que contará con la implementación de tests unitarios para cada una de las funcionalidades que se vayan añadiendo, 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.
- **OBJ-3.** Realizar un aprovisionamiento para la plataforma contenedora de los datos abiertos OpenData UGR que permita una instalación con una base de CKAN más personalizada.

## 3.2. Descripción de los actores

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.

También tenemos al desarrollador, que será el encargado de introducir los elementos de información que serán listado en la plataforma. Como es una página estática, los elementos serán mostrados en la tabla correspondiente de la página correspondiente después de que sean leídos del archivo JSON correspondiente.

## 3.3. Requisitos

Todo software se desarrolla para cubrir una necesidad, por lo que en este apartado vamos a describir los requisitos que se estiman necesarios para cubrir los objetivos propuestos.

### 3.3.1. 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. Todos los requisitos funcionales que tratará este proyecto están únicamente dirigidos a la plataforma UGR Transparente.

- **RF-1.** Introducción de nuevos elementos de información.
- **RF-2.** Acceso a la información:
  - **RF-2.1.** Al seleccionar una categoría esta se despliega y aparecen sus subcategorías.
  - **RF-2.2.** Al seleccionar una subcategoría se muestran todos sus elementos.
  - **RF-2.2.** Cuando se pulsa sobre el enlace de un elemento, automáticamente se accede a la información contenida de ese elemento en OpenData UGR.

- **RF-2.3.** Para obtener información sobre un tema en concreto se introducirá el termino clave relacionado en el buscador.
- **RF-3.** Presentación de la información:
  - **RF-.1.** Generar tablas con los elementos de información.
  - **RF-3.1.** 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-3.2.** Siempre se podrá descargar el archivo con la información contenido en OpenData UGR sobre cualquier elemento listado.

Los aspectos de funcionalidad ya se encuentran implementados en su mayoría (solo existen problemas al cargar la información) de una fase previa al proyecto por lo que esta será la base de la que se partirá para el desarrollo.

### 3.3.2. 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 de la plataforma UGR Transparente.

- **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.

En cuanto a la plataforma OpenData UGR, este proyecto solo contempla un único objetivo no funcional que está relacionado con la administración del sistema más que con que aspectos de desarrollo software.

- **RN-7.** Crear una aprovisionamiento y personalización de CKAN para OpenData UGR.

### 3.3.3. 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.

## 3.4. Modelos de casos de uso

Aunque ya se ha indicado que la parte funcional ya se encuentra implementada de forma previa a este proyecto, se van a incluir unos modelos de caso de uso simples para dar un visión más clara del funcionamiento general de la plataforma UGR Transparente.

### Descripción básica de actores

- **Ac-1.** Usuario
  - Descripción: Persona que usa la plataforma que consulta datos.
  - Características: Es el usuario común que accederá a la página.
  - Relaciones: Ninguna.
  - Atributos: Ninguno.
  - Comentarios: El usuario no es necesario que tenga ningún conocimiento previo al uso de la plataforma, simplemente accederá y consultará los datos que sean de su interés.
- **Ac-1.** Desarrollador
  - Descripción: Encargado de añadir los elementos de información a la plataforma.
  - Características: Su trabajo está en el lado del servidor que genera la página, nunca trabaja desde el lado del cliente.
  - Relaciones: Ninguna.
  - Atributos: Ninguno.
  - Comentarios: Es el encargado de desarrollar las funcionalidades del portal, entre ellas añadir nuevos elementos de información.

### Descripción casos de uso

- **CU-1. Añadir elemento**
  - Actores: Desarrollador
  - Tipo: Primario, Esencial
  - Precondición: Ninguna
  - Postcondición: Nuevos elementos de información serán visualizados en el portal de transparencia
  - Propósito: Son añadidos nuevos elementos de información al portal de transparencia.
  - Resumen: Aparecerán nuevos elementos de información en la categoría que corresponda, acompañado además de su enlace correspondiente al OpenData UGR, un botón para previsualizar la información y otro botón para descargar el elemento.
- **CU-2. Consultar elementos**
  - Actores: Usuario
  - Tipo: Primario, esencial
  - Precondición: Existan elementos de información.
  - Postcondición: Se muestran los elementos de información de un determinado conjunto de datos.
  - Propósito: Obtiene los elementos de información de un determinado conjunto de datos.
  - Resumen: Accediendo a través del panel principal o el buscador se obtiene una tabla con los elementos de información de un determinado conjunto de datos.
- **CU-3. Acceder enlace de elemento**
  - Actores: Usuario
  - Tipo: Primario, esencial
  - Precondición: Se hayan generado las tablas con los elementos de información.
  - Postcondición:
  - Propósito: Accede a la información del elemento contenida en OpenData UGR.
  - Resumen: Cuando se pulsa el enlace, se accede al conjunto de datos que contiene la información del elemento en OpenData UGR, presentándolo de distinta forma en función del formato del elemento y dando la opción de descargar ese mismo elemento en un archivo con su formato original.

- **CU-4.** Previsualizar elemento de información
  - Actores: Usuario
  - Tipo: Secundario
  - Precondición: Se hayan generado las tablas con los elementos de información.
  - Postcondición:
  - Propósito: Previsualiza los datos del elemento.
  - Resumen: Cuando se selecciona el botón de ver un elemento de la tabla se abre una ventana emergente en la que se muestra la información del elemento que se contiene en OpenData UGR.
- **CU-5.** Descargar elemento de información
  - Actores: Usuario
  - Tipo: Secundario
  - Precondición: Se hayan generado las tablas con los elementos de información.
  - Postcondición:
  - Propósito: Descargar el elemento en un archivo con su formato original.
  - Resumen: Cuando se selecciona el botón de descargar un elemento de la tabla se abre una ventana emergente en la que se muestra la información del elemento que se contiene en OpenData UGR.

### Diagramas de casos de uso

La figura 2.1 representa el diagrama de estado de los casos de usos considerados inicialmente para el proyecto.

Figura 3.1: Diagramas de casos de uso

