

**Análisis de Requisitos integración  
FORMIGA-CLOUD / DIRAC (Prototipo II)**

**PROYECTO FORMIGACLOUD  
INTEGRACIÓN CON DIRAC V.2  
(Infraestructura distribuida  
con  
control de agentes remotos)**

Fecha:	24/01/12
Autor:	Víctor Manuel Fernández albor
Órgano al que pertenece:	USC
Versión del documento:	V.2.0

## TABLA DE MODIFICACIONES EN EL DOCUMENTO

Autor	Versión	Tipo de modificación
Víctor Manuel Fernández albor	V2.0	Versión inicial

## INDICE

1. Objetivos .....	4
1.1 Objetivos generales.....	4
1.2 Objetivos específicos.....	4
1.3 Requisitos.....	4
2. Modelo de Casos de uso.....	5
2.1 Actores.....	5
Actor-usuario.....	5
Actor-sistema.....	5
2.2 Clasificación actores.....	5
2.3 Actores y funcionalidades del proyecto .....	6
2.3 Descripción de casos de uso.....	9
2.4. Diagrama de Casos de Uso de Contexto.....	11
3. Matriz de trazabilidad de casos de uso objetivos funcionales.....	12
4. Identificación de otros requisitos.....	13
4.1. Especificación Complementaria.....	13
Anexo I.....	15
Anexo II.....	17

# 1. Objetivos

## 1.1 Objetivos generales

Los objetivos generales del presente documento es obtener ideas preliminares acerca de la creación de la segunda versión en la integración entre DIRAC, plataforma grid de agentes remotos y FORMIGA-CLOUD. En una primera versión en la integración se consiguió el envío de trabajos de investigadores desde DIRAC y utilizando recursos infrautilizados a través de la nube. Esto hace posible que los investigadores sean capaces de submitir sus trabajos y vean fácilmente con su propio navegador el estado de los mismos, y obtengan la salida de los mismos. Por otro lado los recursos computacionales que están en desuso o estén infrautilizados, se pueden ahora utilizar por los investigadores. En esta segunda versión se busca la complementación del gestor de cloud, CloudStack y la herramienta DIRAC, dotándolo de un carácter multiVO, y permitiendo también su fácil utilización y transmisión en sistemas de aulas. Se busca también la interacción con otros sistemas cloud a través de la integración de Interfaces como OCCI, así como la integración con el sistema DIRACVM existente actualmente para la gestión dentro de EC2. Otro de los objetivos en esta segunda versión es la complementariedad del nuevo sistema creado, con los sistemas tradicionales de envíos de trabajos.

## 1.2 Objetivos específicos

- Gestionar y configurar el sistema para múltiples grupos de usuarios de una forma más dinámica.
- Proporcionar interoperabilidad e integración a través de OCCI y DIRACVM
- Proporcionar una forma de implementación y configuración adaptada a aulas, siendo fácil y dinámica
- Complementar sistema de colas existente con sistema de envío de trabajos a través de la infraestructura cloud

## 1.3 Requisitos

En esta sección definiremos los requisitos, capacidades y condiciones que ha de satisfacer el sistema y que se clasifican en 5 tipos:

- Requisitos Funcionales

## 5 Universidad de Santiago de Compostela

- Requisitos de Facilidad de Uso
- Requisitos de Fiabilidad
- Requisitos de Rendimiento
- Requisitos de Soporte

Los requisitos funcionales los recogemos a continuación en el **Modelo de Casos de Uso**. Mientras que el resto pueden captar mediante una **Especificación Complementaria**.

## 2. Modelo de Casos de uso

### 2.1 Actores

Para nuestro nuevo prototipo se definen una serie de actores entre los cuales se tendrán tanto a usuarios como otros sistemas de interacción. Hacemos la clasificación previa de los actores según sean usuarios o sistemas:

#### ***Actor-usuario***

- Administrador DIRAC
- Administrador Cloud
- Gestor software científico
- Investigador

#### ***Actor-sistema***

- FormigaCloud
- DIRAC
- Gestión-Colas
- Recursos

### 2.2 Clasificación actores

Los actores se van a clasificar previamente en función de su importancia en el desarrollo del sistema, clasificándose en tres tipos de relevancia siendo los “actores principales” los más importantes de cara al diseño e implementación del mismo. En los “actores secundarios” su nivel de importancia será más relativa, mientras que en los menos importantes pero no por ello innecesarios, “actores de apoyo” servirán de

## 6 Universidad de Santiago de Compostela

soporte a toda la infraestructura para la mejora de todas las funcionalidades.

En la siguiente tabla vemos el tipo de actores junto con su acrónimo a emplear de aquí en adelante y el nivel de importancia sobre el diseño:

Actor	Acrónimo	Nivel de relevancia
Administrador DIRAC	AD	Principal
Administrador Cloud	AC	Principal
Gestor Software Científico	GSC	Apoyo
Investigador	INV	Apoyo
FormigaCloud	FC	Principal
DIRAC	SD	Principal
Gestión-colas	GC	Secundario
Recursos	RE	Principal

### 2.3 Actores y funcionalidades del proyecto

Un actor es un comportamiento, ya sea una persona, un sistema informático o una organización. Observando la especificación de requisitos descrita en el apartado correspondiente, se podrá identificar los siguientes actores, objetivos y funcionalidades para el nuevo prototipo en la integración.

Actor	Objetivos y Funcionalidades
Administrador DIRAC	F1. Desde el sistema DIRAC se permitirá la configuración de máquinas virtuales para diferentes grupos de usuarios, con una ampliación del prototipo existente.
Administrador Cloud	F2. Deberá de permitir la configuración de varios tipos de instancias a través de un usuario, necesarias para la ejecución de diferentes grupos de investigación
Gestor software científico	F3. El sistema deberá de permitir obtener software de repositorios externos así como poseer su propio repositorio que permita la inclusión de software a mayores dependiendo el grupo de usuarios

## 7 Universidad de Santiago de Compostela

	F4. El sistema tendrá un formato estipulado para la colocación del software
Investigador	<p>F5. El sistema deberá de poder permitir el envío de trabajos y a través del navegador web, permitiendo la delegación de credenciales a través del certificado de proxy previamente subido.</p> <p>F6. El sistema permitirá el registro de usuarios a través de un formulario web, para poder hacer uso de la infraestructura, con la posterior validación del administrador de DIRAC.</p>
FormigaCloud	<p>F7. Deberá de permitir la creación de varios tipos de instancias a través de un usuario, necesarias para la ejecución de diferentes grupos de investigación</p> <p>F8. El sistema debe permitir la obtención de datos de estado de los nodos y MV</p>
DIRAC	<p>F9. El sistema permitirá la gestión dinámica para realizar peticiones de creación de máquinas virtuales para diferentes grupos de usuarios, con una ampliación del prototipo existente, a través del API del gestor Cloud.</p> <p>F10. El sistema se integrará con la versión de DIRACVM existente para el cloud EC2 de Amazon.</p> <p>F11. El sistema permitirá la interoperabilidad a través de la interfaz OCCl</p> <p>F12. El sistema permitirá la salvaguarda de los trabajos ejecutados por los usuarios.</p> <p>F13. El sistema deberá de permitir la salvaguarda del histórico con la relación trabajo y máquina virtual en donde se ejecute.</p> <p>F14. El sistema deberá de ser compatible con el envío de trabajos a gestores de colas.</p> <p>F15. El sistema deberá de permitir el rápido despliegue en aulas.</p>

## 8 Universidad de Santiago de Compostela

	F16. Desde el sistema se podrá monitorizar el estado de las MV
Gestión-colas	F17. Desde el sistema se gestionarán los trabajos que lleguen de DIRAC redistribuyéndolos a través de los recursos existentes.
Recursos	F18. Existirán unos recursos disponibles a través de los cuales se realizarán pruebas para múltiples VOs

### 2.3 Descripción de casos de uso

Tras el estudio de la lista actor-objetivo se decretó crear a nivel contextual los siguientes casos de uso:

<b>Actor</b>	AD
<b>Caso de uso</b>	Gestionar configuración CloudStack en DIRAC para soportar multiVO
<b>Acrónimo</b>	GCDVO
<b>Descripción</b>	En este caso de uso se contempla la gestión y configuración de varias VOs a través de la infraestructura de DIRAC

<b>Actor</b>	AC
<b>Caso de uso</b>	Gestionar instancias cloud
<b>Acrónimo</b>	GIC
<b>Descripción</b>	En este caso de uso se la gestión y configuración de instancias de MV a través del gestor cloud

<b>Actor</b>	GSC
<b>Caso de uso</b>	Proporcionar software multiVO
<b>Acrónimo</b>	PSVO
<b>Descripción</b>	En este caso de uso se contempla la gestión del software para cálculo científico en la que se proporcionará software de una manera escalable y permitiendo la obtención del software para varias VOs, estipulando un formato definitorio en cuanto a la VO dentro del sistema de repositorios



## 9 Universidad de Santiago de Compostela

<b>Actor</b>	INV
<b>Caso de uso</b>	Enviar trabajos al sistema a través del navegador web
<b>Acrónimo</b>	EJWB
<b>Descripción</b>	En este caso de uso se contempla el envío de trabajos de investigación a través del navegador web

<b>Actor</b>	INV
<b>Caso de uso</b>	Solicitar registro a través del navegador web
<b>Acrónimo</b>	SRWB
<b>Descripción</b>	En este caso de uso se contempla la posibilidad de crear solicitudes de registro de usuarios a través del navegador web.

<b>Actor</b>	FC
<b>Caso de uso</b>	Crear varios tipos de instancias
<b>Acrónimo</b>	CVI
<b>Descripción</b>	En este caso de uso se contempla la posibilidad de crear varios tipos de instancias con el fin de dotar a la infraestructura la funcionalidad multiVO.

<b>Actor</b>	SD
<b>Caso de uso</b>	Llamar a la API de CloudStack
<b>Acrónimo</b>	CAC
<b>Descripción</b>	En este caso de uso se contempla la posibilidad de realizar las llamadas necesarias para la creación de máquinas virtuales dentro de los recursos disponibles, y permitiendo ejecutar diferentes tipos de MV dependiendo de la VO

<b>Actor</b>	SD
<b>Caso de uso</b>	Integrar con DIRACVM
<b>Acrónimo</b>	IDVM
<b>Descripción</b>	En este caso de uso se contempla la integración con la versión de DIRAC que actualmente se utiliza para el envío de MV a EC2

## 10 Universidad de Santiago de Compostela

<b>Actor</b>	SD
<b>Caso de uso</b>	Integrar a través de OCCI
<b>Acrónimo</b>	IOCCI
<b>Descripción</b>	En este caso de uso se contempla la interoperabilidad del sistema a través de la integración con OCCI

<b>Actor</b>	SD
<b>Caso de uso</b>	Gestionar user data
<b>Acrónimo</b>	CVO
<b>Descripción</b>	En este caso de uso se contempla el almacenamiento de los trabajos de los usuarios

<b>Actor</b>	SD
<b>Caso de uso</b>	Almacenar información acerca de la relación trabajos-MV
<b>Acrónimo</b>	AITMV
<b>Descripción</b>	En este caso de uso se contempla el almacenamiento de la información de los trabajos de usuarios que hayan sido ejecutados a través de MV

<b>Actor</b>	SD
<b>Caso de uso</b>	Enviar sistemas de colas
<b>Acrónimo</b>	ESC
<b>Descripción</b>	En este caso de uso se contempla la compatibilidad de ejecución de trabajos en MV y envío de trabajos a sistemas de colas

<b>Actor</b>	SD
<b>Caso de uso</b>	Monitorizar recursos
<b>Acrónimo</b>	MR
<b>Descripción</b>	En este caso de uso se contempla la monitorización de los nodos de trabajo existentes

## 11 Universidad de Santiago de Compostela

<b>Actor</b>	GC
<b>Caso de uso</b>	Gestionar colas
<b>Acrónimo</b>	GC
<b>Descripción</b>	En este caso de uso se contempla la gestión de las colas para asignación de recursos en trabajos de usuarios

<b>Actor</b>	RE
<b>Caso de uso</b>	Configurar recursos
<b>Acrónimo</b>	CR
<b>Descripción</b>	En este caso de uso se contempla la configuración de los recursos para las posibles pruebas a realizar.

## 2.4. Diagrama de Casos de Uso de Contexto

Se podrá ilustrar los actores y casos de uso obtenidos en el apartado anterior en el primer diagrama que muestra de forma muy general las funcionalidades, que debe de satisfacer el sistema. Este esquema se denomina Diagrama de Casos de Uso de Contexto.



## 13 Universidad de Santiago de Compostela

F13											X				
F14												X			
F15												X			
F16													X		
F17														X	
F18															X

## 4. Identificación de otros requisitos

Detallar los casos de uso no es suficiente en la etapa de análisis, pues existen otro tipo de requisitos que deben de ser detallados necesitando ser identificados, como los relacionados con la documentación o el empaquetamiento, soporte o licencias, y estos requisitos son dados en la **Especificación Complementaria**.

### 4.1. Especificación Complementaria

En este apartado estarán determinados todos los requisitos no contemplados en los casos de uso, como por ejemplo:

- **Facilidad de instalación y despliegue:** el sistema tendrá que desplegarse fácilmente en aulas o laboratorios, con la utilización de scripts que permitan hacer la instalación y configuración automáticamente.
- **Fiabilidad:** El sistema deberá ser preciso en sus operaciones, y escalable en todos sus componentes debido a la implementación de sistemas que son distribuidos.
- **Restricciones de Implementación:** Será implementado en un lenguaje orientado a objetos compatible con la solución DIRAC existente en este caso será Python, que por otro lado necesitará comunicarse con las funcionalidades de FORMIGA-CLOUD para lo que se plantea la utilización de Web-Services (SOAP) ya que tiene la ventaja de numerosa documentación y APIs para integrar fácilmente con Python o la otra opción es el uso de REST mucho menos documentado hasta el momento y con menos opciones a la hora de elegir APIs y Plugins en programas de desarrollo.

## 14 Universidad de Santiago de Compostela

- **Componentes de Libre distribución:** Se utilizará en la medida de lo posible herramientas de libre distribución para cada una de las distintas partes del proyecto. Así, por ejemplo, la solución software para ello es el sistema de programación Eclipse, integrado con plugins para programación en Python y SVN para interconexión al repositorio DIRAC. Por otro lado, el gestor de base de datos será MySQL.
- **Manuales y guías de usuario:** Se crearán distinto tipo de manuales o guías de usuario, dependiendo del perfil del usuario ya sea Administrador, Investigador o Gestor de Software. Con los manuales se pretende facilitar la instalación, configuración y manejo de las aplicaciones, así como se proporcionarán distinto tipo de enlaces de ayuda en los mismos.
- **Coordinación con DIRAC team:** Se coordinará con el grupo de DIRAC el intercambio de software, y el como subir al repositorio GIT del grupo el desarrollo de la aplicación.