



# FORMIGA/GFLUXO: plataforma distribuida de aulas de informática

## FORMIGA/GFLUXO: distributed platform for IT classrooms

◆ J. López Cacheiro, C. Fernández Iglesias, D. Cordero, C. Fernández Sánchez, E. Gutiérrez, A. Rodríguez, R. Valín, A. Garcia-Loureiro

### Resumen

Los proyectos FORMIGA y G-Fluxo han facilitado la creación de una plataforma distribuida que permite integrar los recursos de las aulas de informática de las universidades gallegas para su uso en computación científica y utilizarlos desde un portal web. Para lograr este objetivo, se ha usado una combinación de distintas tecnologías tales como virtualización, tecnologías grid y redes privadas virtuales (VPN). Una de las principales características de la plataforma creada es que permite la interoperabilidad con grids como EGEE o EELA, así como la NGI española, permitiendo a las aplicaciones utilizar a la vez recursos de ambas infraestructuras.

**Palabras clave:** máquinas virtuales, grid, UPN, portal web, computación científica.

### Summary

The FORMIGA and G-Fluxo projects have led to the creation of a distributed platform that enables IT classroom resources in Galician universities to be integrated for use in computational science and used from a web portal. To achieve this objective, a combination of different technologies has been used, such as virtualisation, grid technologies, and virtual private networks (VPN). One of the main features of the platform created is that it enables interoperability with grids such as EGEE and EELA, and the Spanish NGI, enabling the applications to use resources from both infrastructures at the same time.

**Keywords:** virtual machines, grid, VPN, web portal, computational science.

## 1. Introducción

En la actualidad muchas instituciones como las universidades disponen de aulas de informática que se renuevan periódicamente y que acumulan una gran potencia computacional. Durante las noches, fines de semana y períodos no lectivos estos recursos están siendo desaprovechados, con la consiguiente pérdida económica que esto conlleva, sobre todo si tenemos en cuenta que la vida útil de estos ordenadores es muy corta porque en pocos años quedan desfasados, obligando a su cambio.

El proyecto FORMIGA[1] tiene por objetivo desarrollar una arquitectura distribuida que permita aprovechar los recursos de las distintas aulas de cada una de las universidades gallegas. Este problema ya ha sido abordado con anterioridad sobre todo usando un entorno reducido a unas pocas aulas dentro de la misma red. Históricamente se han usado soluciones basadas en Condor y más recientemente también se ha empleado BOINC. El proyecto FORMIGA propone una solución diferente para aprovechar las aulas de las universidades gallegas basándose en una combinación de tecnologías existentes que engloban grid, virtualización y redes virtuales privadas. Estas tecnologías han permitido crear una infraestructura interoperable con la que es considerada actualmente la mayor infraestructura grid del mundo, el EGEE, permitiendo que el usuario a través del mismo interfaz tenga acceso de forma transparente tanto a los recursos de las aulas como a los recursos grid de EGEE.

Esta plataforma ofrece una enorme cantidad de recursos, pudiendo llegar a más de 2000 nodos distribuidos entre las distintas universidades gallegas, y por tanto en distintas redes, y en muchos casos saliendo al exterior por medio de NAT desde una red privada. Sin embargo, la paralelización de una aplicación para ejecutarse en esta plataforma distribuida supone un reto ya que se debe tener en cuenta

◆  
Esta plataforma permite la interoperabilidad con grids como EGEE o NGI

◆  
El objetivo del proyecto es desarrollar una arquitectura distribuida para aprovechar los recursos de las aulas de informática de las universidades gallegas

tanto la heterogeneidad del sistema como el hecho de que los recursos disponibles varían en función del tiempo.

Se ha utilizado como aplicación de prueba un simulador de nanodispositivos electrónicos de tipo Monte Carlo. Este programa realiza simulaciones en dos dimensiones de transistores MOSFET de doble puerta de silicio sobre aislante (DG-SOI). El código del programa se ha escrito en Fortran. Partiendo de su versión secuencial se han desarrollado diversas adaptaciones para utilizar de la forma más óptima posible la infraestructura grid del FORMIGA.

Para facilitar el uso de la infraestructura y dar una alternativa a los usuarios al envío mediante línea de comandos, se trabajó conjuntamente con el proyecto G-Fluxo[2] para desarrollar un portal web que permite el envío de trabajos de una forma gráfica. Este portal además fue diseñado con el propósito de asegurar una amplia compatibilidad con distintas plataformas, tanto grid como clústers locales y actualmente permite al usuario definir flujos de trabajos (workflows).

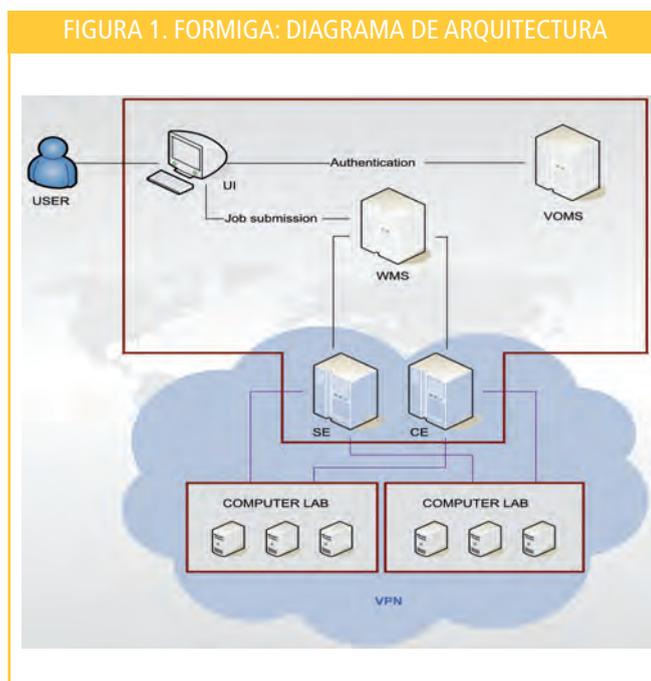
## 2. Arquitectura

En los ordenadores de las aulas se han instalado máquinas virtuales que son utilizadas como nodos de cálculo (WN) y que están conectadas a un Computing Element (CE) situado en el CESGA que se encarga de recibir los trabajos y distribuirlos entre los nodos. Para la distribución de trabajos, el CESGA facilita un Workload Management System (WMS) que se encarga de repartir los trabajos entre los CE disponibles.

Para la autenticación de los usuarios se dispone de un servidor VOMS que comprueba la identidad de los usuarios a partir de sus certificados de usuario. Mediante este servidor también se puede agrupar a los usuarios en distintas comunidades a través de la creación de diferentes Organizaciones Virtuales (VO). Esto permite hacer grupos de usuarios con intereses afines y distribuir de forma flexible los recursos disponibles entre las distintas facultades y grupos de investigación.

Como se puede observar en el esquema de la figura 1 se requiere conectividad bidireccional entre los ordenadores de las aulas y los servidores del CESGA. También se requiere de conectividad bidireccional entre las distintas aulas. Estos requisitos se ven satisfechos mediante la utilización de una red privada virtual (VPN) gestionada por el CESGA.

El diseño actual permite expandir fácilmente esta infraestructura a otros campus y universidades, de forma que estos recursos estén disponibles para todos los investigadores a través de un interfaz de usuario común.



En el proyecto se trabajó conjuntamente con el proyecto G-Fluxo para desarrollar un portal web que permite el envío de trabajos de una forma gráfica

Es necesario tener conectividad bidireccional entre los ordenadores de las aulas y los servidores del CESGA



### 3. Funcionalidades

A continuación se describen las principales funcionalidades disponibles en la plataforma.

**Virtualización con Xen (nodos Linux):** hemos usado Xen como plataforma de virtualización para los nodos Linux por razones de eficiencia, además de sus avanzadas posibilidades de gestión y administración de máquinas virtuales.

**Virtualización con VMWare (nodos Windows):** la utilización de VMWare Player nos permite integrar directamente ordenadores con Windows dentro de la infraestructura.

**VPN:** todos los nodos están interconectados a través de una VPN. El servidor está localizado en el CE. Cada máquina virtual, después de inicializar su conexión de red (con conexión a internet), arranca su servicio VPN. Hay dos posibles casos de conexión a la VPN. El primero, máquinas ya configuradas y trabajando correctamente en la red, usando un certificado creado específicamente para adquirir una IP fija. El segundo caso se usa para la instalación de nuevas máquinas en el sistema. Estas máquinas están pre-configuradas con un certificado genérico para la VPN y se les asigna dinámicamente una IP en una red aislada de los nodos de producción.

**Herramienta de inventario OCS.** Inventory/GLPI: para controlar y recoger los recursos de manera más avanzada se ha incorporado a la infraestructura el control automático de recursos OCS Inventory, que es capaz de recoger la información de todas las máquinas una vez se han conectado a la VPN. Toda esta información se guarda en una base de datos local. Mediante el interfaz GLPI puede ser analizada con mayor detalle.

**Monitorización global con Ganglia, r-Nagios y SAM:** se han implantado servicios genéricos de monitorización, usuales en entornos grid, que permiten detectar el estado de los nodos, comprobar su operatividad y notificar los posibles fallos que puedan tener.

**Monitorización vía web:** se ha desarrollado un pequeño portal para monitorizar el estado de los nodos, colas y trabajos en ejecución, que permite visualizar rápidamente el estado general del sitio. Está desarrollado bajo un servidor http de baja carga y scripts CGI corriendo contra el nodo central de Grid Engine (GE).

**Kernel SMP:** todos los nodos de cálculo tienen un kernel SMP que les permite usar, si es el caso, su capacidad multinúcleo.

**Reenvío de los trabajos en caso de fallo de nodo:** se ha implementado un sistema que permite el reenvío automático de los trabajos que han sufrido un problema en su ejecución a causa del fallo de un nodo de cálculo. Este sistema trabaja directamente con el gestor de colas GE, de modo que es transparente al WMS. Cuando el gestor de colas detecta que un trabajo está corriendo en un nodo que está caído, reenvía directamente el trabajo a colas, con el mismo identificador, pero con una señal de que ha sido reenviado.

**Sensor de GE para medir espacio en disco:** creemos que sería interesante comprobar el espacio libre en disco del nodo de cálculo antes de que un trabajo entre en ejecución en ese nodo. Hemos desarrollado un método mediante el cual el gestor de colas comprueba el espacio libre en disco basándose en la tecnología de sensores ofrecida por GE.

◆  
Dos casos de conexión: uno con máquinas ya configuradas y trabajando en la red, y otro para instalar las nuevas máquinas en el sistema

◆  
OCS Inventory recoge la información de todas las máquinas conectadas a la VPN

**Notificación vía correo electrónico:** como una funcionalidad de posible interés para el usuario hemos implementado una colección de scripts en bash que nos permiten monitorizar todos los trabajos en ejecución a través del WMS, y avisar al usuario de cuales han finalizado correctamente.

## 4. Portal web G-Fluxo

Aún en estos días donde las tecnologías computacionales distribuidas son comunes, el uso de infraestructuras computacionales complejas está ligada fuertemente a la línea de comandos. Incluso el uso de una infraestructura clúster que pueda estar presente en la LAN de un usuario, habitualmente carece de un interfaz gráfico que permita la ejecución de simulaciones complejas que involucran múltiples trabajos interrelacionados.

En este sentido, el proyecto G-Fluxo está dedicado a desarrollar un portal web de fácil actualización que asegure una amplia compatibilidad entre diferentes plataformas del lado del cliente y que permita el uso coordinado de una o varias infraestructuras computacionales distribuidas.

P-GRADE Grid Portal[3] está formado por una combinación de Portlets (definidos por la especificación Sun JSR 168) los cuales son implementaciones de códigos normalmente Java en el lado del servidor que responden a una petición de un cliente web y actualizan una parte concreta de una página web dada.

Desde el punto de vista del usuario, un portlet frecuentemente es representado visualmente por una ventana con sus correspondientes iconos de ayuda, maximización y minimización dentro de una página web cuya descripción está detallada en un fichero XML de configuración.

Muy habitualmente los portlets son programados con una combinación de Java y JSP junto una serie de ficheros XML de configuración. Estos portlets son desplegados en un contenedor de Portlets mantenido en el servidor.

P-GRADE Grid Portal puede ser considerado como un conjunto de portlets que son distribuidos con sus dependencias junto un contenedor llamado GridSphere[4] que se ejecuta a la vez sobre Apache Tomcat. Estos portlets están centrados en computación Grid e implementan la gestión de trabajos (envío y monitorización), la gestión de certificados usando MyProxy así como la gestión de ficheros en los Store Elements (SE).

Esta plataforma es adecuada para trabajar en entornos Grid soportando diferentes Middlewares (interoperabilidad entre diferentes infraestructuras Grid) y puede ser expandida añadiendo nuevas funcionalidades a través de la integración de nuevos portlets.

Un workflow es un conjunto de trabajos interconectados entre ellos de manera que la salida de uno o varios trabajos puede ser la entrada de otros. El editor de Workflows presente en P-Grade, permite diseñar workflows de manera gráfica indicando para cada trabajo la plataforma donde debe ejecutarse, así como los recursos necesarios para su ejecución (tipo de procesador, memoria, tamaño de fichero, etc.). Es posible que una simulación requiera acceder a bases de datos propietarias residentes en un lugar concreto, utilizar un servidor web o hacer una llamada a un servicio web, ejecutar un trabajo altamente demandante en memoria y con necesidad de una red de baja latencia o ejecutar un estudio paramétrico que involucre cientos o miles de trabajos. Esto implica que el portal de usuario debe poder manejar todas estas plataformas de una forma transparente al usuario permitiéndole una ejecución coordinada en todas ellas. El portal desarrollado persigue este fin por lo que varias modificaciones han sido realizadas en el P-GRADE Grid Portal manteniendo el soporte de workflows presente en esta solución.



Los trabajos que han sufrido un problema en su ejecución son reenviados automáticamente

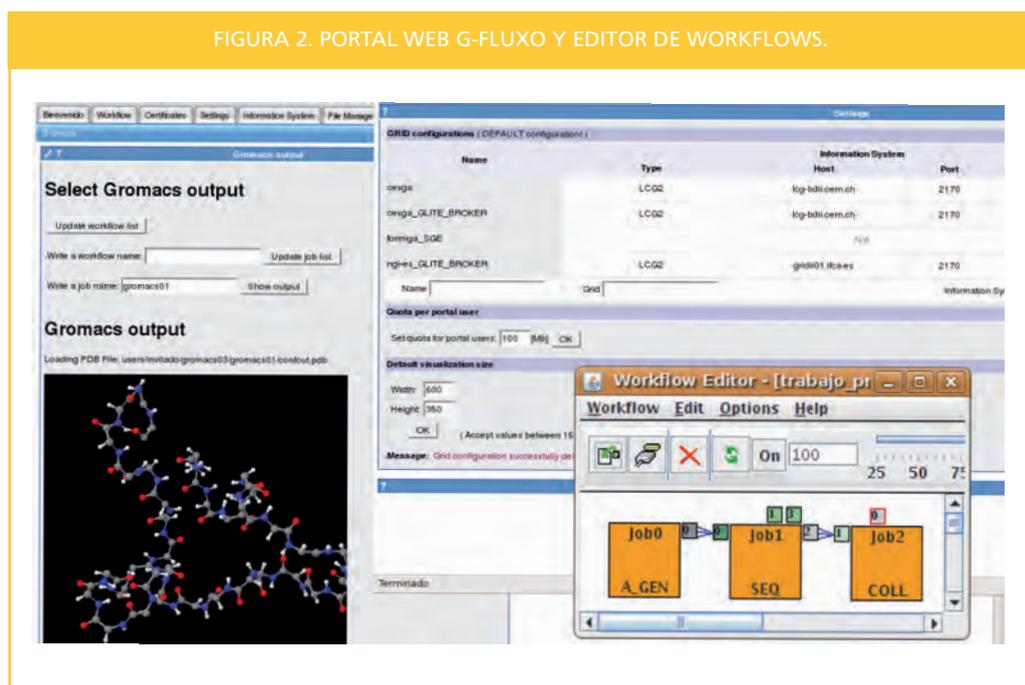


El proyecto G-Fluxo está dedicado a desarrollar el portal web: P-GRADE Grid Portal



P-GRADE Grid Portal presenta una interfaz gráfica para la creación y definición de workflows (Editor de Workflows). Está implementada en java y es ejecutada en lado de cliente a través de Java Web Start.

FIGURA 2. PORTAL WEB G-FLUXO Y EDITOR DE WORKFLOWS.



P-GRADE Grid Portal presenta una interfaz gráfica para la creación y definición de workflows

Tomando como punto de partida del proyecto G-Fluxo la versión 2.7 del P-GRADE Grid Portal varias modificaciones han sido desarrolladas:

1. Integración en el portal de clústers locales que permite el uso de recursos computacionales externos al Grid.
  - Manejo de ficheros y comunicación vía el protocolo SSH.
  - Envío y monitorización de trabajos a los distintos Distributed Resource Management System (DRMS) a través del API DRMAA.
2. Soporte específico de aplicaciones a través del desarrollo de portlets especializados.

El portal desarrollado tiene soporte para la infraestructura del FORMIGA y es una alternativa al envío de trabajos por línea de comandos, permitiendo enviar los trabajos de manera sencilla. Este portal está basado en P-Grade, y permite trabajar con trabajos paramétricos, además de incorporar Servlets que amplían las funcionalidades del portal.

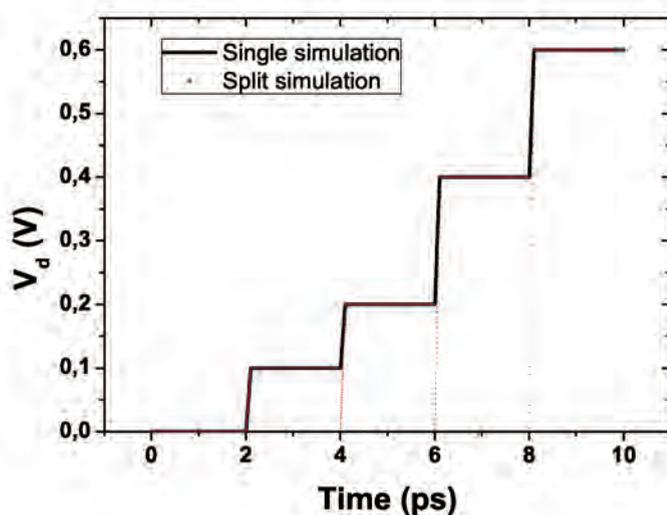
Se ha hecho la integración en el portal de clústers locales, que permite el uso de recursos computacionales externos al Grid

## 5. Adaptación de un Simulador de Nanodispositivos a la Infraestructura Grid del FORMIGA

La VO MOSFET es una organización virtual que ha sido creada para cubrir las necesidades computacionales de la simulación de dispositivos electrónicos utilizando la infraestructura grid que proporciona la red de nacional e-Ciencia[5]. En la actualidad esta VO cuenta con el soporte del Centro de Supercomputación de Galicia, que permite el envío de trabajos a los nodos que forman parte del proyecto FORMIGA y a los nodos de la VO CESGA pertenecientes al proyecto EGEE, mediante la utilización del middleware gLite.

La primera aplicación adaptada a la infraestructura grid del FORMIGA ha sido un simulador Monte Carlo de transistores MOSFET de doble puerta (DG) de silicio sobre aislante (SOI)[6], que permite la realización de estudios paramétricos. El simulador Monte Carlo DG-SOI MOSFET analiza un transitorio en donde la tensión de drenador del dispositivo es conmutada cada 2 ps, tal y como se muestra en la figura 3. La línea continua muestra como sería la polarización del dispositivo cuando la aplicación no está gridificada. En el caso de la aplicación gridificada, línea punteada de la figura, cada intervalo de 2 ps es independiente y puede ser enviado como parámetro a diferentes nodos del grid. La finalidad de esta técnica es reducir el tiempo de ejecución total de un transitorio de 10 ps al tiempo de ejecución de una simulación de 2 ps haciendo uso de un mayor número de recursos.

FIGURA 3. REPRESENTACIÓN DE LA TENSIÓN DE DRENADOR CON EL TIEMPO PARA LA SIMULACIÓN MONTE CARLO SIN GRIDIFICAR Y GRIDIFICADA



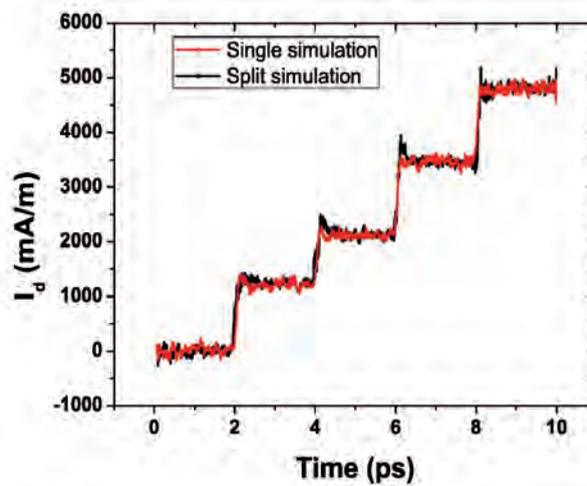
El portal desarrollado tiene soporte para la infraestructura del FORMIGA y es una alternativa al envío de trabajos por línea de comandos

El simulador Monte Carlo DG-SOI-MOSFET realiza estudios paramétricos



Para analizar los resultados se han recopilado los datos de cada una de las simulaciones particionadas, de forma que cada simulación que termina almacena los resultados correspondientes en el Storage Element (SE). Una vez que todas las simulaciones han acabado, los resultados almacenados en el SE se recopilan y se unen formando un fichero de salida único que permite la representación de los datos como se muestra en la figura 4.

FIGURA 4. EVOLUCIÓN DE LA CORRIENTE DE DRENADOR CON EL TIEMPO PARA LA SIMULACIÓN GRIDIFICADA Y SIN GRIDIFICAR



◆  
Cuando las simulaciones han acabado, los resultados se almacenan en el Storage Element (SE)

En la Tabla 1 se muestra el tiempo de ejecución y las características de los nodos de trabajo en donde se han ejecutado cada uno de los intervalos en los que ha sido dividida la simulación transitoria. Los resultados muestran que en el peor de los casos (simulación 2) el tiempo de ejecución es casi un quinto del tiempo total de simulación del transitorio.

◆  
En el peor de los casos, el tiempo de ejecución es casi un quinto del tiempo total de simulación del transitorio

TABLA 1. TIEMPO DE EJECUCIÓN Y CARACTERÍSTICAS DEL NODO DE EJECUCIÓN PARA CADA UNO DE LOS INTERVALOS EN LOS QUE HA SIDO DIVIDIDA LA SIMULACIÓN TRANSITORIA

Id	Time (s)	CPU (GHz)	Ram (MB)
1	6159	Core2 Duo 2.66	512
2	10293	Xeon 1.60	512
3	6447	Core2 Duo 2.66	512
4	6560	Core2 Duo 2.66	512
5	6189	Core2 Duo 2.66	1024
Transient simulation	50655	Xeon 1.60	512

## 6. Conclusiones

En este artículo hemos presentado una descripción de la infraestructura grid del proyecto FORMIGA, que trata de aprovechar los recursos computacionales de las aulas de informática de las universidades gallegas para cálculo científico. Se ha descrito la arquitectura de la misma basada en máquinas virtuales, grid y VPN y las funcionalidades que ofrece a los usuarios.

Adicionalmente también se presenta el portal web que se ha implementado como alternativa al envío de trabajos por línea de comandos que en general resulta excesivamente compleja para muchos usuarios, mediante el cual se pueden mandar los trabajos de una manera más sencilla. Este portal está basado en P-GRADE Portal, y permite enviar trabajos paramétricos, además de incorporar servlets que amplían las funcionalidades del portal.

Para finalizar este trabajo se muestra un caso de uso de la infraestructura, en el que se describe el proceso de adaptación de una aplicación de un simulador Monte Carlo de transistores MOSFET de silicio sobre aislante. De los resultados obtenidos con la ejecución del simulador Monte Carlo en los nodos de computación de la infraestructura, se observa que el tiempo de ejecución de esta aplicación se puede dividir casi por un factor cinco empleando los recursos computacionales disponibles. Además, gracias a la elevada cantidad de ejecuciones que se pueden realizar simultáneamente, este tipo de plataformas son una herramienta muy útil para la simulación de transistores.

El software desarrollado tiene licencia GPL y está disponible a través de la forja de Mancomun (Centro de Referencia e Servicios de Software Libre):

<https://forxa.mancomun.org/projects/formiga/>

<https://forxa.mancomun.org/projects/gfluxo/>

## 7. Agradecimientos

Este trabajo ha sido financiado por la Xunta de Galicia a través de los proyectos PGIDIT07TIC01CT y INCITE08PXIB206094PR y por el Ministerio de Educación y Ciencia bajo el proyecto TIN2007-67537-C03-01.



El proyecto FORMIGA trata de aprovechar los recursos computacionales de las aulas de informática de las universidades gallegas para cálculo científico



El software desarrollado está disponible en la forja de Mancomun



## Referencias:

- [1] Página principal del proyecto FORMIGA: <http://formiga.cesga.es>
- [2] Página principal del proyecto G-FLUXO: <http://gfluxo.cesga.es>
- [3] P. Kacsuk y G. Sipos. *Multi-Grid, Multi-User Workflows in the P-GRADE Portal*. *Journal of Grid Computing*, 3 (3-4). pp. 221-238
- [4] Gridsphere Portal Framework, <http://www.gridisphere.org>
- [5] Wiki e-ciencia VO MOSFET, <http://www.e-ciencia.es/wikilindex.php/mosfet>
- [6] R. Valin, A.Garcia-Loureiro, N. Seoane, M. Aldegunde, C. Sampedro, A. Godoy, y F. Gamiz. *Parallel Implementation of a 2D DGSOI Monte Carlo Simulator Using OpenMP*. Conference on Design of Circuits and Integrated Systems (DCIS), 2008

J. López Cacheiro  
C. Fernández Iglesias  
D. Cordero  
C. Fernández Sánchez  
E. Gutierrez  
A. Rodríguez  
R. Valín

(egee-admin@cesga.es)

Centro de Supercomputación de Galicia (CESGA)

A. Garcia-Loureiro

(egee-admin@cesga.es)

Universidad de Santiago de Compostela