

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ  
FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA  
**UNIVERSIDAD**  
**CATÓLICA**  
DEL PERÚ

**DESARROLLO DE UN FRAMEWORK WEB PARA EL ENVÍO  
REMOTO DE TAREAS, MONITOREO Y RECUPERACIÓN DE  
RESULTADOS PARA DESKTOP GRIDS USANDO UNA  
ARQUITECTURA ORIENTADA A SERVICIOS: CASO BOINC**

**ANEXO A : Prototipos de Pantallas**

## Historial de Revisiones

Historial de revisiones				
Ítem	Fecha	Versión	Descripción	Equipo
1	01/05/10	1	Versión inicial.	Pablo Fonseca
2	13/02/11	1.1	Revisión	Pablo Fonseca
3	01/10/12	2	Revisión	Pablo Fonseca



## Tabla de Contenidos

1. Introducción.....	4
1.1. Alcance.....	4
1.2. Herramienta.....	4
1.3. Reconocimiento.....	4
2. Prototipos de pantallas.....	5
2.1. Inicio.....	5
2.1.1. Login.....	5
2.1.2. Pantalla Inicial de administrador.....	6
2.1.3. Pantalla inicial usuario.....	7
2.2. Creación/Modificación de proyecto.....	8
2.2.1. Datos del proyecto.....	8
2.2.2. Creación Formulario.....	9
2.2.3. Creación Gráfica de Formulario.....	10
2.2.4. Configurar secciones.....	10
2.2.5. Editor de Textbox.....	11
2.2.6. Validación numérica.....	11
2.2.7. Validación de cadena.....	12
2.2.8. Editor de textarea.....	12
2.2.9. Editor de select.....	13
2.2.10. Editor de radio button.....	13
2.2.11. Editor de checkbox.....	14
2.2.12. Editor de file uploader.....	14
2.2.13. Pantalla de usuarios del proyecto.....	15
2.2.14. Pantalla de datos del servicio web.....	16
2.2.15. Pantalla de procesamiento.....	17
2.2.16. Pantalla de archivos del proyecto.....	18
2.3. Administración del sistema.....	19
2.3.1. Parámetros del sistema I.....	19
2.3.2. Parámetros del sistema II.....	20
2.3.3. Parámetros del sistema III.....	21
2.3.4. Parámetros del sistema IV.....	22
2.3.5. Parámetros del sistema V.....	23
2.3.6. Listado de usuarios.....	24
2.3.7. Creación de usuarios.....	25
2.4. Usuarios.....	26
2.4.1. Perfil.....	26
2.4.2. Monitoreo de tareas por proyecto.....	27
2.4.3. Crear Tarea – Formulario Renderizado.....	28
2.4.4. Compartir tarea.....	29

## 1. Introducción

El propósito del presente documento es mostrar los prototipos de pantallas desarrollados como parte del proceso de entendimiento de requerimientos. El fin principal de esta tarea clarificar los requerimientos y retro-alimentar el modelado del sistema en las etapas de análisis y diseño.

Si bien no se incluye una especificación de casos de uso en el desarrollo de este sistema, los prototipos de pantalla se consideraron el medio para validar el entendimiento correcto de los requerimientos del cliente. Esto se cumple en el presente proyecto porque el equipo de desarrollo es pequeño y se tiene acceso inmediato al cliente.

### 1.1. Alcance

El presente documento recoge los prototipos de pantallas con el propósito de clarificar requerimientos, y en segunda instancia representar una guía para el desarrollo del sistema.

### 1.2. Herramienta

Se utilizó la herramienta Balsamiq Mockups por ser de fácil uso, multiplataforma y por contar con una licencia.

### 1.3. Reconocimiento

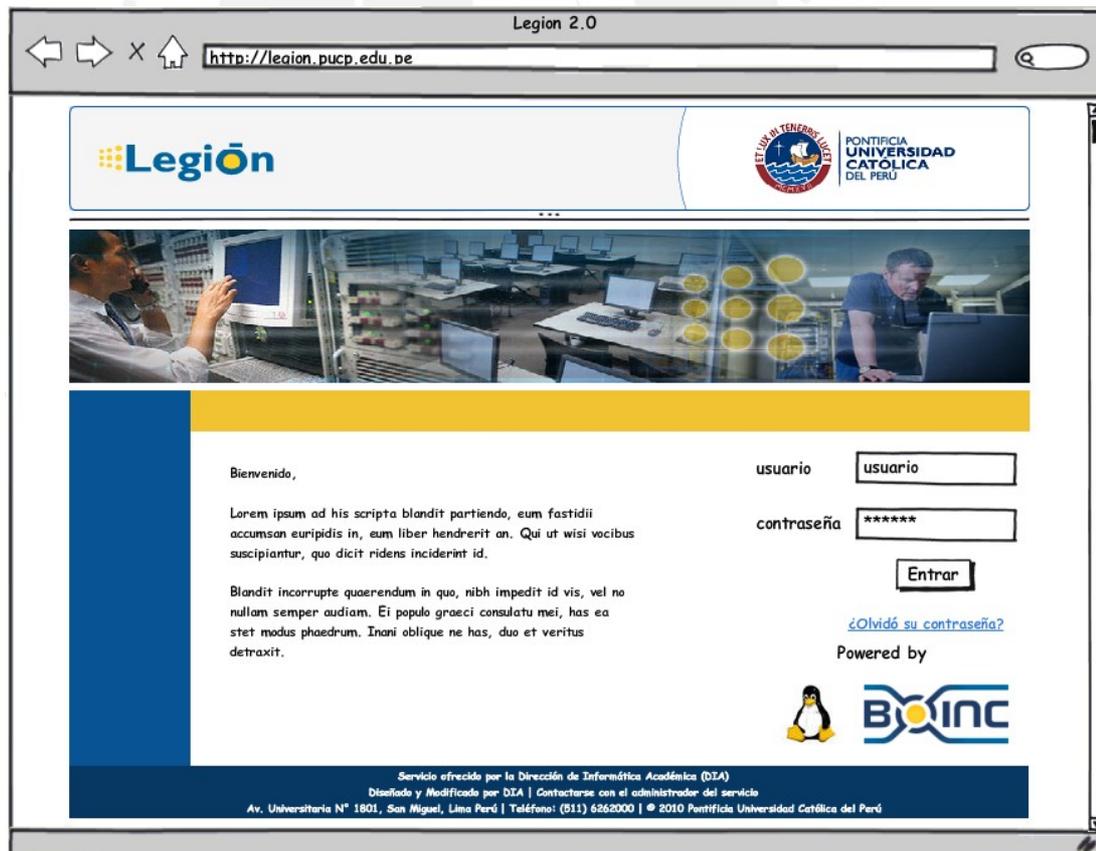
El diseño de la interfaz está basado en los lineamientos del sistema Legión 1.0

## 2. Prototipos de pantallas

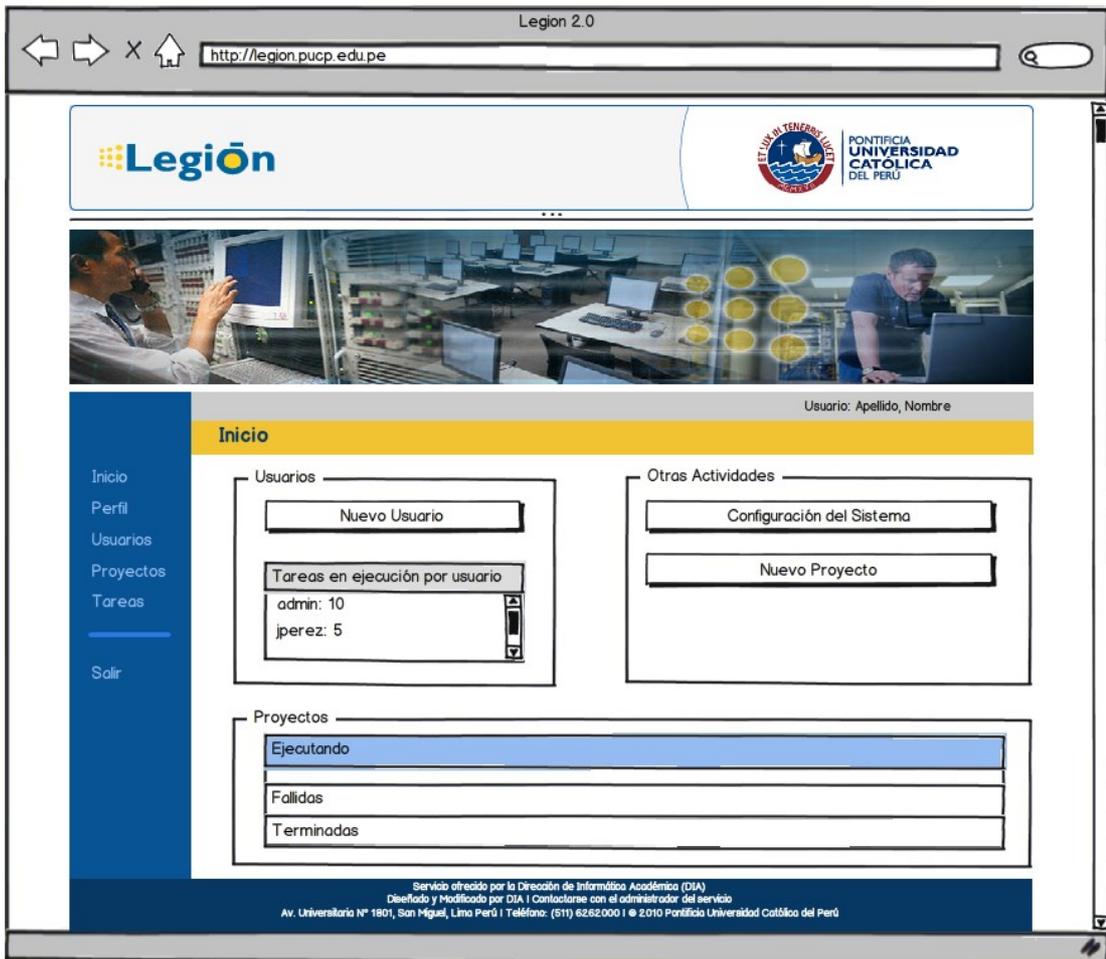
A continuación se presentan los prototipos de pantallas, seguidos en caso requerirse de una breve explicación.

### 2.1.Inicio

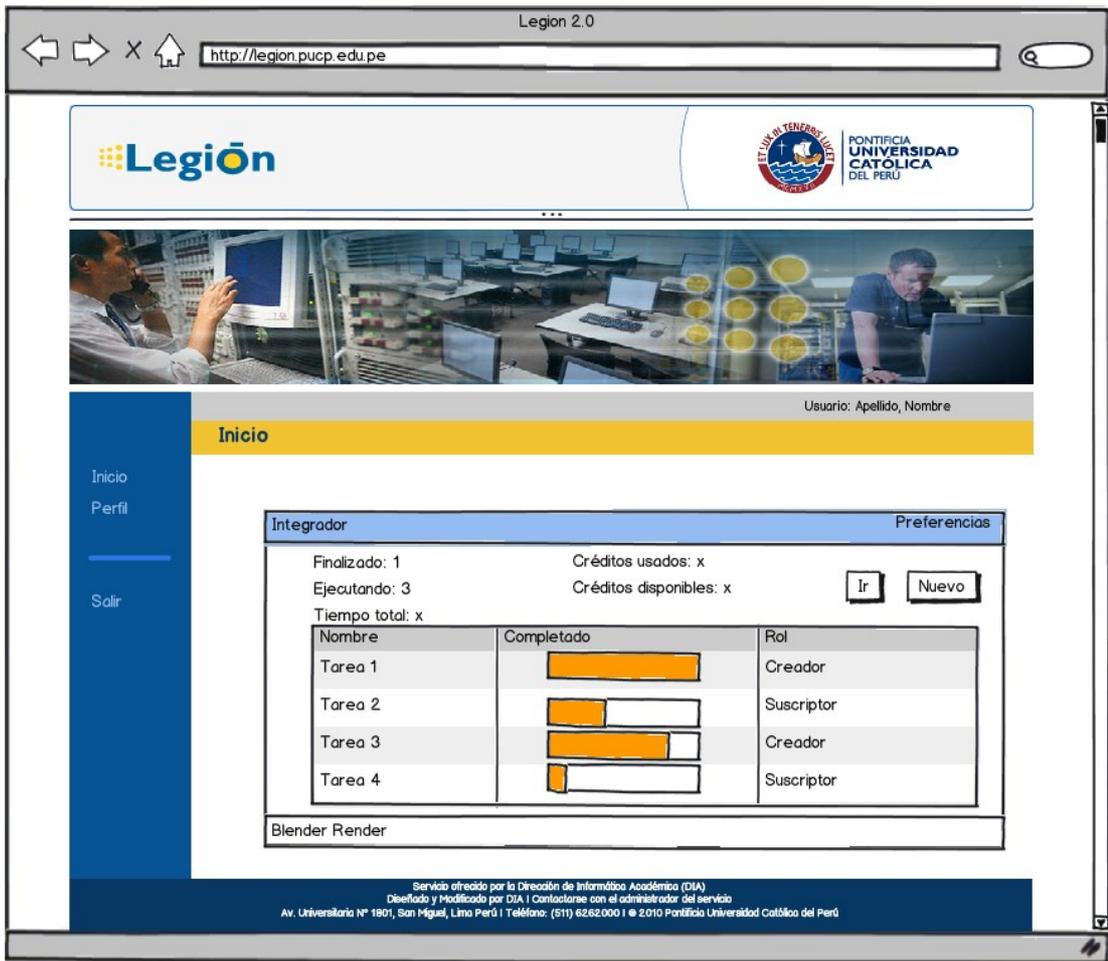
#### 2.1.1.Login



### 2.1.2. Pantalla Inicial de administrador



2.1.3. Pantalla inicial usuario



Legion 2.0

http://legion.pucp.edu.pe

**Legión**

PONTIFICIA  
UNIVERSIDAD  
CATÓLICA  
DEL PERÚ

Usuario: Apellido, Nombre

**Inicio**

Inicio  
Perfil  
Salir

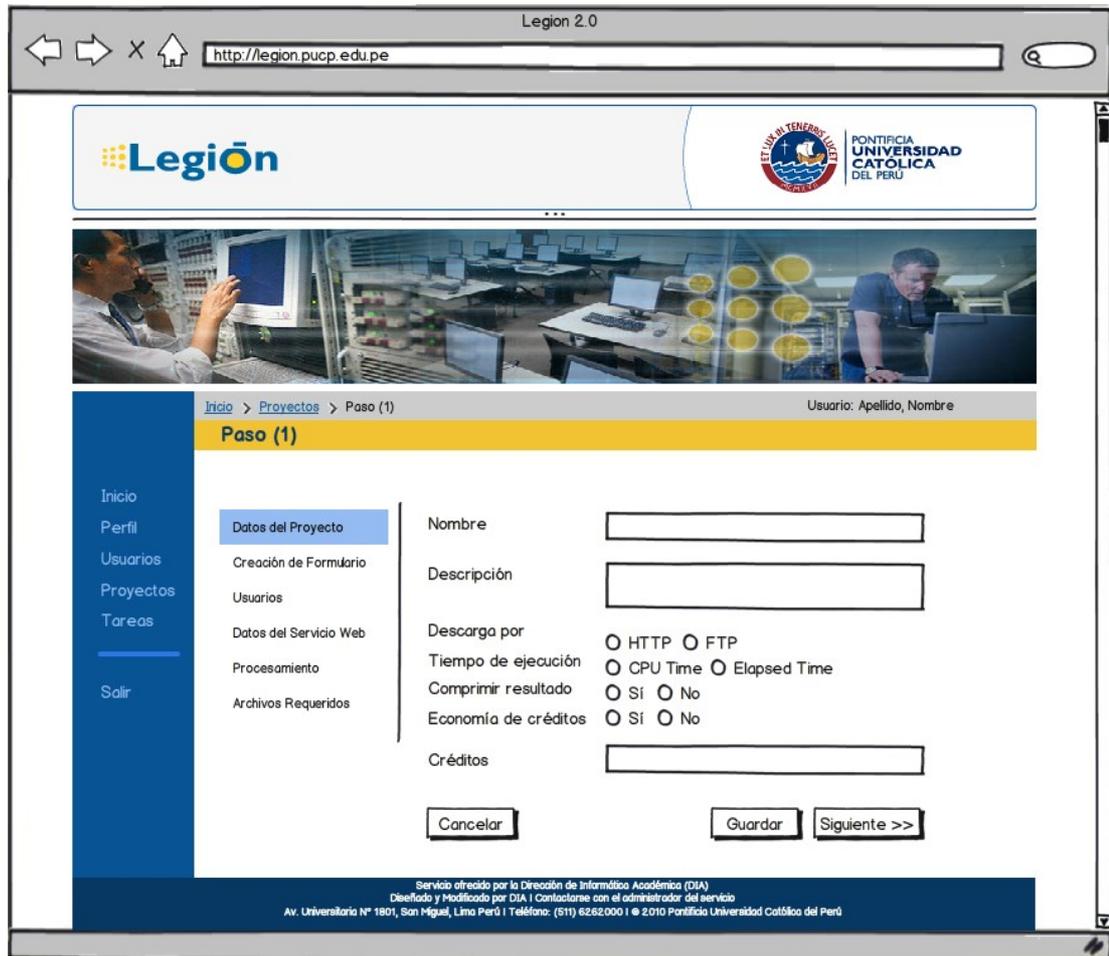
Integrador		Preferencias
Finalizado: 1	Créditos usados: x	<input type="button" value="Ir"/> <input type="button" value="Nuevo"/>
Ejecutando: 3	Créditos disponibles: x	
Tiempo total: x		
Nombre	Completado	Rol
Tarea 1	<div style="width: 80%; background-color: orange;"></div>	Creador
Tarea 2	<div style="width: 20%; background-color: orange;"></div>	Suscriptor
Tarea 3	<div style="width: 70%; background-color: orange;"></div>	Creador
Tarea 4	<div style="width: 10%; background-color: orange;"></div>	Suscriptor

Blender Render

Servicio ofrecido por la Dirección de Informática Académica (DIA)  
Diseñado y Modificado por DIA | Contáctese con el administrador del servicio  
Av. Universitaria N° 1801, San Miguel, Lima Perú | Teléfono: (511) 62.62.000 | © 2010 Pontificia Universidad Católica del Perú

## 2.2.Creación/Modificación de proyecto

### 2.2.1.Datos del proyecto



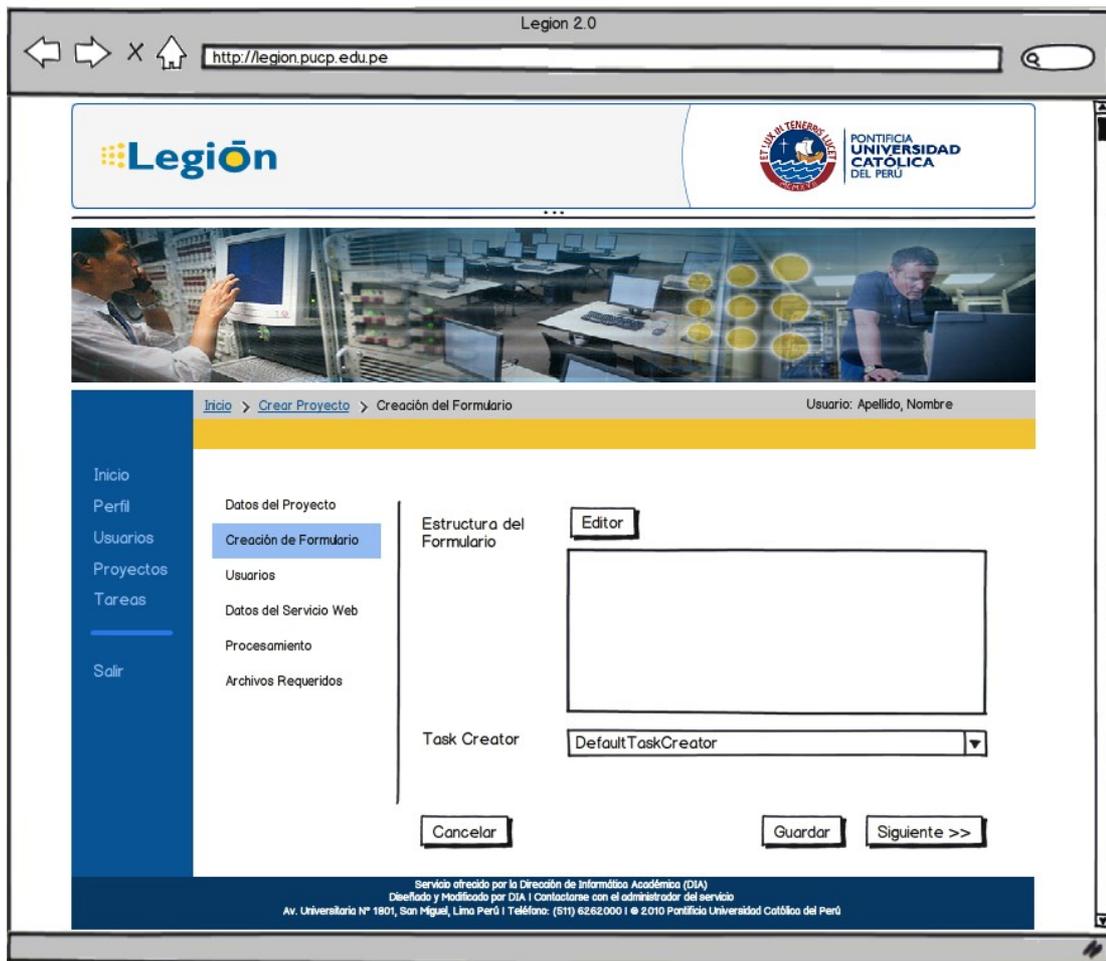
The screenshot shows a web browser window with the URL `http://legion.pucp.edu.pe`. The page header includes the 'Legión' logo and the Pontificia Universidad Católica del Perú logo. Below the header is a banner image of a laboratory. The main content area is titled 'Paso (1)' and contains a form for project data. The form includes a sidebar menu with options like 'Inicio', 'Perfil', 'Usuarios', 'Proyectos', 'Tareas', and 'Salir'. The 'Datos del Proyecto' section contains the following fields and options:

- Nombre:** Text input field.
- Descripción:** Text input field.
- Descarga por:** Radio buttons for  HTTP and  FTP.
- Tiempo de ejecución:** Radio buttons for  CPU Time and  Elapsed Time.
- Comprimir resultado:** Radio buttons for  Sí and  No.
- Economía de créditos:** Radio buttons for  Sí and  No.
- Créditos:** Text input field.

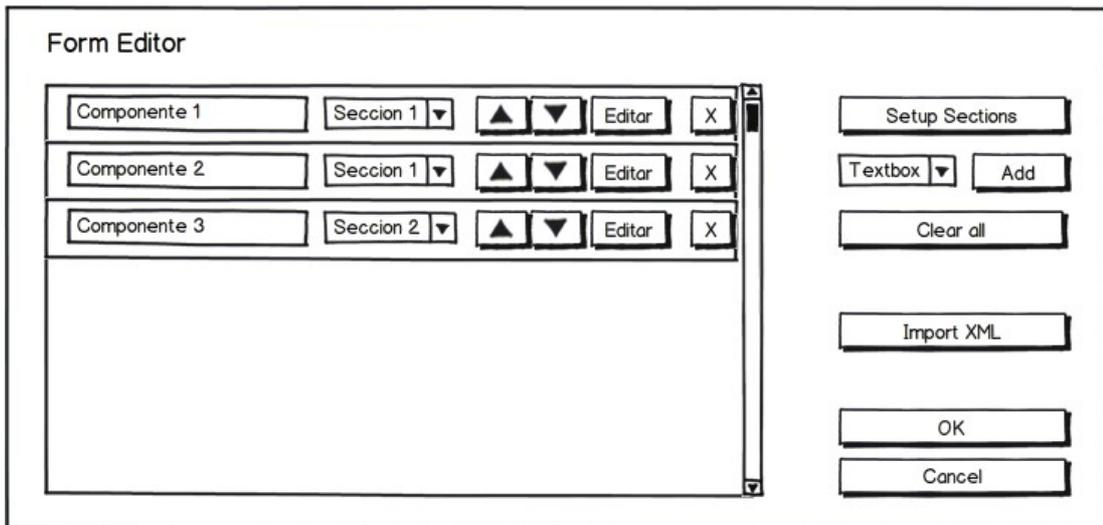
At the bottom of the form are three buttons: 'Cancelar', 'Guardar', and 'Siguiente >>'. The footer of the page contains the following text:

Servicio ofrecido por la Dirección de Informática Académica (DIA)  
Diseñado y Modificado por DIA | Contactarse con el administrador del servicio  
Av. Universitaria N° 1801, San Miguel, Lima Perú | Teléfono: (511) 6262.000 | © 2010 Pontificia Universidad Católica del Perú

2.2.2.Creación Formulario

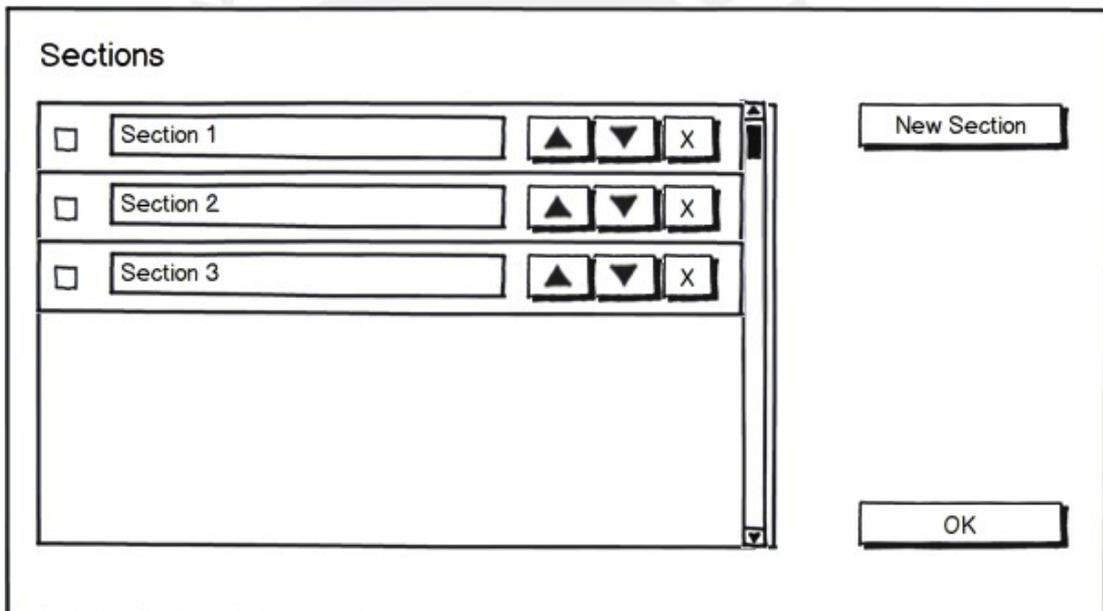


### 2.2.3. Creación Gráfica de Formulario



The screenshot shows a 'Form Editor' window. On the left, there is a list of three components: 'Componente 1', 'Componente 2', and 'Componente 3'. Each component has a dropdown menu for 'Seccion' (Seccion 1 for the first two, Seccion 2 for the third), up and down arrow buttons, an 'Editar' button, and an 'X' button. On the right side, there are several control buttons: 'Setup Sections', 'Textbox' (with a dropdown arrow) and 'Add', 'Clear all', 'Import XML', 'OK', and 'Cancel'.

### 2.2.4. Configurar secciones



The screenshot shows a 'Sections' configuration window. On the left, there is a list of three sections: 'Section 1', 'Section 2', and 'Section 3'. Each section has a checkbox, a text input field, up and down arrow buttons, and an 'X' button. On the right side, there are two buttons: 'New Section' and 'OK'.

## 2.2.5. Editor de Textbox

**Textbox Editor**

HTML id

Label

Message

Validation

## 2.2.6. Validación numérica

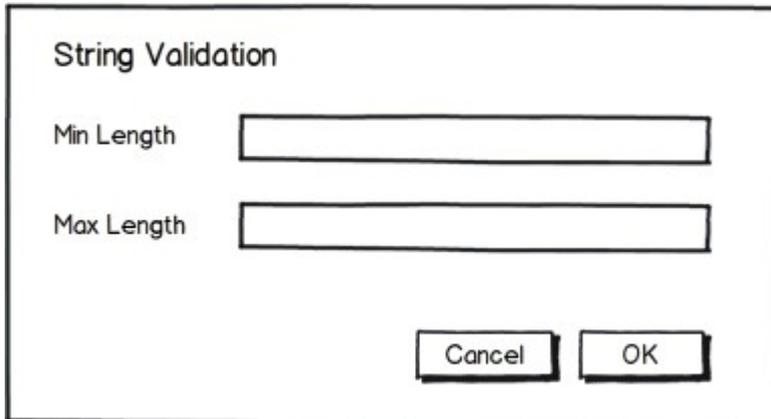
**Number Validation**

Min Value

Max Value

Type

## 2.2.7. Validación de cadena

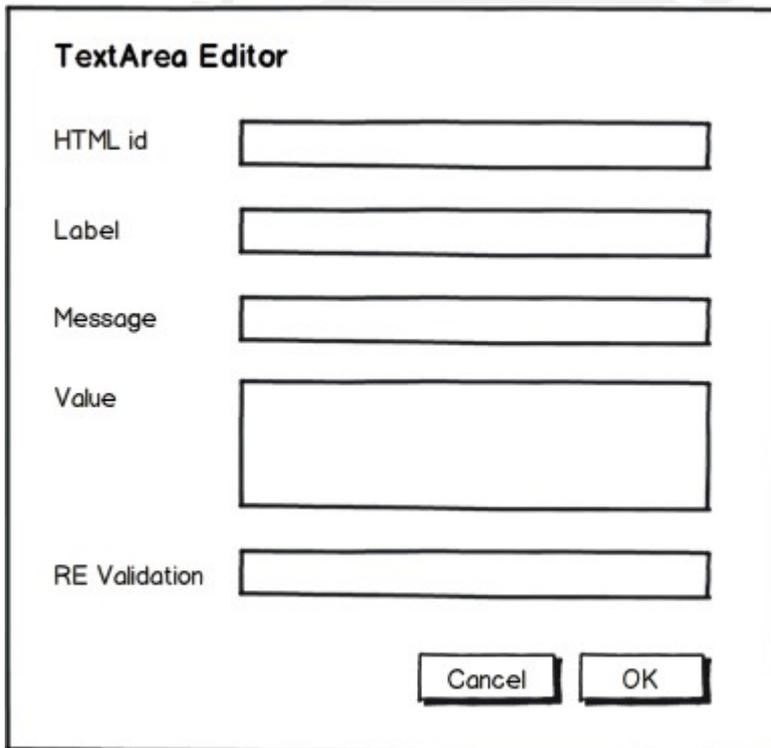


**String Validation**

Min Length

Max Length

## 2.2.8. Editor de textarea



**TextArea Editor**

HTML id

Label

Message

Value

RE Validation

2.2.9. Editor de select

**Select Editor**

HTML id

Label

Message

Options

2.2.10. Editor de radio button

**Radio Editor**

HTML id

Label

Message

Options

2.2.11. Editor de checkbox

**Checkbox Editor**

HTML id

Label

Message

Default  Checked

2.2.12. Editor de file uploader

**FileUpload Editor**

HTML id

Label

Message

2.2.13. Pantalla de usuarios del proyecto



Legion 2.0

http://legion.pucp.edu.pe

**Legión**

PONTIFICIA  
UNIVERSIDAD  
CATÓLICA  
DEL PERÚ

Inicio > Proyectos > Paso (3) Usuario: Apellido, Nombre

**Paso (3)**

Inicio  
Perfil  
Usuarios  
Proyectos  
Tareas  
Salir

Datos del Proyecto  
Creación de Formulario  
**Usuarios**  
Datos del Servicio Web  
Procesamiento  
Archivos Requeridos

Usuarios

Usuario 1  
Usuario 2

Usuarios creadores

Usuario 3

Usuarios suscriptores

Usuario 4

Cancelar Guardar Siguiente >>

Servicio otorgado por la Dirección de Informática Académica (DIA)  
Diseñado y Modificado por DIA | Consultarse con el administrador del servicio  
Av. Universitaria N° 1801, San Miguel, Lima Perú | Teléfono: (511) 62.62.000 | © 2010 Pontificia Universidad Católica del Perú

2.2.14. Pantalla de datos del servicio web



The screenshot shows a web browser window with the URL `http://legion.pucp.edu.pe`. The page title is "Legión 2.0". The header includes the "Legión" logo and the Pontificia Universidad Católica del Perú logo. Below the header is a banner image of a laboratory. The main content area is titled "Paso (4)" and shows a navigation menu on the left with options: Inicio, Perfil, Usuarios, Proyectos, Tareas, and Salir. The "Proyectos" menu item is selected. The main content area is divided into two columns. The left column contains a list of options: Datos del Proyecto, Creación de Formulario, Usuarios, Datos del Servicio Web (highlighted), Procesamiento, and Archivos Requeridos. The right column contains a form with the following fields: Backend (dropdown menu with "Nuevo" selected), Nombre, Descripción, Legion WS, Legion WS User, Legion WS Email, Legion WS Password, and Legion WS Result URL. At the bottom of the form are three buttons: "Cancelar", "Guardar", and "Siguiente >>".

Legion 2.0

`http://legion.pucp.edu.pe`

**Legión**

PONTIFICIA  
UNIVERSIDAD  
CATÓLICA  
DEL PERÚ

Inicio > **Proyectos** > Paso (4) Usuario: Apellido, Nombre

**Paso (4)**

Inicio  
Perfil  
Usuarios  
Proyectos  
Tareas  
**Sair**

Datos del Proyecto  
Creación de Formulario  
Usuarios  
**Datos del Servicio Web**  
Procesamiento  
Archivos Requeridos

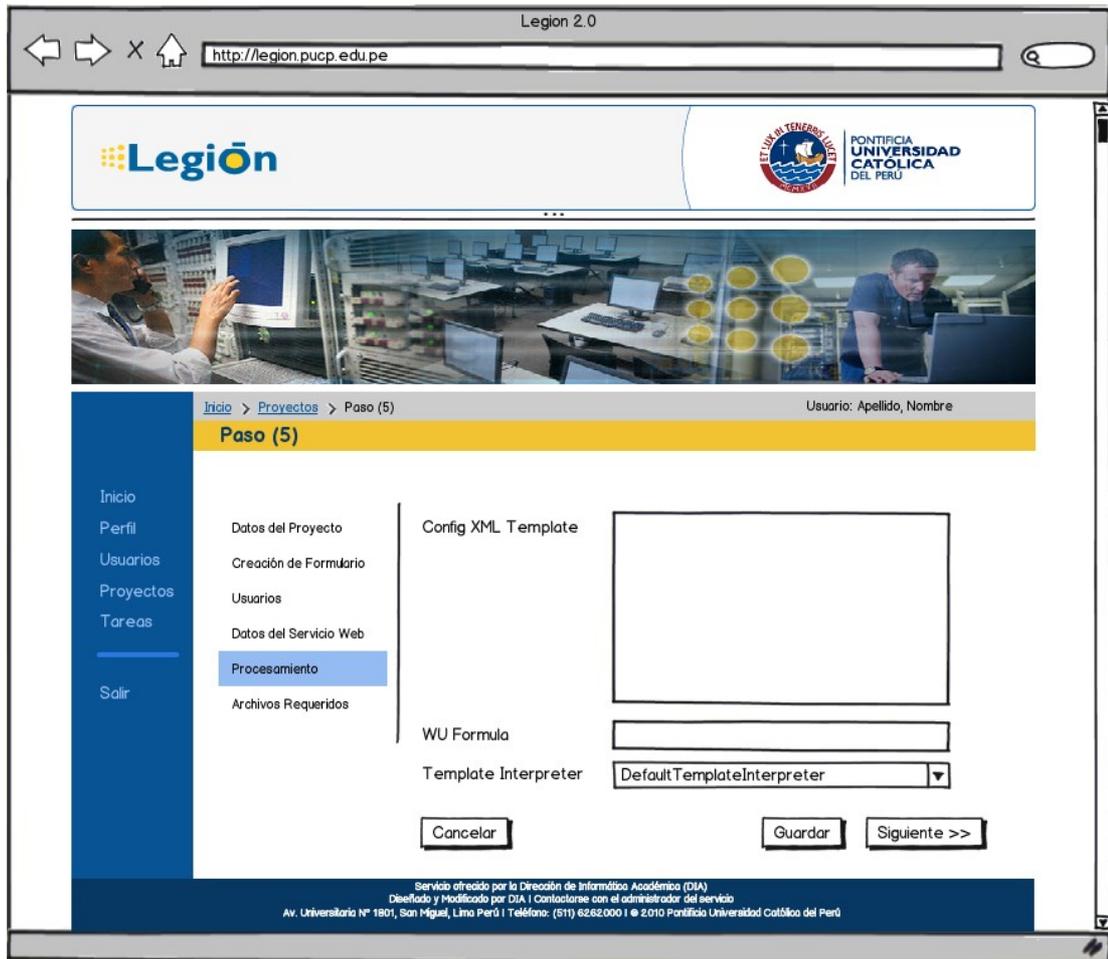
Backend  
Nombre  
Descripción  
Legion WS  
Legion WS User  
Legion WS Email  
Legion WS Password  
Legion WS Result URL

Nuevo

Cancelar Guardar Siguiente >>

Servicio otorgado por la Dirección de Informática Académica (DIA)  
Diseñado y Modificado por DIA | Contáctese con el administrador del servicio  
Av. Universitaria N° 1801, San Miguel, Lima Perú | Teléfono: (511) 62.62.000 | © 2010 Pontificia Universidad Católica del Perú

2.2.15. Pantalla de procesamiento



2.2.16. Pantalla de archivos del proyecto



The screenshot shows a web browser window with the URL <http://legion.pucp.edu.pe>. The page header includes the 'Legión' logo and the Pontificia Universidad Católica del Perú logo. A navigation menu on the left lists: Inicio, Perfil, Usuarios, Proyectos, Tareas, and Salir. The main content area is titled 'Paso (6)' and shows a list of project files under the heading 'Archivos Requeridos'. The files are organized into two sections: 'Exec Files' and 'Other Files'. Each file entry includes a name, an 'Examinar' button, an 'Orden' dropdown menu, and an 'Eliminar' checkbox. At the bottom of the main area are 'Cancelar', 'Guardar', and 'Siguiente >>' buttons. A footer contains contact information for the Dirección de Informática Académica (DIA).

Legion 2.0

<http://legion.pucp.edu.pe>

**Legión**

PONTIFICIA  
UNIVERSIDAD  
CATÓLICA  
DEL PERÚ

Inicio > **Proyectos** > Paso (6) Usuario: Apellido, Nombre

**Paso (6)**

Inicio  
Perfil  
Usuarios  
Proyectos  
Tareas  
Salir

Datos del Proyecto  
Creación de Formulario  
Usuarios  
Datos del Servicio Web  
Procesamiento  
Archivos Requeridos

Exec Files [Nuevo](#)

Archivo	Orden	Eliminar
<a href="#">File1</a> ... <input type="button" value="Examinar"/>	1	<input type="checkbox"/>

Other Files [Nuevo](#)

Archivo	Orden	Eliminar
<a href="#">OtherFile1</a> ... <input type="button" value="Examinar"/>	1	<input type="checkbox"/>
<a href="#">OtherFile2</a> ... <input type="button" value="Examinar"/>	2	<input type="checkbox"/>

Servicio otorgado por la Dirección de Informática Académica (DIA)  
Diseñado y Modificado por DIA | Contactarse con el administrador del servicio  
Av. Universitaria N° 1801, San Miguel, Lima Perú | Teléfono: (511) 62.62.000 | © 2010 Pontificia Universidad Católica del Perú

## 2.3. Administración del sistema

### 2.3.1. Parámetros del sistema I



The screenshot shows a web browser window titled "Legion 2.0" with the URL "http://legion.pucp.edu.pe". The page header includes the "Legión" logo and the Pontificia Universidad Católica del Perú logo. Below the header is a banner image of a server room. The main content area is titled "Opciones del sistema" and features a navigation menu on the left with items: Inicio, Perfil, Usuarios, Proyectos, Tareas, and Salir. The "Opciones del sistema" section has tabs for "Directorio", "Vista", "Seguridad", "Correo", and "Estado del hilo". The "Directorio" tab is active, showing three configuration fields: "Directorio de subida de archivos" with the value "/var/legion/files", "Directorio config.xml" with the value "/var/legion/config", and "Directorio Exec Files" with the value "/var/legion/exec". There are "Cancelar" and "Guardar" buttons at the bottom of the configuration area. A footer at the bottom of the page contains contact information for the Dirección de Informática Académica (DIA).

2.3.2. Parámetros del sistema II



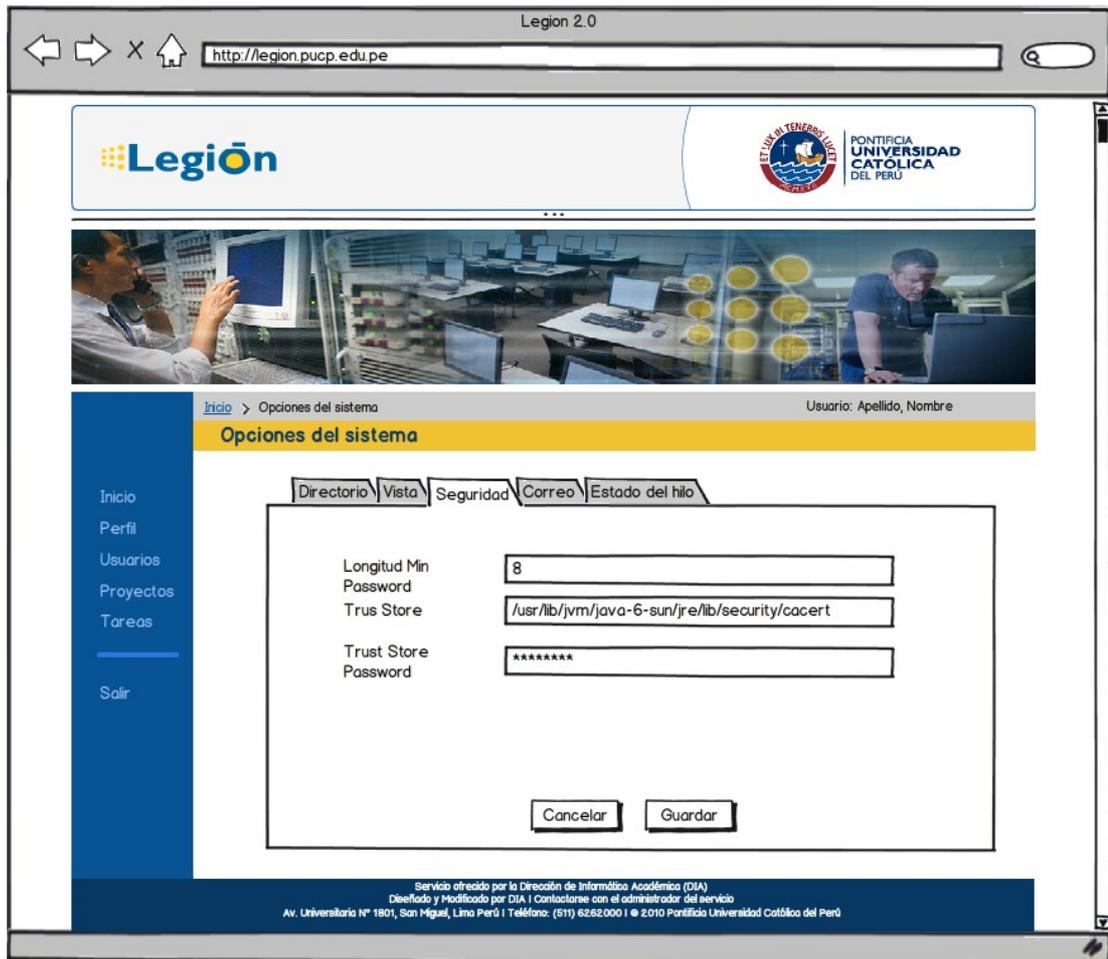
The screenshot shows a web browser window with the URL `http://legion.pucp.edu.pe`. The page header includes the 'Legión' logo and the Pontificia Universidad Católica del Perú logo. Below the header is a banner image of a laboratory. The main content area is titled 'Opciones del sistema' and features a navigation menu on the left with items: Inicio, Perfil, Usuarios, Proyectos, Tareas, and Salir. The 'Opciones del sistema' section has tabs for 'Directorio', 'Vista', 'Seguridad', 'Correo', and 'Estado del hilo'. The 'Directorio' tab is active, showing a form with the following fields:

- Dirección WWW:
- Cantidad filas:
- Cantidad filas-Home Admin:
- Cantidad filas-Home Usuario:
- Idioma:

At the bottom of the form are 'Cancelar' and 'Guardar' buttons. A footer at the bottom of the page contains the following text:

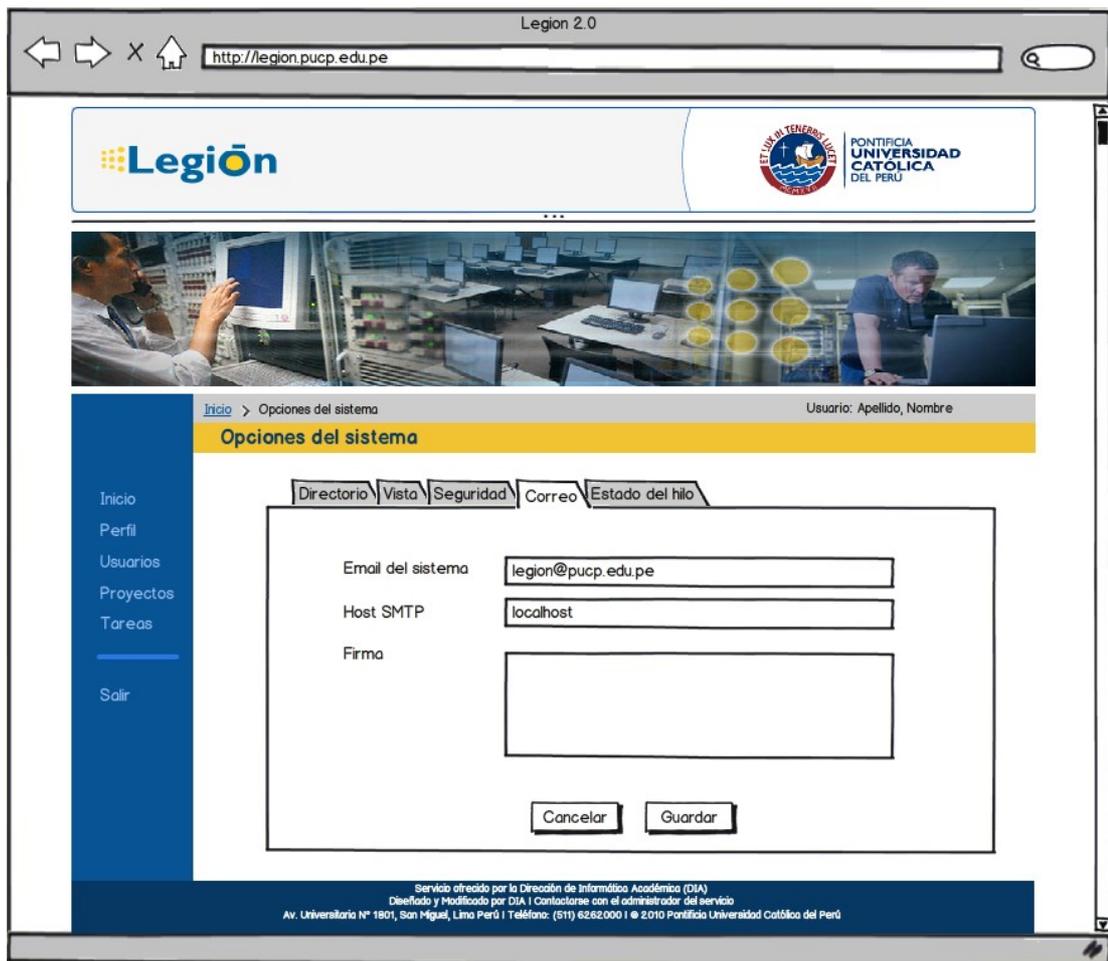
Servicio ofrecido por la Dirección de Informática Académica (DIA)  
Diseñado y Modificado por DIA | Contactarse con el administrador del servicio  
Av. Universitaria N° 1801, San Miguel, Lima Perú | Teléfono: (511) 6262.000 | © 2010 Pontificia Universidad Católica del Perú

### 2.3.3. Parámetros del sistema III

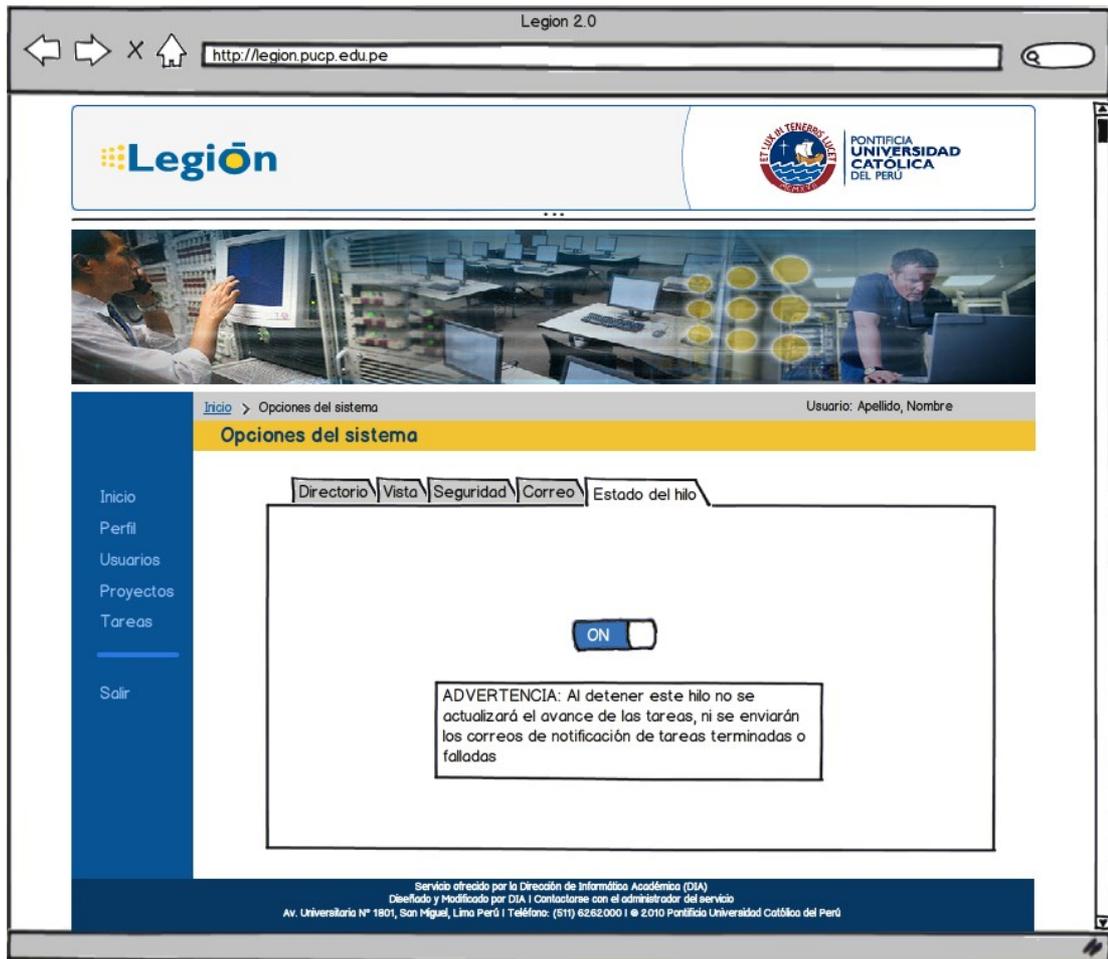


The screenshot shows a web browser window with the URL `http://legion.pucp.edu.pe`. The page title is "Legion 2.0". The header includes the "Legión" logo and the Pontificia Universidad Católica del Perú logo. Below the header is a banner image showing a laboratory setting. The main content area is titled "Opciones del sistema" and includes a navigation menu on the left with items: Inicio, Perfil, Usuarios, Proyectos, Tareas, and Salir. The "Seguridad" tab is selected, showing configuration fields for "Longitud Min Password" (value: 8), "Trus Store" (value: `/usr/lib/jvm/java-6-sun/jre/lib/security/cacert`), and "Trust Store Password" (value: `*****`). "Cancelar" and "Guardar" buttons are at the bottom of the form. A footer contains contact information for the Dirección de Informática Académica (DIA).

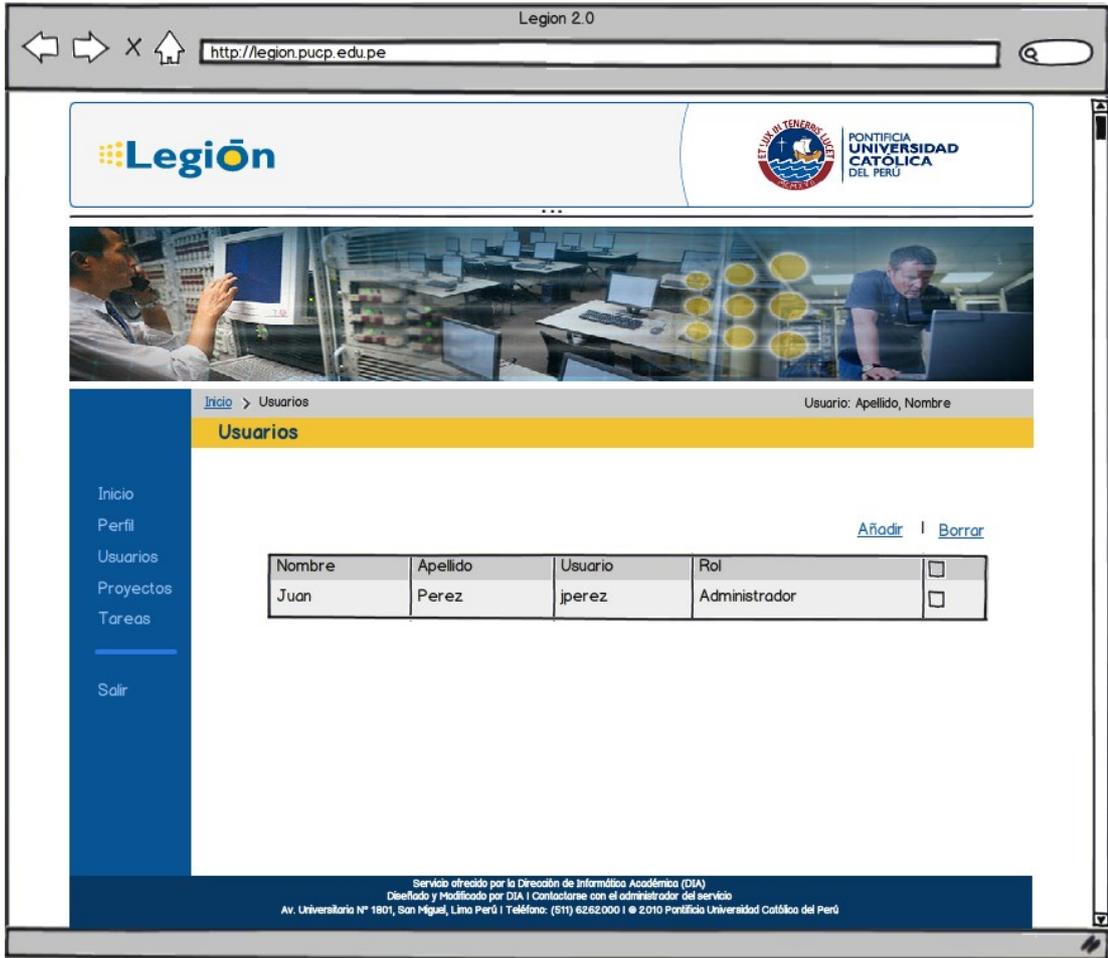
### 2.3.4. Parámetros del sistema IV



2.3.5. Parámetros del sistema V



2.3.6. Listado de usuarios



The screenshot shows a web browser window with the URL `http://legion.pucp.edu.pe`. The page header includes the 'Legión' logo and the Pontificia Universidad Católica del Perú logo. Below the header is a banner image of a laboratory. The main content area is titled 'Usuarios' and features a navigation menu on the left with options: Inicio, Perfil, Usuarios, Proyectos, Tareas, and Salir. The 'Usuarios' section contains a table with one user entry and two action links: 'Añadir' and 'Borrar'.

Inicio > Usuarios Usuario: Apellido, Nombre

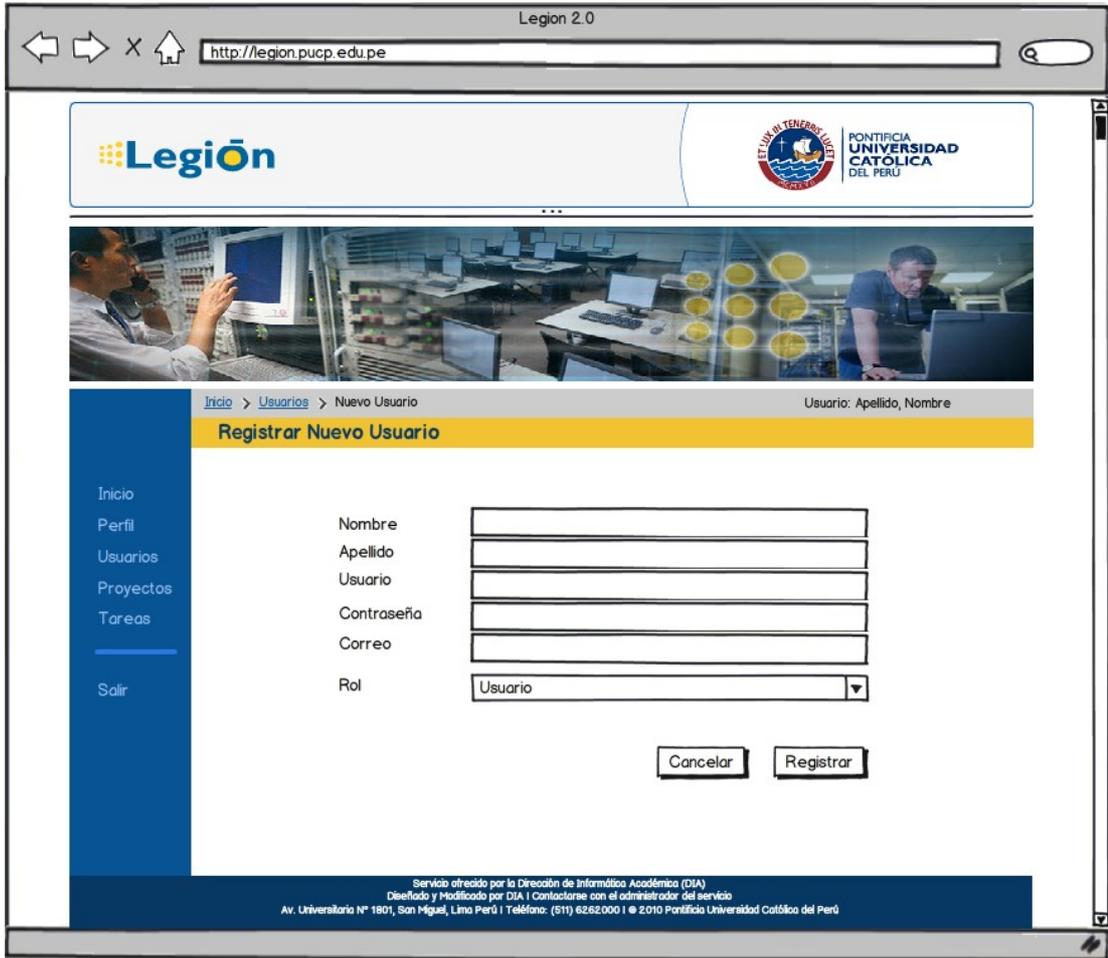
**Usuarios**

[Añadir](#) | [Borrar](#)

Nombre	Apellido	Usuario	Rol	
Juan	Perez	jperez	Administrador	<input type="checkbox"/>

Servicio ofrecido por la Dirección de Informática Académica (DIA)  
 Diseñado y Modificado por DIA | Contactarse con el administrador del servicio  
 Av. Universitaria N° 1801, San Miguel, Lima Perú | Teléfono: (511) 6262.000 | © 2010 Pontificia Universidad Católica del Perú

2.3.7. Creación de usuarios



The screenshot shows a web browser window with the URL <http://legion.pucp.edu.pe>. The page header includes the 'Legión' logo and the Pontificia Universidad Católica del Perú logo. Below the header is a banner image of a laboratory. The main content area is titled 'Registrar Nuevo Usuario' and contains a form with the following fields:

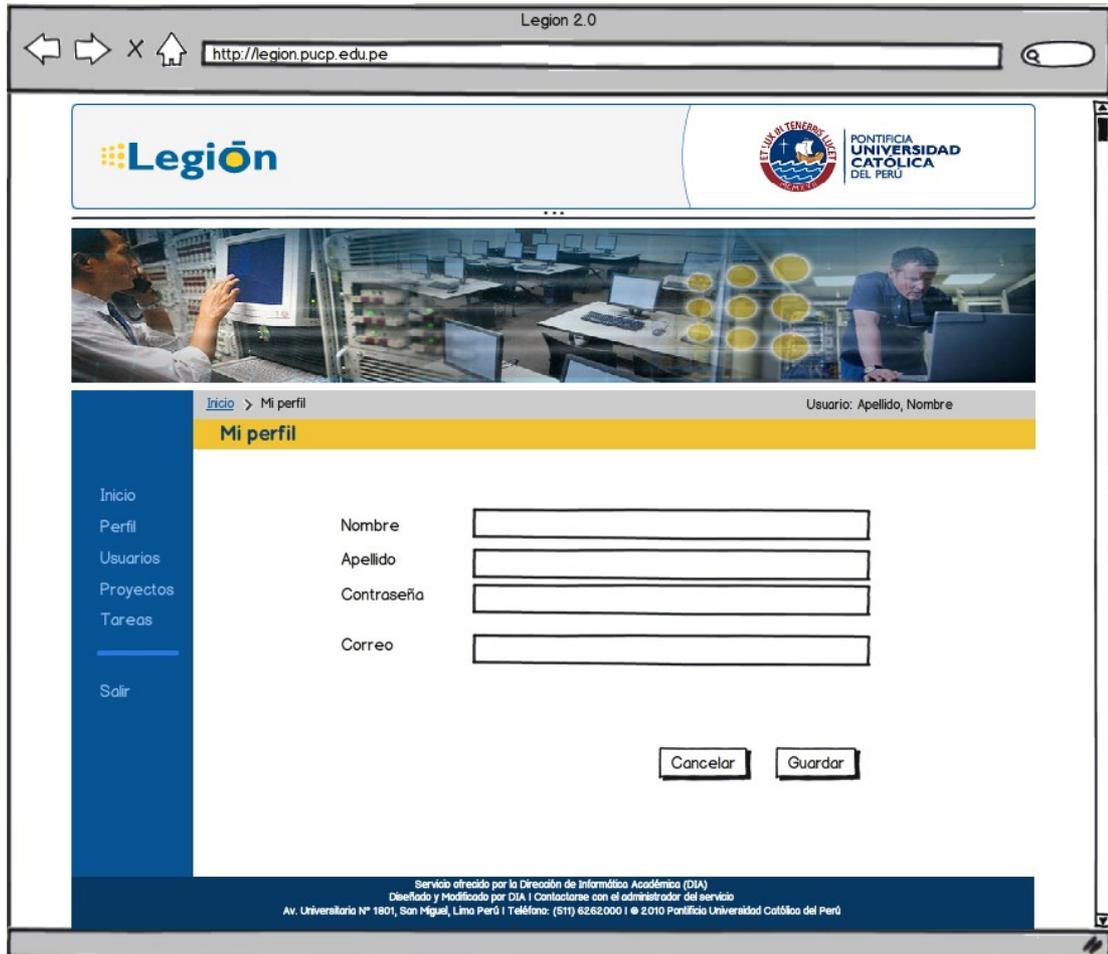
- Nombre:
- Apellido:
- Usuario:
- Contraseña:
- Correo:
- Rol:

At the bottom of the form are 'Cancelar' and 'Registrar' buttons. A footer at the bottom of the page contains the following text:

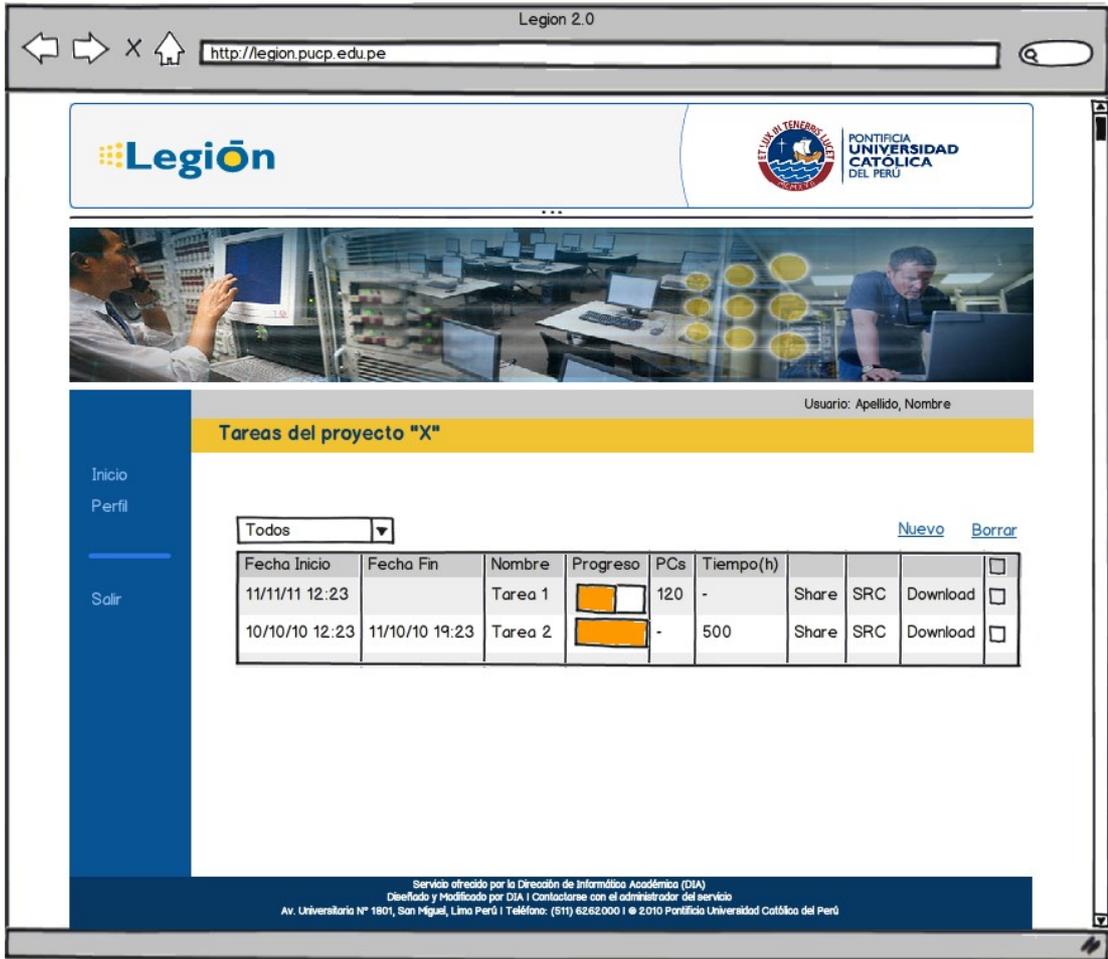
Servicio ofrecido por la Dirección de Informática Académica (DIA)  
Diseñado y Modificado por DIA | Contactarse con el administrador del servicio  
Av. Universitaria N° 1801, San Miguel, Lima Perú | Teléfono: (511) 62.62.000 | © 2010 Pontificia Universidad Católica del Perú

## 2.4. Usuarios

### 2.4.1. Perfil



2.4.2. Monitoreo de tareas por proyecto

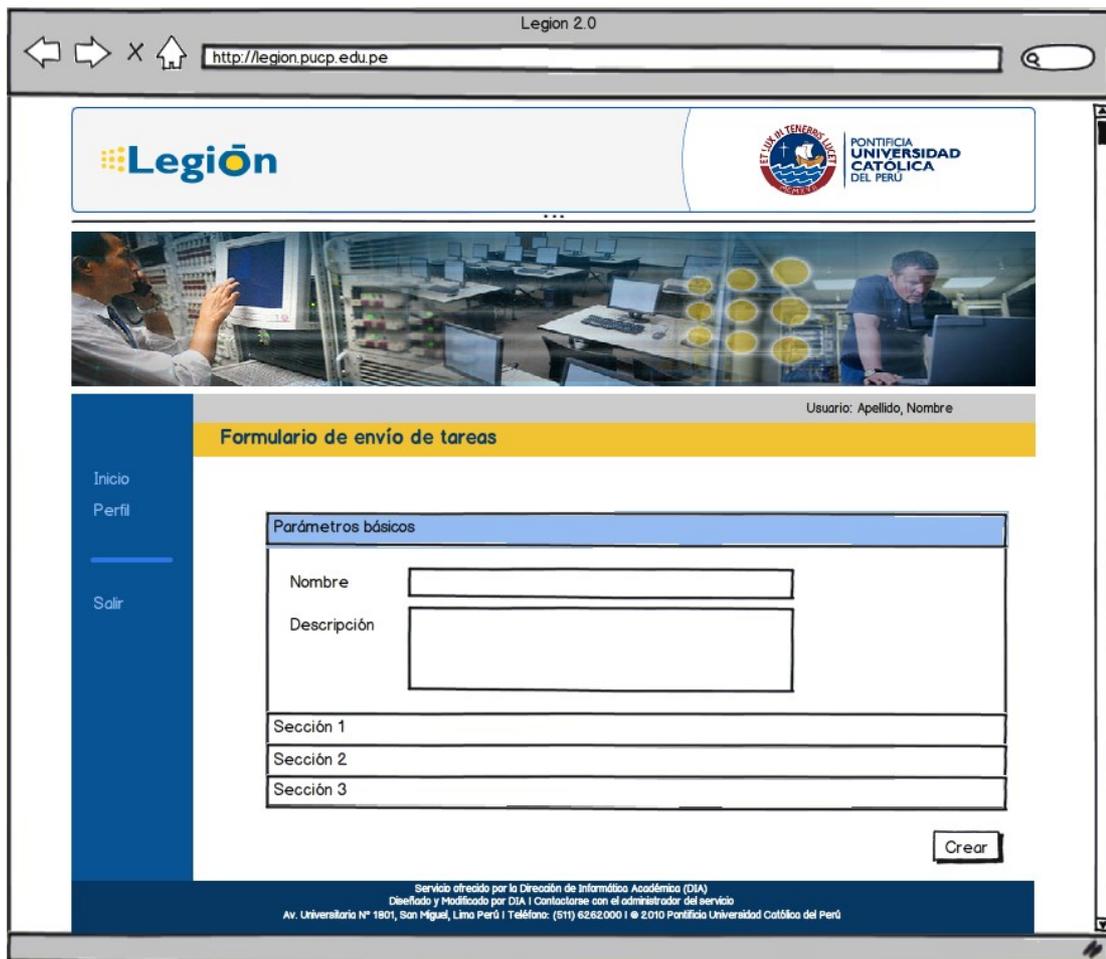


The screenshot shows a web browser window with the URL `http://legion.pucp.edu.pe`. The page header includes the "Legión" logo and the Pontificia Universidad Católica del Perú logo. Below the header is a banner image of a laboratory. The main content area is titled "Tareas del proyecto 'X'" and includes a sidebar with "Inicio", "Perfil", and "Salir" options. A table displays task details:

Fecha Inicio	Fecha Fin	Nombre	Progreso	PCs	Tiempo(h)	Share	SRC	Download	
11/11/11 12:23		Tarea 1	<div style="width: 25%; background-color: orange;"></div>	120	-	Share	SRC	Download	<input type="checkbox"/>
10/10/10 12:23	11/10/10 19:23	Tarea 2	<div style="width: 100%; background-color: orange;"></div>	-	500	Share	SRC	Download	<input type="checkbox"/>

At the bottom of the page, there is a footer with contact information: "Servicio otorgado por la Dirección de Informática Académica (DIA). Diseñado y Modificado por DIA. Contáctese con el administrador del servicio. Av. Universitaria N° 1801, San Miguel, Lima Perú | Teléfono: (511) 6262.000 | © 2010 Pontificia Universidad Católica del Perú".

2.4.3. Crear Tarea – Formulario Renderizado



### 2.4.4. Compartir tarea



PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ  
FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA  
**UNIVERSIDAD**  
**CATÓLICA**  
DEL PERÚ

**DESARROLLO DE UN FRAMEWORK WEB PARA EL ENVÍO  
REMOTO DE TAREAS, MONITOREO Y RECUPERACIÓN DE  
RESULTADOS PARA DESKTOP GRIDS USANDO UNA  
ARQUITECTURA ORIENTADA A SERVICIOS: CASO BOINC**

**ANEXO B : Arquitectura**

## Historial de Revisiones

Historial de revisiones				
Ítem	Fecha	Versión	Descripción	Equipo
1	24/04/10	1	Versión inicial.	Pablo Fonseca
2	06/05/10	1.1	Correcciones y añadidos	Pablo Fonseca
3	01/10/12	2.0	Correcciones	Pablo Fonseca



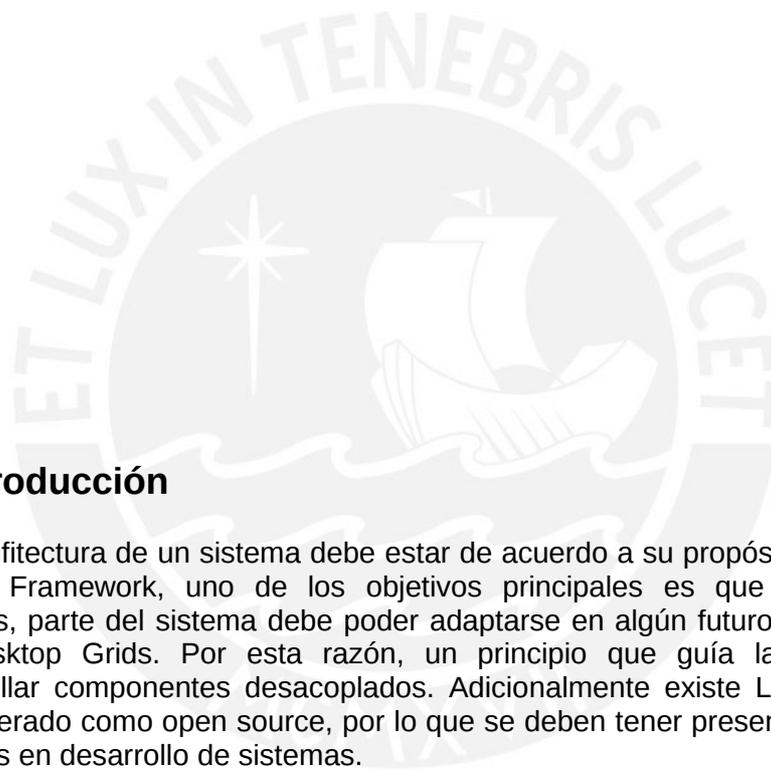
## Tabla de Contenidos

1. Introducción.....	5
1.1. Alcance.....	5
1.2. Herramienta.....	5
2. Especificación de la arquitectura.....	6
2.1.Representación de la arquitectura.....	6
2.2.Metas y restricciones de la arquitectura.....	7
2.2.1.Metas.....	7
2.2.2.Restricciones.....	7
2.3.Despliegue y componentes.....	7
2.3.1.Nodo Cliente.....	8
2.3.2.Nodo Servidor Web .....	8
2.3.3.Nodo Servidor Backend.....	8
2.3.4.Componente: Legión Web Interface .....	8
2.3.5.Componente: Legión Web Services.....	8
2.3.6.Interacción del sistema.....	9
2.3.7.Lineamientos para interacción de las capas.....	9
3. Validación de la arquitectura.....	10
3.1.Pruebas de implementación realizadas.....	10
3.1.1.Servicios Web SOAP en Python.....	10
3.1.2.Servicios Web SOAP en Java.....	11
3.1.3.Servicios Web SOAP en PHP.....	11
3.1.4. Servicios Web REST en Python.....	11
3.2.Validación de casos de uso.....	11
3.2.1.Envío de tareas.....	11
3.2.2.Recuperación de resultados.....	12
4. Conclusiones y direcciones.....	14

## Tabla de Imágenes

Imagen 2.1: Diagrama de arquitectura.....	7
Imagen 3.1: Diagrama de secuencia – Envío de tareas.....	12
Imagen 3.2: Diagrama de secuencia – Recuperar Resultado.....	13





## 1. Introducción

La arquitectura de un sistema debe estar de acuerdo a su propósito. En el caso de Legión Framework, uno de los objetivos principales es que sea mantenible. Además, parte del sistema debe poder adaptarse en algún futuro a otros sistemas de Desktop Grids. Por esta razón, un principio que guía la arquitectura es desarrollar componentes desacoplados. Adicionalmente existe Legión Framework será liberado como open source, por lo que se deben tener presente las tendencias actuales en desarrollo de sistemas.

### 1.1. Alcance

El propósito del presente documento es mostrar la arquitectura del sistema y demostrar que los principales casos de uso son soportados por esta. Adicionalmente, este documento tiene por intención el especificar los patrones que se seguirá en la división de tareas entre los componentes presentados. Se especificará la arquitectura del sistema y se demostrará la conveniencia de su diseño.

### 1.2. Herramienta

Se utilizó la herramienta StarUML para el diseño de diagramas UML.

## 2. Especificación de la arquitectura

En la presente sección se muestra la especificación y detalles de la arquitectura del sistema, así como los criterios que se tomaron en cuenta para ello.

### 2.1.Representación de la arquitectura

#### **Orientado a Servicios**

La arquitectura estará basada en el paradigma orientado a servicios, de modo que en el futuro se puedan incluir nuevos componentes que interactúen con los servicios definidos.

#### **Orientado a Objetos**

El desarrollo de los componentes se hará haciendo uso de la programación orientada a objetos, incluso en la capa web, haciendo uso de tecnología reciente.

#### **Repositorio**

Tanto la capa Legión Web Interface como la capa Legión Web Services usan sistemas administradores de base de datos para persistir los datos.

## Web

Se utilizará un paradigma web, pues es importante que la aplicación se fácilmente accedida, y desde múltiples locaciones.

## 2.2. Metas y restricciones de la arquitectura

Se presentan las metas de la arquitectura como líneas guía para su desarrollo, así mismo como las restricciones siguientes.

### 2.2.1. Metas

Se listan las metas.

- Mantenable – El sistema debe ser mantenible, ya que se espera una vida duradera del sistema.
- Componentes desacoplados – Los componentes deben estar desacoplados, de modo que se pueda trabajar independientemente y adicionalmente puedan ser fácilmente reemplazables.
- Seguridad – El sistema debe ser seguro, pues los datos y resultados de las tareas pueden ser sensibles.
- Arquitectura actual – Se debe usar una arquitectura actual.

### 2.2.2. Restricciones

- Por seguridad sólo dos métodos son permitidos para transferencia de archivos hacia el usuario. Estos son, retransmisión y acceso directo/FTP.
- El sistema funcionará en el sistema operativo RHEL 6 por lo que se debe considerar funcionalidades y paquetes disponibles en esta familia de sistemas.

## 2.3. Despliegue y componentes

Se ha planteado una arquitectura basada en tres componentes, nodo cliente, nodo servidor web y nodo de servicios web. Este último incluye el servidor BOINC.

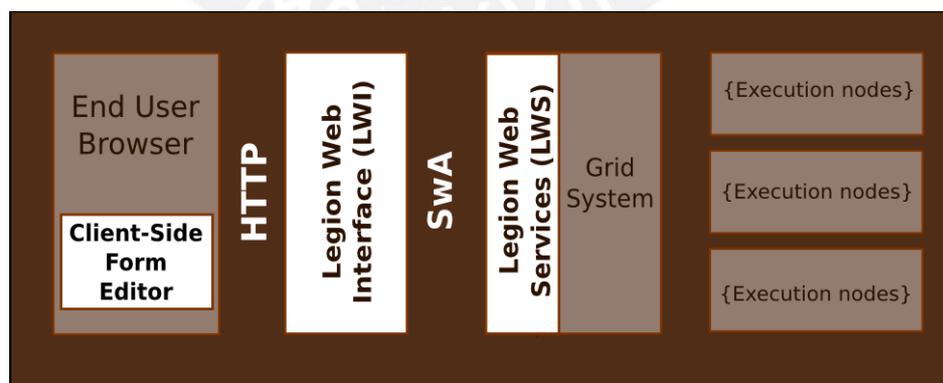


Imagen 2.1: Diagrama de arquitectura

### 2.3.1. Nodo Cliente

Es la computadora del usuario del sistema y contiene únicamente un navegador web, el cual deberá tener habilitadas funcionalidades que permitan ejecutar código en el lado del cliente.

### 2.3.2. Nodo Servidor Web

Es el encargado de recibir y atender las peticiones del cliente a través del protocolo HTTP. Contiene, adicionalmente, un sistema administrador de base de datos, del que cual se vale para obtener y persistir la información generada durante la interacción con el cliente. También, depende del nodo de Legión Web Services que le brinda los servicios que requiere para atender al cliente.

### 2.3.3. Nodo Servidor Backend

Es el nodo más importante del sistema como conjunto porque expone servicios web, que serán consumidos por la capa de Legión Web Interface. Es importante notar que BOINC no tiene servicios web, sino que depende de la capa Legión Web Services la que representa el enlace con las operaciones de bajo nivel expuestas por la API del Servidor BOINC. Ambos componentes, LWS y BOINC dependen de una única instancia de un sistema administrador de bases de datos MySQL.

### 2.3.4. Componente: Legión Web Interface

La capa web maneja el contacto del usuario con el sistema. Por ello es necesario poner especial cuidado en su diseño.

Es conocido que hay muchas formas de realizar el diseño de aplicaciones web. En este caso es necesario contar con generación gráfica de formularios, lo que exige alta interactividad. Debido a esto, se decidió optar por tener funcionalidades del lado del cliente.

Sin embargo, la generación gráfica de formularios es la única parte de la capa web que requiere de alta interactividad. Es posible emular el comportamiento de las aplicaciones de escritorio en aplicaciones web usando frameworks o librerías de javascript.

### 2.3.5. Componente: Legión Web Services

La capa de Legión Web Services expondrá servicios web SOAP. Estos servicios serán consumidos por LWI, pero representa una abstracción mayor y es susceptible a la integración con cualquier cliente que pueda consumir los servicios. Una de las ideas principales al diseñar esta capa es brindar a la comunidad de usuarios BOINC una capa que pueda integrarse con sistemas propios.

Finalmente estos servicios harán uso de la API de BOINC para ejecutar acciones sobre el servidor.

### 2.3.6. Interacción del sistema

Es importante validar que la arquitectura propuesta soporta los flujos principales del sistema. Consideramos importantes:

- Flujo de envío de tareas
- Flujo de recuperación de resultados

Para lograr esto se presenta en la sección 3.2 del presente documento los diagramas de secuencias que permiten observar la interacción entre las capas en los flujos que pueden representar problemas en el sistema.

### 2.3.7. Lineamientos para interacción de las capas

Se presentan algunos lineamientos para la interacción de los componentes del sistema.

- El usuario, cuando recupera el resultado de una tarea, puede hacerlo por retransmisión desde la capa web en caso de tratarse de un solo archivo o también, puede recibir una dirección ftp, de donde puede descargar los archivos.
- La capa web recibe una URL a la que puede acceder con el fin de retransmitir el contenido de algún archivo hacia el usuario.
- Cuando se envía una tarea, dependiendo del número de unidades de trabajo que implique, puede demorar mucho en el proceso de creación. Por eso no es factible confirmar la creación de la tarea, debido a eso se considera un estado "En creación".

### **3. Validación de la arquitectura**

Con el propósito de validar la arquitectura, se realizaron pruebas de implementación y se realizaron diagramas de secuencia entre capas.

#### **3.1. Pruebas de implementación realizadas**

Se realizaron pruebas de implementación, de modo que se pueda evaluar la adaptación de tecnologías para la resolución del problema.

##### **3.1.1. Servicios Web SOAP en Python**

Se implementaron servicios web en Python con dos librerías. La primera es The Zolera SOAP Infrastructure (ZSI 2.0). Se logró desarrollar un servidor y se realizaron pruebas con un cliente en Java. Se logró transferir archivos binarios codificados como base64 dentro de la respuesta SOAP. Sin embargo no se logró usar el estándar SwA (SOAP with Attachments) que usa MIME/Multipart.

La segunda librería usada es soaplib. Esta librería es más reciente que ZSI, y tiene implementado el soporte para SwA. El uso de este estándar es recomendado debido a que el tamaño de los archivos binarios a transferir puede oscilar entre 1 y 15 MB. Se logró implementar un servicio que se accedió con clientes remotos programados en java y en python.

### 3.1.2. Servicios Web SOAP en Java

Se utilizó la librería Apache AXIS2 y el contenedor de servlets Apache Tomcat 7 en un entorno Linux. Se logró transferir archivos binarios con un cliente desarrollado en java. Soporta el estándar SwA. Tiene una característica importante: Los archivos binarios son volcados a disco duro mientras son transferidos, es decir no residen en la memoria RAM durante este proceso.

### 3.1.3. Servicios Web SOAP en PHP

Se compiló e instaló la librería WSO2 WSF/PHP en un entorno Linux. Es la única librería en PHP que soporta actualmente el estándar SwA. Sin embargo, se realizaron pruebas con esta librería como cliente únicamente.

### 3.1.4. Servicios Web REST en Python

Se utilizó la librería Piston, que es una aplicación del framework de desarrollo web de Django. Esta librería provee una implementación de servicios web REST. Se logró transferir archivos. Se implementó en un entorno Linux y se evaluó el consumo de memoria en un contexto de concurrencia. El resultado indica que durante la transferencia de un archivo, se guarda en memoria la parte transferida hasta que se completa. Esto puede representar problemas en un contexto de alta concurrencia.

## 3.2. Validación de casos de uso

Con el propósito de validar la arquitectura, y entender de una mejor manera la responsabilidad de tareas entre las capas se presentan dos flujos importantes para el sistema: el envío de tareas y la recuperación de resultados. La presentación es en forma de diagramas de secuencia entre las capas.

### 3.2.1. Envío de tareas

El envío de tareas es una de las mayores funcionalidades del sistema junto con la recuperación de resultados. A continuación se explica el flujo.

1. El usuario ingresa al sistema y selecciona la opción de ver el formulario de creación de una tarea en el contexto de un proyecto específico.
2. El formulario se genera/renderiza dinámicamente, es decir, la capa web ejecuta una acción para lograr esto. Finalmente se devuelve el formulario HTML solicitado al navegador web.
3. El usuario llena el formulario, se valida su correctitud y finalmente se le indica a la capa Gateway que debe crear las unidades de trabajo.
4. Sin embargo, el proceso de creación de tareas puede ser largo (algunas veces hay que crear unos millones de unidades de trabajo). Es por esto que no se puede esperar que se termine la creación para responder al usuario. Por ello, la capa gateway indica el *id* de la tarea, con el que posteriormente se puede recuperar su estado. Este *id* es almacenado en la capa web.
5. Se crean las unidades de trabajo sobre el servidor BOINC.

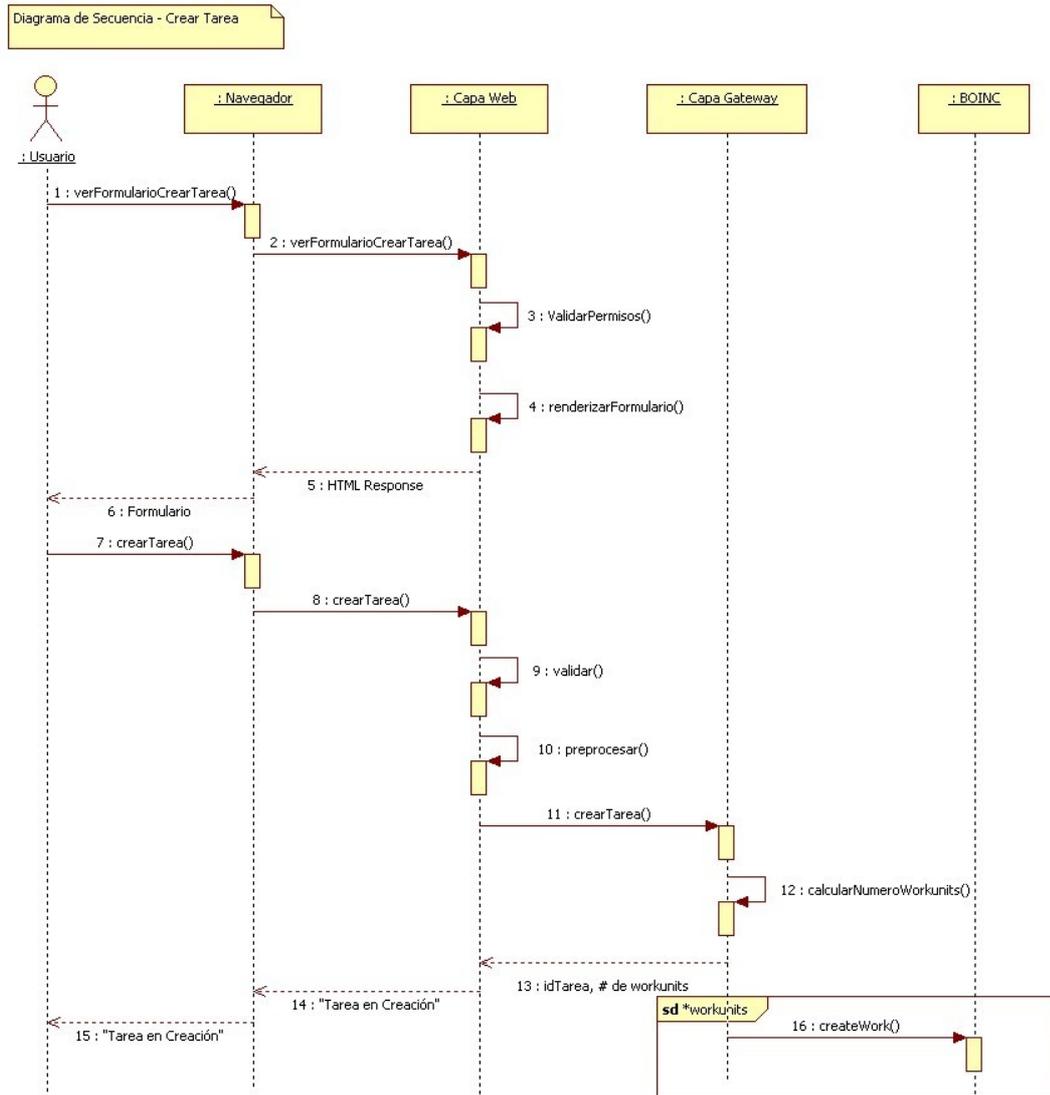


Imagen 3.1: Diagrama de secuencia – Envío de tareas

### 3.2.2. Recuperación de resultados

Cuando una tarea ha terminado de ejecutarse satisfactoriamente en el Grid, es importante que los investigadores tengan acceso a los resultados de la misma. Sin embargo, este resultado puede contener información sensible. Se muestra el flujo.

1. El usuario intenta recuperar los resultados de una tarea.
2. Se valida al usuario, y se pide a la capa gateway o LWS el resultado de la tarea.
3. Si se trata de un solo archivo, se puede descargar directamente, para esto la capa LWS le da una URL a la capa web, y esta hace una retransmisión del archivo resultado. Esto mejora la seguridad del sistema, ya que el recurso solo estará disponible a través de esta retransmisión.
4. Si se trata de varios archivos, se le brinda al usuario una dirección a la que podrá acceder con los credenciales correspondientes.

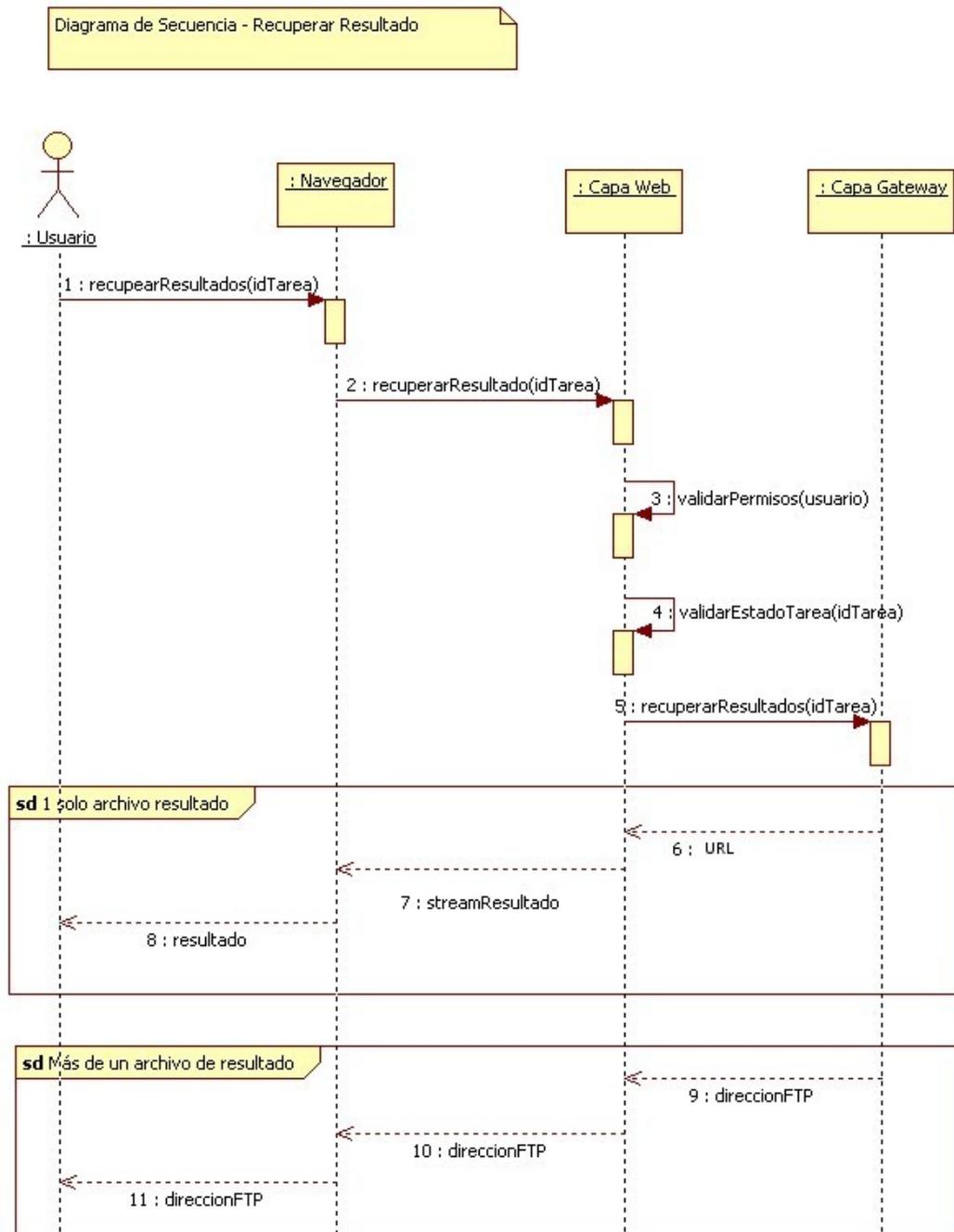


Imagen 3.2: Diagrama de secuencia – Recuperar Resultado

#### 4. Conclusiones y direcciones

Se ha mostrado que la arquitectura soporta los flujos más importantes del sistema y se ha demostrado la bondad de la arquitectura.

Se demuestra que la arquitectura cumple las metas establecidas:

- Mantenable – Los cambios propios al Backend están centralizados en el componente LWS.
- Componentes desacoplados – Las capas están desacopladas, se suscriben a un contrato y pueden ser reemplazadas fácilmente.
- Seguridad – El sistema cuida de los datos sensibles del usuario.
- Arquitectura actual – Se usa una arquitectura orientada a servicios.

Por lo tanto, se puede considerar que la arquitectura cumple las metas establecidas.

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA  
**UNIVERSIDAD**  
**CATÓLICA**  
DEL PERÚ

DESARROLLO DE UN FRAMEWORK WEB PARA EL ENVÍO  
REMOTO DE TAREAS, MONITOREO Y RECUPERACIÓN DE  
RESULTADOS PARA DESKTOP GRIDS USANDO UNA  
ARQUITECTURA ORIENTADA A SERVICIOS: CASO BOINC

**ANEXO C: Manual de usuario**



## Legión Framework v. 1.4

### Manual de Usuario

## Tabla de Contenidos

Introducción.....	5
Arquitectura de Legión Framework.....	6
Instalación.....	8
Instalación de Legión Web Services (LWS).....	8
Prerrequisitos.....	8
Procedimiento de la instalación.....	9
Instalación de Legión Web Interface.....	13
Prerrequisitos.....	13
Procedimiento de instalación.....	13
Otras opciones de instalación.....	13
Uso de Legión Framework.....	15
Entorno de trabajo.....	15
Tareas administrativas.....	16
Pantalla de Inicio del Administrador.....	16
Configuración del Sistema.....	17
Administración de Usuarios.....	23
Gestión de proyectos.....	25

Asistente de creación de proyectos .....	25
Editor gráfico de formularios .....	34
Nuevo TextBox .....	35
Validación numérica.....	36
Validación de cadenas.....	37
Nuevo TextArea.....	37
Nuevo Select.....	38
Nuevo FileUpload .....	40
Nuevo Radio .....	40
Nuevo Checkbox.....	41
Intérprete de Plantillas (Template Interpreter) .....	43
Reemplazo de variables de formulario .....	44
La evaluación matemática.....	44
Operador “Split” .....	44
Implementación de un nuevo “TemplateInterpreter” .....	46
Construcción del descriptor de tareas .....	47
Etiquetas del descriptor de tareas.....	47
Tareas simples .....	55
Extended Mode .....	57
Ejemplo: Barrido de parámetros para renderizar videos en 3D.....	59
Basic Mode.....	60
Búsqueda avanzada de tareas.....	61
Acceso de usuarios finales .....	62
Pantalla de Inicio del Usuario .....	62
Envío de tareas.....	62
Proceso de creación de tareas .....	64
<i>Monitoreo y descarga de tareas</i> .....	65
Anexo I : Instalación de Berkeley Open Infrastructure for Network Computing [Basado en un Manual Interno de la Organización].....	66
Prerrequisitos .....	66
Compilación de BOINC .....	66
Creación de un proyecto BOINC .....	68



## Introducción

---

**Legión Framework** es un software desarrollado y mantenido por la **Dirección Informática Académica** de la **Pontificia Universidad Católica del Perú** que tiene el propósito de facilitar el acceso a recursos computacionales de una Desktop Grid a través de una interfaz web sencilla. En especial, está preparado para trabajar con Berkeley Open Infrastructure for Network Computing (**BOINC**). Este último es un proyecto de código abierto mantenido por la Universidad de California en Berkeley, que permite aprovechar la capacidad libre de cómputo, y de otra forma desperdiciada, de computadoras personales a través de las conexiones de red. Dado que su uso y

administración no pueden ser considerados triviales, estas tareas son automatizadas casi en su totalidad por Legión Framework. Es así como en la interacción de Legión Framework y BOINC que es posible dotar a una institución de capacidad computacional para realizar cálculos intensivos de fácil acceso y de bajo costo si se asume la preexistencia de recursos como laboratorios de cómputo.

El inicio del Proyecto Legión data del año 2007 cuando se entendió la necesidad de contar con una infraestructura de gran escala en la institución. Desde entonces, ha permitido a los investigadores de la universidad realizar cálculos complejos; que de darse en una sola computadora, habría tomado varios años en completarse. Muchos de estos cálculos han originado publicaciones internacionales por parte de los científicos involucrados, lo que ha repercutido favorablemente en un mejor posicionamiento de la institución respecto a sus pares nacionales e internacionales.

A lo largo de este manual se presentan las consideraciones y métodos que posibilitan la instalación y uso de Legión Framework.

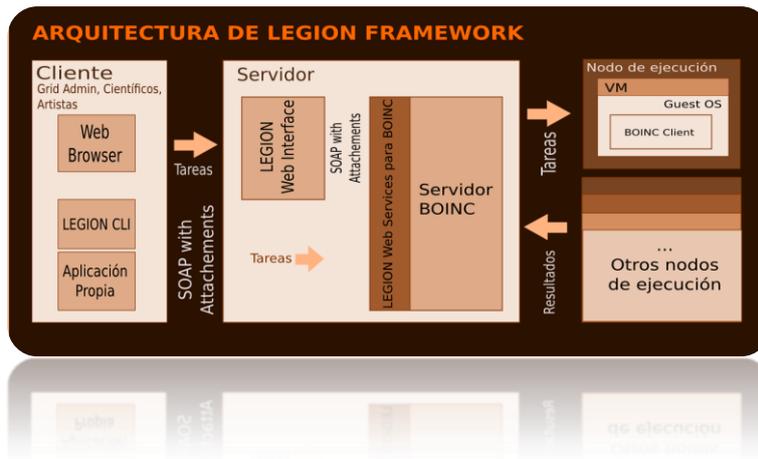
## ***Arquitectura de Legión Framework***

Legión Framework está compuesto por dos componentes principales:

- **Legión Web Interface**
- **Legión Web Services**

El primero, es la interfaz web que permite a los usuarios finales beneficiarse de los recursos de cómputo distribuidos provistos por una instalación de BOINC, de manera remota y sencilla a través de la web. A la vez, Legión Web Services se acopla al servidor BOINC, y permite controlarlo totalmente a través de servicios web, proveyendo así el acceso a los recursos de cómputo.

Los componentes mencionados interactúan siguiendo un protocolo bien definido, especificado a través de un archivo WSDL, usado para describir servicios web SOAP. Al usarse una arquitectura orientada a servicios, es posible que los componentes sean intercambiables siempre y cuando los componentes se suscriban a un “contrato”. Así, Legión Web Interface (qué realiza el envío de tareas) podría ser remplazado por un cliente de línea de comandos o una aplicación propia, use los recursos de la Desktop Grid directamente. Por otro lado, también sería factible remplazar Legión Web Services para BOINC, y por ende, permitir el acceso a recursos de otras Desktop Grids de manera transparente para Legión Web Interface.



### Arquitectura de Legión Framework

Los clientes BOINC son los que finalmente realizan los cálculos enviados. Pueden estar instalados en diversos sistemas operativos como Windows, Linux, Mac OSX entre otros. Sin embargo, se recomienda usar máquinas virtuales con un sistema operativo “guest” Linux. De este modo, los nodos de ejecución serán homogéneos, lo que facilitará el envío de archivos compilados para una arquitectura y conjunto de librerías en particular. Así mismo, las operaciones de actualización y prueba se verán reducidas en complejidad.

## Instalación

---

En esta sección se revisan los métodos, requisitos y consideraciones para instalar satisfactoriamente Legión Framework, para ello primero se verá la instalación de Legión Web Services y luego Legión Web Interface.

### *Instalación de Legión Web Services (LWS)*

Debido a que Legión Web Services (LWS) interactúa directamente con el Servidor BOINC, es importante tener en cuenta que este debe haber sido instalado previamente. Para una referencia de cómo instalar BOINC, puede ver el Anexo I.

### Prerrequisitos

- **Se recomienda una versión de Linux que tenga los siguientes paquetes instalados:** make, m4, libtool, autoconf, automake, gcc, openssl, mod\_ssl, sqlite, pkgconfig, mysql-connector-odbc, mysql, mysql-devel, php-mysql, mod\_auth\_mysql, php-gd, php-cli, php-common, openssl-devel, python-setuptools, MySQL-python, mysql++, mysql++-devel, gcc-c++, python-cherrypy, php, python-lxml, libxml2-devel, libxslt-devel, Cython, python-devel.
- **El servidor BOINC debe estar instalado en su versión estable.** Disponible en [http://boinc.berkeley.edu/svn/branches/server\\_stable](http://boinc.berkeley.edu/svn/branches/server_stable) usando la revisión 26042. Más detalles en el Anexo I.

## Procedimiento de la instalación

1. Primero, ejecutando la terminal como usuario **boincadm** se descomprime la aplicación LWS (la creación de este usuario se da durante la instalación del Servidor BOINC).

```
$ su boincadm
$ cd
$ tar -xzf Service-1.3.tar.gz
```

2. Luego, se procede a instalar LWS indicando el nombre de la aplicación y del proyecto configuradas en la instalación del Servidor BOINC (ver Anexo I).

```
$ cd
$ cd Service
$ python install.py --appname {Nombre app} --project_dir
/home/boincadm/projects/{Proyecto}/
Successful installation
```

Steps you should take now:

- Check the file 'ws\_config.py'.
- Add the assimilator 'legion\_assimilator' in the config.xml file of the project.
- Start web service with './bin/ws\_start' (to stop use './bin/ws\_stop').

3. Para cambiar alguna opción del servicio web se debe modificar el archivo **ws\_config.py** ubicado en la carpeta **/home/boincadm/projects/{Proyecto}/py/Service**.

```
$ cd
$ cd /home/boincadm/projects/{Proyecto}
$ more py/Service/ws_config.py
#
ADDRESS = "0.0.0.0"
PORT = 7789

# directory
APP_NAME = "{Nombre app}"
RESULT_DIRECTORY_BASE = "/home/boincadm/projects/{Proyecto} //task/result"
TEMP_DIRECTORY_BASE = "/home/boincadm/projects/{Proyecto} //task/tmp"
```

```
LOG_DIRECTORY = "/home/boincadm/projects/{Proyecto}//task/logs"
DELAY = "3"

#log DEBUG=10, INFO=20, ERROR=40
LOG_LEVEL=10

# SSL support
SSL_SUPPORT = "False"
SSL_CERTIFICATE = "/path_cert/localhost.crt"
SSL_PRIVATE_KEY = "/path_key/localhost.key"

# mail
SEND_MAIL = "False"
SUBJECT_MAIL = "[Develop]"
FROM_MAIL = "localhost@localdomain"
TO_DEFAULT_MAIL = "localhost@localdomain"
SMTP_MAIL = "smtp.server.com"
BOTTOM_MAIL = "\n\nSincerely,\n\nSistema de Super C"
BOTTOM_MAIL += str('\xf3')+"mputo Legi"+str('\xf3')+"n\nDirecci"+str('\xf3')+"n de
Inform"+str('\xe1')+"tica Acad"+
str('\xe9')+"mica"

# default
MAX_DISK_SPACE = str(20*1024*1024)
MAX_CPU_TIME = str(3*60*60)
```

El archivo **ws\_config.py** presenta varias opciones para personalizar el comportamiento de LWS. Estas opciones se describen a continuación.

- **PORT** – Indica el puerto que utilizará el servicio web.
- **APP\_NAME** - Indica el nombre de la aplicación a la que estará asociada el servicio web.
- **LOG\_LEVEL** - Indica el nivel de log, debe ser uno de los siguientes valores:
  - **Debug = 10**
  - **Info = 20**
  - **Error = 40**
- **RESULT\_DIRECTORY\_BASE** – Indica el directorio donde se guardan los resultados.
- **TEMP\_DIRECTORY\_BASE** - Indica el directorio temporal que será utilizado por el servicio web.
- **LOG\_DIRECTORY\_BASE** - Indica el directorio donde se guardará el log.
- **DELAY** - Es el máximo tiempo que durará una sesión (en minutos).
- Opciones para usar SSL.
- El envío de correo al finalizar una ejecución.
- Valores por defecto: máximo espacio de disco y tiempo de cpu máximo por workunit.

#### 4. Luego, iniciar el proyecto y el servicio web

```
$ cd /home/boincadm/projects/{Proyecto}/  
$ ./bin/start  
...  
$ ./bin/ws_start  
Start Web Service
```

5. Para verificar el correcto funcionamiento se puede acceder a la siguiente URL (usar dominio o IP del servidor).

<http://{dominio o IP}:7789/?wsdl>

*Se podrá observar el archivo*

WSDL del

servicio web.

6. Luego, configurar el servidor Apache para habilitar la descarga de resultados que, de acuerdo a la configuración, se encuentran en la carpeta:

**/home/boincadm/projects/{Proyecto}/task/result**

Para ello, crear el archivo **{Proyecto}\_task.conf** en la carpeta **/etc/httpd/conf.d** con el siguiente contenido<sup>1</sup> siguiente:

```
alias /download_task_{Proyecto} /home/boincadm/projects/{Proyecto}/task/result

# Note: projects/*/keys/ should NOT be readable!

<Directory "/home/boincadm/projects/{Proyecto}/task/result">
  Options FollowSymLinks
  AllowOverride None
</Directory>
```

- 
- 1 Las rutas deben ser las que han sido configuradas en el archivo de configuración de LWS `ws_config.py`. Los valores que se proporcionan podrían variar en su instalación.

## Instalación de Legión Web Interface

Legión Web Interface es una aplicación web que no necesariamente debe residir en el mismo servidor que Legión Web Services.

### Prerrequisitos

El sistema donde se instale Legión Web Interface debe tener las siguientes características:

- Sistema Operativo Linux
- Java Development Kit versión 1.6 o superior
- Apache Ant versión 1.8.2 o superior
- Apache Tomcat 7.x
- Base de datos MySQL 5.x

### Procedimiento de instalación

1. Puede ser necesario exportar las variables de entorno \$JAVA\_HOME y \$ANT\_HOME.
2. Ejecutar las siguientes líneas.

```
tar -xzf WebInterface-1.3.tar.gz
python install.py --db_user {usuario BD} --db_passwd {password BD} tomcat_home
/opt/tomcat/
```

3. De ser correcta la instalación, debe observar el siguiente mensaje de confirmación

```
Successful installation
```

### Otras opciones de instalación

```
# python install.py --help
```

```
install.py --db_user <user> --db_passwd <password> --tomcat_home <home of tomcat>
```

Required parameters:

<code>--db_user</code>	user database
<code>--db_passwd</code>	password database
<code>--tomcat_home</code>	location of tomcat

**Optional parameters:**

**--database** name of the database, default: Legion2WI  
**--db\_host** address of the database, default: localhost  
**--db\_port** port of the database, default: 3306  
**--webapps** location where the application will be installed,  
default : \$tomcat\_home/webapps  
**--app\_name** name of the application, default: Legion2WI  
**--update** only the application updates



## Uso de Legión Framework

---

Legión Framework tiene el objetivo de facilitar el acceso a la infraestructura de cálculo que ofrece BOINC mediante interfaces web sencillas. A la vez, facilitar la administración del sistema en su conjunto. Es por estas razones que presenta dos tipos acceso: administradores y usuarios finales.

Los usuarios finales solo pueden acceder a los proyectos en los que han sido registrados por un administrador de Legión Framework.

### *Entorno de trabajo*

Legión Framework está disponible vía web, esto quiere decir que debe usarse un navegador para poder acceder a las funcionalidades del framework. Los navegadores soportados son: © Google Chrome y © Mozilla Firefox. Además, funciona correctamente en © Microsoft Internet Explorer 9.

Cuando inicie Legión Framework por primera vez después de la instalación las credenciales de acceso serán las siguientes:

- **Nombre de usuario** : admin
- **Password** : admin

Se le recomienda cambiar de contraseña cuanto antes (puede hacerse desde la opción

correspondiente en el perfil).



**¿Que es Legión PUCP?**

La Dirección de Informática Académica (DIA) presenta el nuevo sistema de super cómputo Legión, el cual posee actualmente una potencia de cálculo equivalente a la de 250 procesadores Intel Core 2 Duo, con una capacidad máxima estimada de 10 12 (10 a la 12) operaciones matemáticas por segundo. A fin de lograr la capacidad mencionada, Legión hace uso concurrente del potencial disponible en las computadoras instaladas en los laboratorios a cargo de la DIA, lo cual implica que quienes estén haciendo uso de dichos equipos no se verán afectados cuando el sistema esté en funcionamiento.

Legión se basa en tecnología de computación en malla (grid computing) y posee una interfaz amigable que ayuda al investigador la complejidad del sistema, facilitando así el desarrollo de proyectos de investigación con procesos que requieren de grandes cálculos.

La tecnología de cómputo de alto rendimiento permitirá nuevas posibilidades a los investigadores de nuestra universidad en campos como predicción climatológica, química cuántica, física de altas energías, exploración petrolera y sísmica, validación de modelos estadísticos, bioinformática, análisis de estructuras de proteínas, simulaciones de fluidos, manejo (renderizado) intensivo de animaciones e imágenes en 3D, minería de datos, entre otros, favoreciendo así el incremento de la producción científica y el posicionamiento académico de nuestra universidad.

La DIA pone a Legión a disposición de los investigadores de la Universidad y ofrece todo el apoyo técnico necesario para la migración de aplicaciones que requieran de cómputo intensivo al esquema de grid computing. Si desea conocer más acerca del Sistema Legión y cómo puede utilizarlo en sus proyectos de investigación puede escribirnos a la dirección [legion@pucp.edu.pe](mailto:legion@pucp.edu.pe).

**ingresar**

Usuario:

Contraseña:

[¿Olvidó su contraseña?](#)

Powered by:

BOINC  
APACHE  
MYSQL  
JAVA  
PYTHON

Servicio creado por la Dirección de Informática Académica (DIA)  
Diseñado y Modificado por DIA | Contacto con el administrador del sistema por correo electrónico:  
A: Universidad1801, San Miguel, Lima - Perú | Phone: +51 1 4262300 | © 2013 Pontificia Universidad Católica del Perú

## Pantalla de inicio de Legión Framework

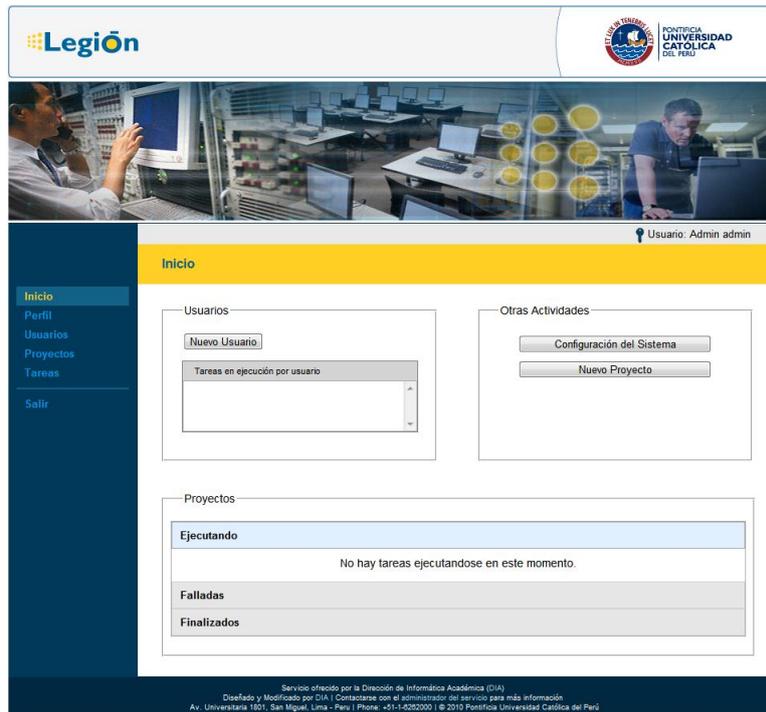
### Tareas administrativas

Las operaciones administrativas en Legión Framework están relacionadas a la administración de usuarios, monitoreo del estado del sistema y por consiguiente: el estado de las tareas enviadas. Cabe resaltar que la mayoría de conceptos en Legión Framework gira en torno al concepto de tarea.

### Pantalla de Inicio del Administrador

La pantalla de inicio del administrador está diseñada para mostrar un vistazo general del estado del sistema. En la parte superior izquierda se puede observar un listado los usuarios que tienen tareas en ejecución, junto a un conteo de la cantidad de estas.

Junto a este componente, al lado derecho se encuentra un conjunto de botones que permiten acceso a las funciones del sistema: Configuración y creación de nuevo proyecto.



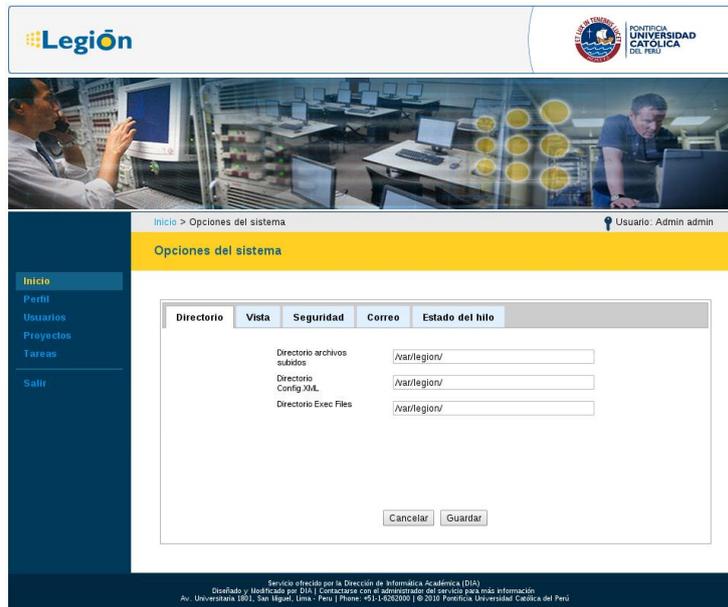
Pantalla de Inicio del Administrador

## Configuración del Sistema

La configuración del sistema permite personalizar el comportamiento de Legión Framework.

## Opciones de Directorio

Las opciones de directorio permiten configurar dónde Legión Framework almacenará los archivos.



### Opciones de Directorio

- **Directorio de archivos subidos** – Indica dónde se almacenarán los archivos que envíe el usuario. Es importante que este directorio exista previamente y que el usuario que ejecuta el servidor Apache Tomcat tenga permisos para crear nuevos archivos.

*Valor por defecto:* /tmp/

*Nota:* Debe terminar en slash (/)

- **Directorio ConfigXML** – Indica dónde se almacenarán los archivos Config XML después de ser procesados por el TemplateInterpreter. Estos archivos describen una tarea que será creada eventualmente, por lo que son muy importantes. Es necesario que este directorio exista previamente y que el usuario que ejecuta el servidor Apache Tomcat tenga permisos para crear nuevos archivos

*Valor por defecto:* /tmp/

*Nota:* Debe terminar en slash (/)

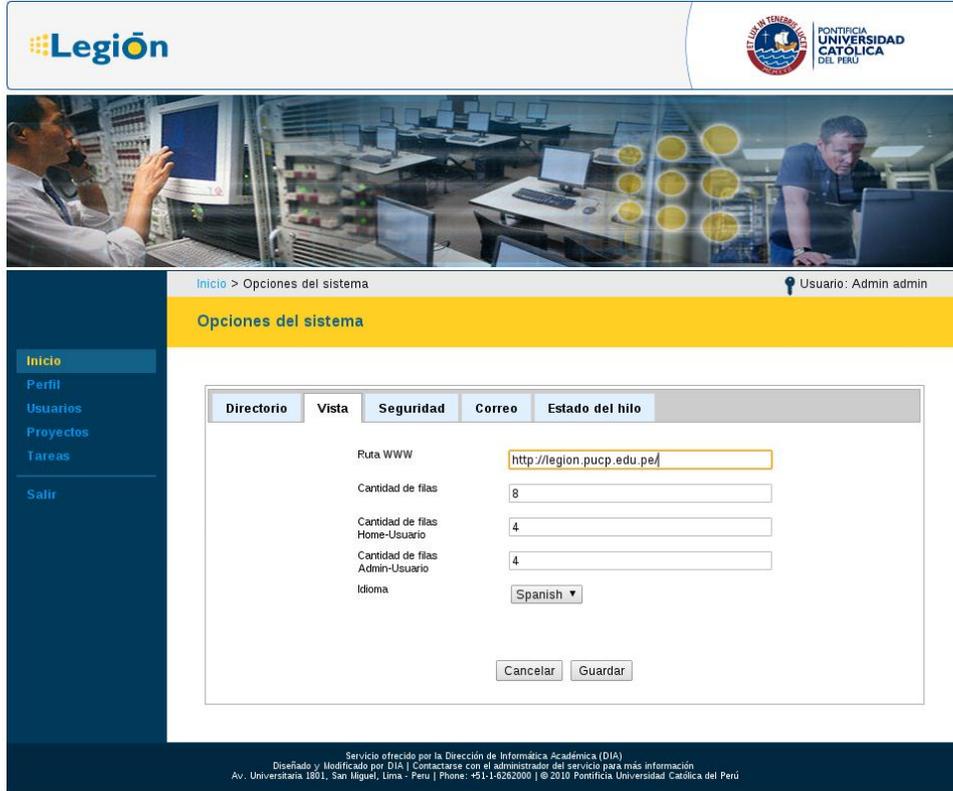
- **Directorio Exec Files** - Indica dónde se almacenarán los archivos que se usarán para ejecutar las tareas de los proyectos. Estos archivos describen una tarea que será creada eventualmente, por lo que son muy importantes. Es necesario que este directorio exista previamente y que el usuario que ejecuta el servidor Apache Tomcat tenga permisos para crear nuevos archivos

*Valor por defecto:* /tmp/

*Nota:* Debe terminar en slash (/)

## Opciones de Vista

Las opciones de vista permiten configurar el comportamiento visual que tendrá Legión Framework.



Inicio > Opciones del sistema Usuario: Admin admin

Opciones del sistema

Directorio	Vista	Seguridad	Correo	Estado del hilo
Ruta WWW	<input type="text" value="http://legion.pucp.edu.pe/"/>			
Cantidad de filas	<input type="text" value="8"/>			
Cantidad de filas Home-Usuario	<input type="text" value="4"/>			
Cantidad de filas Admin-Usuario	<input type="text" value="4"/>			
Idioma	<input type="text" value="Spanish"/>			

Cancelar Guardar

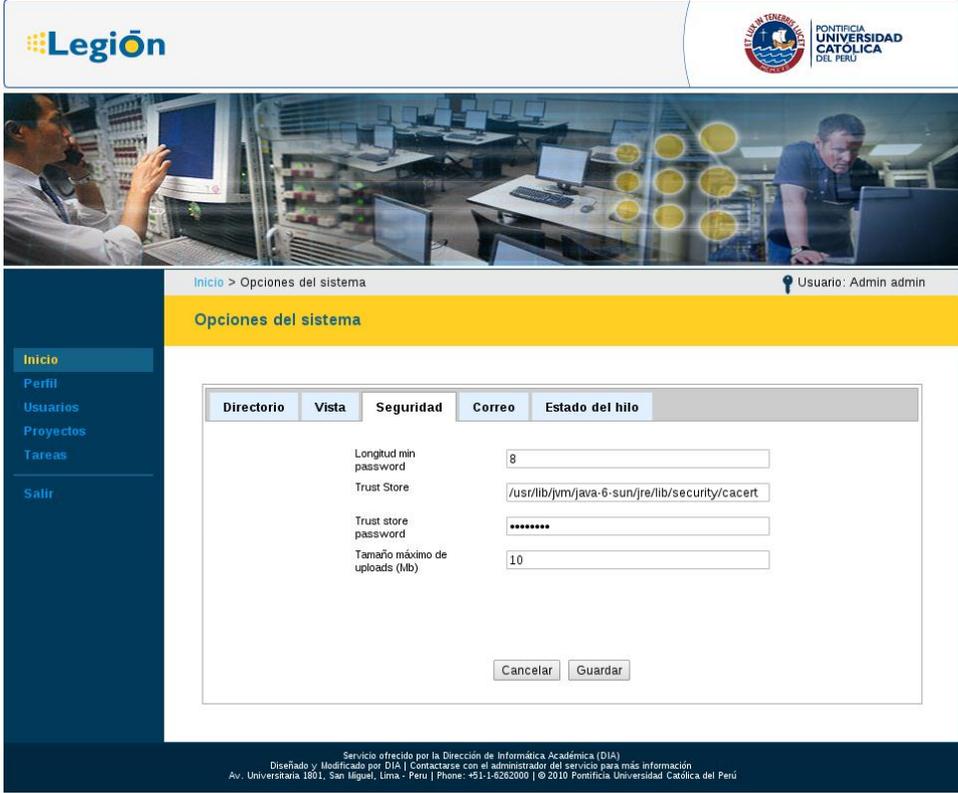
Servicio ofrecido por la Dirección de Informática Académica (DIA)  
Diseñado y Modificado por DIA | Contactarse con el administrador del servicio para más información  
Av. Universitaria 1801, San Miguel, Lima - Perú | Phone: +51-1-6262000 | © 2010 Pontificia Universidad Católica del Perú

### Opciones de Vista

- **Ruta WWW** – Ruta que se usará para mostrar la dirección correcta, incluso si se accediera a través de un proxy.
- **Cantidad de filas** – Indica la cantidad de filas que tendrán las tablas.
- **Cantidad de filas Home-Usuario** – Indica la cantidad de filas que tendrán las tablas en el Home de usuario.
- **Cantidad de filas Admin-Usuario** - Indica la cantidad de filas que tendrán las tablas en el Home del Administrador.
- **Idioma** – Actualmente se soporta: Inglés y Español.

## Opciones de seguridad

Las opciones de seguridad permiten configurar ciertos niveles de protección para Legión Framework.



Inicio > Opciones del sistema Usuario: Admin admin

Opciones del sistema

Directorio	Vista	Seguridad	Correo	Estado del hilo
		Longitud min password: <input type="text" value="8"/>		
		Trust Store: <input type="text" value="/usr/lib/jvm/java-6-sun/jre/lib/security/cacert"/>		
		Trust store password: <input type="password" value="*****"/>		
		Tamaño máximo de uploads (Mb): <input type="text" value="10"/>		

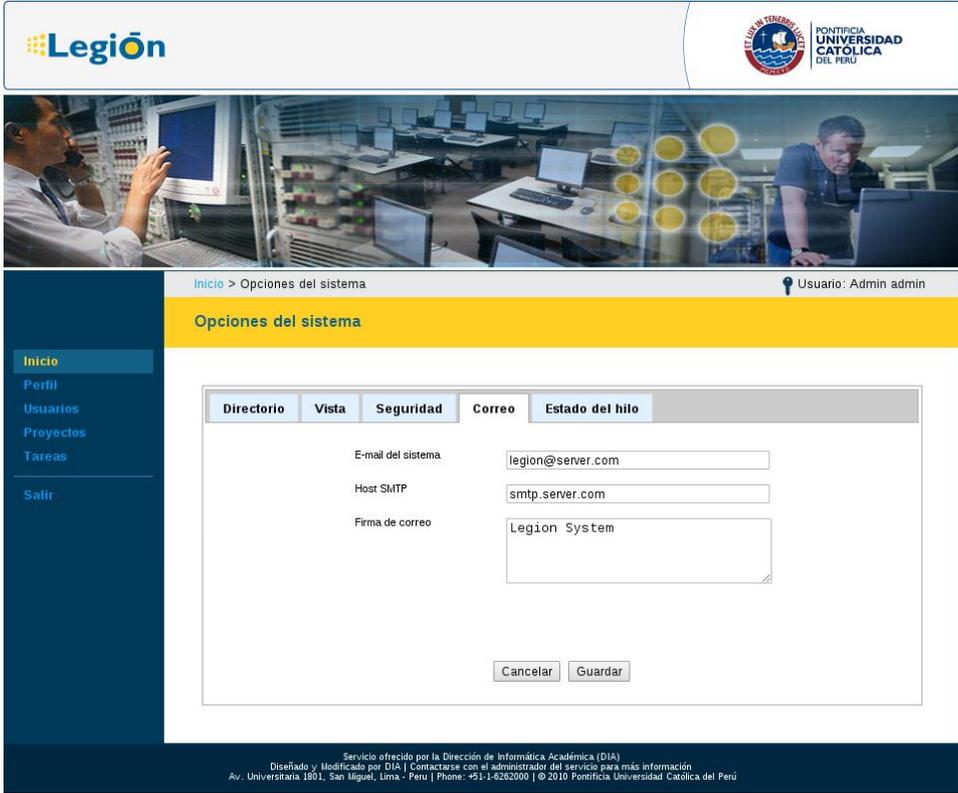
Servicio ofrecido por la Dirección de Informática Académica (DIA)  
Diseñado y Modificado por DIA | Contactarse con el administrador del servicio para más información  
Av. Universitaria 1801, San Ildefonso, Lima - Perú | Phone: +51-1-4222000 | © 2010 Pontificia Universidad Católica del Perú

### Opciones de seguridad

- **Longitud min password** – La longitud mínima del contraseña que se exigirá a los usuarios.
- **Trust Store** – Contenedor de los certificados que garantizan la conexión con Legión Web Services cuando funciona sobre HTTPS.
- **Trust Store password** – Contraseña para acceder al Trust Store.
- **Tamaño máximo de uploads** – Tamaño máximo en Megabytes a subir a Legión.

## Opciones de correo

Las opciones de correo permiten configurar el comportamiento del sistema respecto a las notificaciones de correo electrónico.



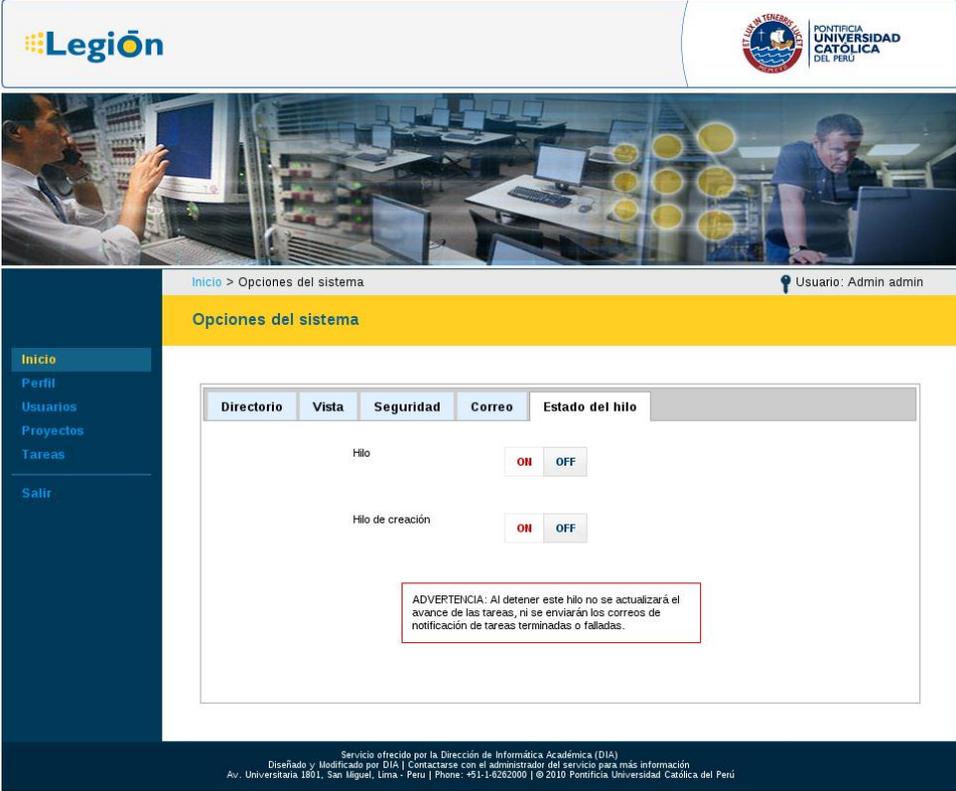
The screenshot displays the 'Legión' system administration interface. At the top, there is a navigation bar with the 'Legión' logo on the left and the university logo on the right. Below this is a banner image showing a server room. The main content area is titled 'Opciones del sistema' and includes a breadcrumb trail 'Inicio > Opciones del sistema' and a user indicator 'Usuario: Admin admin'. A sidebar on the left contains navigation links: Inicio, Perfil, Usuarios, Proyectos, Tareas, and Salir. The main panel has tabs for 'Directorio', 'Vista', 'Seguridad', 'Correo', and 'Estado del hilo', with 'Correo' selected. The 'Correo' configuration form includes three input fields: 'E-mail del sistema' (containing 'legion@server.com'), 'Host SMTP' (containing 'smtp.server.com'), and 'Firma de correo' (containing 'Legion System'). At the bottom of the form are 'Cancelar' and 'Guardar' buttons. A footer at the very bottom contains small text: 'Servicio ofrecido por la Dirección de Informática Académica (DIA) | Diseñado y Modificado por DIA | Contactarse con el administrador del servicio para más información | Av. Universitaria 1891, San Ildefonso, Lima - Perú | Phone: +51-1-4252000 | © 2010 Pontificia Universidad Católica del Perú'.

## Opciones de correo

- **E-mail del sistema** – Dirección de e-mail desde la que serán enviados los correos del sistema.
- **Host SMTP** – Servidor SMTP que se usará para enviar los correos.
- **Firma de correo** – Firma que se anexará al final de cada mensaje (multilínea).

## Opciones del hilo

Esta pantalla permite monitorear el estado del hilo de monitoreo y del hilo de creación de tareas.



Inicio > Opciones del sistema Usuario: Admin admin

### Opciones del sistema

Directorio	Vista	Seguridad	Correo	Estado del hilo
				Hilo <input type="button" value="ON"/> <input type="button" value="OFF"/>
				Hilo de creación <input type="button" value="ON"/> <input type="button" value="OFF"/>

ADVERTENCIA: Al detener este hilo no se actualizará el avance de las tareas, ni se enviarán los correos de notificación de tareas terminadas o falladas.

Servicio ofrecido por la Dirección de Informática Académica (DIA)  
Diseñado y Modificado por DIA | Conéctese con el administrador del servicio para más información  
Av. Universitaria 1801, San Miguel, Lima - Perú | Phone: +51-1-6262000 | © 2010 Pontificia Universidad Católica del Perú

## Opciones del hilo

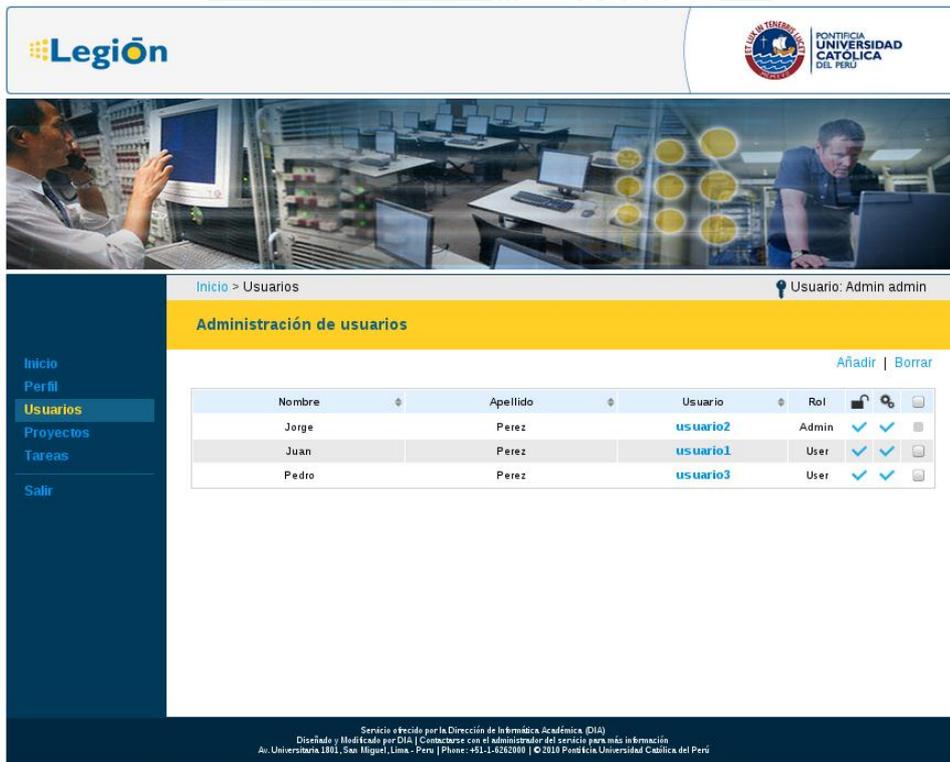
Se puede cambiar el estado “encender” o “apagar” usando el botón “ON/OFF” correspondiente a cada hilo.

## Administración de Usuarios

La pantalla de administración de usuarios es accesible solo a los usuarios con el rol de administrador. La forma de accederla es a través del link que se encuentra en el menú lateral “Usuarios”. Al abrir esta página, se mostrará una lista de todos los usuarios en una tabla indicando los siguientes datos “Nombre”, “Apellido”, “Usuario” y “Rol”. Puede realizar las siguientes operaciones desde esta ventana:

- **Modificar el usuario** – haciendo click al nombre de usuario de la fila correspondiente.
- **Bloquear acceso** – haciendo click sobre el ícono  de la fila correspondiente, de la columna . Este ícono cambiará a  una vez realizado el proceso.
- **Bloquear el envío de tareas** – haciendo click sobre el ícono  de la columna . Este ícono cambiará a  una vez realizado el proceso.
- **Borrar** – para eliminar un usuario definitivamente del sistema, deberá seleccionar uno o varios usuarios mediante el checkbox que se encuentra en cada fila de la tabla, y luego hacer click en “Borrar”.

**NOTA:** El usuario “admin” no podrá ser alterado a través de esta pantalla.



Inicio > Usuarios Usuario: Admin admin

### Administración de usuarios

[Añadir](#) | [Borrar](#)

Nombre	Apellido	Usuario	Rol			<input type="checkbox"/>
Jorge	Perez	usuario2	Admin			<input type="checkbox"/>
Juan	Perez	usuario1	User			<input type="checkbox"/>
Pedro	Perez	usuario3	User			<input type="checkbox"/>

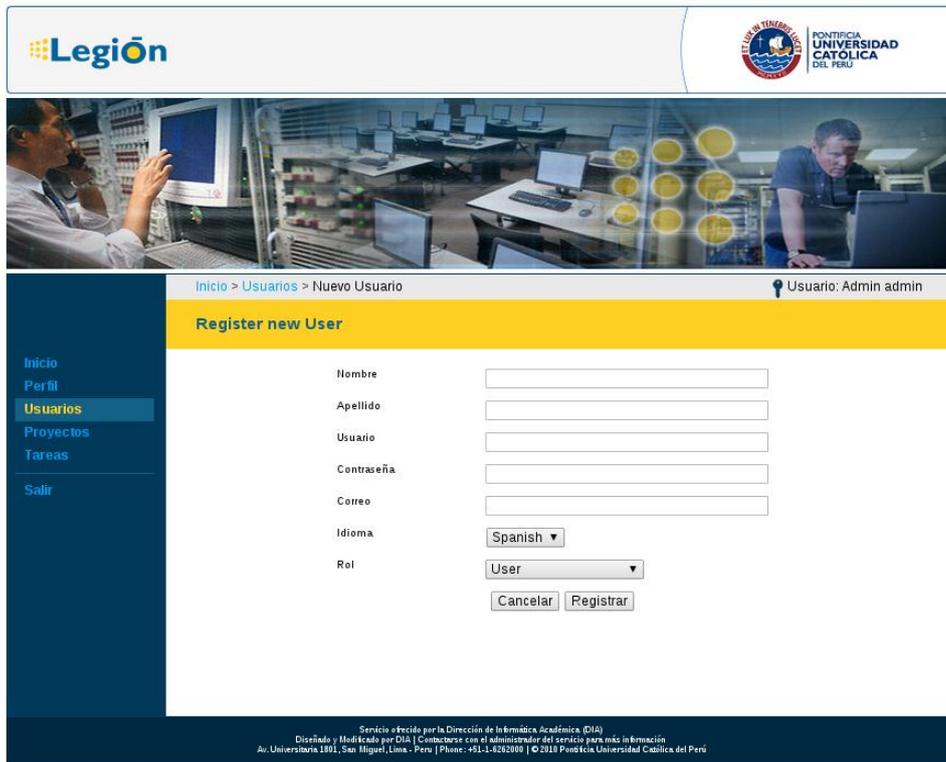
Servicio ofrecido por la Dirección de Informática Académica (DIA)  
 Diseñado y Mantenido por DIA | Contactarse con el administrador del servicio para más información  
 Av. Universitaria 1801, San Miguel, Lima - Perú | Phone: +51-1-6262000 | © 2010 Pontificia Universidad Católica del Perú

### Pantalla de Administración de Usuarios

Para registrar un nuevo usuario en el sistema diríjase a la opción “Añadir” que se encuentra sobre la tabla.

### Creación de Usuarios

Para crear un nuevo usuario consigne los siguientes datos “Nombre”, “Apellidos”, “Usuario”, “Contraseña”, “Correo”, “Idioma”, “Rol” y proceda a presionar el botón “Registrar”. Si deseara cancelar la operación, seleccione el botón del mismo nombre.



Inicio > Usuarios > Nuevo Usuario

Usuario: Admin admin

**Register new User**

Nombre

Apellido

Usuario

Contraseña

Correo

Idioma

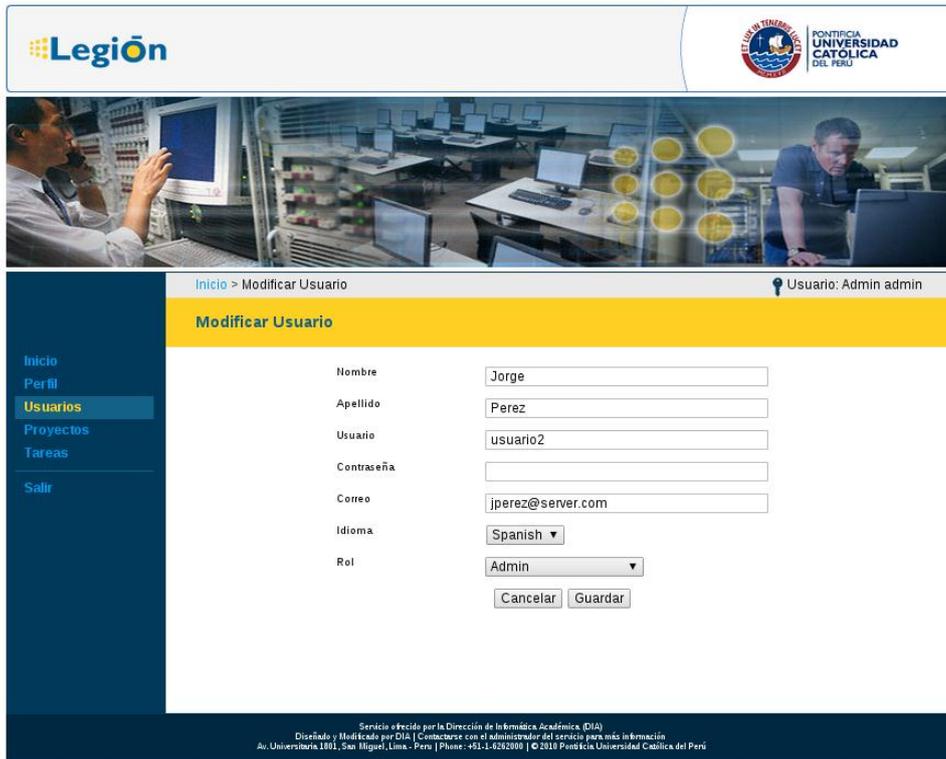
Rol

Servicio ofrecido por la Dirección de Informática Académica (DIA)  
Diseñado y Modificado por DIA | Contactarse con el administrador del servicio para más información  
Av. Universitaria 1801, San Miguel, Lima - Perú | Phone: +51-1-6262000 | © 2010 Pontificia Universidad Católica del Perú

Pantalla de creación de usuarios

### Modificación de Usuarios

Al realizar la modificación de un usuario, los siguientes campos pueden ser modificados: “Nombre”, “Apellido”, “Usuario”, “Contraseña”, “Correo”, “Idioma” y “Rol”. Una vez realizadas estas modificaciones, seleccione el botón “Guardar” para que los cambios se reflejen en el sistema. Si no deseara realizar la operación, seleccione el botón “Cancelar”.



Legión

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Inicio > Modificar Usuario Usuario: Admin admin

### Modificar Usuario

Nombre:

Apellido:

Usuario:

Contraseña:

Correo:

Idioma: Spanish ▾

Rel: Admin ▾

Servicio ofrecido por la Dirección de Informática Académica (DIA)  
Diseñado y Mantenido por DIA | Contactarse con el administrador del servicio para más información  
Av. Universitaria 1801, San Miguel, Lima - Perú | Phone: +51-1-6292000 | © 2010 Pontificia Universidad Católica del Perú

Pantalla de modificación de usuario

## Gestión de proyectos

Si bien es cierto que las tareas son la unidad central en Legión Framework, estas existen únicamente en el contexto de los proyectos. Las tareas dentro de un proyecto comparten muchas características como los programas ejecutables, listado de parámetros y forma de crear las unidades de trabajo. Difieren, por otro lado, en los parámetros y eventualmente en archivos adicionales que son enviados por el usuario final.

## Asistente de creación de proyectos

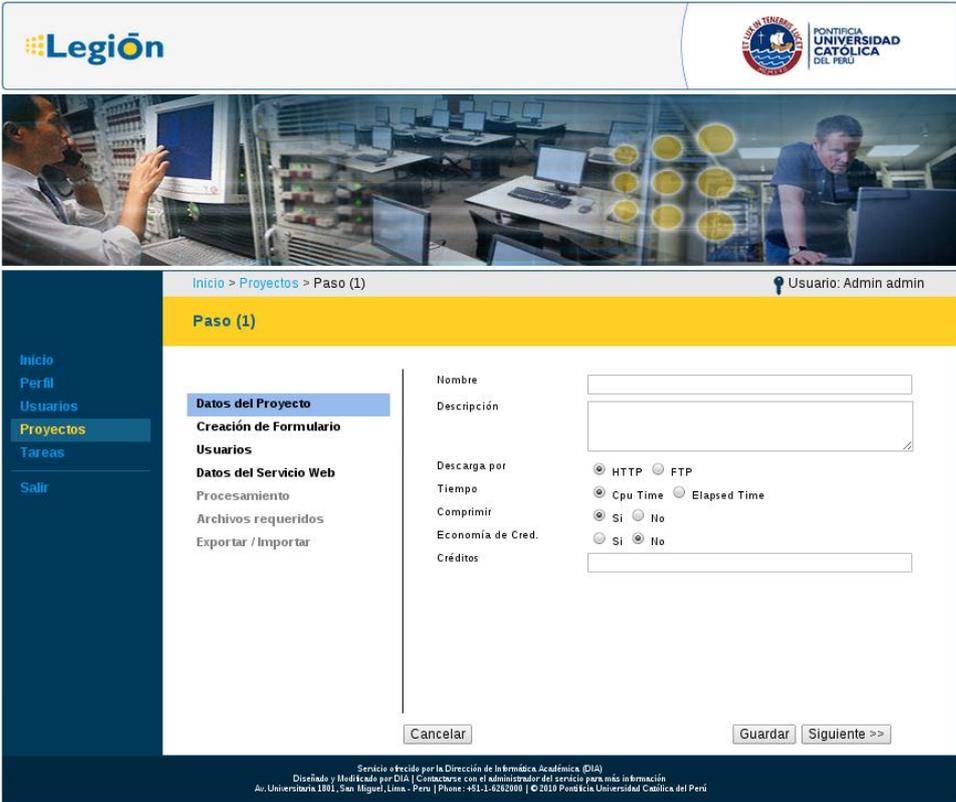
Para crear un proyecto en Legión Framework se tiene que seguir una serie de 6 pasos. A lo largo de estos, se tiene que especificar como se creará la tarea, los archivos requeridos y definir el formulario de envío de tareas. A continuación se revisan estos pasos.

### Paso 1: Datos del proyecto

En este paso se definen los datos generales del proyecto:

- **“Nombre”** - indica el nombre que se usará para identificar al proyecto. Esta información es pública a los usuarios inscritos en el proyecto.
- **“Descripción”** - indica información útil para identificar el proyecto, sin embargo esta información no es pública.

- **“Descarga por”** - indica, si se selecciona HTTP, que las descargas de resultados son retransmitidas por Legión Web Interface, es decir que el usuario final no conocerá la ruta donde reside realmente el archivo resultado; en cambio, si se seleccionara FTP solamente se redirigirá al usuario a la ruta de los resultados.
- **“Tiempo”** - indica como se contabilizará el tiempo de ejecución: o bien usando el tiempo estimado de CPU (estandarizado por los FLOPs de una estación), o simplemente el tiempo acumulado que transcurrió para las unidades de trabajo.
- **“Comprimir”** - indica si se comprimirán los resultados, el valor seleccionado pasará a ser accesible en la variable **{Project:compress}**
- **“Economía de créditos”** - indica si la economía de créditos estará activa, lo que implica que al proyecto le es asignada una cantidad de créditos, siendo estos consumidos a razón de un ratio equivalente a las horas de ejecución.
- **“Créditos”** - indica la cantidad de créditos con los que cuenta el proyecto.



Legión

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Inicio > Proyectos > Paso (1) Usuario: Admin admin

**Paso (1)**

**Datos del Proyecto**

Creación de Formulario

Usuarios

Datos del Servicio Web

Procesamiento

Archivos requeridos

Exportar / Importar

Nombre

Descripción

Descarga por  HTTP  FTP

Tiempo  Cpu Time  Elapsed Time

Comprimir  Si  No

Economía de Cred.  Si  No

Créditos

Cancelar Guardar Siguiente >>

Servicio ofrecido por la Dirección de Informática Académica (DIA)  
Diseñado y Mantenido por DIA | Comunicarse con el administrador del servicio para más información  
Av. Universitaria 1801, San Miguel, Lima - Perú | Phone: +51-1-6262000 | © 2010 Pontificia Universidad Católica del Perú

## Paso 1- Datos del Proyecto

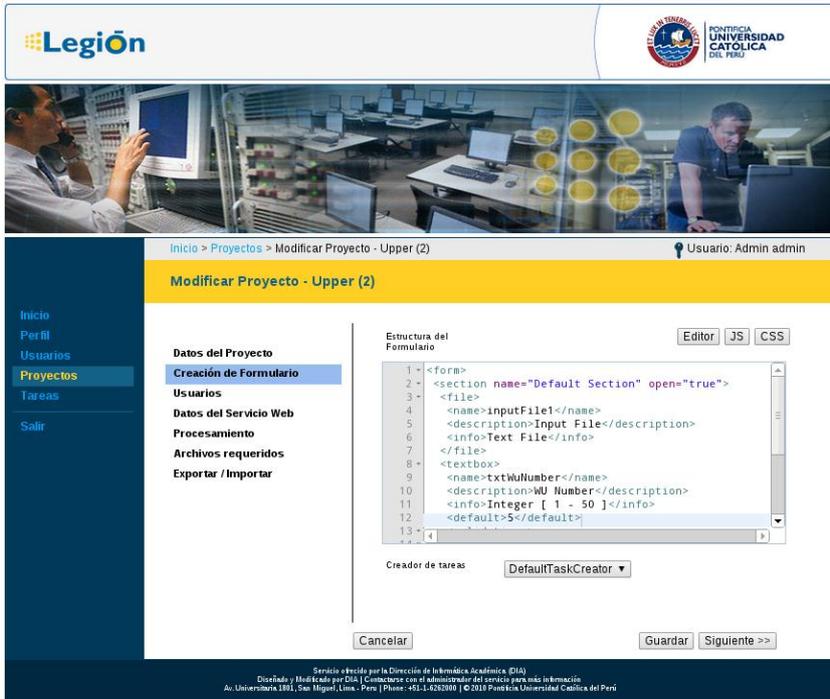
## Paso 2: Creación del formulario

La creación del formulario es una etapa muy importante, porque es la interfaz que se le presentará al usuario final.

Los elementos soportados son:

- **Textbox**
- **Textarea**
- **Select**
- **FileUpload**
- **Radio**
- **Checkbox**

Al seleccionar la opción “Editor” se desplegará una ventana que permite la edición del formulario de forma gráfica.



Legión

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Inicio > Proyectos > Modificar Proyecto - Upper (2) Usuario: Admin admin

Modificar Proyecto - Upper (2)

Inicio  
Perfil  
Usuarios  
Proyectos  
Tareas  
Salir

Datos del Proyecto

Creación de Formulario

Usuarios

Datos del Servicio Web

Procesamiento

Archivos requeridos

Exportar / Importar

Estructura del Formulario

```

1 <form>
2 <section name="Default Section" open="true">
3 <file>
4 <name>inputFile1</name>
5 <description>Input File</description>
6 <info>Text File</info>
7 </file>
8 <textbox>
9 <name>txtWUNumber</name>
10 <description>WU Number</description>
11 <info>Integer [ 1 - 50 ]</info>
12 <default>5</default>
13

```

Creador de tareas: DefaultTaskCreator

Cancelar Guardar Siguiente >>

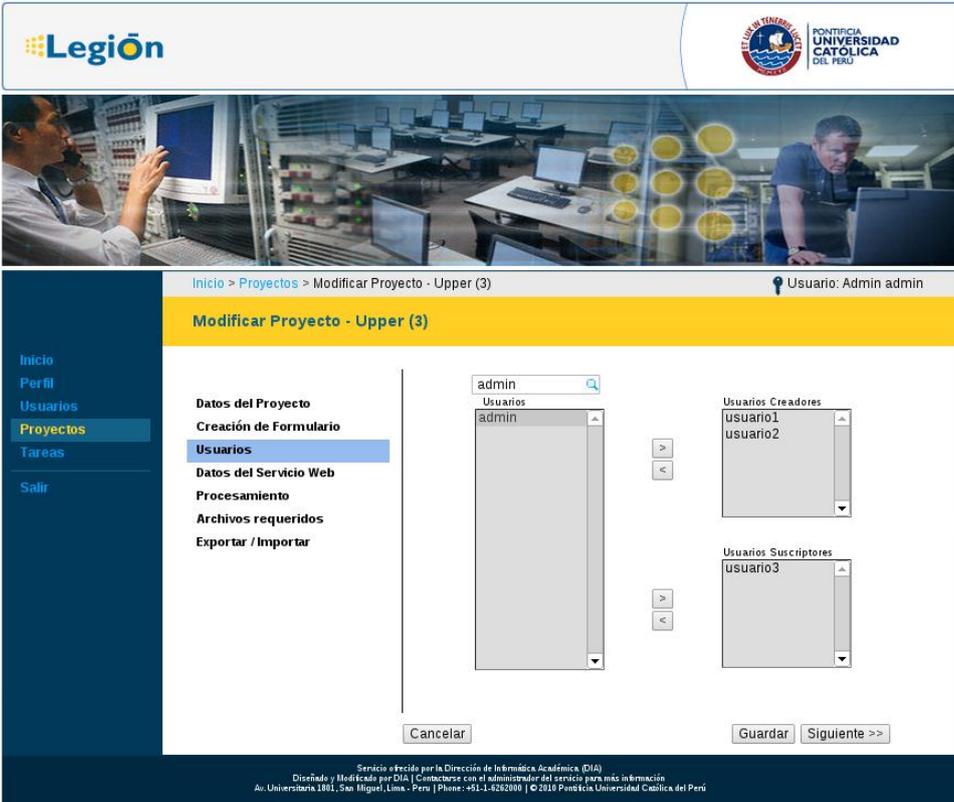
Servicio ofrecido por la Dirección de Informática Académica (DIA)  
Diseñado y Modificado por DÍA | Conectarse con el administrador del servicio para más información  
Av. Universidad 1801, San Miguel, Lima - Perú | Phone: +51-1-6252000 | © 2019 Pontificia Universidad Católica del Perú

## Paso 2- Creación del formulario

Ver la sección “Editor Gráfico de formularios” para más detalles.

### Paso 3: Añadir usuarios

Con el fin de seleccionar a los usuarios que participarán en el proyecto con uno de los dos roles posibles, se presenta un listado de usuarios (con posibilidad de filtrarlos por: nombre de usuario, nombre, apellidos e e-mail) y las dos listas inicialmente vacías. Los botones “>” y “<” que se encuentran paralelos a “Usuarios Creadores” / “Usuarios Suscriptores” permiten mover usuarios de una lista a otra. Un usuario creador tiene el privilegio de crear tareas y un usuario suscriptor solo puede ver tareas que le hayan sido compartidas.



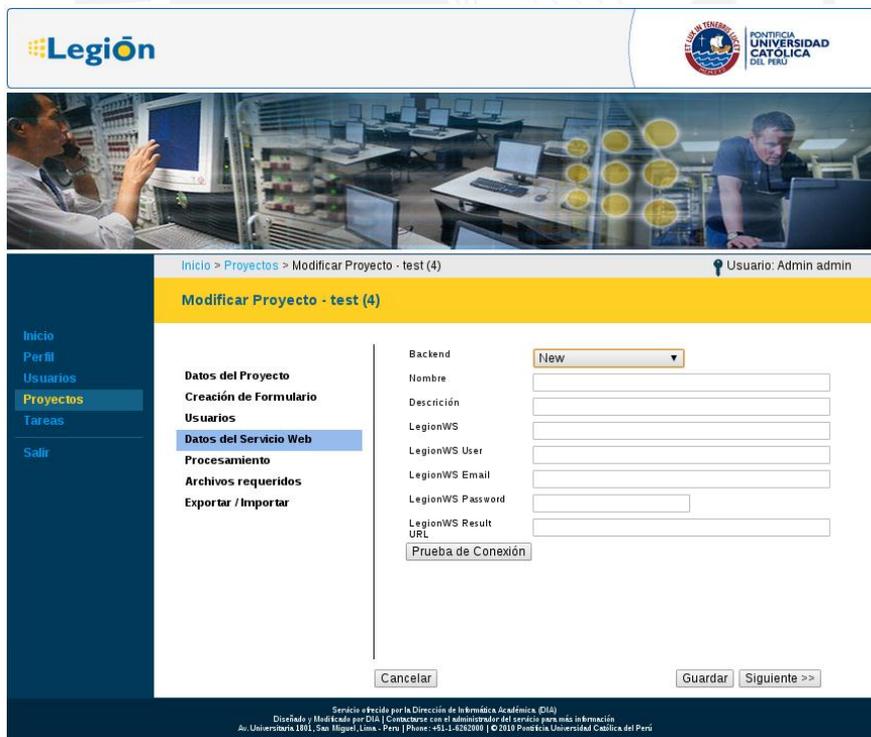
The screenshot displays the 'Legión' web application interface. At the top, there is a navigation bar with the 'Legión' logo and the Pontificia Universidad Católica del Perú logo. Below this is a banner image showing a laboratory setting. The main content area is titled 'Modificar Proyecto - Upper (3)' and shows a search bar with 'admin' entered. Below the search bar is a list of users, currently containing 'admin'. To the right, there are two lists: 'Usuarios Creadores' containing 'usuario1' and 'usuario2', and 'Usuarios Suscriptores' containing 'usuario3'. Navigation buttons '>' and '<' are placed between these lists. At the bottom, there are 'Cancelar', 'Guardar', and 'Siguiente >>' buttons. A footer contains small text: 'Servicio creado por la Dirección de Informática Académica (DIA) | Diseñado y mantenido por DIA | Colaboro con el administrador del servicio para más información | Av. Universitaria 1891, San Miguel, Lima - Perú | Phone: +51-1-4252000 | © 2013 Pontificia Universidad Católica del Perú'.

### Paso 3 - Usuarios

## Paso 4: Datos del Servicio Web

Es importante configurar el servicio web con el que trabajará el proyecto, porque este provee acceso al entorno donde se ejecutarán las tareas enviadas. Legión Framework puede trabajar con un Backend distinto por proyecto, aunque es factible que también varios proyectos usen el mismo.

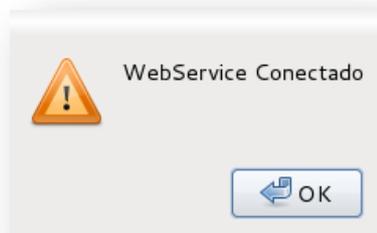
- **Backend** –Es una plataforma de cómputo intensivo a la que Legión Framework tendrá acceso a través de servicios web. Si ya se encontraran registrados algunos Backends, se podría seleccionar cualquiera de ellos.
- **Descripción** – Información útil para identificar el Backend.
- **LegionWS** – Dirección del Servicio web, debe incluir IP y puerto.
- **LegionWS User** – Nombre de usuario en el servidor BOINC (primer usuario creado) con el propósito de validar el permiso de acceso al Backend.
- **LegionWS Email** – Email del usuario en el servidor BOINC (primer usuario creado) con el propósito de validar el permiso de acceso al Backend.
- **LegionWS Password** - Password del usuario en el servidor BOINC (primer usuario creado) con el propósito de validar el permiso de acceso al Backend.
- **LegionWS Result URL** – Dirección donde se guardarán los resultados. Esta se genera al crear un “alias” al directorio donde se guardan los resultados (ver Legion Web Services).



The screenshot displays the 'Legión' web interface. At the top, there is a navigation bar with the 'Legión' logo and the university's name. Below this is a banner image showing a laboratory setting. The main content area is titled 'Modificar Proyecto - test (4)' and contains a form for configuring web service data. The form includes a dropdown menu for 'Backend' (set to 'New'), and input fields for 'Nombre', 'Descripción', 'LegionWS', 'LegionWS User', 'LegionWS Email', 'LegionWS Password', and 'LegionWS Result URL'. A 'Prueba de Conexión' button is located below the input fields. At the bottom of the form are 'Cancelar', 'Guardar', and 'Siguiente >>' buttons. A sidebar on the left contains navigation links: 'Inicio', 'Perfil', 'Usuarios', 'Proyectos', 'Tareas', and 'Salir'. The footer of the page contains small text: 'Servicio creado por la Dirección de Informática Académica (DIA) Diseñado y Modificado por DIA | Contactarse con el administrador del servicio para más información Av. Universidad 1801, San Miguel, Lima - Perú | Phone: +51-1-4262000 | © 2010 Pontificia Universidad Católica del Perú'.

## Paso 4 – Datos del Servicio Web

Una vez ingresados los valores, se puede probar que las credenciales sean correctas haciendo click en el botón “Prueba de Conexión”.

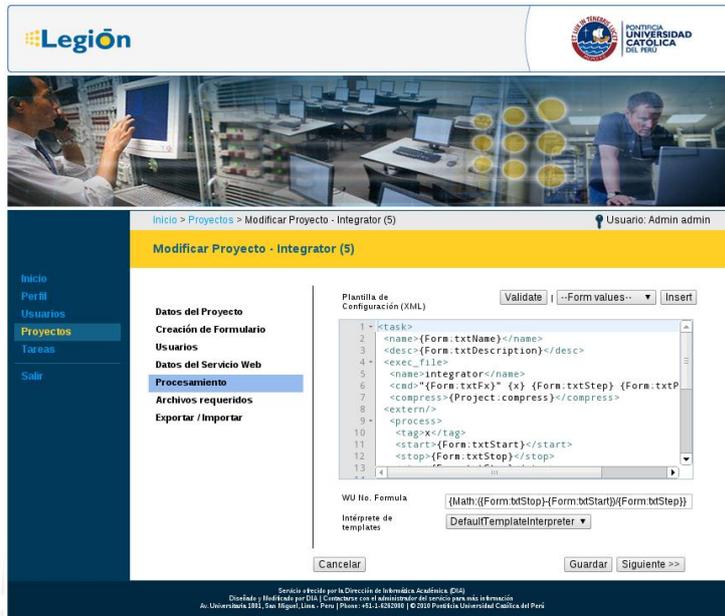


### Mensaje de confirmación (conexión)



## Paso 5: Procesamiento

En este paso se ingresará la plantilla o “Template” para el descriptor de tareas. Más detalles de como construir este archivo lo puede observar en la sección “Intérprete de Plantillas” y “Descriptor de tareas”



The screenshot shows the 'Legión' web application interface. The main content area is titled 'Modificar Proyecto - Integrator (5)'. On the left, there is a navigation menu with options: Inicio, Perfil, Usuarios, **Proyectos**, Tareas, and Salir. The 'Proyectos' section is expanded, showing sub-options: Datos del Proyecto, Creación de Formulario, Usuarios, Datos del Servicio Web, **Procesamiento**, Archivos requeridos, and Exportar / Importar. The main area displays the 'Plantilla de Configuración (XML)' editor. It includes a 'Validate' button, a dropdown for 'Form values', and an 'Insert' button. The XML code is as follows:

```

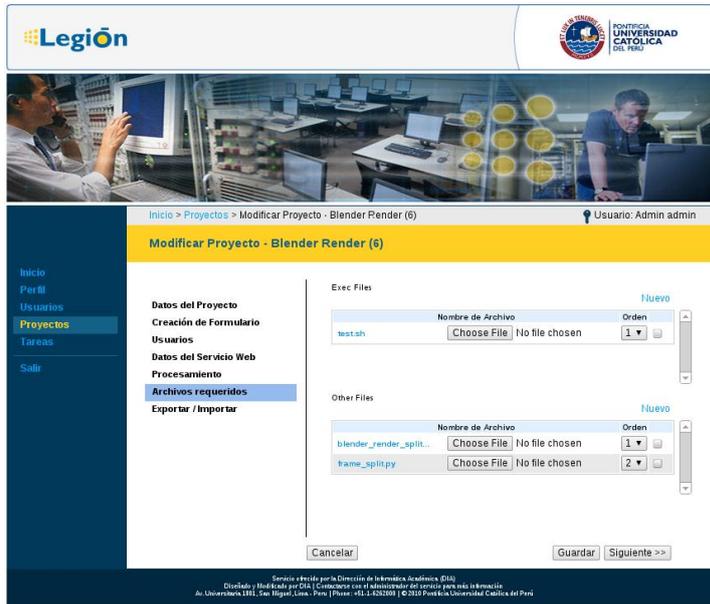
1 <task>
2   <name>{Form.txtName}</name>
3   <desc>{Form.txtDescription}</desc>
4   <exec_file>
5     <name>Integrator</name>
6     <cmd>{Form.txtEx} {x} {Form.txtStep} {Form.txtP
7     <compress>{Project.compress}</compress>
8   </exec_file>
9   <process>
10    <tag></tag>
11    <start>{Form.txtStart}</start>
12    <stop>{Form.txtStop}</stop>
13  </task>
  
```

Below the XML editor, there are fields for 'WU No. Formula' with the value '{Math.({Form.btStop}-{Form.btStart}){Form.btStep}' and 'Intérprete de plantillas' with the value 'DefaultTemplateInterpreter'. At the bottom, there are 'Cancelar', 'Guardar', and 'Siguiente >>' buttons.

## Paso 5 – Procesamiento

- **Validate** – Permite verificar que el documento xml sea correcto, mostrando dónde se encuentran los errores en caso contrario (usando la sintaxis del *Config.XML*).
- **Insert** – Permite añadir el elemento del formulario seleccionado en el lugar donde se encuentre el cursor.

## Paso 6: Archivos Requeridos



Inicio > Proyectos > Modificar Proyecto - Blender Render (6) Usuario: Admin admin

**Modificar Proyecto - Blender Render (6)**

Inicio  
Perfil  
Usuarios  
**Proyectos**  
Tareas  
Salir

Datos del Proyecto  
Creación de Formulario  
Usuarios  
Datos del Servicio Web  
Procesamiento  
**Archivos requeridos**  
Exportar / Importar

Exec Files Nuevo

Nombre de Archivo	Orden
test.sh	1

Other Files Nuevo

Nombre de Archivo	Orden
blender_render_split...	1
frame_split.py	2

Cancelar Guardar Siguiente >>

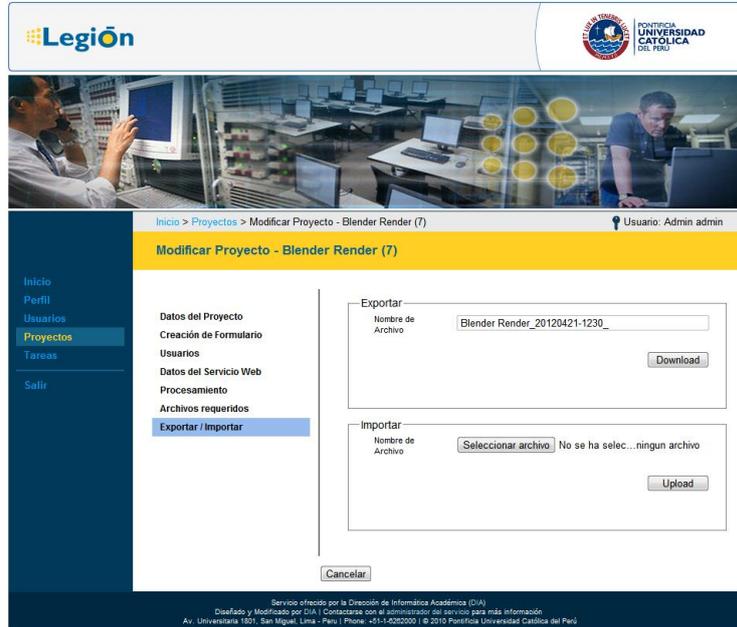
Desarrollado por la Dirección de Informática Académica (DIA)  
Diseñado y Mantenido por DTA. Contacto con el responsable de soporte por esta información:  
Av. Universitaria 1801, San Miguel, Lima - Perú | Phone: +51 1 6252000 | © 2010 Pontificia Universidad Católica del Perú

### Paso 6 – Archivos requeridos

Esta pantalla permite subir los archivos de ejecución y soporte que se usarán en el proyecto luego de presionar el botón “Guardar”. El select al costado de cada fila, permite reordenar los archivos.

## Exportar/Importar

Esta pantalla permite exportar o importar un proyecto.



Legión

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Inicio > Proyectos > Modificar Proyecto - Blender Render (7) Usuario: Admin admin

Modificar Proyecto - Blender Render (7)

Datos del Proyecto  
Creación de Formulario  
Usuarios  
Datos del Servicio Web  
Procesamiento  
Archivos requeridos  
**Exportar / Importar**

Exportar  
Nombre de Archivo: Blender Render\_20120421-1230\_  
Download

Importar  
Nombre de Archivo: Seleccionar archivo No se ha selec...ningun archivo  
Upload

Cancelar

Servicio ofrecido por la Dirección de Informática Académica (DIA)  
Diseñado y Modificado por DIA | Contactarse con el administrador del servicio para más información  
Av. Universitaria 1801, San Miguel, Lima - Perú | Phone: +51-1-4232000 | © 2010 Pontificia Universidad Católica del Perú

## Exportar / Importar proyectos

En la opción de Exportar permite descargar el proyecto en un archivo comprimido tipo zip. Si se importa un proyecto este sobre escribirá el existente.



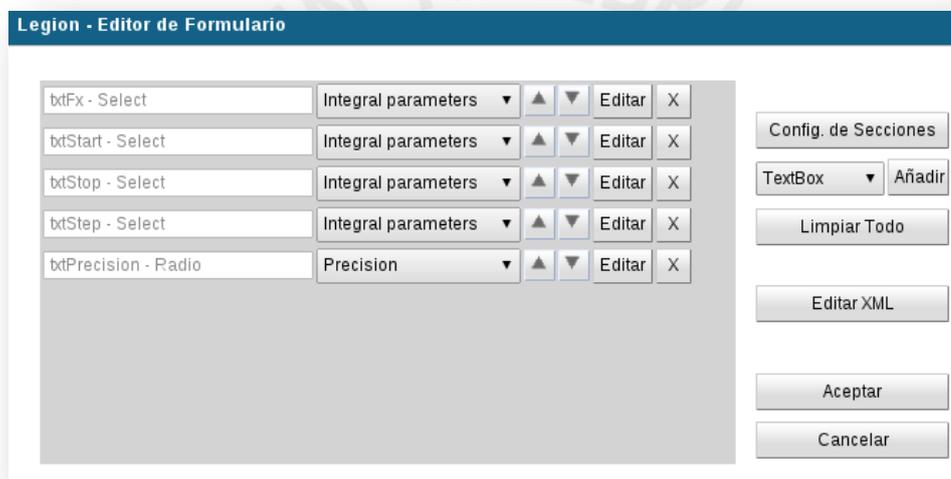
## Editor gráfico de formularios

---

Para acceder al editor gráfico de formularios se tiene que seleccionar el botón “Editor”. Hecho esto, se desplegará una ventana que permite diseñar el formulario de envío de tareas. Esta ventana presenta las siguientes opciones:

- **“Configuración de secciones”** - Permite crear y ordenar las secciones. Estas permiten agrupar componentes en pestañas en el formulario que se mostrará.

- **“Añadir componentes”** - Añade un nuevo componente, del tipo que se encuentra seleccionado en la lista.
- **“Limpiar todo”** - Elimina todos los componentes en el listado.
- **“Editar XML”** - Permite editar el formulario desde su representación XML. Los cambios se reflejarán en la ventana principal.
- **“Aceptar”** - Permite actualizar el formulario.
- **“Cancelar”** - Cancela la operación, y las modificaciones no se guardan.



### Editor gráfico de formularios

#### Nuevo TextBox

Al insertar un nuevo elemento del tipo “TextBox”.

- **HTML ID** – es el id con el que se podrá recuperar el valor del componente.
- **Etiqueta** – será la etiqueta que identificará el campo.
- **Mensaje** – un breve mensaje clarificador para el usuario; puede incluir indicaciones sobre el contenido del campo.

- **Por Defecto** – valor por defecto para el campo.
- **Validación** – indica cómo debe ser validado el campo. Para un “TextBox” se puede seleccionar validación numérica o para cadenas. La validación es ejecutada tanto en el navegador (función javascript) como en el servidor.



TextBox Editor

HTML ID\*

Etiqueta\*

Mensaje

Por Defecto

Validación

### TextBox Editor

#### Validación numérica

La validación numérica permite restringir el contenido ingresado en un campo “TextBox” donde se espere un valor numérico, ya sea entero o de punto flotante.

- **Min** – Representa el valor mínimo admitido para este campo. De quedar vacío, no se ejecuta la validación por el criterio de “valor mínimo admitido”.
- **Max** - Representa el valor máximo admitido para este campo. De quedar vacío, no se ejecuta la validación por el criterio de “valor máximo admitido”.
- **Type** – Representa que tipo de número se espera. Valores admitidos: Integer (Entero) o Float (Número de punto flotante).



### Validación numérica

#### Validación de cadenas

La validación de cadenas permite restringir el contenido ingresado en un campo “TextBox” en términos de longitud mínima y máxima.

- **Min Longitud** – Longitud mínima admitida. De quedar vacío, no se ejecuta la validación por el criterio de “Longitud mínima admitida”.
- **Max Longitud** – Longitud máxima admitida. De quedar vacío, no se ejecuta la validación por el criterio de “Longitud máxima admitida”.



### Validación de cadenas

#### Nuevo TextArea

Al insertar un nuevo elemento del tipo “TextArea”.

- **HTML ID** - Es el id con el que se podrá recuperar el valor del componente.
- **Etiqueta** - Es la etiqueta que identificará el campo.

- **Mensaje** – Es un breve mensaje clarificador para el usuario; puede incluir indicaciones sobre el contenido del campo.
- **Por Defecto** - Valor por defecto para el campo.
- **RE Validación** – Expresión regular por la que se validará el campo, tanto en el navegador (javascript) como en el servidor.



### TextArea Editor

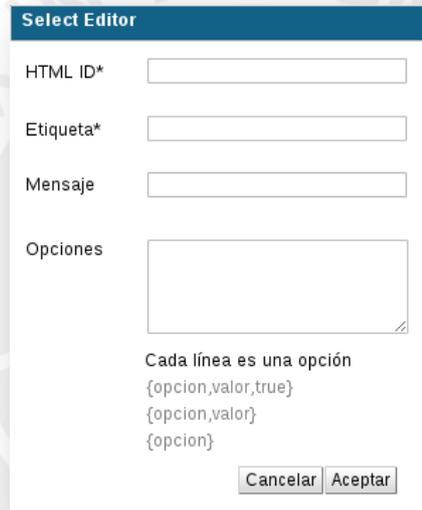
#### Nuevo Select

Al insertar un nuevo elemento del tipo “**Select**”.

- **HTML ID** - Es el id con el que se podrá recuperar el valor del componente.
- **Etiqueta** - Es la etiqueta que identificará al campo.
- **Mensaje** – Es un breve mensaje clarificador para el usuario; puede incluir indicaciones sobre el contenido del campo.
- **Opciones** – Las opciones que estarán presentes en el Select. Es importante notar que los valores que se renderizan ocultan el contenido del campo “valor”, así no será posible cambiarlos modificando el DOM. Se debe ingresar una opción por línea y con el siguiente formato:
  - {opción} : Se creará una opción con el mismo nombre y valor (por defecto no

estará seleccionada).

- {opción, valor} : Se creará una opción con nombre y valor eventualmente distintos. Se reemplazará el contenido de “valor” como resultado del campo select.
- {opción, valor, [true | false] } : Se creará una opción igual que en la alternativa anterior. Sin embargo, si se coloca “true” en el tercer parámetro, esa línea representará la opción por defecto, y se encontrará seleccionada cuando se cargue el formulario.



**Select Editor**

HTML ID\*

Etiqueta\*

Mensaje

Opciones

Cada línea es una opción

{opcion,valor,true}

{opcion,valor}

{opcion}

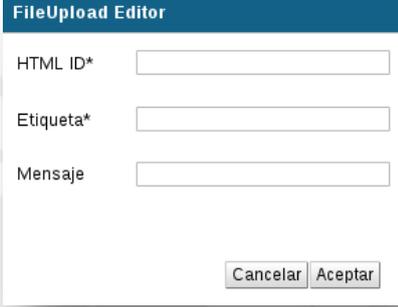
Cancelar Aceptar

### Select Editor

## Nuevo FileUpload

Al insertar un nuevo elemento del tipo “FileUpload”.

- **HTML ID** - Es el id con el que se podrá recuperar el valor del componente.
- **Etiqueta** - Es la etiqueta que identificará el campo.
- **Mensaje** – Es un breve mensaje clarificador para el usuario; puede incluir indicaciones sobre el contenido del campo.



FileUpload Editor

HTML ID\*

Etiqueta\*

Mensaje

Cancelar Aceptar

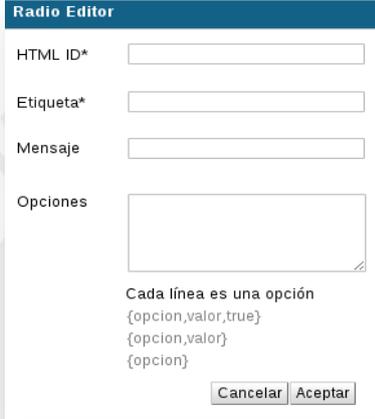
## FileUpload Editor

## Nuevo Radio

Al insertar un nuevo elemento del tipo “Radio”.

- **HTML ID** - Es el id con el que se podrá recuperar el valor del componente.
- **Etiqueta** - Es la etiqueta que identificará el campo.
- **Mensaje** – Es un breve mensaje clarificador para el usuario; puede incluir indicaciones sobre el contenido del campo.
- **Opciones** – Las opciones que estarán presentes en el Grupo de RadioButton. Es importante notar que los valores que se renderizan ocultan el contenido del campo “valor”, así no es posible cambiarlos modificando el DOM. Se debe ingresar una opción por línea y con el siguiente formato:
  - {opción} : Se creará una opción con el mismo nombre y valor (por defecto no estará seleccionada).

- {opción, valor} : Se creará una opción con nombre y valor eventualmente distintos. Se reemplazará el contenido de “valor” como resultado del campo Radio.
- {opción, valor, [true | false] } : Se creará una opción igual que en la alternativa anterior. Sin embargo, si se coloca “true” en el tercer parámetro, esa línea representará la opción por defecto, y se encontrará seleccionada cuando se cargue el formulario.



### Radio Editor

#### Nuevo Checkbox

Al insertar un nuevo elemento del tipo “**Checkbox**”

- **HTML ID** - Es el id con el que se podrá recuperar el valor del componente.
- **Etiqueta** - Es la etiqueta que identificará el campo.
- **Mensaje** – Es un breve mensaje clarificador para el usuario; puede incluir indicaciones sobre el contenido del campo.
- **Por defecto** – Si se selecciona el checkbox “Seleccionado”, el checkbox renderizado en el formulario aparecerá con ese valor inicial.

**Checkbox Editor**

HTML ID\*

Etiqueta\*

Mensaje

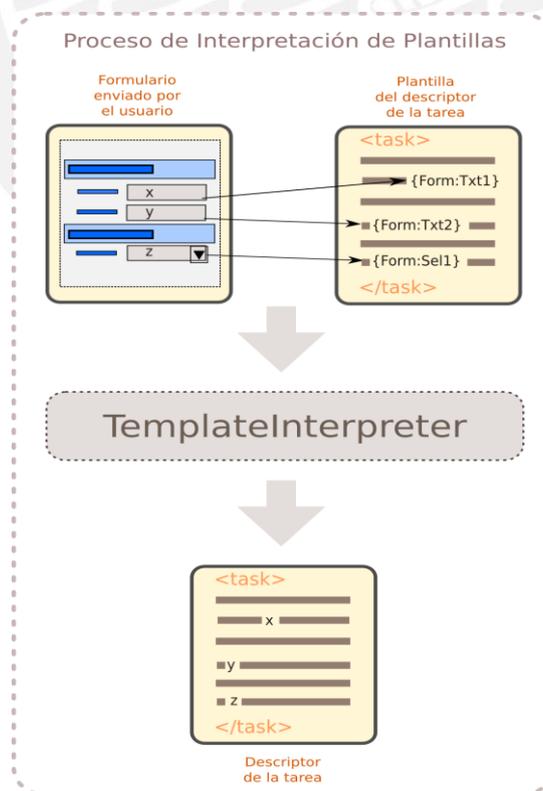
Por Defecto  Seleccionado

### Checkbox Editor



## Intérprete de Plantillas (Template Interpreter)

Legión Framework incluye un “DefaultTemplateInterpreter” para la generación del archivo descriptor de la tarea que es requerido por Legión Web Services. Este descriptor indica como se creará la tarea; es por ello que debe contener información que sólo está disponible en el momento del envío de estas. Se proporcionan etiquetas para acceder a los parámetros enviados en el formulario para la construcción del archivo **config.xml**. En la etapa de creación del proyecto se debe especificar la plantilla y qué intérprete de plantilla se va a utilizar.



### Funcionamiento del TemplateInterpreter

## Remplazo de variables de formulario

Si se ingresa un componente HTML en el formulario y se identifica, por ejemplo, por el id "txtSomething"; entonces, sólo se tendrá que utilizar la etiqueta **{Form:txtSomething}** en la plantilla o "template" para obtener el valor de este componente al realizar el remplazo de la etiqueta.

## La evaluación matemática

Si fuera necesario calcular un valor antes de sustituirlo, solo se tiene que añadir una etiqueta de evaluación matemática: **{Math:operación matemática}**. Se puede utilizar los operadores básicos: suma (+), resta (-), multiplicación (\*) y división (/) en conjunción con el remplazo de valores del formulario. Es válido también el uso de paréntesis como signos de colección.

Por ejemplo:

**{Math:({Form:txtLast} - {Form:txtFirst}) / {Form:txtStep}}**

Calculará:

$Reemplazo = \frac{txtLast - txtFirst}{txtStep}$ , usando valores enviados por el usuario final a través del formulario de envío de tareas.

## Operador "Split"

A veces será necesario separar los valores de un componente HTML. Por ejemplo, si se tuviera un conjunto de parámetros para ejecutar en cada línea de un "TextArea", probablemente sería necesario separar el valor devuelto por este campo, usando el carácter de cambio de línea "\n" como separador. Para conseguir esto, si el campo se llamara por ejemplo: txtCommands, bastaría usar la etiqueta "Split" de la siguiente forma:

**<workunit>{Split:txtCommands: \n}</workunit>**

*Cabe resaltar que toda la línea se repite. Es decir, si la etiqueta está rodeada por texto, como es el caso de la etiqueta "workunit" en el ejemplo, este también se repetirá. Además, se puede usar cualquier carácter o cadena de caracteres como separador.*

**<workunit> línea 1 </workunit>**

**<workunit> línea 2 </workunit>**

**<workunit> línea 3 </workunit>**

...



Esta etiqueta puede ser usada en conjunto con la etiqueta “**SplitCount**”, que permite contar la cantidad de elementos generados al realizar la operación de separación o “Split”.

Por ejemplo:

```
{SplitCount:txtCommands:\n}
```

Esta etiqueta será remplazada por el valor numérico de la cantidad de líneas no vacías en el componente txtCommands.

## Implementación de un nuevo “TemplateInterpreter”

Con el fin de sobrescribir el comportamiento por defecto del intérprete de plantillas provisto, se puede colocar una clase que implemente la interfaz `ITemplateInterpreter` en el paquete `pe.edu.pucp.legion.webinterface.templateinterpreter.extensions`, el nombre de esta clase debe terminar en “`TemplateInterpreter`” ej. “`ExtendedTemplateInterpreter`”. Después de esta operación, la clase estará disponible para ser seleccionada en el paso 5 de la creación de proyectos, pues la lista es cargada por inspección del mencionado paquete.



### Listado de TemplateInterpreters disponibles

Esta operación permite que se implementen un conjunto de etiquetas que pueden ser usadas en un tipo particular de creación de tareas, o también podría ser útil al momento de implementar la integración con otro Backend.

Un nuevo `TemplateInterpreter` requiere que se implementen los siguientes métodos:

```
public interface ITemplateInterpreter {
    public String interpret(String template);
    public void setForm(Form form);
    public void setFormValues(HashMap<String,String> formValues);
    public void setProjectValues(HashMap<String,String> projectValues);
    public void setOtherValues(HashMap<String,Object> otherValues);
}
```

## Construcción del descriptor de tareas

Existen dos modos principales para la creación de tareas:

- Grupo de tareas, o
- Tareas simples.

Es importante notar que una tarea estará compuesta por una o muchas unidades de trabajo. A su vez, un grupo de tareas estará compuesto por una o más tareas.

Para especificar una tarea es necesario conocer la información que solo está disponible desde el momento en que las tareas son enviadas, por ello el **TemplateInterpreter** permite especificar una plantilla del descriptor de la tarea que tenga “información por completar” de modo que sea genérica. Así, después de ejecutar un remplazo de etiquetas, se obtendrá el descriptor de la tarea que se desea enviar.

Cabe resaltar que el concepto de grupo de tareas al igual que el **TemplateInterpreter**, no están implementados en Legión Web Services y constituyen conceptos propios de Legión Web Interface. De ese modo, si fuera necesario enviar tareas a Legión Web Services directamente, por ejemplo desde una aplicación propia mediante el uso de los bindings para Java y PHP disponibles, se tendría que enviar un archivo descriptor de la tarea ya “interpretada”. El descriptor de la tarea puede ser accedido por el administrador desde la búsqueda avanzada de tareas.

## Etiquetas del descriptor de tareas

A continuación se presentan todas las etiquetas admitidas en el Descriptor de Tareas. Para más detalle de como crear este archivo, se puede consultar el diagrama de flujo de “Extended Mode” o “Basic Mode”.

Etiqueta	Contenido	Descripción
<b>ROOT (raíz)</b>	<code>task   group</code>	Nodo raíz. El archivo XML debe empezar con una etiqueta <code>task</code> o <code>group</code> .
<b>ROOT &gt; group</b>	<code>repeat</code>	Representa un grupo de tareas. Donde cada una de estas es creada siguiendo un criterio de repetición consignado en la etiqueta “repeat” y usando la estructura base que se encuentra descrita en “task”.
	<code>task</code>	Una lista de parámetros que son accedidos usando {Task}, y

		que varían por cada tarea del grupo, permiten diferenciar las tareas entre sí ( <i>ver Template Interpreter - Reemplazo de {Task}</i> ).
<b>ROOT &gt; task</b>	<b>name</b>	Representa una tarea, indicando como se creará. Una
	<b>desc</b>	tarea es una colección de unidades de trabajo o
	<b>compress?</b>	“workunits” y produce necesariamente uno o más
	<b>queue?</b>	archivos resultados, usando archivos ejecutables y
	<b>((basic_mode, works?)   exec_file*)</b>	eventualmente otros archivos de apoyo.
	<b>other_file*</b>	
	<b>out_file+</b>	
	<b>boinc?</b>	
<b>task &gt; name</b>	<i>Cadena de texto</i>	Nombre de la tarea, debe ser descriptivo y no necesariamente único.
<b>task &gt; desc</b>	<i>Cadena de texto</i>	Descripción de la tarea.
<b>task &gt; compress</b>	<i>Cadena de texto</i> <i>( 0   1 )</i>	<b>Posibles valores:</b> 0 : no se comprime 1 : sí se comprime
<b>task &gt; queue</b>	<i>Cadena de texto</i>	Número de veces que se repetirá la tarea (El mismo workunit es enviado múltiples veces - muy útil para simulaciones de Montecarlo).

<b>task &gt; basic_mode</b>	<Vacío>	Esta etiqueta no presenta contenido, pero de encontrarse presente indica que se usará el modo básico o “basic mode”.
<b>task &gt; works</b>	workunit*	Esta etiqueta contiene tantos “workunit” como se requiera para la tarea.
<b>task &gt; works &gt; workunit</b>	<i>Cadena de texto</i>	Esta etiqueta contiene la línea de comandos correspondientes al workunit que se creará.
<b>task &gt; exec_file</b>	<b>name</b>	Esta etiqueta permite especificar la ejecución de un programa.
	<b>out_filename?</b>	
	<b>path?</b>	
	<b>chk_filename?</b>	
	<b>cmd?</b>	
	<b>compress?</b>	
	<b>extern?</b>	
<b>process*</b>		
<b>task &gt; exec_file &gt; name</b>		Nombre que tomará el ejecutable
<b>task &gt; exec_file &gt; out_filename</b>		
<b>task &gt; exec_file &gt; path</b>		
<b>task &gt; exec_file &gt; chk_filename</b>		Nombre del checkpoint.

<p><b>task &gt; exec_file &gt; cmd</b></p>		<p>La línea de comandos que será enviada al programa.</p>
<p><b>task &gt; exec_file &gt; compress</b></p>		
<p><b>task &gt; exec_file &gt; extern</b></p>	<p>&lt;Vacío&gt;</p>	<p>Indica si el archivo será enviado por la interfaz web, o si se asume que reside en el sistema operativo destino. Por ejemplo, si se quisiera usar un programa en Python (que se encuentra instalado en el SO) no se debería incluir esta etiqueta. Por otro lado, si se envía el programa a ejecutar sí debería usar.</p>
<p><b>task &gt; exec_file &gt; process</b></p>	<p>tag</p>	<p>La etiqueta “<b>process</b>” permite ejecutar un barrido de parámetros. Esto permitirá que el parámetro tome un valor distinto en cada línea de comandos del “<b>exec_file</b>”, generando así múltiples unidades de trabajo. La cantidad de workunits está dada por:</p> $ Process_x  = \frac{stop_x - start_x}{step_x}$ <p>Por otro lado, es posible combinar múltiples tags “process”. La cantidad de workunits será en total:</p>
	<p>start</p>	
	<p>stop</p>	
	<p>step</p>	

		$\text{NúmeroWorkunits} = \prod_i  \text{Process}_i $
<b>task &gt; exec_file &gt; process &gt; tag</b>	<i>Cadena de texto</i>	Nombre por el cuál se identificará al parámetro que se hará variar en el rango indicado.
<b>task &gt; exec_file &gt; process &gt; start</b>	<i>Cadena de texto</i>	Valor numérico del primer valor que tomará el barrido de un parámetro en particular.
<b>task &gt; exec_file &gt; process &gt; stop</b>	<i>Cadena de texto</i>	Valor numérico del límite exclusivo del barrido de un parámetro en particular.
<b>task &gt; exec_file &gt; process &gt; step</b>	<i>Cadena de texto</i>	Valor numérico del “paso” que separará a dos elementos consecutivos del barrido de parámetros. Si fuera necesario un barrido de parámetros de paso no constante, este debe ser implementado por la aplicación a ejecutar de modo que el valor usado sea $y = f(x)$ , donde $x$ es el valor del parámetro para un workunit y $f$ es una función que realiza el mapeo o una función de hash.
<b>task &gt; other_file</b>	<i>name</i>	Referencia a un archivo adicional que será subido por el administrador en el apartado “Otros Archivos” o enviado por el usuario mediante un campo “File Uploader” en el formulario de envío de tareas. Puede ser útil, por ejemplo, para agregar archivos script,
	<i>path?</i>	

		<p>imágenes a procesar, etc.</p> <p><b>NOTA:</b> El orden de los archivos en el documento XML es de suma importancia, pues esto permite identificarlos correctamente.</p> <p>Deben encontrarse en el mismo orden que los archivos en el listado de “Otros Archivos” seguidos por los componentes “File Uploader” en el orden en el que se encuentran en el paso 6 de la creación de proyectos y en el formulario de envío de tareas, respectivamente.</p>
<b>task &gt; other_file &gt; name</b>	<i>Cadena de texto</i>	Indica el nombre con el que será copiado en la carpeta de trabajo. Esto sin importar el nombre del archivo que se subió.
<b>task &gt; other_file &gt; path</b>	<i>Cadena de texto</i>	
<b>task &gt; out_file</b>	<i>type</i>	Indica que archivos se deben recuperar después de la ejecución satisfactoria de la unidad de trabajo.
	<i>name</i>	
<b>task &gt; out_file &gt; type</b>	<i>Cadena de texto</i>  <i>(None   cat )</i>	<p><b>Posibles valores:</b></p> <p><b>None</b> : no se ejecutará operación adicional</p> <p><b>cat</b> : se concatenarán todos los archivos, en el orden en que</p>

		fueron creados.
<b>task &gt; out_file &gt; name</b>	<i>Cadena de texto</i> <i>ej. result.txt</i>	Nombre del archivo que se recuperará. Es importante notar que este archivo debe haber sido creado durante la ejecución de la unidad de trabajo con exactamente el mismo nombre.
<b>task &gt; boinc</b>	<i>max_cpu_time?</i> <i>max_disk_space?</i> <i>min_quorum?</i> <i>target_nresults?</i> <i>max_error_results?</i> <i>max_total_results?</i> <i>max_success_results?</i> <i>rsc_fposts_bound?</i> <i>rsc_fposts_est?</i> <i>rsc_memory_bound?</i>	Parámetros propios de BOINC. Referirse a la documentación de BOINC para un mejor entendimiento de la aplicación de los parámetros.
<b>task &gt; boinc &gt; max_cpu_time</b>	<i>Cadena de texto</i>	Parámetros propios de BOINC.
<b>task &gt; boinc &gt; max_disk_space</b>	<i>Cadena de texto</i>	Parámetros propios de BOINC.
<b>task &gt; boinc &gt; min_quorum</b>	<i>Cadena de texto</i>	Parámetros propios de BOINC.
<b>task &gt; boinc &gt; target_nresults</b>	<i>Cadena de texto</i>	Parámetros propios de BOINC.
<b>task &gt; boinc &gt; max_error_results</b>	<i>Cadena de texto</i>	Parámetros propios de BOINC.

<b>task &gt; boinc &gt; max_total_results</b>	<i>Cadena de texto</i>	Parámetros propios de BOINC.
<b>task &gt; boinc &gt; max_success_results</b>	<i>Cadena de texto</i>	Parámetros propios de BOINC.
<b>task &gt; boinc &gt; rsc_fpop_bound</b>	<i>Cadena de texto</i>	Parámetros propios de BOINC.
<b>task &gt; boinc &gt; rsc_fpop_est</b>	<i>Cadena de texto</i>	Parámetros propios de BOINC.
<b>task &gt; boinc &gt; rsc_memory_bound</b>	<i>Cadena de texto</i>	Parámetros propios de BOINC.



## Tareas simples

Una tarea simple queda definida por: un formulario, archivos ejecutables, otros archivos de soporte y el descriptor de la tarea, que es un archivo XML que indica como se crearán las unidades de trabajo en base al formulario que enviará eventualmente el usuario. Hay dos modos principales para el descriptor de una tarea simple: el “**Basic Mode**” o modo básico y el “**Extended Mode**” o modo extendido. El “Basic Mode” permite enviar un conjunto de líneas de comando a un proyecto BOINC que exista previamente, es decir, Legión Framework solo funcionará como interfaz web. Por otro lado, el “Extended Mode” permite describir la tarea, enviar los archivos ejecutables y ejecutar operaciones como barrido de parámetros.

- Elemento Raíz: `<task></task>`
- Nombre (Obligatorio): `<name></name>`
- Descripción (Obligatorio): `<desc></desc>`
- Archivos ejecutables (en el orden en que se ejecutarán y en el orden que aparecen en el paso que pide subir los archivos ejecutables – en caso de discrepancia, reordenar el listado de archivos):
  - Nombre : `<name>ruta en el Sistema Operativo destino | nombre </name>`
  - Línea de Comandos: `<cmd></cmd>` - Parámetros que se entregarán a la aplicación
  - Compress : Si se comprimirán los resultados.
  - Extern: `<extern/>` estará presente si el archivo ejecutable se envía por el formulario de creación de proyectos; en caso contrario, debería residir en la computadora destino.

```

<task>
  <name>{Form:txtName}</name>
  <desc>{Form:txtDescription}</desc>
  <exec_file>
    <name>integrador</name>
    <cmd>"{Form:txtFx}" {x} {Form:txtStep} {Form:txtPrecision}</cmd>
    <compress>{Project:compress}</compress>
    <extern/>
  </exec_file>
  <process>
    <tag>x</tag>
    <start>{Form:txtStart}</start>
    <stop>{Form:txtStop}</stop>
    <step>{Form:txtStep}</step>
  </process>
  <out_file>
    <type>cat</type>
    <name>result.txt</name>
  </out_file>

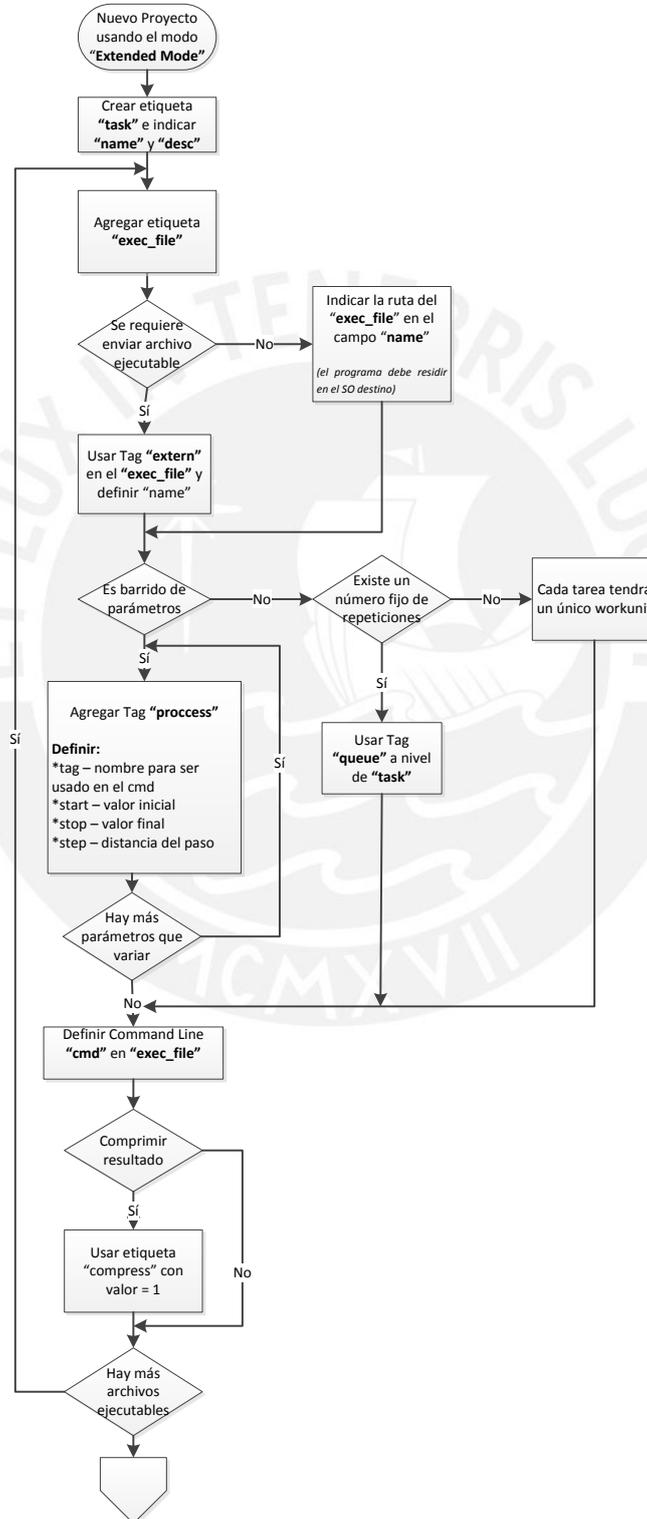
```

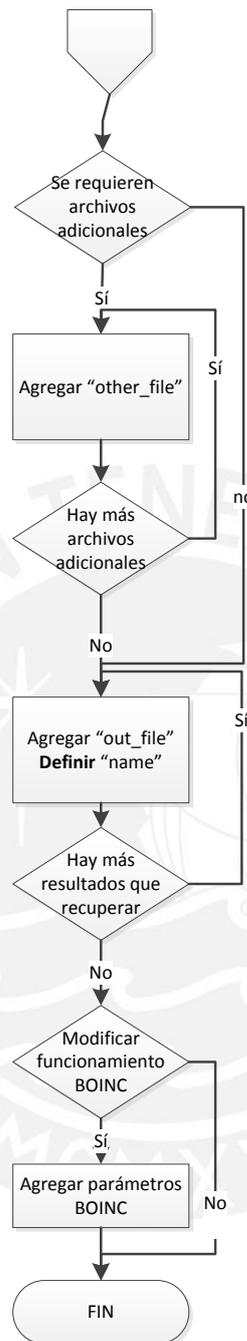
</task>



## Extended Mode

Para crear un descriptor de tareas usando el “Extended Mode” o modo extendido, puede basarse en el siguiente diagrama de flujo.



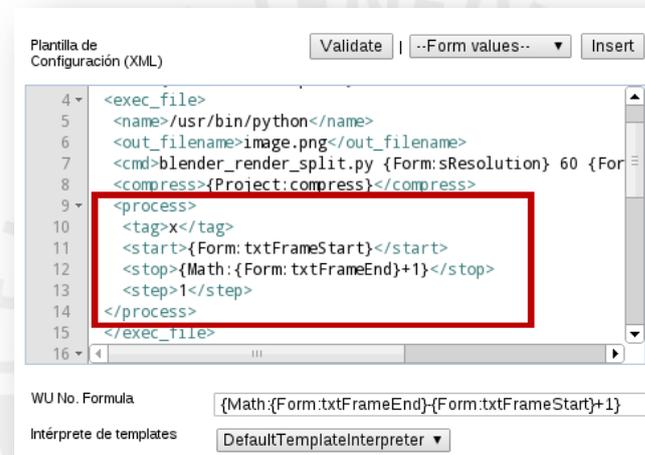


La principal funcionalidad del modo extendido o “Extended Mode” es permitir que uno de los administradores de Legión Framework pueda seleccionar los programas que se ejecutarán sin necesidad de crear un nuevo proyecto BOINC. Al permitir que múltiples proyectos de Legión Framework trabajen con un único proyecto y aplicación de BOINC, reduce la necesidad de configuraciones adicionales. Además, permite la ejecución de barrido de parámetros para crear tareas (las simulaciones de Montecarlo, por ejemplo, encajan en este tipo de proyectos). Muchas aplicaciones como análisis de sensibilidad pueden ejecutarse mediante un barrido de parámetros con el tag “process”. Por otro lado, cuando cada ejecución presente un comportamiento aleatorio sin variar ningún

parámetro, sería posible usar la etiqueta “queue” que repite un mismo workunit muchas veces.

### Ejemplo: Barrido de parámetros para renderizar videos en 3D

Otro ejemplo de barrido de parámetros es el renderizado de películas 3D con Blender. En la imagen a continuación se puede observar una plantilla para crear tareas de *renderización* de videos 3D con Blender, haciendo un barrido entre un fotograma inicial y uno final, ambos valores enviados por el usuario a través de un formulario diseñado con ese fin. Como se puede observar, esto se implementa usando la etiqueta “process” que hace el barrido desde “txtFrameStart” hasta “txtFrameEnd” inclusive.



```

Plantilla de Configuración (XML)
Validate 1 --Form values-- Insert

4 <exec_file>
5 <name>/usr/bin/python</name>
6 <out_filename>image.png</out_filename>
7 <cmd>blender_render_split.py {Form:sResolution} 60 {For
8 <compress>{Project:compress}</compress>
9
10 <process>
11 <tag>x</tag>
12 <start>{Form:txtFrameStart}</start>
13 <stop>{Math:{Form:txtFrameEnd}+1}</stop>
14 <step>1</step>
15 </process>
16 </exec_file>

WU No. Formula {Math:{Form:txtFrameEnd}-{Form:txtFrameStart}+1}
Intérprete de templates DefaultTemplateInterpreter
  
```

**Plantilla usando “Extended Mode”**

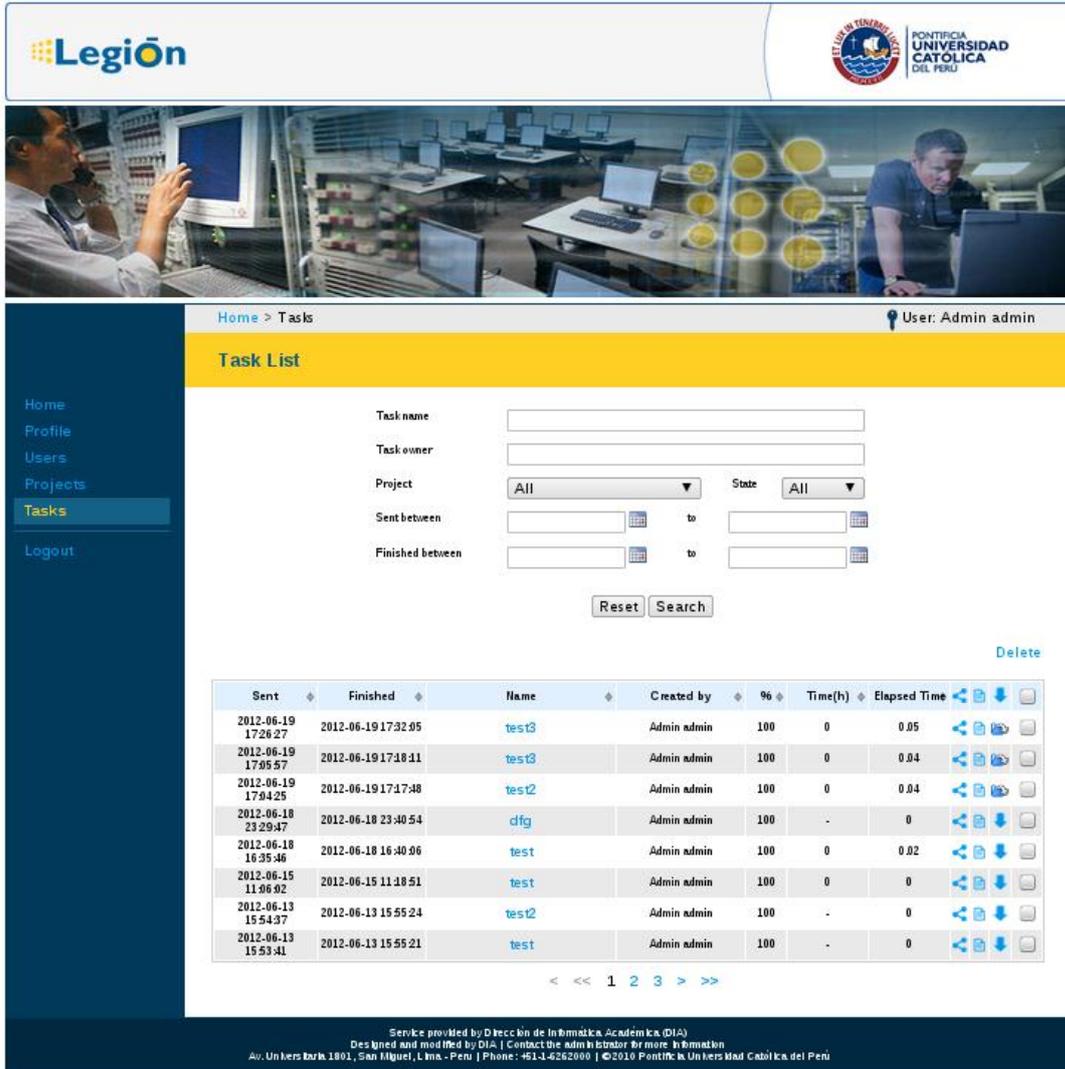
## Basic Mode

En el modo básico, Legión Framework actúa únicamente como interface para un proyecto existente en BOINC. Por lo que será únicamente necesario el envío de las líneas de comando que formaran la tarea.

### Por ejemplo:

```
<task>
  <name>{Form:txtName}</name>
  <desc>{Form:txtDescription}</desc>
  <basic_mode/>
  <works>
    <workunit>{Split:txtTasks:\n}</workunit>
  </works>
  <out_file>
    <type>cat</type>
    <name>out</name>
  </out_file>
</task>
```

## Búsqueda avanzada de tareas



The screenshot shows the 'Legión' web application interface. At the top, there is a navigation bar with the 'Legión' logo and the university's name. Below this is a banner image of a computer lab. The main content area is titled 'Task List' and includes a sidebar with navigation options: Home, Profile, Users, Projects, Tasks (selected), and Logout. The 'Task List' section features search filters for Task name, Task owner, Project (set to 'All'), State (set to 'All'), Sent between, and Finished between. There are 'Reset' and 'Search' buttons. Below the filters is a table of tasks with columns for Sent, Finished, Name, Created by, %, Time(h), and Elapsed Time. The table contains 10 rows of task data. At the bottom of the page, there is a footer with contact information for the Dirección de Informática Académica (DIA).

Sent	Finished	Name	Created by	%	Time(h)	Elapsed Time
2012-06-19 17:26:27	2012-06-19 17:32:05	test3	Admin admin	100	0	0.05
2012-06-19 17:05:57	2012-06-19 17:18:11	test3	Admin admin	100	0	0.04
2012-06-19 17:04:25	2012-06-19 17:17:48	test2	Admin admin	100	0	0.04
2012-06-18 23:29:47	2012-06-18 23:40:54	cfg	Admin admin	100	-	0
2012-06-18 16:35:46	2012-06-18 16:40:06	test	Admin admin	100	0	0.02
2012-06-15 11:06:02	2012-06-15 11:18:51	test	Admin admin	100	0	0
2012-06-13 15:54:37	2012-06-13 15:55:24	test2	Admin admin	100	-	0
2012-06-13 15:53:41	2012-06-13 15:55:21	test	Admin admin	100	-	0

### Búsqueda avanzada de tareas

La búsqueda de tareas permite filtrar por nombre de la tarea, usuario creador, proyecto, estado, fecha de envío y fecha de finalización. Se podrá descargar, compartir y ver archivos enviados para la creación de la tarea.

## Acceso de usuarios finales

El acceso a usuarios finales es más limitado que para los administradores. Para comenzar, sus pantallas iniciales son distintas. Un usuario final podrá enviar, monitorear la ejecución, y descargar los resultados de las tareas.

### Pantalla de Inicio del Usuario

La pantalla de inicio para usuarios finales muestra un listado de los proyectos en los que está registrado el usuario, junto con las tareas más recientes de cada uno de estos. Se puede enviar inmediatamente una nueva tarea haciendo click en “nuevo” y visitar la página de monitoreo de tareas de un proyecto haciendo click en “ir”.



User: Normal User

**Home**

- Home
- Profile
- Logout

Name	Completed	Role
test_bd_xml_form by Admin Admin		Subscriber
test100 by Admin Admin	100%	Subscriber
test A4 by Admin Admin	100%	Subscriber
Sin_x_squared_integration	100%	Creator

Service provided by Dirección de Informática Académica (DIA)  
Designed and modified by DIA | Contact the administrator for more information  
Av. Universitaria 1801, San Miguel, Lima - Peru | Phone: +51-1-6262000 | © 2010 Pontificia Universidad Católica del Perú

### Pantalla de inicio para usuarios

#### Envío de tareas

Durante el envío de tareas, se mostrará el formulario “renderizado” que fue diseñado durante la fase de creación del proyecto. Una vez hecho click en “crear” la tarea se prepara para ser creada y se coloca en una cola por backend.



Inicio > Proyectos > Integrator > Nuevo Usuario: Admin admin

### Envío de tareas de Integrator

**Parametros Básicos**

Name

Description

---

**Integral parameters**

f(x)  Function to integrate

Lower Bound  Choose a value for the lower bound

Upper Bound  Choose a value for the upper bound

Step  Choose a value for step

**Precision**

Servicio ofrecido por la Dirección de Informática Académica (DIA)  
 Diseñado y Modificado por DIA | Contactarse con el administrador del servicio para más información  
 Av. Universitaria 1801, San Miguel, Lima - Peru | Phone: +51-1-6262000 | © 2010 Pontificia Universidad Católica del Perú

## Envío de tareas

## Proceso de creación de tareas

Las tareas en Legión Framework son enviadas a través de una interfaz web. Sin embargo, su creación en el servidor BOINC no es inmediata. Las tareas una vez enviadas por el usuario, son almacenadas en un “buffer” de tareas por Backend esperando su creación. Sin embargo, el buffer en mención es un concepto lógico, porque las tareas finalmente residen en la base de datos. Un hilo es iniciado por cada Backend que tenga tareas en espera de ser creadas. Este hilo las va ir creando, una por una, en el Servidor BOINC a través de los servicios SOAP ofrecidos por Legión Web Services.

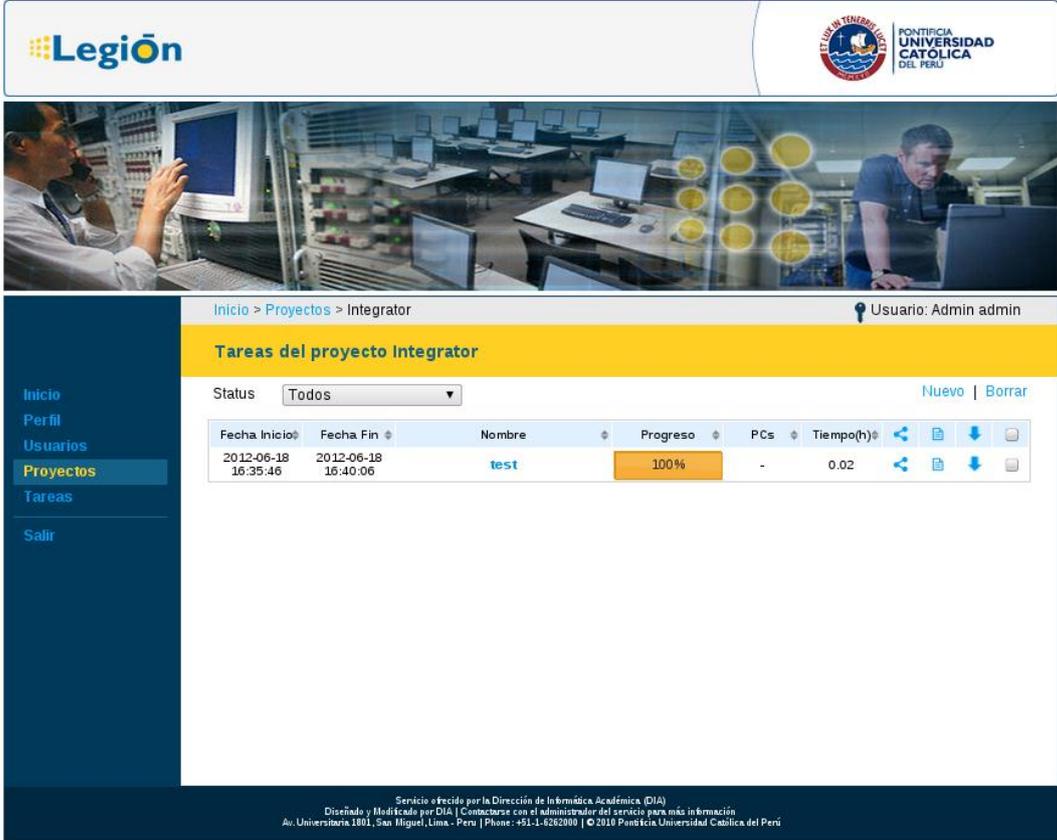


### Proceso de creación de tareas

El proceso de creación de una tarea se inicia en el momento en que el usuario, tras haber abierto el formulario de envío de tareas, hace click en crear. Los parámetros son recibidos por el TaskCreator designado en el paso 2 de la creación de proyectos.

Por defecto se usa “DefaultTaskCreator”. Sin embargo, de requerirse una modificación del comportamiento estándar, es posible usar una implementación propia.

## Monitoreo y descarga de tareas



The screenshot displays the 'Legión' web application interface. At the top, there is a navigation bar with the 'Legión' logo on the left and the university's name and logo on the right. Below the navigation bar is a banner image showing a server room with a person working at a computer. The main content area is titled 'Tareas del proyecto Integrator' and shows a table of tasks. The table has columns for 'Fecha Inicio', 'Fecha Fin', 'Nombre', 'Progreso', 'PCs', and 'Tiempo(h)'. A single task named 'test' is listed with a 100% progress bar. The interface also includes a sidebar with navigation options like 'Inicio', 'Perfil', 'Usuarios', 'Proyectos', 'Tareas', and 'Salir'. At the bottom, there is a footer with contact information for the Dirección de Informática Académica (DIA).

Fecha Inicio	Fecha Fin	Nombre	Progreso	PCs	Tiempo(h)
2012-06-18 16:35:46	2012-06-18 16:40:06	test	100%	-	0.02

Servicio o hecho por la Dirección de Informática Académica (DIA)  
Diseñado y Modificado por DIA | Contactarse con el administrador del servicio para más información  
Av. Universitaria 1801, San Miguel, Lima - Perú | Phone: +51-1-6262000 | © 2010 Pontificia Universidad Católica del Perú

### Monitoreo de tareas

La página del monitoreo de tareas permite visualizar la lista completa de tareas enviadas, junto con su progreso que será refrescado automáticamente en caso de tratarse de una tarea en ejecución. Se podrá asimismo suscribir a otros usuarios a las tareas, ver los archivos enviados para la creación de tarea y descargar los resultados.

## **Anexo I : Instalación de Berkeley Open Infrastructure for Network Computing [Basado en un Manual Interno de la Organización]**

---

En este anexo se tratará la instalación de Berkeley Open Infrastructure for Network Computing (BOINC), procedimiento que es prerequisite para la instalación de Legión Framework y consta aquí como referencia.

### ***Prerrequisitos***

Para instalar BOINC se tendrá que tener un servidor físico o virtual con un sistema operativo Linux instalado. Para propósitos prácticos de este manual, se menciona el proceso usando como referencia la distribución basada en RHEL, Scientific Linux 6. Si se usara una distribución distinta a la mencionada, es posible que alguno de los pasos tenga que ser ligeramente modificado.

### ***Compilación de BOINC***

Previo a la instalación de BOINC, será necesario contar con los paquetes requeridos por BOINC y Legión Framework. Para eso ejecute la siguiente línea de comandos en una

distribución de Linux basada en RHEL 6.

```
# yum install make m4 libtool autoconf automake gcc openssl mod_ssl sqlite m4
make gcc libtool m4 autoconf automake pkgconfig mysql-connector-odbc mysql
mysql-devel php-mysql mod_auth_mysql php-gd php-cli php-common openssl-
devel python-setuptools MySQL-python mysql++ mysql++-devel gcc-c++
python-cherrypy php python-lxml libxml2-devel libxslt-devel Cython python-
devel#
```

1. Ingresar al **Servidor** como usuario **root** y crear un usuario con nombre **"boincadm"**. Así mismo crear el grupo **"boinc"**, añadiendo a este grupo a los usuarios **"apache"** y **"boincadm"**.

```
# useradd boincadm
# groupadd boinc
# groupmems -a apache -g boinc
# groupmems -a boincadm -g boinc
```

Es importante notar que con el usuario **boincadm** se compilará y creará los proyectos en BOINC.

2. Ingresar como usuario **boincadm** y en la carpeta `/home/boincadm` proceder a descargar el código fuente de la versión estable de BOINC (r25760)

```
$ su boincadm
$ cd /home/boincadm
$ svn co http://boinc.berkeley.edu/svn/branches/server_stable
```

3. A continuación, proceder con la compilación:

```
$ cd /home/boincadm/server_stable/
$ ./_autosetup
...
Done, now run ./configure
./configure -C          to enable caching
$ ./configure --enable-server --disable-client --without-x
...
--- Configuring BOINC 7.1.0 (Release) ---
--- Build Components: ( manager libraries server) ---
$ make
...
make[1]: se sale del directorio `/home/boincadm/server_stable'
```

Con los comandos ejecutados anteriormente, el sistema ya se encuentra preparado para crear un proyecto BOINC.

## Referencias

<http://boinc.berkeley.edu/trac/wiki/SourceCode>

## Creación de un proyecto BOINC

1. Antes de empezar con la creación de proyecto se debe tener un usuario en la base de datos MySQL. El presente manual trabaja con el usuario “**boincadm**” con password “**clave1234**”, sin embargo esto solo es con propósitos demostrativos. Esta operación se puede realizar desde el cliente de línea de comandos de MySQL.

```
$ mysql
mysql> GRANT ALL ON BoincPrueba.* TO 'boincadm'@'localhost';
mysql> SET PASSWORD FOR 'boincadm'@'localhost' = PASSWORD('clave1234');
```

Es importante notar que el usuario debe tener todos los permisos necesarios para crear el esquema del proyecto, insertar, actualizar y borrar datos.

2. Como usuario **boincadm** se ingresará a la carpeta **/home/boincadm/server\_stable**, y luego, ejecutar la aplicación **make\_project** para crear un proyecto cuyo nombre será **BoincPrueba** (este nombre es solamente con propósitos demostrativos).

```
make_project [opciones] nombre_del_proyecto [ 'Nombre completo del proyecto' ]
```

```
$ cd
$ cd /home/boincadm/server_stable
$ ./tools/make_project --delete_prev_inst --user_name boincadm --drop_db_first --db_user
boincadm --db_passwd clave1234 --url_base http://{dominio o IP}/ BoincPrueba 'Proyecto
Boinc de Prueba'
```

Creating project 'Proyecto Boinc de Prueba' (short name 'BoincPrueba'):

```
PROJECT_ROOT = /home/boincadm/projects/BoincPrueba/
URL_BASE = http://{dominio o IP}/
HTML_USER_URL = http://{dominio o IP}/BoincPrueba/
HTML_OPS_URL = http://{dominio o IP}/BoincPrueba_ops/
CGI_URL = http://{dominio o IP}/BoincPrueba_cgi/
```

```

KEY_DIR = /home/boincadm/projects/BoincPrueba/keys/
DB_NAME = BoincPrueba
DB_HOST =
Continue? [Y/n] Y
Creating directories
Generating encryption keys
Copying files
Setting up database
Writing config files
Linking CGI programs
update_translations finished
Done installing default daemons.
Done creating project. Please view
/home/boincadm/projects/BoincPrueba/BoincPrueba.readme
for important additional instructions.

```

Las opciones utilizadas:

- delete\_prev\_inst**: Borra el directorio si existiera una instalación previa.
- drop\_db\_first**: Borra la base de datos si existiera una instalación previa.
- db\_user**: El nombre del usuario de la base de datos
- db\_password**: La contraseña del usuario de la base de datos.
- url\_base**: La dirección URL del proyecto. Puede ser una IP o un dominio, sin embargo es importante que esta sea fija, pues un cambio de dirección afecta el funcionamiento del proyecto.

Para ver más opciones ejecutar: **make\_project -help** . Si no se coloca nombre de la base de datos se crea una con el nombre corto del Proyecto.

3. Si el proyecto se hubiera creado correctamente, se podrá verificar que la carpeta **/home/boincadm/projects/BoincPrueba** se encuentra presente con el contenido mostrado a continuación. Así mismo, la base de datos “**BoincPrueba**” debe haber sido creada.

```

BoincPrueba
├── apps/
├── bin/
├── BoincPrueba.cronjob
├── BoincPrueba.httptd.conf
├── BoincPrueba.readme
├── cgi-bin/
├── config.xml
├── db_dump_spec.xml
├── db_revision
├── download/
├── html/
├── keys/
├── local.revision
└── log_boinclegion/

```

```

├── project.xml
├── py/
├── templates/
└── upload/
  
```

- Luego, se añade el permiso de acceso a la interfaz web de administración del proyecto. Las credenciales usadas son únicamente referenciales.

```

$ cd /home/boincadm/projects/BoincPrueba/html/ops/
$ htpasswd -c .htpasswd Administrador
  
```

New password: **clave1234**

Re-type new password: **clave1234**

Adding password for user Administrador

- Ejecutar como usuario **root** un cambio de permisos de las siguientes carpetas dentro del proyecto **BoincPrueba** con la finalidad de permitir el acceso web.

```

# cd /home/boincadm/projects/BoincPrueba
# chgrp -R apache html/languages
# chmod -R g+w html/languages
# chgrp -R apache upload
# chmod -R g+w upload
# chgrp -R apache log_boinclegion
# chmod -R g+w log_boinclegion
# chgrp -R apache html/cache
# chmod -R g+w html/cache/
# chgrp -R apache html/user_profile
# chmod -R g+w html/user_profile
# chmod -R o+rx html/inc
# chgrp apache download
# chmod g+rxs download
# chgrp apache bin/feeder
# chmod g+xs bin/feeder
# chmod 755 /home/boincadm
  
```

- Estando como usuario **root** se configura el servidor web de modo que dé acceso al proyecto. Para ello se hace uso del archivo **BoincPrueba.htpd.conf** que incluye una serie de directivas por defecto.

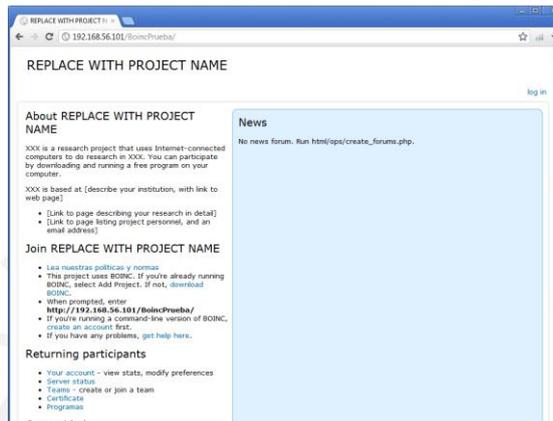
```

# cd /home/boincadm/projects/BoincPrueba
# cp BoincPrueba.htpd.conf /etc/httpd/conf.d/BoincPrueba.conf
  
```

## 7. Se reinicia el servidor web para que tomen efecto los cambios

```
# service httpd restart
```

Se ingresa a [http://ip\\_del\\_servidor/BoincPrueba/](http://ip_del_servidor/BoincPrueba/) que es la interfaz de usuario del proyecto.



Para la interfaz administrativa se ingresa a [http://ip\\_del\\_servidor/BoincPrueba\\_ops/](http://ip_del_servidor/BoincPrueba_ops/)

