



**IMPLEMENTACIÓN DE APLICACIONES Y  
HERRAMIENTAS DE POST-PROCESAMIENTO DE  
IMÁGENES MÉDICAS VÍA WEB**

Memoria del Proyecto de Fin de Carrera  
de Ingeniería Informática realizado por

**Jordi Casals Hernández**

y dirigido por

**Diego Javier Mostaccio Mancini**

Universitat Autònoma de Barcelona

Bellaterra, 25 de Junio de 2010



El sotasignat, Diego Javier Mostaccio Mancini, professor de la Escola Tècnica Superior d'Enginyeria de la UAB,

**CERTIFICA:**

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en Jordi Casals Hernández.

I per tal que consti firma la present.

Signat: Diego Javier Mostaccio Mancini  
Bellaterra, 22 de Juny de 2010

La sotasignada, Yolanda Vives Gilabert, de l'empresa PIC (*Port d'Informació Científica*)

**CERTIFICA:**

Que el treball a què correspon aquesta memòria ha estat realitzat en l'empresa sota la seva supervisió mitjançant un conveni entre PIC (*Port d'Informació Científica*) i DACSO (*Departament d'Arquitectura de Computadors i Sistemes Operatius*) de la Universitat Autònoma de Barcelona.

Així mateix, l'empresa en té coneixement i dóna el vist-i-plau al contingut que es detalla en aquesta memòria.

Yolanda Vives Gilabert

Bellaterra, 22 de Juny de 2010

*A la mare.*

# Índice general

<b>1. Introducción</b>	<b>7</b>
<b>2. Estado del arte</b>	<b>9</b>
2.1. Plataforma existente: PICNIC . . . . .	9
2.2. Propuesta de proyecto . . . . .	10
<b>3. Recursos y Planificación</b>	<b>11</b>
3.1. Recursos Hardware . . . . .	11
3.2. Recursos Software . . . . .	12
3.3. Planificación de tareas . . . . .	13
<b>4. Diseño de la Aplicación</b>	<b>16</b>
4.1. Arquitectura de la Plataforma . . . . .	16
4.2. Diseño de la Interfaz Web . . . . .	18
<b>5. Implementación de herramientas</b>	<b>20</b>
5.1. SPHARM . . . . .	21
5.1.1. Descripción del Software . . . . .	21
5.1.2. Proceso de Implementación . . . . .	22
5.1.3. Integración en PICNIC . . . . .	25
5.1.4. Problemas y Soluciones . . . . .	27
5.1.5. Análisis y Resultados . . . . .	28
5.2. FSL . . . . .	30
5.2.1. Descripción del Software . . . . .	30
5.2.2. Proceso de Implementación . . . . .	32

## ÍNDICE GENERAL

---

5.2.3.	Integración en PICNIC . . . . .	35
5.2.4.	Problemas y Soluciones . . . . .	38
5.2.5.	Análisis y Resultados . . . . .	38
5.3.	SPM con Máquinas Virtuales . . . . .	41
5.3.1.	Descripción del Software . . . . .	41
5.3.2.	Proceso de Implementación . . . . .	43
5.3.3.	Integración en PICNIC . . . . .	48
5.3.4.	Problemas y Soluciones . . . . .	51
5.3.5.	Análisis y Resultados . . . . .	52
5.4.	Otras Herramientas . . . . .	55
5.4.1.	ITK-Snap . . . . .	55
<b>6.</b>	<b>Trabajo Futuro</b>	<b>57</b>
<b>7.</b>	<b>Conclusiones</b>	<b>59</b>
<b>A.</b>	<b>Casos de uso</b>	<b>64</b>

# 1

## Introducción

El PIC [1] (Port d'Informació Científica), donde se ha desarrollado este proyecto, es un consorcio entre el DEiU (*Departament d'Educació i Universitats*), el IFAE (*Institut de Física d'Altes Energies*), el CIEMAT (*Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas*) y la UAB (*Universitat Autònoma de Barcelona*). Se trata de un centro innovador, situado en el campus de la UAB, que da soporte a grupos que necesitan recursos de computación y de almacenamiento de gran cantidad de datos mediante la tecnología Grid.

El PIC opera como centro español Tier-1 para procesamiento de datos del LHC (*Large Hadron Collider*) en el proyecto WLCG (*Worldwide LHC Computing Grid*). También trabaja en otros proyectos de ámbito científico como la investigación médica, grupo en el que se sitúa este proyecto.

Con la creciente generación de resonancias magnéticas, los servicios de radiología necesitan aplicaciones que les faciliten el trabajo de acceso remoto a los datos y a las herramientas que utilicen para la extracción de datos para realizar sus diagnósticos. El objetivo de este proyecto es el de estudiar e integrar en la plataforma web del grupo de Imagen Médica del PIC llamada PICNIC (*PIC NeuroImaging Center*) [2] un conjunto de aplicaciones para el estudio y procesamiento de neuroimagen con la implementación de herramientas software en la plataforma grid del PIC.

El factor más importante que se quiere aprovechar para la realización de este

## *1. INTRODUCCIÓN*

---

proyecto es la utilización de recursos de computación de alto rendimiento de los que se dispone en el PIC. Estos recursos son, a parte de la gran cantidad de espacio de almacenamiento del que se dispone, el cluster de máquinas que se usará para mejorar la velocidad de procesamiento de los datos necesarios.

A todo esto se le añade que se quiere crear un sistema centralizado y de fácil acceso y utilización para aquellos usuarios que no tengan un conocimiento informático tan avanzado (investigadores clínicos), por lo que se decidió crear una plataforma única de acceso a las utilidades y herramientas que se ponen a disposición de los usuarios finales.



## 2

# Estado del arte

El proyecto se desarrolla dentro del ámbito de la Biomedicina y más concretamente dentro del campo de estudio de MRI (*Magnetic Resonance Imaging*). A continuación se explicará el punto de partida y la propuesta del proyecto que se quiere realizar.

### 2.1. Plataforma existente: PICNIC

Al iniciar este proyecto, se contaba con una primera versión de la plataforma que se usará como base. Esta plataforma, llamada PICNIC [2] (*PIC Neuro Imaging Center*) ya ofrecía una serie de herramientas para el estudio y procesamiento de neuroimágenes.

PICNIC permite a los usuarios la utilización de estas herramientas de una manera más sencilla y a distancia en un entorno más conocido por todos como es la web. La aplicación elegida fue un portal web, ya que así, además, se puede integrar una base de datos con la cual interactuar y guardar información que no es necesaria para el procesamiento de los datos, pero sí para el estudio estadístico.

El punto de partida de este proyecto respecto a la plataforma ya existente disponía de varios software de tratamiento de imágenes ya funcionales integrando sus scripts en el portal aportando una interfaz gráfica a aquellos que se ejecutan a través de la consola de Linux, y otros que se pueden usar mediante la integración de aplicaciones remotas.

## 2.2. Propuesta de proyecto

Con esto se ha situado el proyecto en el contexto en el que se encuentra. Ahora se puede hacer una valoración sobre los puntos que se tendrán que tener en cuenta para realizarlo y qué es lo que se quiere conseguir de él.

En primer lugar se deberá investigar sobre distintos paquetes software de estudio y tratamiento de imagen médica, concretamente dentro del marco de la neuroradiología. De esta manera se podrá decidir cuales son las aplicaciones que aportan más a la plataforma o cuales pueden ser más útiles para los usuarios. Esta decisión se tomará conjuntamente con ellos. Los principales paquetes susceptibles a ser integrados en PICNIC son SPHARM-PDM [3], FSL [4], y SPM [5], entre otros.

A continuación se realizará un estudio de paquetes y de la mejor manera de integrarlo a la plataforma web. Una vez se haya entendido el funcionamiento de las herramientas, se optimizará su rendimiento aprovechando el clúster del que se dispone en el PIC, de manera que su funcionamiento pueda ser rápido, además de fiable, y que tenga una buena usabilidad pensando en un usuario que no debería por qué tener conocimientos extensos sobre informática.

También se hará una comparación entre los tiempos de ejecución iniciales y finales, para poder ver como realmente se han aprovechado los recursos de los que se dispone en el PIC, y para ver que el trabajo hecho ha dado de sí todo lo que se esperaba.

# 3

## Recursos y Planificación

Este apartado pretende listar los recursos de los que se dispone, tanto hardware como software, para la realización del proyecto.

### 3.1. Recursos Hardware

Para llevar a cabo todas las tareas de desarrollo del proyecto, se necesita tener una infraestructura con la cual trabajar, y sobre la cual explotar las posibilidades que ofrece el PIC.

Para uso exclusivo dentro del grupo de Imagen Médica del PIC y desarrollo del trabajo se dispone de los siguientes componentes hardware:

1. Un PC Dell PowerEdge 400SC, para todas las tareas de desarrollo de software dentro del departamento de neuroimagen del PIC.
2. Dos Dell Workstation con sistema operativo Linux CentOS 5.4 para post-procesamiento de imágenes médicas en local.
3. Una cabina FAS 2020 con 700GB reservados para el grupo de imagen médica para almacenamiento de datos.
4. Un servidor (*neuroweb01*) Intel Xeon CPU 5160@3.00GHz como centro de operaciones de todo el sistema creado con la plataforma web y el lanzamiento de Jobs al cluster del PIC.

### 3. RECURSOS Y PLANIFICACIÓN

---

5. Un portátil Apple MacBook Pro con sistema operativo Mac OS X 10.6 para el desarrollo de software fuera del departamento y para realización de documentación.

Para uso compartido con el resto de departamentos del PIC se dispone de los siguientes recursos:

1. **160 Worker Nodes** HP BL460 / BL280 de los cuales se destina un 1% de su dedicación total a los Jobs que se envíen desde el grupo de imagen médica. Estos nodos de trabajo tienen una arquitectura x86\_64 con 2 procesadores de 4 núcleos y 16GB de RAM cada uno.
2. Una **capacidad de disco** de 3 PetaBytes, disponible en el PIC para datos y distribuido en un conjunto de Thumpers SunFire X4500 y en dos bloques de discos Data Direct Network S2A 9900, cada uno con 600 discos de 2TB cada uno.

## 3.2. Recursos Software

Dentro de la explicación de los recursos software de los que se dispone, se puede diferenciar dos casos: los recursos que son necesarios y de los que se dispone para realizar el trabajo de implementación de aplicaciones en PICNIC, y las aplicaciones que se querrán implementar en PICNIC.

El listado de software necesario para la realización de las tareas de implementación contiene todo aquello que se ha usado para la programación de scripts y el software necesario para el funcionamiento correcto de la plataforma web consta de los siguientes componentes:

1. **Sistema operativo** Linux Ubuntu 9.10 para la **máquina personal** sobre la que se han desarrollado los scripts usando la propia consola de Linux y el editor de textos Vim.

2. **Sistemas operativos** Linux CentOS 5.4 para las **Workstation Dell XPS** y para el servidor que contiene la plataforma web.
3. Un **servidor** Apache dentro del que se alojan todos los ficheros de la plataforma web.
4. Una **base de datos** realizada con MySQL [6], que facilita el trabajo entre el PHP [7] y el servidor Apache [8].
5. Un sistema de **gestión de colas** TORQUE / MAUI, una versión libre de PBS [9] (Portable Batch System).

Mientras que a la ya existente lista de aplicaciones implementadas en PICNIC para el tratamiento de neuroimagen (FreeSurfer, LCMoel y SPM sobre NX) hay que añadir las tres siguientes:

1. SPHARM-PDM.
2. FSL (*FMRIB Software Library*).
3. SPM8 (*Statistical Parametric Mapping*) con máquinas virtuales.
4. ITK-Snap [10].

### 3.3. Planificación de tareas

Una vez realizado el estudio sobre los recursos software y hardware de los que se dispone, y de las aplicaciones software que se quieren añadir a PICNIC, se ha hecho un estudio del tiempo del que se dispone y de la estimación de tiempo que se le deberá dedicar a cada una de las partes.

El diagrama de la Figura 3.1 se puede ver la planificación que inicialmente se decidió llevar a cabo junto a las reuniones con el tutor del proyecto cuyas fechas están establecidas desde el inicio.

### 3. RECURSOS Y PLANIFICACIÓN

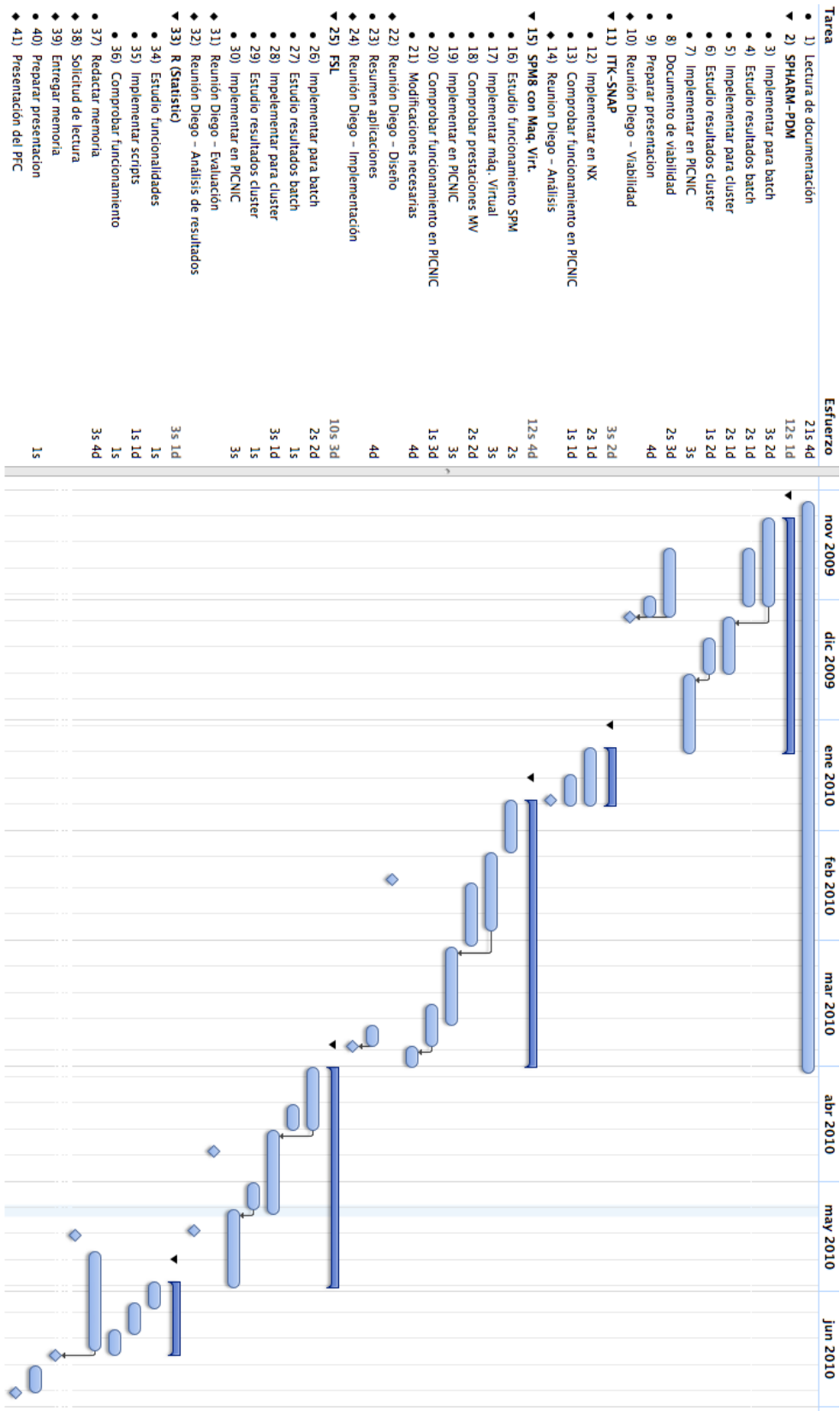


Figura 3.1: Planificación inicial

### 3. RECURSOS Y PLANIFICACIÓN

---

Finalmente la planificación inicial no se ha podido cumplir viéndose afectado el desarrollo e implementación del paquete estadístico R [11]. La causa ha sido que se le ha dedicado más tiempo del planeado inicialmente a los otros paquetes, pero no por problemas e imprevistos, con los cuales ya se contaba en la planificación, sino porque se ha considerado oportuno incluir ciertas aplicaciones que contienen estos que pueden ser útiles para los estudios que actualmente se realizan en el grupo de imagen médica.

# 4

## Diseño de la Aplicación

Esta parte de la memoria pretende hacer una explicación detallada de la estructura de la aplicación completa, explicando sus componentes principales y los métodos usados para su implementación. También se presentará una valoración de las herramientas utilizadas y del diseño de la plataforma web.

### 4.1. Arquitectura de la Plataforma

Para comenzar con la explicación de la plataforma, se explicará la arquitectura y estructura que tiene el sistema completo, hablando de las partes más importantes de las que está formado. Para el post-procesamiento de imágenes a través de la plataforma web se realiza el acceso a través del servidor **Neuroweb** (<https://neuroweb01.pic.es>), quien además de proporcionar una interfaz gráfica, coordina las acciones que se realizan con los tres grandes bloques del sistema, como se puede ver en la Figura 4.1:

1. **Post-procesamiento:** Se puede considerar el elemento más importante del conjunto. Está formado por el conjunto de herramientas que se han implementado en el cluster del PIC y posteriormente integrado en PICNIC. Se accede a través del servidor de la plataforma a través de una interfaz web. Cada uno de los servicios tiene unas características distintas en función de las necesidades de cada herramienta. Se ha integrado el sistema de visualización remota de aplicaciones NX [12], y un sistema de ejecución de scripts, que puede ser ejecutado localmente en el servidor, bien lanzado al clúster para obtener un

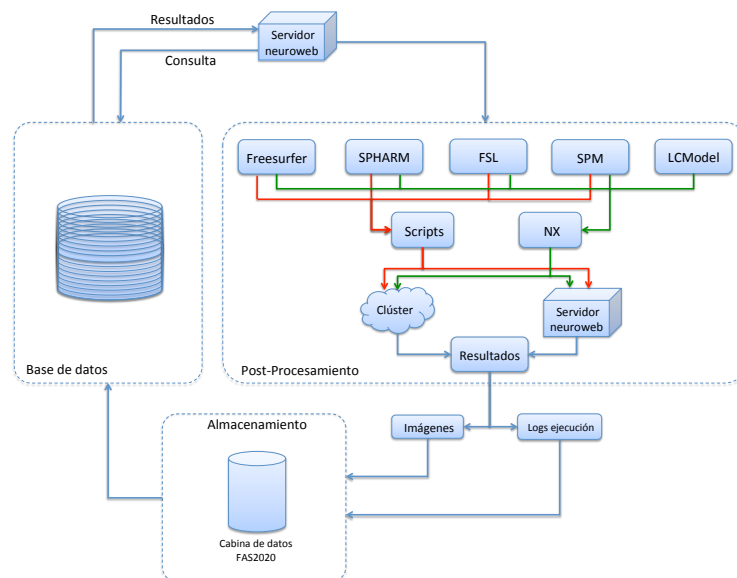


#### 4. DISEÑO DE LA APLICACIÓN

---

rendimiento mayor en el caso de procedimientos que se pueden ejecutar en paralelo o bien para no sobrecargar el servidor con ejecuciones que limiten sus recursos.

- Base de datos:** Se encuentra integrada en el servidor junto a la interfaz web. Se tiene una serie de metadatos almacenados correspondientes a la información personal de cada paciente. Pueden ser útiles para estudios específicos, por ejemplo en estudios de pacientes comprendidos dentro de un rango de edades, o de un mismo sexo. Por otro lado, es muy importante ya que la interfaz web se apoya en la base de datos para mostrar algunos de sus datos, que incluso se usan como parámetros de las ejecuciones. Podrá ser consultada o modificada por usuarios con permisos.
- Almacenamiento:** El sistema de datos es muy importante, ya que es donde se guardan tanto los datos originales como los generados en la zona de post-procesamiento. Está conectada al servidor con el protocolo NFS (*Network File System*) [13] y es inaccesible desde la plataforma web, de manera que tan sólo el administrador puede acceder a ella y así evitar problemas en la integridad de los datos.



**Figura 4.1:** Arquitectura de la plataforma

## 4. DISEÑO DE LA APLICACIÓN

En la Figura 4.1 se puede ver como existe una conexión entre los tres bloques. El servidor sirve para mostrar la interfaz gráfica y para realizar la ejecución de los scripts que lanzan Jobs a la granja en la parte de post-procesamiento. Además de eso, también está conectado al centro de almacenamiento para guardar los datos creados en las ejecuciones y los logs de salida y de error.

### 4.2. Diseño de la Interfaz Web

El conjunto de herramientas a desarrollar se integrarán en una interfaz web *user-friendly* para facilitar el uso de esas herramientas y conseguir una centralización de las mismas en un punto común. La web está dividida en dos partes, la zona pública y la zona privada.

La zona pública sirve de escaparate y descripción del portal para posibles usuarios, pero no da acceso a ninguna herramienta ni zona de uso restringido. La zona privada consta de un menú lateral con las opciones correspondientes a los permisos otorgados por el administrador.

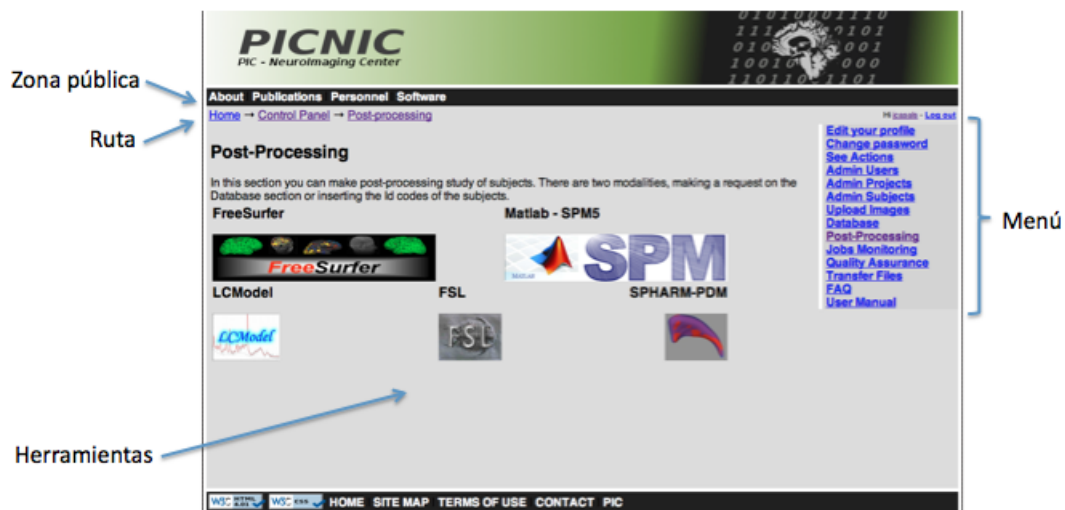


Figura 4.2: Esquema interfaz post-procesamiento

La parte de post-procesamiento, sobre la que se ha centrado el proyecto, dispone de una página principal con enlaces a las zonas de cada aplicación implementada. Como se puede ver en la Figura 4.2, en la parte superior del marco aparece la ruta de

#### 4. DISEÑO DE LA APLICACIÓN

enlaces seguida por el usuario, para que sepa en todo momento donde se encuentra, y a la derecha el menú completo de la plataforma, para poder acceder a las opciones disponibles desde cualquier punto.

Cada aplicación tiene su implementación en PICNIC con el fondo de un color distinto para que el usuario identifique fácilmente el lugar en el que está. También se ha separado todo el proceso completo por pestañas, cada una representando un paso de todo el conjunto.

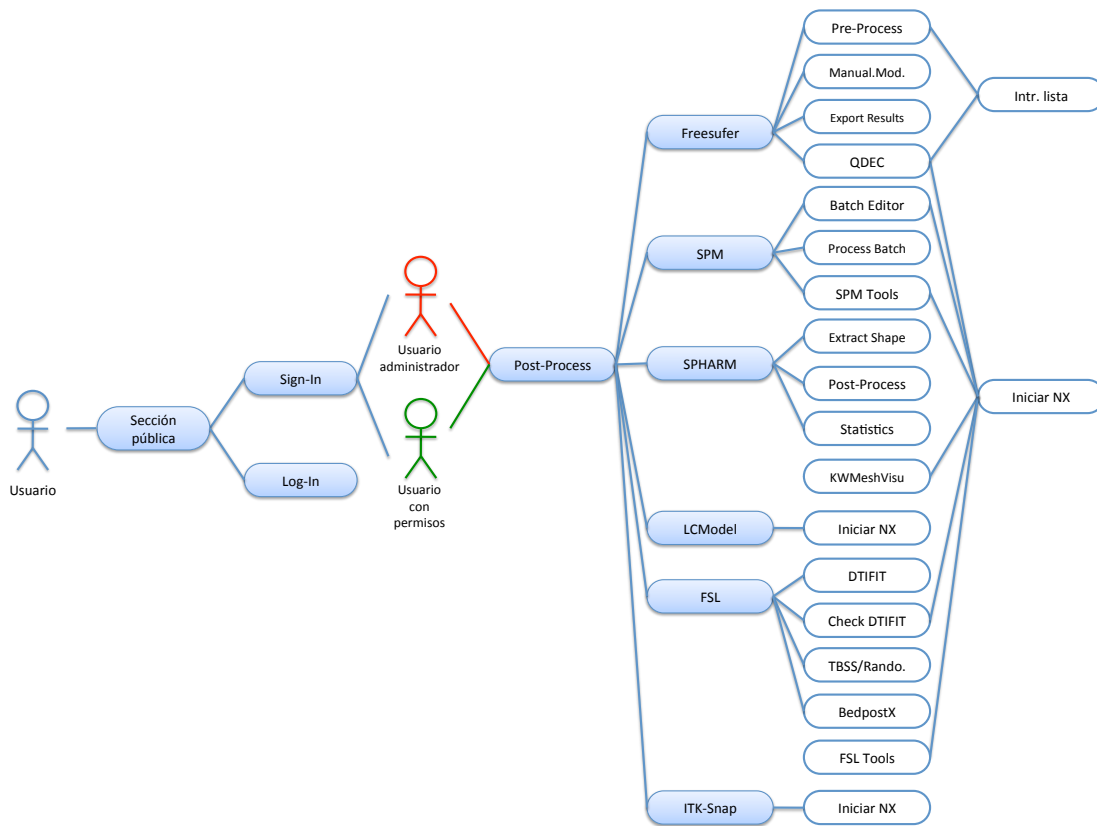


Figura 4.3: Diagrama de casos de uso

Para decidir la estructura en la que se ha separado toda la aplicación, antes se han estudiado las diferentes posibilidades y se han diseñado todos los casos de uso (Anexo A) para realizar todos los pasos de cada una de las aplicaciones. En el diagrama de la Figura 4.3 están representados todos los casos de uso que se han desarrollado o que afectan a la aplicación, dejando sin representar aquellos que no se han modificado o que no afectan al desarrollo de todo lo creado para este proyecto.

# 5

## Implementación de herramientas

Este apartado se centrará en explicar cada uno de los software que se han usado para la realización de este proyecto. Constará de cuatro partes. Las tres primeras se centrarán en los tres software que más trabajo requerían para el proyecto (SPHARM, FSL y SPM con Máquinas Virtuales) y una cuarta que explicará varios trabajos que se han hecho sin tanta importancia como los tres anteriores.

Para la descripción de cada software se va a seguir una estructura similar para facilitar la comprensión del proceso. Cada software va a disponer de cuatro partes:

1. Descripción de las funcionalidades del software, de tal manera que se pueda entender mejor el por qué de la aplicación de este software a PICNIC.
2. Objetivos principales y proceso seguido en la implementación de los componentes necesarios para el desarrollo y buen funcionamiento del paquete.
3. Descripción modular del apartado del software dentro del portal web de PICNIC, con una descripción de cada uno de los módulos.
4. Valoración de las dificultades que se han tenido en el proceso de implementación y las soluciones que se han adoptado para solucionarlos.
5. Resultados obtenidos para evaluar las mejoras alcanzadas con respecto al software original.

## 5.1. SPHARM

### 5.1.1. Descripción del Software

SPHARM-PDM [14] es uno de los paquetes software que se van a implementar en PICNIC. Para el análisis de estructuras morfológicas cerebrales a menudo se utilizan mediciones volumétricas. Estas mediciones sirven para demostrar que se produce una atrofia o dilatación causada por la enfermedad del paciente. Estos cambios de volumen no siempre reflejan la atrofia, y es por eso por lo que el análisis de la forma genera un creciente interés para la comunidad de neuroimagen, ya que permite detectar con precisión los cambios de forma entre las estructuras sanas y las dañadas.

Lo que permite este software es realizar una parametrización de la forma de una estructura subcortical para así poder hacer un estudio estadístico y comparar el grado de deformación de dicha estructura en grupos de pacientes con diferentes grados de evolución de una enfermedad determinada.

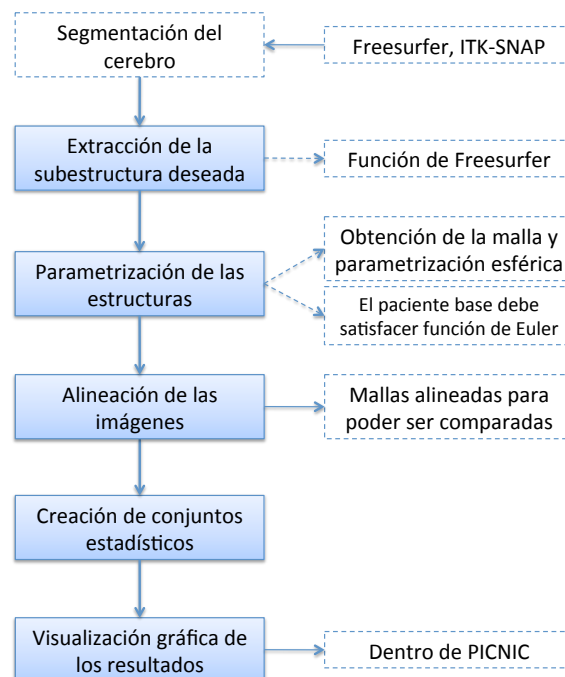


Figura 5.1: Esquema SPHARM

El proceso de tratamiento de imágenes a través de SPHARM-PDM y el paquete ShapeTools [15], como se puede ver en la Figura 5.1 necesita primero una segmentación completa del cerebro de cada paciente, proceso que se realizará previamente con Freesurfer [16] (proceso automático) o con ITK-SNAP (proceso semi-automático). Con esta imagen de entrada, se hará un muestreo de la estructura subcortical elegida y se comprobará que se dispone de la topología esférica necesaria para los siguientes pasos.

Después de este paso, se pasa a obtener la malla y la parametrización de la estructura de un solo paciente para tener una plantilla que facilite la normalización geométrica de las imágenes, para así poder hacer la posterior comparación. Una vez se tengan las imágenes alineadas, se hace el estudio estadístico que permitirá hacer diversos tipos de comparaciones.

Por último, será interesante contar con un visualizador gráfico que permita ver los resultados, reflejados en una imagen tridimensional, de la estructura que se haya analizado. Este visualizador es KWMeshVisu [17] y está incluido en el paquete de SPHARM-PDM. Este software genera mapas estadísticos con las zonas de la estructura subcortical en donde hay un cambio significativo de su forma respecto al conjunto. Estas diferencias son las que se podrán apreciar en los mapas estadísticos visualizados con KWMeshVisu.

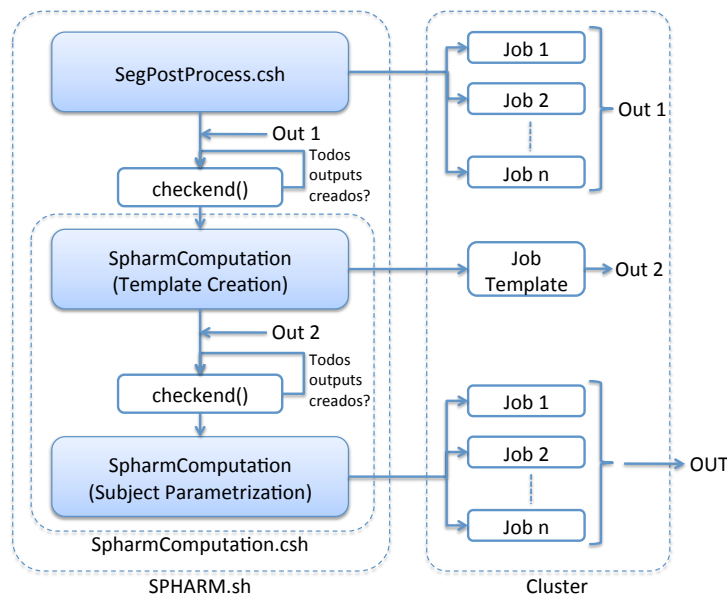
### 5.1.2. Proceso de Implementación

El objetivo principal que justifica la implementación de este software en PICNIC es que la aplicación funcione de la mejor manera posible. Para ello se ha conseguido alternar ejecuciones en local con las estaciones de trabajo de altas prestaciones disponibles en el PIC, así como la ejecución en paralelo, para obtener un procesamiento más rápido en aquellos procesos que lo requieran. Además, también se quiere conseguir que la aplicación permita hacer ejecuciones desde cualquier máquina que no esté establecida en el PIC para que, por ejemplo, se pueda trabajar con ella desde el hospital.

## 5. IMPLEMENTACIÓN DE HERRAMIENTAS

Para la parte de procesamiento, se han desarrollado una serie de scripts para aprovechar las funcionalidades incluidas en el paquete SPHARM-PDM. Para ello se han usado los lenguajes de programación Bash (*Bourne Again Shell*) [18] y C-Shell [19]. Se han elegido estos lenguajes para aprovechar las características presentadas anteriormente. Estos scripts son los encargados de realizar todo el proceso descrito en este apartado, y se ejecutarán por detrás del portal web.

Tras haber estudiado el software y haber visto las características principales de los procesos que ejecuta, se ha creído conveniente que para optimizar el tiempo de ejecución, esta se haga en paralelo en función de los pacientes.



**Figura 5.2:** Esquema SPHARM.sh

Estos scripts se usarán para controlar el flujo de ejecución del proceso. El primero de ellos es `extract_shape.sh`, que realiza la extracción de la máscara de la estructura subcortical deseada de los pacientes especificados. Después se ejecutará el script `SPHARM.sh`, cuyo esquema se puede ver en la Figura 5.2, que contiene la ejecución de otros dos scripts más: `SegPostProcess.csh` y `SpharmComputation.csh`. El script que controla el flujo ejecuta primero `SegPostProcess.csh` y mediante una comprobación iterativa, no deja que se ejecute `SpharmComputation.csh` hasta que no se hayan generado todos los outputs del primero. Y por último, el fichero

## 5. IMPLEMENTACIÓN DE HERRAMIENTAS

---

`StatsNonParamTest.csh`, que se encarga de ejecutar todo el proceso estadístico, comparando todos los sujetos seleccionados de cada estudio, en todas las posibles combinaciones de dos elementos con los grupos existentes. Este script permite realizar la estadística con o sin el escalado de las imágenes calculado en el script `icv.sh`, también desarrollado para este proyecto, que permite normalizar las imágenes al volumen total de cerebro de cada sujeto en función de una fórmula establecida.

Además de la creación de scripts y de la interfaz web, se ha añadido la opción de visualizar gráficamente los resultados para poder hacer una comparación en función de los resultados que se quieren observar. Para la visualización gráfica se usará la herramienta `KWMeshVisu`, introducida anteriormente.

El desarrollo e implementación de este software en `PICNIC` se ha realizado en dos fases. La primera corresponde a la creación de los scripts para un funcionamiento correcto en local, y la segunda corresponde a la modificación de los scripts anteriormente implementados para obtener su funcionamiento desde `PICNIC` y utilizando el sistema de colas del `PIC` para conseguir un funcionamiento correcto con ejecución en paralelo. El proceso que se ha seguido es el siguiente:

### **Fase 1: Instalación y ejecución secuencial**

- Estudio de las funcionalidades del paquete con la ayuda de los neuropsicólogos y otros usuarios.
- Programación de scripts en bash, aprovechando las funcionalidades aportadas por el paquete, para obtener ejecución del software en batch.
- Estudio de las prestaciones del software en batch.
- Instalación de librerías y otros componentes necesarios en *NFS* de manera que el software sea accesible y funcional para todos los nodos de la granja, evitando así la instalación del software en cada nodo.
- Test funcional con un solo sujeto.



### **Fase 2: Ejecución en paralelo**

- Desarrollo de nuevos scripts en bash que permitan la ejecución del software SPHARM-PDM en paralelo (1 job = 1 proceso / sujeto).
- Test en local y pruebas con su adaptación a PICNIC.
- Validación de su ejecución en paralelo usando el clúster del PIC.
- Comparación de las mejoras aportadas por la ejecución en paralelo respecto a los resultados obtenidos en secuencial.
- Estudio de los parámetros de entrada y de su efecto en los resultados finales.
- Optimización de la ejecución del software.

Una vez realizados los pasos anteriores ya se dispone de una interfaz web con la que poder trabajar y que está totalmente operativa.

### **5.1.3. Integración en PICNIC**

Para integrar SPHARM-PDM en PICNIC, se ha utilizado PHP para la creación de la interfaz gráfica, y se ha seguido con la estructura por pestañas utilizada en la implementación de otros software ya incluidos en PICNIC tal y como se ha explicado anteriormente.

Este apartado en concreto, consta de cuatro pestañas con las diferentes partes del proceso:

1. **Extracción de la estructura subcortical** deseada, de la cual se procederá a extraer la topología esférica de la estructura elegida de los pacientes que se hayan seleccionado. Para poder extraer una determinada estructura subcortical del cerebro de un paciente esta se tendrá que haber segmentado previamente utilizando otro software (*Freesurfer*). La interfaz realiza una consulta a la base de datos para obtener la lista de proyectos en los que el usuario tiene permiso para trabajar, y con una consulta a un fichero se obtiene la lista con las distintas estructuras subcorticales, tal y como muestra la Figura 5.3.

## 5. IMPLEMENTACIÓN DE HERRAMIENTAS

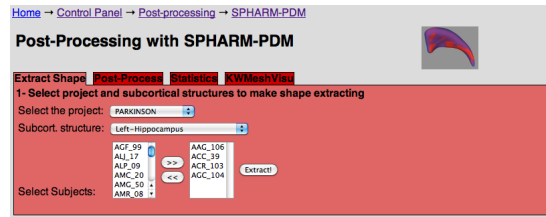


Figura 5.3: Extract Shape

2. **Parametrización de estructuras**, para posteriormente alinear las mallas en función de las posiciones globales calculadas y de los parámetros que se especifiquen en los formularios que contiene. Inicialmente se daba la posibilidad de realizar el análisis solamente de una estructura subcortical dentro de un mismo grupo, pero finalmente, gracias a la interacción con los usuarios, se ha conseguido adaptar el sistema para poder analizar comparativamente dos estructuras subcorticales dentro de un mismo grupo, como se puede ver en la Figura 5.4.

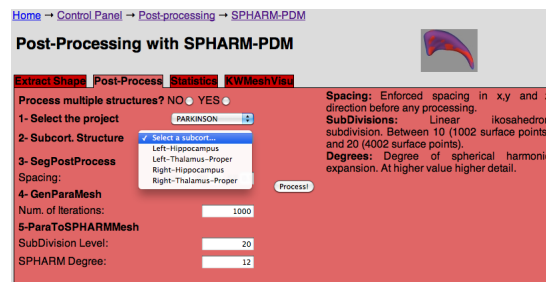


Figura 5.4: Post-Procesamiento con SPHARM

3. La Figura 5.5 corresponde a la **creación de conjuntos estadísticos** en función de los pacientes analizados anteriormente, y que una vez terminado, será posible analizar con el visualizador que ofrece SPHARM-PDM. También se da la posibilidad de usar el factor de escalado o no.

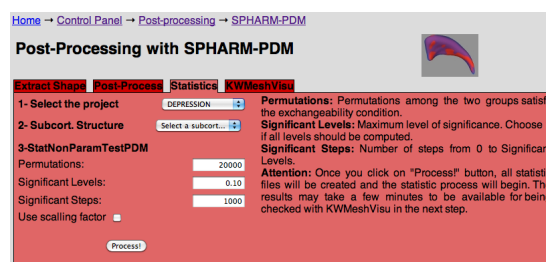


Figura 5.5: Creación estudio estadístico

## 5. IMPLEMENTACIÓN DE HERRAMIENTAS

4. El visor gráfico **KWMeshVisu**, mostrado en la Figura 5.6, permitirá ver los resultados obtenidos. En este apartado se aprovecha la herramienta NX, que ofrece la posibilidad de abrir aplicaciones remotas como si estuvieran en la máquina del cliente.

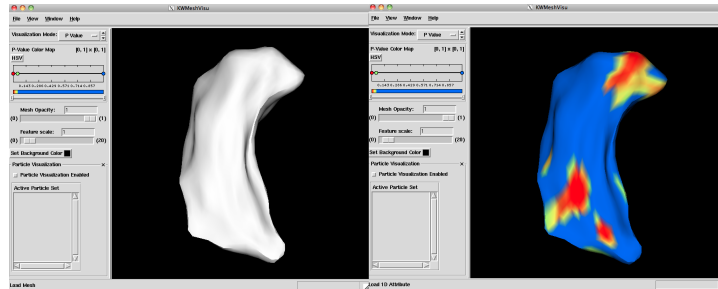


Figura 5.6: Visualización con KWMeshVisu

Además de esta estructura, se ha implementado un sistema de notificación al usuario por e-mail cuando terminan los procesos que el mismo ha lanzado. También se dispone de un sistema de monitorización de Jobs lanzados a través de la web, que permite ver el estado en el que se encuentran (R: *Running* o Q: *Queued*) e incluso pararlos y eliminarlos de la cola de ejecución.

### 5.1.4. Problemas y Soluciones

Una de las dificultades encontradas durante la implementación de SPHARM-PDM en PICNIC ha sido el aprendizaje de toda la arquitectura de PICNIC y de los sistemas que utiliza, como el sistema de datos y el gestor de colas PBS. Además, también han ido apareciendo problemas de integración entre lenguajes, que afectaban al proyecto temporalmente.

La mejor manera de solucionar los problemas ha sido ir desarrollando la parte final del software a la vez que se realizaban las pruebas de funcionamiento. De esta manera es más fácil ver cuando los resultados obtenidos no son los esperados o si falla la ejecución, poder solucionarlo para la versión final. Esta metodología también ha permitido ver qué puntos eran más o menos importantes para decidir que diseño debía tener la interfaz web.

Un factor muy importante que ha influido muy positivamente para decidir que estructura debía tener la interfaz web ha sido tener la posibilidad de trabajar conjuntamente con los usuarios de la aplicación, de manera que se ha podido mejorar y adaptar según su criterio.

### 5.1.5. Análisis y Resultados

Como se podrá ver reflejado en los resultados, y cumpliendo uno de los objetivos principales de todo el proyecto, se han aprovechado las prestaciones del clúster y las estaciones de trabajo de alto rendimiento disponibles lo máximo que ha permitido el software.

En este apartado se presenta un análisis de los tiempos de ejecución de SPHARM-PDM aplicado a un proyecto de la enfermedad de Parkinson [20] con demencia. El 83 % de los pacientes enfermos de Parkinson desarrollan demencia en el transcurso de la enfermedad. Los enfermos de Parkinson con demencia (*EPCD*) tienen cambios estructurales que se pueden apreciar en la imagen de la resonancia magnética, como la disminución del grosor cortical o pérdida de volumen de ciertas estructuras subcorticales. Los hipocampos son una de estas estructuras que pueden verse afectadas (dado que son estructuras relacionadas con la memoria y por tanto que se ven afectadas en pacientes con demencia), y por eso se va a realizar la prueba de funcionamiento con esta estructura, más concretamente, para el hipocampo izquierdo.

Esta prueba se ha llevado a cabo sobre un grupo de 94 pacientes provenientes del Hospital de la Santa Creu i Sant Pau [21], divididos en cuatro grupos: Control (personas que no tienen ningún rastro de la enfermedad), EPSD (*Enfermos de Parkinson Sin Demencia*), EPDCL (*Enfermos de Parkinson con Deterioro Cognitivo Leve*) y EPCD (*Enfermos de Parkinson Con Demencia*).

En la Tabla 5.1 se presenta una comparativa entre los tiempos de ejecución que se tenían antes de la realización del proyecto (*ejecución secuencial*) y los obtenidos una vez se han hecho toda una serie de modificaciones para optimizar el rendimiento

## 5. IMPLEMENTACIÓN DE HERRAMIENTAS

---

del software (*ejecución en paralelo*), sin tener en cuenta el tiempo de espera en la cola.

Las pruebas de ejecución secuencial se han realizado en una máquina con las mismas características hardware y software que los Worker Nodes con el fin de obtener unos tiempos de ejecución comparables.

—		Secuencial	Paralelo	Speed-Up
Extract Shape		11 min.	11 min.	1
Spharm	SegPostProcess	10 h. 58 min.	7 min.	94 ( <i>#pacientes</i> )
	Create Template	57 min.	57 min.	1
	Parametrization	2 d. 6 h. 45 min.	45 min.	65
Estadística		2 h. 15 min.	2 h. 15 min.	1
Total		2 d. 21 h. 6 min.	4 h. 15 min.	16.25

**Cuadro 5.1:** Resultados SPHARM-PDM para 94 pacientes

En el proceso **Extract Shape** no ha habido variación de tiempo, ya que como el tiempo de extracción de la estructura de cada paciente era de pocos segundos, se decidió hacer este proceso en local, para no perder prioridad en la cola de ejecución por culpa de un proceso tan corto.

En el Post-Procesamiento sí se puede apreciar la diferencia de tiempo. Esta parte consta de tres pasos. En **SegPostProcess** se necesitan siete minutos por paciente, lo que eleva enormemente el tiempo de la ejecución secuencial respecto al tiempo en paralelo. La primera parte de **Spharm Computation** consiste en la creación de una plantilla con una máscara de la estructura y utiliza todos los pacientes a la vez, así que no hay posibilidad de paralelización y el tiempo de ejecución será el mismo para ambos casos. Sin embargo, este proceso también se ejecuta en un nodo de la granja del PIC para no absorber demasiados recursos del servidor. Ya en el último paso de esta segunda parte, en la parametrización dentro de **Spharm Computation**, también se aprovecha la paralelización por sujetos. Como se puede ver en la tabla de resultados, lo que en secuencial significaría más de 2 días de ejecución, en paralelo sería a penas una hora. Hay que tener en cuenta que en el caso de la ejecución secuencial, en este paso, los sujetos que no cumplen las condiciones necesarias no se analizan. Así pues, se han desechado varios pacientes, y al ejecutar una parte de

los 94 disponibles, el tiempo en secuencial es un poco menos de lo que podría haber sido.

En la parte final, se lleva a cabo el proceso estadístico. No se puede mejorar el tiempo de ejecución, ya que se necesita trabajar con todo el conjunto. De manera que será el mismo para los dos métodos de ejecución. Aún así, se enviará a la granja como un solo Job para no sobrecargar el servidor.

Una vez terminado el proceso, se pueden apreciar las mejoras de rendimiento de la herramienta en la columna *Speed-Up* de la tabla de resultados. En `SegPostProcess`, la mejora es equivalente al número de pacientes y en `Parametrization` también, pero en este caso hay que restar los pacientes descartados. Por lo tanto, cuanto mayor sea el número de pacientes, mayor será la mejora de tiempo respecto a la ejecución secuencial, hasta llegar al punto en que el número de pacientes sea mayor que el número de nodos disponibles, ya que se perderá tiempo en la cola de ejecución. En este caso se ha conseguido un tiempo de ejecución 16 veces más rápido que el secuencial.

## 5.2. FSL

### 5.2.1. Descripción del Software

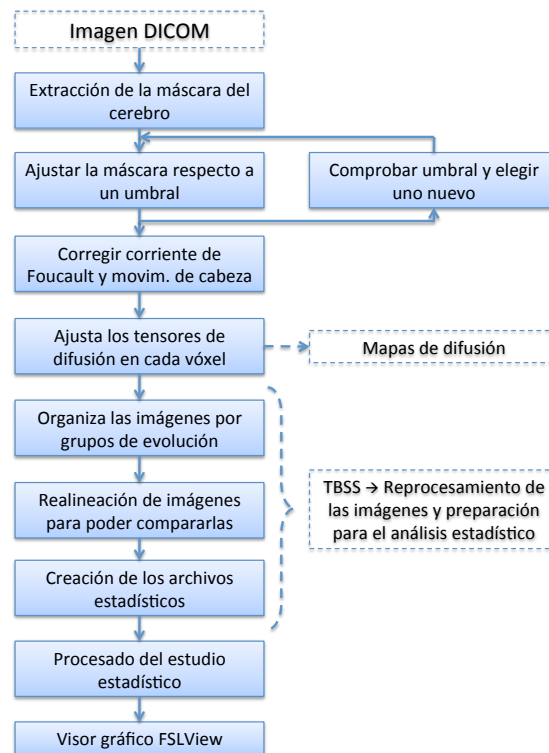
El siguiente software que se ha querido que forme parte de la plataforma web de PICNIC es FSL. Su implementación va a seguir una estructura similar a la seguida en SPHARM. Aunque FSL tiene muchas utilidades, en este caso se va a usar para DTI (*Diffusion Tensor Imaging*) para el estudio de la materia blanca en el cerebro.

La secuencia de pasos a seguir para hacer un post-procesamiento de imágenes con FSL representada en la Figura 5.7 consistirá, primeramente, en recibir las imágenes en formato DICOM [22] y convertirlas a NIFTI [23] (`dcm2nii`), formato con el que trabaja FSL. Una vez convertido, con los ficheros creados se procederá a extraer la máscara de la imagen (`fslroi`) y se ajustará a partir de un cierto umbral predeterminado (`bet`). Para que la imagen quede lista para su procesamiento hay

## 5. IMPLEMENTACIÓN DE HERRAMIENTAS

que aplicarle una corrección para compensar las corrientes de Foucault inducidas en las bobinas así como los posibles movimientos de cabeza (`eddy_correct`). El resultado se deberá comprobar visualmente para que, en caso de no haber obtenido la máscara deseada, se pueda reprocesar con un nuevo umbral (`bet2`).

Una vez ajustadas las imágenes, se crearán una serie de mapas de difusión (`dtifit`) con los cuales se podrá preparar el análisis estadístico con el paquete TBSS (*Tract-Based Spatial Statistics*) [24]. El proceso estadístico con TBSS mide la difusión anisotrópica en los tractos de sustancia blanca para estudiar la conectividad anatómica del cerebro.



**Figura 5.7:** Esquema FSL

Para hacer el estudio estadístico se debe hacer una reagrupación de las imágenes para que el software acceda a ellas por grupos. Después se hará una realineación y reajuste de las imágenes para que tengan todas el mismo tamaño y poder compararlas entre sí. En el siguiente paso se proyectarán las imágenes de cada sujeto sobre los esqueletos globales obtenidos y se hará una comparación de las diferencias de cada sujeto con la media.

A todo el estudio con TBSS, se le añade la posibilidad de realizar un estudio con BEDPOSTX (*Bayesian Estimation of Diffusion Parameters Obtained using Sampling Techniques*) [25], otra de las funcionalidades de FSL. Bedpostx utiliza la cadena de muestreo de Markov, Monte Carlo, para crear distribuciones con los parámetros de difusión de cada vóxel y crea los ficheros necesarios para realizar un estudio de tractografía probabilística.

### 5.2.2. Proceso de Implementación

Como el resto de paquetes software que se han implementado en PICNIC, lo que se busca al introducirlo en la plataforma es que los usuarios puedan trabajar fácilmente con él y con una interfaz más *user-friendly* para ellos que la terminal de Linux, sobre la cual se ejecutan los scripts creados.

Otro de los objetivos principales es la explotación de los servicios de clustering que ofrece la infraestructura del PIC. Así, aprovechando la posibilidad de ejecución en paralelo que ya se ha usado en SPHARM, se pretende reducir de manera considerable el tiempo de ejecución de todo el proceso, y por tanto poder realizar más estudios en menos tiempo.

Para el desarrollo de este apartado se ha decidido seguir con el mismo sistema que el anterior. Para la creación de scripts se seguirá usando *bash* ya que se ha ganado mucha experiencia en la creación de este tipo de scripts con este lenguaje. Otra razón es que se ha debido modificar uno de los scripts incluidos en el paquete de FSL, hecho en *bash*.

Se han tenido que desarrollar una serie de scripts, encargados de paralelizar el proceso descrito anteriormente para la ejecución de este software. Por una parte se tienen los correspondientes al procesado con DTIFIT y el proceso estadístico TBSS con Randomise [26], y por otra parte se tiene el script correspondiente a Bedpostx.

Para el primer grupo de scripts se creó `preproc.sh`. Este script contiene las acciones necesarias para el procesamiento de las imágenes de entrada con DTIFIT.



## 5. IMPLEMENTACIÓN DE HERRAMIENTAS

---

Está dividido en tres partes similares pero con objetivos distintos:

1. Permite al usuario procesar las imágenes de todos los pacientes disponibles para cada proyecto de golpe.
2. Procesar tan sólo los escogidos por el usuario, lo cual será lo más común.
3. Reprocesar la imagen con otro umbral diferente tras comprobar si el usado por defecto dejaba la imagen correctamente para poder trabajar con ella.

En los dos primeros casos de este script, primero se convierten las imágenes recibidas de formato DICOM a NIFTI, con el que se trabajará en el proceso. Extrae la máscara del cerebro con el umbral por defecto, corrige los errores provocados por las corrientes de Foucault y los movimientos de la cabeza y devuelve los mapas de difusión.

El siguiente script, `tbss.sh`, controla el flujo de la ejecución de los cuatro pasos de TBSS y del proceso estadístico Randomise. El proceso de TBSS está formado por cuatro scripts contenidos que representan los cuatro pasos del proceso: `tbss_1_preproc`, `tbss_2_reg`, `tbss_3_postreg` y `tbss_4_prestats`. El segundo, `tbss_2_reg`, ha sido modificado para obtener mejores resultados. Cuando se empezó a trabajar con el software, el proceso tardaba tres horas por paciente. El script estaba preparado para un sistema de ejecución en paralelo, SGE [27] (*Sun Grid Engine*) distinto a PBS (*Portable Batch System*) que es el que se tiene en el PIC, lo que lo convertía en una ejecución secuencial que elevaba el tiempo de ejecución a semanas si se contaba con un grupo de pacientes numeroso. Al modificarlo, como se podrá ver en el Análisis de resultados, se redujo ese tiempo de ejecución de una manera considerable, pero además, la publicación de una nueva versión del software mientras se trabajaba en él hizo que el tiempo fuera aún menor. En la parte final se llama al proceso Randomise, que será el encargado de finalizar el estudio estadístico que se había empezado con el proceso TBSS.

El último script es `bedpostx.sh`. Este script permite ejecutar en paralelo el comando `Bedpostx` sobre cada uno de los sujetos en lugar del proceso original en

secuencial, lo que supondrá un ahorro de tiempo en la ejecución final.

Además de la interfaz web y los scripts, se quiere añadir la opción de tener por una parte un visualizador para ver los resultados del pre-procesamiento con DTIFIT. Se va a usar FLS View, herramienta incluida en el paquete de FSL. Por otra parte se pondrá a disposición del usuario el resto de funcionalidades de FSL con su versión gráfica integrada en la web. Estas dos aplicaciones se implementarán vía NX.

La implementación de FSL, al igual que SPHARM-PDM, presentado anteriormente, se divide en dos fases. La primera consistirá en aprender a usar el software en local para entender sus funcionalidades y parámetros, y la segunda implicará su integración en PICNIC, así como el desarrollo para su ejecución en paralelo.

### **Fase 1: Instalación y ejecución secuencial**

- Estudio de las funcionalidades del paquete
- Análisis de los parámetros que se deben especificar al software para su ejecución.
- Diseño de las funcionalidades necesarias en función de lo observado anteriormente.
- Creación de los scripts necesarios para su funcionamiento en batch y secuencial.
- Estudio de los resultados obtenidos que permite decidir el plan de desarrollo de la siguiente fase.

### **Fase 2: Ejecución en paralelo**

- Desarrollo de nuevos scripts en bash que permitan la ejecución del software FSL en paralelo.
- Integración de los scripts creados en la plataforma PICNIC.
- Validación de su ejecución en paralelo una vez implementado en PICNIC.
- Valoración de la aportación del proceso realizado a la ejecución en paralelo de FSL.

## 5. IMPLEMENTACIÓN DE HERRAMIENTAS

- Comparativa de los resultados obtenidos por los scripts desarrollados en comparación a las ejecuciones “manuales” de los usuarios.

Tras este proceso de desarrollo, se dispone de una plataforma amigable con la que poder trabajar los usuarios finales, los neuropsicólogos del hospital.

### 5.2.3. Integración en PICNIC

Para la integración de la herramienta FSL en PICNIC también se ha utilizado PHP para la interfaz gráfica y bash y C-shell para los scripts de procesamiento que se han creado. Se ha seguido la estructura por pestañas aplicada a toda la plataforma, y este apartado en concreto cuenta con cinco pestañas:

1. **Preprocesamiento con DTIFIT.** Se especificarán los sujetos que se quieren procesar con DTIFIT. A la derecha de la pestaña, que se ve en la Figura 5.8, se ha introducido una lista con los pacientes que aún no han sido procesados de cada proyecto, para que el usuario no reprocese ningún sujeto ya procesado. Se puede escoger una lista de pacientes a procesar o si se quiere se puede reprocesar todos los pacientes de un proyecto.

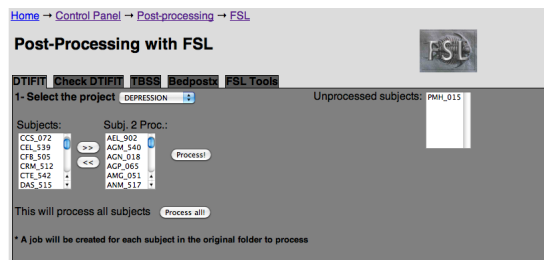


Figura 5.8: DTIFIT

2. **Revisión de los sujetos que ya han sido procesados.** Una vez seleccionado el proyecto se mostrará una lista de sujetos procesados a modo de lista como se ve en la Figura 5.9, para seleccionar si están procesados con el umbral correcto o no. En caso de tener algún paciente que no esté bien procesado, se podrá **reprocesar con un nuevo umbral** especificado por el usuario. El umbral por defecto se ha decidido después de varias pruebas para ver cual era la mejor opción, pero puede ser que alguna de las imágenes necesite otro

## 5. IMPLEMENTACIÓN DE HERRAMIENTAS

distinto. Para visualizar los resultados obtenidos con DTIFIT se usará FSL View, un visor de imágenes que se incluye en el paquete de FSL y que se ha integrado a la web con el visor de aplicaciones remotas NX.

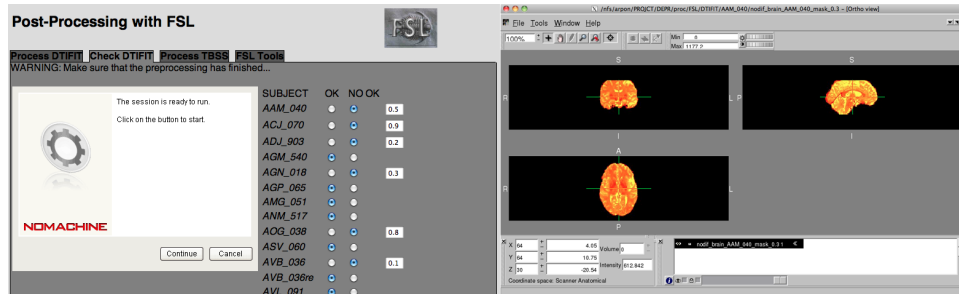


Figura 5.9: Comprobar y reprocesar DTIFIT

- Estudio estadístico.** Primero se procederá a preparar todos los datos necesarios para realizar el estudio estadístico, que se hace con TBSS. El contenido de esta pestaña está dividido en dos partes. La Figura 5.10 muestra la **parte de la izquierda** permite escoger el proyecto y el nombre del estudio que se quiere crear, así como la lista de pacientes que se querrán utilizar y el grupo al que pertenecen. **La parte de la derecha** llevará al apartado de Randomise, el proceso estadístico para el cual se han preparado los ficheros en el paso anterior. Antes de ejecutar el proceso Randomise, se deberá preparar la matriz de valores necesaria con la aplicación de GLM (*General Linear Model*) incluida en FSL, que se abrirá vía NX. Esta aplicación también es accesible a través de la pestaña FSL Tools. Una vez preparado se podrá ejecutar y después ver los resultados con FSL View de nuevo.

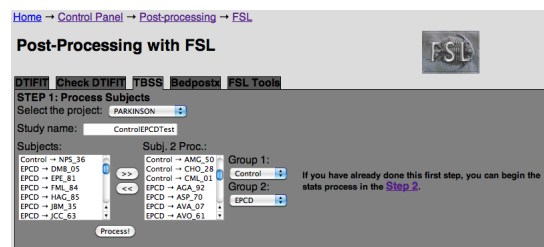


Figura 5.10: Procesamiento TBSS

- Bedpostx.** Este proceso es independiente de todo el flujo que forman DTIFIT, TBSS y Randomise. Con Bedpostx se realiza un estudio estadístico sobre todos los pacientes de un mismo proyecto.

## 5. IMPLEMENTACIÓN DE HERRAMIENTAS

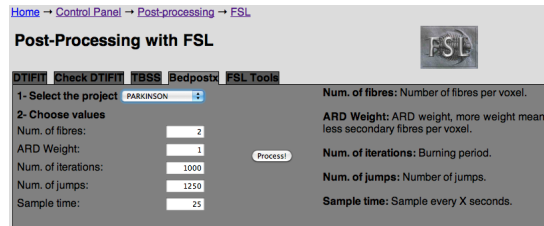


Figura 5.11: BedpostX

5. **FSL Tools** es una opción que se ha querido incluir en la última pestaña, en la Figura 5.12. Se trata de la **aplicación completa de FSL** mediante NX, por si el usuario quiere usar alguna opción de FSL que no esté implementada en la plataforma web.

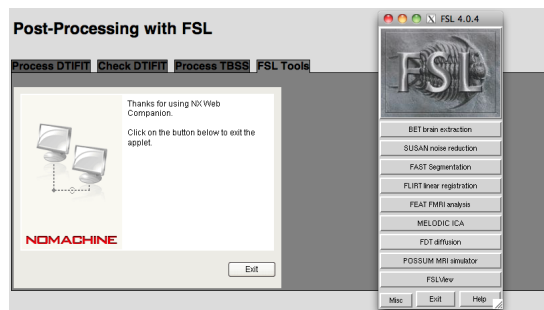


Figura 5.12: FSL Tools

También se ha añadido la funcionalidad de aviso por e-mail al usuario una vez finalice cada uno de los procesos que haya lanzado. Recibirá un e-mail con remitente “PICNIC” y con asunto diferente en función del proceso finalizado a notificar y del estado en que haya terminado. El cuerpo del mensaje será correspondiente también al proceso, pero se especificará además el nombre del fichero del paciente procesado o el nombre del estudio y a qué proyecto pertenece.

Además de la parte visible, a la plataforma se le ha añadido el mismo sistema de Monitorización de Jobs que al software SPHARM. De esta manera el usuario puede gestionar los Jobs que ha lanzado al clúster sin necesidad de tener una terminal con línea de comandos delante, sino que puede borrar los Jobs automáticamente con varios clicks.

### 5.2.4. Problemas y Soluciones

Esta parte del proyecto quizás ha llevado más problemas para desarrollarse que la anterior. Aunque la anterior requirió de la comprensión de la arquitectura sobre la que se trabajaba, esta tenía más dificultad de implementación. Además, el hecho de tener más utilidades, hacía que surgieran más problemas.

El principal problema con el que se ha tenido que trabajar ha sido con la realización del script que modificaba la segunda parte del proceso TBSS. Ese script estaba pensado para trabajar con la ejecución en paralelo, pero aprovechando los recursos que pueden ofrecer los sistema de SGE (*Sun Grid Engine*). En el PIC se dispone del sistema PBS de Torque, lo que no permitía aprovechar el script para la ejecución en paralelo tal y como era. Después de estudiar el script y entender su funcionamiento, se hicieron las modificaciones necesarias para que funcionara con el sistema PBS y se pudo comprobar, como se verá en el apartado de Análisis de resultados, que el tiempo invertido en realizar las modificaciones valió la pena.

Otro problema encontrado fue la actualización del software usado de su versión 4.0 a la versión 4.1 durante el proceso de desarrollo de esta parte. Algunas de las funcionalidades que ya se usaban, y que ya estaban integradas en la plataforma web, sufrieron pequeños cambios que obligaron a revisar el trabajo hecho y readaptarlo a la nueva versión.

Sin duda, el punto más importante y que más ha ayudado en el momento de desarrollar la plataforma web y los scripts necesarios para el funcionamiento del software ha sido que se ha trabajado en todo junto a los usuarios finales, lo que hacía más fácil el desarrollo final de la interfaz y que a la larga supone más tiempo para dedicar a otros aspectos del proyecto.

### 5.2.5. Análisis y Resultados

A continuación se procederá a realizar un estudio e interpretación de los resultados obtenidos tras las ejecuciones de la parte realizada en comparación con el punto del que se partía.

## 5. IMPLEMENTACIÓN DE HERRAMIENTAS

---

Los pacientes de Cushing [28] pueden presentar una serie de deformaciones cerebrales, pero existen pocas evidencias provenientes de estudios sobre MRI que demuestren esta serie de atrofas. El objetivo será estudiar las alteraciones corticales que puedan haber, como la disminución del grosor cortical. Estas comparaciones se realizan con pacientes con síndrome de Cushing y con pacientes de Depresión, de los grupos de control y crónicos, debido a la similitud de síntomas de ambas enfermedades. En el caso de los resultados mostrados en la Tabla 5.2, se han tenido en cuenta enfermos de Cushing y pacientes de Control, sumando un total de 53 individuos.

Igual que en SPHARM, como se ha comentado en el apartado 5.1.5, los resultados obtenidos con la ejecución en secuencial se han obtenido de la ejecución en una máquina con las mismas características que los Worker Nodes para poder hacer una comparación objetiva.

—		Secuencial	Paralelo	Speed-Up
DTIFIT		7 h. 57 min.	9 min.	53 ( <i>#pacientes</i> )
TBSS	tbss_1_preproc	5 min.	5 min.	1
	tbss_2_reg	9 h. 42 min.	11 min.	53 ( <i>#pacientes</i> )
	tbss_3_postreg	10 min.	10 min.	1
	tbss_4_prestats	9 min.	9 min.	1
Randomise		8 h. 20 min.	8 h. 20 min.	1
Total		1 d. 2 h. 23 min.	9 h. 4 min.	2.91

**Cuadro 5.2:** Resultados DTIFIT + TBSS para 53 pacientes

Como se puede ver en la tabla de resultados, en el primer proceso (*DTIFIT*) se ha aprovechado muy bien el sistema de ejecución en paralelo y se ha reducido el tiempo de ejecución de casi ocho horas hasta menos de diez minutos. Es una mejora muy importante debido a que gran parte del procesamiento de FSL es la parte estadística (*TBSS* y *Randomise*), y son este tipo de procesos los que resultan difíciles de paralelizar.

El proceso TBSS está dividido en cuatro partes, de las cuales una (*tbss\_2\_reg*) si que se ha podido paralelizar, y se ha obtenido una muy buena mejora logrando un tiempo de ejecución de diez minutos, muy poco tiempo comparado con las casi diez horas que necesita el mismo número de pacientes para el mismo proceso pero

## 5. IMPLEMENTACIÓN DE HERRAMIENTAS

---

con ejecución secuencial. El resto de procesos de TBSS no se pueden ejecutar en paralelo, pero tienen una ejecución relativamente rápida, así que la valoración de los resultados es muy positiva.

En cuanto a la ejecución de Randomise, es la que más tiempo necesita para completarse, pero no se ha podido implementar un método de ejecución en paralelo por sujetos, por lo que la mejora final se verá penalizada por culpa de este proceso.

Igual que en la herramienta anterior, nos encontramos en que la mejora, reflejada en la columna *Speed-Up*, se puede apreciar como en algunos de los procesos (`dtifit` y `tbss_2_reg`) el tiempo de ejecución es proporcional al número de sujetos a procesar. De esta manera se conseguirá una mejora mayor cuanto mayor sea el número de pacientes mientras sea menor que el número de nodos disponibles.

Además de esta tabla de resultados, hay que añadir la mejora que se ha realizado en la ejecución del Bedpostx tras paralelizar el proceso que realiza donde, aunque solo tenga un paso, se ha conseguido una gran mejora en cuanto al tiempo de ejecución. Los resultados se muestran en una tabla distinta, ya que no forman parte del mismo proceso que el anterior, aunque ambos son utilidades de FSL. Esta prueba se ha realizado sobre los pacientes del proyecto de Depresión, que suman un total de 123 pacientes.

—	Secuencial	Paralelo	Speed-Up
Bedpostx	47 d. 20 h.	9 h. 20 min.	123 ( <i>#pacientes</i> )

**Cuadro 5.3:** Resultados Bedpostx para 123 pacientes

Como se puede ver en la Tabla 5.3, los resultados suponen una gran mejora respecto al tiempo que se necesitaba para realizar el proceso sobre todos los pacientes. Debido a la larga duración del proceso con el método de ejecución secuencial, aún no se habían procesado todos los sujetos de un proyecto, y tan solo se habían hecho pruebas con algún sujeto, pero tras implementar la ejecución en paralelo se ha observado que la mejora de tiempo crece linealmente mientras el número de pacientes sea menor que el número de nodos.



## 5.3. SPM con Máquinas Virtuales

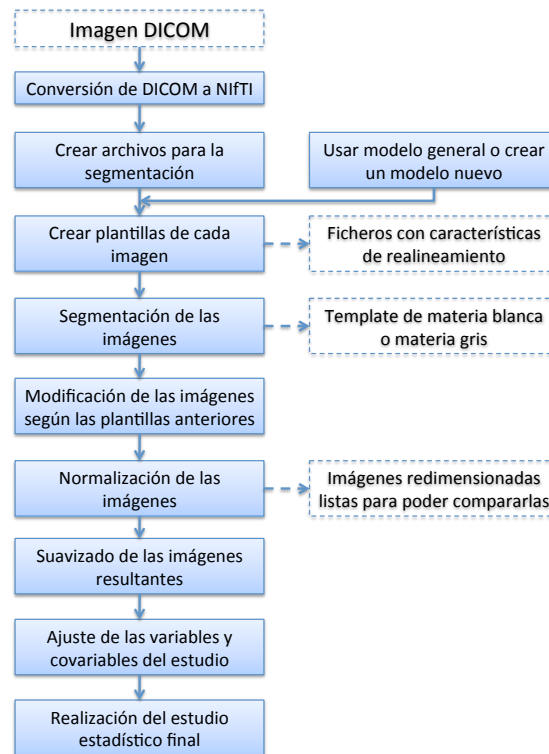
### 5.3.1. Descripción del Software

Este componente del proyecto tiene unas particularidades distintas a SPHARM y FSL, por lo que se seguirá un proceso de desarrollo distinto, aunque el objetivo final también será implementarlo en PICNIC.

SPM8 (*Statistical Parametric Mapping*) es una herramienta de procesamiento de resonancias magnéticas que incluye un paquete estadístico y se utiliza para examinar diferentes modalidades de estudios de MRIs (funcional, estructural, electroencefalogramas, etc). Se distribuye como software libre, pero está escrito en Matlab [29] y se ejecuta sobre él.

El principal problema de una aplicación Matlab en un clúster viene dado principalmente por las licencias. En una configuración clásica, se deberían tener tantas licencias de programa como usuarios o máquinas estuvieran trabajando con SPM y sólo tendrían licencia unos pocos nodos, además de obligar a ocupar espacio con la instalación en ese nodo. Pero con el uso de máquinas virtuales se puede aprovechar mejor el número de licencias de las que se dispone y de esta manera disponer de una plataforma más flexible. La idea sería que al enviar un Job a la granja con los mismos comandos para PBS que se usan hasta el momento, en lugar de ejecutar un script dentro del nodo al que se le asigna el Job, este se ejecutaría dentro de una máquina virtual, arrancada por el nodo que se le ha asignado al Job.

En el procesamiento con SPM, representado en el esquema de la Figura 5.13, en primer lugar se convierten las imágenes que están en formato DICOM a NIfTI. A continuación se hace una reorientación manual para que el software pueda hacer el registro a un *template*. El siguiente paso es la segmentación de las imágenes reorientadas en diferentes tejidos (materia gris, blanca y líquido cefalorraquídeo). Para hacer esto, se crea un *template* con los individuos del estudio o bien se utiliza el que se tiene por defecto (Template ICBM: [30] *International Consortium of Brain Mapping*).



**Figura 5.13:** Esquema SPM

Seguidamente se usará el paquete DARTEL (*Diffomorphic Anatomical Registration Through Exponentiated Lie Algebra*) [31] para realizar un registro más preciso de las imágenes resultantes y permitir obtener mejores resultados. Finalmente, se aplicará un filtro gaussiano para suavizar las imágenes.

En este punto del análisis ya se tendrán los ficheros necesarios para hacer el estudio estadístico, que se realiza mediante un estudio vóxel (píxel volumétrico) [32] a vóxel. Los resultados se nos mostrarán en función de las variables, que son los grupos del estudio según el grado de evolución de la enfermedad que se está estudiando, y una serie de covariables relacionadas con el curso de la enfermedad y que pueden influir en los resultados, como por ejemplo la edad del paciente.

La segunda parte de este apartado consiste en la creación de un modelo de máquina virtual que funcione correctamente con Matlab y que además sea ligera. A continuación se creará un Parser que editará el Job creado en SPM para crear los Jobs individuales que se envían a la granja.

### 5.3.2. Proceso de Implementación

El proceso de implementación de este apartado del proyecto es ligeramente diferente a los anteriores. Se diferenciarán dos partes: en primer lugar se creará el modelo de máquina virtual a utilizar, y después se estudiará el software para crear los Jobs necesarios para su ejecución.

#### Creación de la máquina virtual

El objetivo principal es ejecutar SPM con Matlab en una máquina virtual. Una vez se consiga ejecutar Jobs correctamente se estudiarán qué procesos y drivers no son necesarios para la ejecución de los Jobs enviados y se evitará que se carguen al iniciar la máquina, de manera que se pueda obtener una arrancada de la máquina virtual más rápida y un pequeño ahorro de recursos del sistema.

En el proceso de creación de la máquina virtual se ha utilizado Qemu [33] para generar el espacio de disco y KVM [34] para la emulación, ya que permite realizar virtualización completa. Para el sistema operativo se ha utilizado CentOS 5.4, dado que en el grupo se dispone de una máquina real funcionando con Matlab y SPM8 sobre este sistema operativo y se ha aprovechado para hacer una réplica del entorno.

Se han realizado pruebas con máquinas de 32 bits y de 64 bits, ambas con la misma configuración base. Para la máquina de 32 bits se ha conseguido un funcionamiento correcto sin ningún problema, pero para la de 64 bits se tuvo que solucionar algún problema que no permitía la ejecución. Dado que la ejecución en una máquina de 64 bits es más rápida que en una de 32 bits, se ha tenido especial interés en obtenerla. Además, las máquinas de 64 bits permiten el uso del driver para emulación de discos duros Virtio. Este driver tiene velocidad de lectura y escritura mayor que el driver por defecto de KVM. Tras muchas pruebas con cambios de sistema operativo, de versiones de Java, con distintos drivers de disco (*SCSI*, *Virtio*, *HDA*) se ha descubierto que en los sistemas operativos probados para máquinas de 64 bits, viene activado por defecto el sistema de seguridad SELinux [35], siendo este sistema quien causaba problemas.

## 5. IMPLEMENTACIÓN DE HERRAMIENTAS

---

Una vez solucionado el inconveniente de SELinux, se ha procedido a realizar una metodología para crear una máquina igual que la que se tiene ahora. Los pasos son los siguientes:

1. Creación del espacio de disco necesario para la instalación de la máquina.

```
/usr/bin/qemu-img create -f qcow2 /home/vm/test.img 16G
```

2. Arranque de la máquina virtual con el driver de disco duro Virtio y con la imagen de CentOS 5.4 en la unidad de CD-Rom para instalarla.

```
/usr/bin/kvm -cpu qemu64 -M pc -m 2048 -smp 1 -name vmneuro64  
-boot dc -cdrom /home/vm/CentOS-5.4-x86_64-netinstall.iso -drive  
file=/home/vm/vmneuro64b.img, if=virtio, index=0, boot=on, snap-  
shot=off ,media=disk -k es -vnc 0.0.0.0:2 -no-fd-bootchk -clock  
unix
```

3. Instalación de Java (*versión 18*). Aunque se ha probado que funciona a partir de la versión 6 update 16, en el momento de las pruebas se hizo con el update 18.

```
wget "http://cds.sun.com/.../jre-6u18-linux-x64.bin"  
./jre-6u18-linux-x64.bin  
cd /usr/bin  
ln -s /usr/jre1*/bin/java
```

4. Instalación de X Window System. Este paso es muy importante para poder usar aplicaciones que tengan entorno gráfico como Matlab y SPM.

```
yum groupinstall "X Window System"
```

5. Desactivación SELinux. Cambiar la línea SELINUX deja el sistema desactivado para no tener conflicto en la ejecución de aplicaciones.

```
vim /etc/selinux/config  
SELINUX=disabled
```

6. Instalación libXp para 64 bits. Dependencia para poder instalar Matlab.

```
yum install libXp-1.0.0-8.1.el5.x86_64
```

7. Copia de la carpeta del software de SPM8 en `/usr/local/spm8`.
8. Reinicio de la máquina con la imagen de Matlab cargada en la unidad de CD-Rom para poder instalarla. Es muy importante reiniciar antes de instalar Matlab. De esta manera tendrán efectos los cambios realizados en el fichero de configuración SELinux.
9. Introducción de la carpeta donde se ha guardado SPM8 anteriormente en los paths conocidos por Matlab.

### Creación de Jobs

Ahora se tendrá que realizar el estudio sobre el funcionamiento de SPM8 y como crea sus Jobs para poder crear los scripts necesarios para su ejecución en paralelo. Para este proceso se observarán los Jobs creados por SPM8 para los siguientes procesos del software: *Segment*, *Initial Import*, *Create Template (DARTEL)*, *Create Warped* y *Smooth*. Estos procesos se han explicado en el esquema de la Figura 5.13.

SPM8 se ejecuta a través de un entorno gráfico dentro del cual se especifican los procesos que se quieren ejecutar. De momento se tendrán en cuenta tan solo los cinco nombrados anteriormente. Para su implementación se mantendrá el sistema gráfico original, con lo que el usuario introducirá los parámetros de manera convencional. Una vez especificados los procesos que quiere realizar, el usuario guardará el Job en forma de script con la opción *Save Job and Script* de SPM. Se han desactivado la opción de ejecución a través de la interfaz gráfica para evitar que el usuario lance el Job desde la ventana de SPM. Al guardar el script se crean dos ficheros: uno con el flujo de ejecución del Job y otro con los datos de configuración. En la Figura 5.14, viene representado todo el procedimiento del proceso de ejecución. Consiste en que el usuario cree su Job completo desde la aplicación gráfica de SPM.

Para esta implementación se ha trabajado sobre cinco procesos. Una vez guardados los Jobs, se crean los dos ficheros antes nombrados que contienen todo el flujo de ejecución, pero secuencialmente. El objetivo es ejecutar en paralelo (por pacientes) los procesos que lo permitan. En este caso son *Segment*, *Initial Import*, *Create War-*

## 5. IMPLEMENTACIÓN DE HERRAMIENTAS

---

*ped* y *Smooth*, mientras que *Create Template* al ejecutarse sobre todo el conjunto de usuarios a la vez, se tendrá que hacer secuencialmente.

Con estos ficheros y la lista de sujetos que se lee del Job completo, y con la ejecución de un Parser (*editjobs.sh*) creado especialmente para estos procesos, se obtiene tres tipos de scripts:

1. Un fichero *SegInit.m* para cada sujeto de la lista, que contendrá los necesario para la ejecución de los procesos *Segment* e *Initial Import*.
2. Un fichero *Temp.m*, que se tendrá uno para todo el conjunto de pacientes a procesar, correspondiente al proceso *Create Template*.
3. Un fichero *WarpSmo.m* para cada sujeto que contiene el flujo de ejecución para los procesos *Create Warped* y *Smooth*.

La razón por la cual se han separado con esta estructura es que los ficheros de entrada de *Initial Import* dependen de la salida de *Segment* y que la entrada de *Smooth* depende de *Create Warped*. *Create Template* es un proceso secuencial y es el punto central del proceso completo. Las imágenes que recibe son las de salida de *SegInit*, y las que genera son las que espera *WarpSmo* para comenzar su ejecución. Se ha aprovechado la paralelización por sujetos para optimizar los tiempos de ejecución de las diferentes funciones.

Una vez creados todos los Jobs, se procede a realizar su ejecución mediante el script *runspm.sh*. Los Jobs son enviados al cluster para su ejecución en paralelo y para mantener el flujo de ejecución deseado se usan una serie de comprobaciones (*checkend()*) para ver si se han generado los outputs necesarios para comenzar la ejecución del siguiente proceso. Para evitar la sobrecarga del servidor, este tan solo se utiliza para controlar el flujo de ejecución, también con el script *runspm.sh* mientras que el procesamiento se realiza en el cluster.

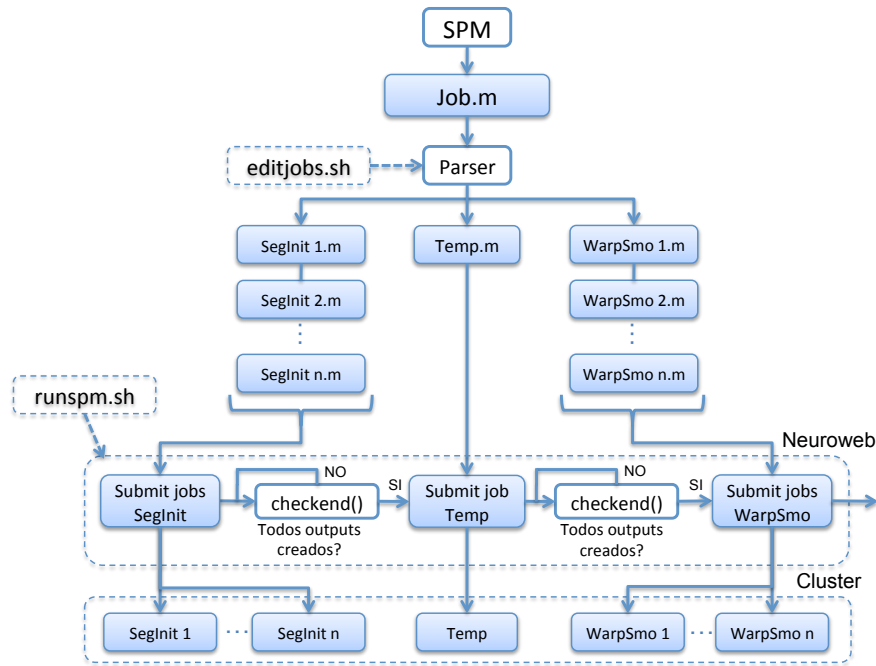


Figura 5.14: Estructura SPM

Para que los Jobs que se han enviado a la granja se ejecuten con la máquina virtual creada, se deberá añadir a la cabecera de cada Job la siguiente línea:

```
#PBS -l other=vmneuro64b.img
```

Una vez puesto a punto todo el sistema de creación de la máquina virtual y de la ejecución de Jobs en la máquina virtual creada, se ha realizado el diseño de la interfaz gráfica para integrar SPM en PICNIC.

### Versión compilada del software

En la parte final del proyecto, se ha tenido contacto con los desarrolladores de SPM8 y se ha podido obtener una versión beta del software compilado. Como se ha dicho anteriormente, este software necesita Matlab para funcionar, pero existe una alternativa, que es realizar una versión compilada con una utilidad que ofrece Mathworks.

Se trata de Matlab Compiler [36], una herramienta que permite compilar los ficheros creados con Matlab y convertirlos en una versión *standalone* del software. Este proceso crea archivos ejecutables que hacen que el software deje de ser depen-

## 5. IMPLEMENTACIÓN DE HERRAMIENTAS

---

diente de Matlab para funcionar. Mathworks permite la libre distribución de estos ficheros ejecutables.

Para poder ejecutar el software compilado se debe instalar un paquete facilitado por Mathworks llamado MCR (*Matlab Compiler Runtime*) en *NFS*, para que todos los Worker Nodes puedan acceder a él. Este paquete es el encargado de interpretar los ficheros compilados y puede ser distribuido legalmente por la persona que haya compilado el software que se quiere utilizar. De esta manera se podrán enviar Jobs a la granja de la misma manera que se hace con las otras herramientas, ejecutando el comando de PBS `qsub` que ejecute SPM8, especificando la ruta del Matlab Compiler Runtime y con el comando que se quiere ejecutar dentro del software compilado. El comando utilizado para ejecutar los jobs lanzados a la granja con el SPM compilado es:

```
$PATH_SPM8/run_spm.sh $PATH_MatlabMCR/v711 run seginit.m
```

En cuanto al proceso de creación de Jobs, se seguirá usando el mismo *Parser* que se tenía, ya que lo único que cambia es la plataforma sobre la cual se ejecuta el software, y los Jobs que se envían a la granja son exactamente los mismos y son tratados de la misma manera.

Actualmente se están realizando pruebas de rendimiento con este software, y de momento no se tiene implementado ya que no ha sido lanzado oficialmente, siendo el trabajo que se realiza en el grupo de prueba y de estudio de rendimiento en colaboración con los desarrolladores.

En el apartado de Resultados se mostrarán una serie de tablas con los tiempos obtenidos con diferentes características de ejecución, y comparándolos con los tiempos obtenidos con la ejecución usando máquinas virtuales.

### 5.3.3. Integración en PICNIC

Para crear la interfaz de usuario de SPM en PICNIC, se pensaron dos posibilidades. La primera opción era seguir con la misma metodología que se ha usado en



## 5. IMPLEMENTACIÓN DE HERRAMIENTAS

SPHARM y FSL, donde el usuario especifica los parámetros más importantes necesarios para cada paso, y los que no se suelen cambiar dejarlos como valores por defecto siendo pasados al programa para que opere con ellos. La otra opción era integrar con NX el entorno gráfico de SPM para que el usuario creara Jobs necesarios y después separarlos en las partes necesarias para su ejecución. Finalmente se optó por la segunda. En el análisis de problemas se analiza el por qué de esta decisión, y porque finalmente se ha decidido no seguir la misma estructura que en los anteriores paquetes.

De esta manera la interfaz gráfica resulta más sencilla de implementar que en los casos anteriores, aunque se sigue usando la estructura por pestañas, quedando en este caso en tres:

1. **Creación de un nuevo proyecto** de SPM. En la pestaña de la Figura 5.15 lo que se hace es asignar un nombre a un nuevo estudio de SPM a uno de los proyectos existentes. Se tendrá que especificar el proyecto y un nuevo nombre de estudio.

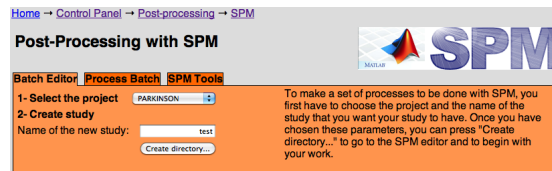


Figura 5.15: Creación nuevo proyecto SPM

Al hacer click en el botón para Crear el Directorio se creará todo el árbol de carpetas necesario para el proceso de estudio, aparecerá el pop-up de la Figura 5.16 indicando la ruta en la cual se deben guardar los datos y se abrirá el Batch Editor (Figura 5.17) de SPM vía NX. Al nombre indicado se le añade la fecha actual, para así no tener conflictos de creación de estudios, y para tener conocimiento de cuando se realizó el estudio.

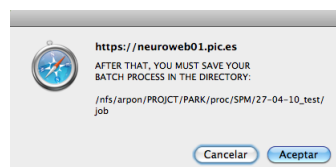


Figura 5.16: Aviso SPM

## 5. IMPLEMENTACIÓN DE HERRAMIENTAS

Además, con tal de reducir la aparición de errores, se ha editado la interfaz gráfica de SPM integrada en PICNIC eliminando la posibilidad de usar las características y funciones del software que no son funcionales en nuestro caso.

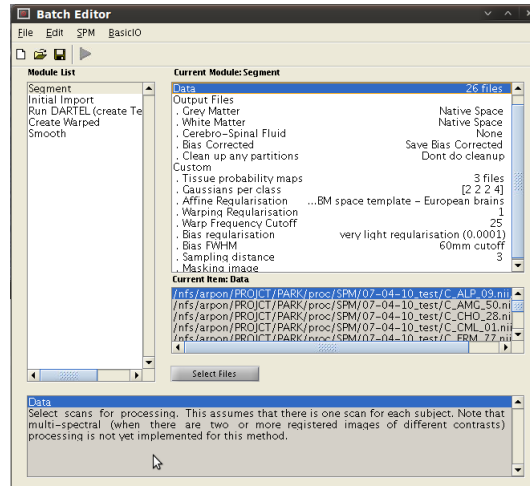


Figura 5.17: SPM Batch Editor

2. **Ejecución de los scripts del proyecto**, después de haberlos separado en sus partes correspondientes. Una vez seleccionado el proyecto y el estudio que se ha creado y donde se ha tenido que guardar el Job completo desde el *Batch Editor*, se procesan los ficheros generados para separarlos en sus Jobs correspondientes, y ser enviados a la granja ejecutándose sobre la máquina virtual creada.

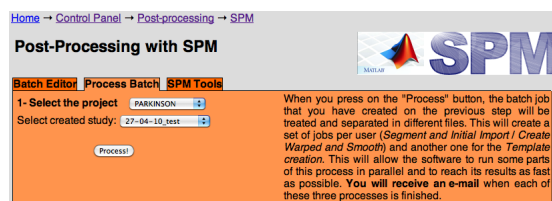


Figura 5.18: Process Batch Jobs

3. **SPM Tools**, al igual que en FSL, donde se añade la aplicación completo vía NX, con SPM también se da al usuario la posibilidad de cargar toda la aplicación gráficamente por si quiere usar alguna otra de las posibilidades que ofrece.

### 5.3.4. Problemas y Soluciones

El hecho de que este paquete debía implementarse de manera distinta a los anteriores y usando herramientas distintas, ha hecho que los problemas encontrados fueran un poco desconocidos. Aún así se han podido solucionar de manera satisfactoria.

Uno de los problemas encontrados ha sido en la **creación del modelo de máquina virtual** que se quería utilizar. Se llevaron a cabo pruebas con Linux de 32 bits y de 64 bits. Se empezó con la de 64 bits para recrear el entorno de la máquina real de la que se disponía en el grupo de Imagen Médica, pero no se ha conseguido ejecutar el software. Entonces se ha probado con otra versión del mismo sistema operativo pero para arquitecturas de 32 bits. En este caso el software ha funcionado. Investigando sobre un posible error se ha descubierto que la versión del Kernel de la máquina de 32 bits era inferior al 2.6, mientras que la de 64 bits era mayor. Buscando en las ayudas de Matlab y del sistema operativo, se ha encontrado que las versiones de Kernel superiores a la 2.6.x, llevan incorporado el sistema de seguridad SELinux, y que se debía desactivar o añadir excepciones para utilizar el software sin problemas.

SELinux (*Security-Enhanced Linux*) es un sistema desarrollado por la NSA (*National Security Agency*) y Red Hat e integrado en los Kernel de Linux con versión 2.6 y posterior. Provee al sistema operativo de un conjunto de políticas de seguridad que pueden ser cambiadas por el administrador del sistema. Estas políticas consisten, a grandes rasgos, en limitar el acceso del sistema a aquellos procesos que cree que pueden no ser seguros. De manera que cuando Matlab se ejecuta, y SPM se abre, pero no se llega a ejecutar ninguno de sus procesos. Para conseguir la ejecución se ha optado por desactivar completamente SELinux, en lugar de editar sus excepciones, ya que nadie va a tener acceso al interior de la máquina, y así se evitan errores similares en el futuro.

Una vez resuelto este problema, y con las dos máquinas creadas, se hicieron pruebas de rendimiento entre una y otra. La máquina de 32 bits, en algunas aplicaciones necesita casi el doble de tiempo en ejecutar lo mismo que la de 64 bits. Esto se

podrá ver más detalladamente en las tablas de resultados.

En el apartado de **integración en PICNIC** se ha comentado el hecho de tener que tomar una decisión sobre qué se creía mejor para el usuario a la hora de integrar el paso de parámetros, y finalmente se creyó mejor para el usuario integrar con NX el *Batch Editor*. El problema surgió al pensar la mejor manera de integrar el paso de parámetros en la interfaz web. La gran cantidad de parámetros que se le deben pasar a SPM fue determinante, ya que se pensó en intentar escoger los más importantes y los otros dejarlos constantes, pero si el usuario quisiera cambiar alguno de ellos, debería modificar el código, y no es algo muy recomendable. Entonces se decidió seguir con la interfaz gráfica a la cual los usuarios ya estaban habituados, y así a ellos también se les facilitaría el trabajo.

Para complementar la decisión anterior, al decidir mantener la interfaz gráfica, se ha limitado a los usuarios las acciones que pueden realizar, modificando la interfaz gráfica de SPM, programada en Matlab. Debido a que SPM tiene una gran cantidad de ficheros, resultaba difícil encontrar el encargado de la GUI. Al ser software libre y de código abierto, se contactó con los desarrolladores (*Wellcome Trust Center for NeuroImaging de Londres*) para saber qué ficheros debían ser editados. Para realizar estas modificaciones se ha usado el editor de GUI's de Matlab, GUIDE (*Creador de Interfaces Gráficas de Usuario*).

Igual que con el resto de implementaciones, en la integración de SPM a PICNIC, ha sido muy importante también para el desarrollo final el haber podido trabajar tan cerca de los usuarios finales, ya que de esta manera se ha podido adaptar la aplicación a sus necesidades, lo que ha motivado la adaptación de los 5 procesos descritos (*Segment, Initial Import, Create Template, Create Warped, Smooth*) a PICNIC, y ha permitido acelerar el proceso de aprendizaje de sus funciones.

### 5.3.5. Análisis y Resultados

Para el análisis de resultados se ha observado primero la diferencia entre las ejecuciones de los mismos procesos con las máquinas de 32 y 64 bits, y después se ha

## 5. IMPLEMENTACIÓN DE HERRAMIENTAS

---

realizado un estudio sobre distintas posibilidades de ejecución con la máquina de 64 bits, la que finalmente ha sido elegida.

—	32 bits	64 bits
Arrancada	61 seg.	54 seg.
Segment	15 min.	6 min.
Initial Import	3 min.	46 seg.
Create Template	54 min.	41 min.
Create Warped	10 seg.	5 seg.
Smooth	1 seg.	1 seg.
Apagada	33 seg.	28 seg.

**Cuadro 5.4:** Comparativa entre máquinas para un solo paciente

En los resultados de tiempos de la Tabla 5.4, obtenido del procesado de la secuencia de imágenes *T13D* separada en 170 cortes de un solo paciente, se puede ver como la máquina de 64 bits es mucho más rápida que la de 32 bits en algunos de sus procesos. Los tiempos de puesta en marcha y cierre de ambas es similar, aunque sigue siendo mejor la de 64 bits. Estos tiempos se han calculado con el sistema original, por lo que una vez realizado el estudio de necesidades se conseguirá rebajar el tiempo de puesta en marcha y cierre de la máquina virtual.

En el proceso de *Segment* y en *Create Warped*, se puede ver como el tiempo de ejecución es la mitad en la máquina de 64 bits y casi una tercera parte en el caso de *Initial Import*. La diferencia entre las dos máquinas con el proceso de *Smooth* es inapreciable y el tiempo registrado ha sido de un segundo. En el caso de *Create Template* la diferencia no es tan notable, aunque llega casi a 15 minutos menos en la máquina de 64 bits. La razón de que esta diferencia no sea proporcional a las anteriores puede deberse a que en este proceso se dedica mayor tiempo a escritura en fichero que a lectura.

Para el proceso implementado con el uso del software compilado, se han hecho pruebas para los mismos procesos, pero forzando el uso de un procesador por Job en un caso o dos en otro, para comparar las velocidades y poder decidir cual sería el adecuado en caso de terminar implementando este sistema. Se realizarán pruebas con 8 pacientes del proyecto de Parkinson (4 del grupo de *Control* y 4 del grupo

## 5. IMPLEMENTACIÓN DE HERRAMIENTAS

---

*EPCD*) y con los tiempos medios de espera en cola y de ejecución para cada proceso. El proceso `temp.sh` tiene un Job para todos los sujetos, así que el tiempo es tan sólo el de un único Job.

Observando los resultados presentados en la Tabla 5.5, se puede ver como con dos procesadores la media de tiempos es más baja, pero se necesita más tiempo de espera en cola, ya que hay que esperar a que hayan disponibles dos procesadores por cada Job enviado. Esto implica que con una cantidad mayor de sujetos a estudiar este tiempo de espera sea mayor. Esta prueba se ha realizado debido a que con ejecuciones individuales, se ha comprobado que la ejecución con dos procesadores es mucho más rápida que con uno (salvo en el proceso intermedio, `Temp`) pero al lanzar muchos sujetos las medias entre todos los resultados igualan un poco el resultado final, por lo que se usará la versión que utiliza un sólo procesador.

—		1 CPU / Job	2 CPU / Job
Seg Init	Espera en cola	1 min.	11 min.
	Ejecución	9 min.	6 min.
Temp	Espera en cola	1 min.	6 min.
	Ejecución	71 min.	68 min.
Warp Smo	Espera en cola	2 min.	15 min.
	Ejecución	2 min.	1 min.

**Cuadro 5.5:** Comparativa tiempos 1 CPU vs. 2 CPU

Una vez decidido que las mejores opciones para cada caso son, para el caso de máquinas virtuales la de 64 bits, y para el caso de la versión compilada aquella que se ejecuta con tan sólo un procesador por proceso, se obtiene la comparativa que se muestra en la Tabla 5.6. Los tiempos de espera en cola son iguales, y no se han incluido ya que no se consideran relevantes para la toma de una decisión final.

—	Versión Compilada	Máq. Virtual 64 bits
Seg Init	9 min.	7 min.
Temp	71 min.	41 min.
Warp Smo	1 min.	30 seg.
Total	81 min.	42 min.

**Cuadro 5.6:** Comparativa entre mejores versiones de cada modelo

## 5. IMPLEMENTACIÓN DE HERRAMIENTAS

Aunque la versión compilada es un poco más lenta que la versión que se ejecuta sobre máquinas virtuales, esta ofrece la ventaja de no requerir Matlab, y por lo tanto no requiere la adquisición de una licencia. En cambio, la versión que funciona con máquinas virtuales consigue una ejecución más rápida, pero el factor de que necesite una licencia de Matlab puede suponer un problema y un gasto económico adicional.

### 5.4. Otras Herramientas

#### 5.4.1. ITK-Snap

ITK-Snap [10] es una aplicación que permite la segmentación de órganos del cuerpo humano a partir de imágenes médicas en tres dimensiones. Esta aplicación permite segmentación semiautomática a partir de técnicas basadas en contornos activos, así como navegación visual a través de la imagen. Ofrece soporte para diferentes formatos de imágenes 3D.

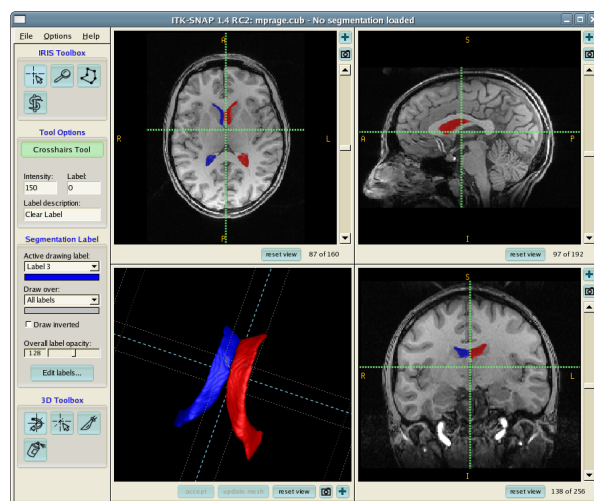


Figura 5.19: ITK-Snap

En este caso se ha implementado en PICNIC con el uso de NX. Se ha querido añadir para que se tenga acceso remoto, pero de momento tan solo se ha introducido la interfaz gráfica original.

No se han tenido excesivos problemas ya que en el apartado correspondiente se ha realizado una implementación simple con tan solo una pestaña en la interfaz web

## *5. IMPLEMENTACIÓN DE HERRAMIENTAS*

---

con el visor de aplicaciones remotas NX. Se estudiará la posibilidad de realizar una implementación más completa.



# 6

## Trabajo Futuro

El objetivo principal de PICNIC es conseguir una plataforma completamente escalable y con la posibilidad de añadir fácilmente todo tipo de herramientas adicionales a las ya existentes.

Para posibles mejoras se podría estudiar el conjunto de herramientas que ofrece el paquete FSL e intentar añadir alguna otra opción a las ya disponibles de DTIFIT, TBSS, Randomise y Bedpostx (como por ejemplo *fslmeans* o *fslmaths*).

En cuanto a SPM, se ha mantenido contacto con los creadores del software y han explicado que están terminando una versión *stand-alone* del conjunto de aplicaciones, la cual se ha podido probar durante la realización de este proyecto. Puede ser un paso muy importante, ya que no sería necesario Matlab para ejecutar SPM, lo que además de suponer un ahorro económico supondría el poder usarlo como un software independiente y quizás no tener que usar las máquinas virtuales creadas para su ejecución. Antes de tomar una decisión se deberá realizar un estudio, considerando las ventajas y desventajas de cada caso para considerar cual de los dos sistemas.

Un apartado distinto a los anteriores que se podría implementar es el de un sistema de creación de informes. Este sistema permitiría a los médicos realizar informes radiológicos de los pacientes que se han sometido a pruebas de neurorradiología contrastando sus datos con los de los pacientes ya procesados. Ayudaría a realizar

## 6. TRABAJO FUTURO

---

un diagnóstico precoz o como mínimo a situar el estado del paciente en un punto en comparación con el resto del grupo de investigación. Además, permitiría generar informes con la misma estructura y facilitando la tarea de los médicos en el momento de redactarlos.

Por último, otra posible mejora sería la de añadir a la plataforma un sistema de monitorización de las actividades del grupo de imagen médica, de manera gráfica y numérica, de la cantidad de Jobs enviados a la granja, del estado en que han terminado, o de la cantidad de memoria que han necesitado para su ejecución, entre otros.

Como se ha visto, PICNIC es una plataforma modular, por lo que favorece la implementación de nuevas herramientas de procesamiento y análisis online de MRIs.

# 7

## Conclusiones

Dado que PICNIC ofrece una estructura modular, se han podido implementar más herramientas de post-procesamiento de imagen médica online satisfactoriamente, como son SPHARM, FSL, SPM y ITK-Snap

En cuanto a SPHARM y FSL se ha optado por la realización de una interfaz gráfica de paso de parámetros para ejecutar una serie de scripts que realizan el proceso por detrás de la interfaz, obteniendo una buena interacción entre la plataforma y el usuario. Estos scripts aprovechan las posibilidades de paralelización que ofrece el PIC gracias al número elevado de máquinas y de la potencia de procesamiento de datos de que se dispone, reduciendo de manera notable los tiempos de ejecución de los que se partía.

En cuanto a SPM finalmente se ha decidido mantener la interfaz gráfica del programa dado a la gran cantidad de parámetros de entrada. Esta configuración es más ventajosa para el usuario ya que sigue trabajando con la interfaz a la que está acostumbrado, pero realizando una serie de ejecuciones en *background* de la plataforma web para mejorar la ejecución original, y conseguir aprovechar la paralelización de datos.

También se ha añadido la herramienta ITK-Snap de manera gráfica. No se ha modificado su forma de funcionar ni de uso, pero se da la posibilidad de usarla remotamente a través de la plataforma web.

## 7. CONCLUSIONES

---

De esta manera, PICNIC resulta ser una aplicación muy potente si se comparan sus requisitos con las posibilidades de procesamiento que ofrece. Para usar PICNIC en cualquier ordenador tan solo se necesita tener una versión actual de Java, un navegador web (Explorer, Firefox, Safari, Chrome...) y conexión a Internet. Estos requisitos los cumplen la mayoría de ordenadores existentes y al estar implementada en una interfaz web, lo cual la mayoría de usuarios conoce, la convierte en una herramienta potente y fácil de usar.

Durante la realización de este proyecto se ha podido comprobar como el uso de clústers y de computación distribuida aporta una gran cantidad de posibilidades de procesamiento, pero sobretodo un aumento de las prestaciones de las aplicaciones que aprovechan esta característica. Mientras que las aplicaciones implementadas se suelen usar fuera de PICNIC, y por tanto sin los recursos de computación de los que se dispone en el PIC (lo que implica estudios de muy larga duración), aquí sí se han aprovechado, y convierte a PICNIC en una herramienta innovadora dentro del campo de la neuroimagen ya que permite realizar post-procesamiento de imágenes no solo online, sino que además realiza gran parte de los procesos en un transcurso de tiempo mucho menor al que se requería hasta el momento.

En cuanto a la aportación de la ingeniería a este proyecto, hay que remarcar que se ha seguido una metodología que ha facilitado su desarrollo. Para cada uno de los bloques de los que constaba el proyecto se ha realizado un análisis y estudio de los objetivos, un conjunto de propuestas de solución y la implementación necesaria. Posteriormente se han realizado una serie de pruebas para validar el trabajo hecho.

También añadir que el trabajo realizado el año anterior por el creador de PICNIC y su soporte en las tareas de desarrollo de las partes correspondientes a este proyecto han sido muy importantes, y sin lo cual no habría sido posible el estado final al que se ha llegado.

# Bibliografía

- [1] PIC, *Port d'Informació Científica*, <http://www.pic.es>, Noviembre 2009.
- [2] Medical NeuroImaging Group at PIC, *PIC NeuroImaging Center*, <http://neuroweb01.pic.es>, Julio 2010.
- [3] Mark A. McPeck, *SPHARM Morphometric Software*, <http://www.enallagma.com/SPHARM.php>, Noviembre 2009.
- [4] FMRIB, *FSL: FMRIB Software Library*, <http://www.fmrib.ox.ac.uk/fsl>, Enero, 2010.
- [5] Wellcome Trust Centre, *SPM: Statistical Parametric Mapping*, <http://www.fil.ion.ucl.ac.uk/spm>, Mayo, 2010.
- [6] MySQL, *The world's most popular open source database*, <http://www.mysql.com>.
- [7] PHP: *Hypertext Preprocessor*, <http://www.php.net>.
- [8] Apache, *The Apache Software Foundation*, <http://www.apache.org>.
- [9] PBS: *Portable Batch System*
- [10] P. Yuskevich, G. Gerig, *ITK-SNAP*, <http://www.itksnap.org>, Noviembre, 2009.
- [11] R-Project, *The R Project for Statistical Computing*, <http://www.r-project.org/>, Mayo, 2010.

- [12] Nomachine NX, *Desktop Virtualization and Remote Management Software*, <http://www.nomachine.com/>.
- [13] NFS, *Network File System*, <http://www.nomachine.com/>.
- [14] M. Styner, I. Oguz, S. Xu, C. Brechbühler, D. Pantazis, J. J. Levitt, M. E. Shenton, G. Gerig, *Framework for the Statistical Analysis of Brain Structures using SPHARM-PDM*, 11 de Julio, 2006.
- [15] G. Gerig, M. Styner, D. Jones, D. Weinberger, J. Lieberman, *Shape analysis of brain ventricles using SPHARM*, Diciembre, 2001.
- [16] Athinoula A. Martinos Center for Biomedical Imaging, *FreeSurfer*, <http://surfer.nmr.mgh.harvard.edu>, Noviembre, 2009.
- [17] I. Oguz, G. Gerig, S. Barre, M. Styner, *KWMeshVisu: A Mesh Visualitzation Tool fot Shape Analysis*, 13 de Julio, 2006.
- [18] Brian Fox, Chet Ramey, Berkeley University, Bash: *Bourne-Again Shell*, <http://es.wikipedia.org/wiki/Bash>.
- [19] B. Joy, *C Shell*, [http://es.wikipedia.org/wiki/C\\_Shell](http://es.wikipedia.org/wiki/C_Shell).
- [20] *Enfermedad de Parkinson*, <http://es.wikipedia.org/wiki/Parkinson>.
- [21] *Hospital de la Santa Creu i Sant Pau*, <http://www.santpau.es>.
- [22] DICOM, *Digital Imaging and Communication In Medicine*, <http://medical.nema.org>.
- [23] NIFTI, *Neuroimaging Informatics Technology Initiative*, <http://nifti.nih.gov/>.
- [24] S.M. Smith, M. Jerkinson, H. Johansen-Berg, D. Rueckert, T.E. Nichols, C.E. Mackay, K.E. Watkins, O. Ciccarelli, M. Zaheer Cader, P.M. Matthews, T.E.J. Behrens, *Tract-Based Spatial Statistics: Voxelwise Analysis of Multi-Subject Diffusion Data* 2006

- [25] T.E.J. Behrens, H. Johansen-Berg, S. Jbabdi, M.F. Rushworth, M.W. Woolrich, *Probabilistic diffusion tractography with multiple fibre orientations: What can we gain?*, 2007.
- [26] T.E. Nichols, A.P. Holmes, *Nonparametric Permutation Tests for Functional Neuroimaging: A Primer with Examples*, 2002.
- [27] SGE, *Sun Grid Engine*, <http://www.sun.com/software/sge>.
- [28] *Síndrome de Cushing*, [http://es.wikipedia.org/wiki/Síndrome\\_de\\_Cushing](http://es.wikipedia.org/wiki/Síndrome_de_Cushing).
- [29] The MathWorks, *Matlab*, <http://www.mathworks.com>, Abril, 2010.
- [30] ICBM, *International Consortium of Brain Mapping*, <http://www.loni.ucla.edu/ICBM>.
- [31] J. Ashburner, *A Fast Diffeomorphic Image Registration Algorithm*, Julio, 2007.
- [32] Vóxel, *Volumetric Pixel*, <http://es.wikipedia.org/wiki/Vóxel>, Mayo 2010.
- [33] Qemu: *Open Source Processor Emulator*, <http://wiki.qemu.org>.
- [34] KVM, *Kernel-based Virtual Machine*, <http://www.linux-kvm.org>.
- [35] SELinux: *Security-Enhanced Linux*, <http://www.nsa.gov/research/selinux/index.shtml>.
- [36] Matlab Compiler: *Build standalone executables and software components from Matlab code*, <http://www.mathworks.com/products/compiler>.

# Anexo A

## Casos de uso

### **Log-in**

Descripción: Permite entrar en el sistema.

Actores: Usuario de Internet.

Pre-Condiciones:

Flujo Normal:

- El usuario hace clic en “log-in”.
- El usuario introduce su nombre y contraseña.
- Aprieta al botón “Submit”.
- El sistema comprueba los datos.
- El usuario accede al sistema.

Flujo Alternativo:

- En caso de que los datos no sean correctos, el sistema nos avisa.
- Si el usuario no recuerda la contraseña puede usar el apartado “recordar contraseña”.
- Si el usuario no está registrado, puede ir al apartado “Not a user? Sign in!”.

Post-Condiciones: Mostrar el menú correspondiente y mostrar un mensaje de bienvenida.

### **Iniciar sesión NX**

Descripción: El cliente NX se utiliza para poder utilizar el visualizador que se encuentra en la máquina servidor.

Actores: Administrador y usuario con permisos.

Pre-Condiciones:

- Tener una sesión iniciada.
- Tener permisos para acceder a ese proyecto.

Flujo Normal:

- El usuario accede a la sesión de NX mediante el botón “Continue”.
- Se inicia la sesión y se muestra el objeto cargado.

Flujo Alternativo:

- El usuario aprieta “Cancel” y vuelve al menú principal.

Post-Condiciones:



### **Post-procesamiento**

Descripción: El usuario accede al menú de post-procesamiento de imágenes.

Actores: Administrador y usuario con permisos para procesar.

Pre-Condiciones:

- Tener una sesión iniciada

Flujo Normal:

- Se muestran las diferentes posibilidades de post-procesamiento.
- El usuario hace clic en el tipo de post-procesamiento que quiere utilizar.

Flujo Alternativo:

- Si el usuario elige Freesurfer se ejecutará el flujo “Post-procesamiento con Freesurfer”.
- Si el usuario elige LCModel se ejecutará el flujo “Post-procesamiento con LCModel”.
- Si el usuario elige SPM se ejecutará el flujo de “Post-procesamiento con SPM”.
- Si el usuario elige SPHARM se ejecutará el flujo “Post-procesamiento con SPHARM”.
- Si el usuario elige FSL se ejecutará el flujo “Post-procesamiento con FSL”.
- Si el usuario elige ITK-Snap se ejecutará el flujo “Post-procesamiento con ITK-Snap”.

Post-Condiciones:

### **Post-procesamiento con Freesurfer**

#### **Processing**

Descripción: Este apartado nos permite realizar una acciones de post-procesamiento con el programa Freesurfer.

Actores: Administrador y usuario con permisos.

Pre-Condiciones:

- Tener una sesión iniciada.
- Tener permisos sobre el proyecto.

Flujo Normal:

- Se nos muestran unas pestañas cada una con una parte del procesamiento con Freesurfer.
- Por defecto se nos carga la pestaña de procesamiento, el primer paso.
- El actor rellenará todos los parámetros que se nos piden correspondientes al post-procesamiento.
- Hace clic sobre el botón “Process”.
- Se ejecutará un script de Freesurfer con los parámetros anteriores.

Flujo Alternativo:

- En la pestaña “Processing” se nos da la opción de cargar una lista de sujetos para automatizar el proceso de selección de sujetos.
- Si accedemos a la pestaña “Manual Modifications” se ejecutará el flujo descrito seguidamente.
- Si se accede a la pestaña “Export Results”, indicando el proyecto y el tipo (ASEG, LH, RH) procesado con Freesurfer se puede exportar un fichero con los resultados deseados.

- Si se accede a la pestaña QDEC, tras rellenar los datos necesarios, haciendo clic en el botón “Run QDEC”, se abrirá el visualizador.

Post-Condiciones: En la pestaña “Processing”, si los sujetos que se van a procesar ya están procesados, se nos mostrará un mensaje de advertencia.

### **Modificaciones manuales**

Descripción: En este caso se permite aplicar una serie de funcionalidades a sujetos que han sido previamente procesados con Freesurfer y que requieren alguna modificación.

Actores: Administrador y usuario con permisos.

Pre-Condiciones:

- Tener una sesión iniciada.
- Tener permisos sobre el proyecto.

Flujo Normal:

- El usuario debe seleccionar un proyecto y un sujeto.
- Desde aquí se tienen varias posibilidades.

Flujo Alternativo:

- Se puede elegir entre hemisferios del cerebro y abrir el visor TK-Surfer mediante una sesión NX.
- Se puede abrir el visor TK-Medit mediante NX.
- Si el usuario selecciona la casilla “White Matter Modifications” se lanzará un script con los datos de proyecto y sujeto seleccionados.
- Si el usuario selecciona la casilla “Pial Modifications” se lanzará un script con los datos de proyecto y sujeto seleccionados.
- Si el usuario selecciona la casilla “Control Point Modifications” se lanzará un script con los datos de proyecto y sujeto seleccionados.

Post-Condiciones:

### **Post-procesamiento con SPHARM-PDM**

#### **Extract Shape**

Descripción: Este escenario muestra las opciones necesarias para realizar la segmentación de las estructuras subcorticales.

Actores: Administrador y usuario con permisos.

Pre-Condiciones:

- Tener una sesión iniciada.
- Tener permisos sobre el proyecto.

Flujo Normal:

- Se nos muestran unas pestañas cada una con una parte del procesamiento con SPHARM.

- La primera pestaña es la primera fase del proceso y se nos muestra por defecto.
- El usuario debe elegir un proyecto en el que trabajar, la estructura subcortical y el conjunto de sujetos sobre el que quiera trabajar para realizar la primera parte del proceso.
- Una vez haga clic sobre el botón “Extract!” se lanzará un script que procesará las estructuras deseadas.

Flujo Alternativo:

- Si el usuario se dirige a la pestaña “Post-Process” se ejecutará el flujo correspondiente al Post-Procesamiento.
- Si el usuario se dirige a la pestaña “Statistics” se ejecutará el flujo correspondiente al proceso estadístico.
- Si el usuario hace clic en la pestaña “KWMeshVisu” se le abrirá una sesión de NX con el visualizador KWMeshVisu para ver los resultados obtenidos.

Post-Condiciones:

### **Post-Process**

Descripción: En este escenario es donde se realiza el post-procesamiento de las imágenes de cada estructura subcortical.

Actores: Administrador y usuario con permisos.

Pre-Condiciones:

- Tener una sesión iniciada.
- Tener permisos sobre el proyecto.

Flujo Normal:

- El usuario debe seleccionar un proyecto sobre el que trabajar y después la estructura que desea analizar y si quiere analizar varias estructuras o no.
- Se muestran cuatro parámetros con unos valores por defecto. Se dispone de una explicación sobre cada parámetro.
- Después de haber cambiado o no estos parámetros, se aprieta sobre el botón “Process!” y se ejecutarán una serie de scripts para generar los archivos necesarios para seguir con el estudio estadístico.

Flujo Alternativo:

- Si el usuario se dirige a la pestaña “Statistics” se ejecutará el flujo correspondiente al proceso estadístico.
- Si el usuario hace clic en la pestaña “KWMeshVisu” se le abrirá una sesión de NX con el visualizador KWMeshVisu para ver los resultados obtenidos.

Post-Condiciones:

### **Statistics**

Descripción: En este escenario es donde se realiza el procesamiento estadístico correspondiente a SPHARM.

Actores: Administrador y usuario con permisos.

Pre-Condiciones:

- Tener una sesión iniciada.
- Tener permisos sobre el proyecto.

Flujo Normal:

- El usuario debe seleccionar un proyecto sobre el que trabajar y después la estructura que desea analizar.
- Se muestran seis parámetros con unos valores por defecto. Se dispone de una explicación sobre cada parámetro.
- Después de haber cambiado o no estos parámetros, se aprieta sobre el botón “Process!” y se ejecutarán una serie de scripts para generar los resultados del estudio estadístico.

Flujo Alternativo:

- Si el usuario hace clic en la pestaña “KWMeshVisu” se le abrirá una sesión de NX con el visualizador KWMeshVisu para ver los resultados obtenidos.

Post-Condiciones:

### **Post-procesamiento con FSL**

#### **DTIFIT**

Descripción: Este apartado nos permite realizar una acciones de post-procesamiento con el programa SPHARM.

Actores: Administrador y usuario con permisos.

Pre-Condiciones:

- Tener una sesión iniciada.
- Tener permisos sobre el proyecto.

Flujo Normal:

- Se muestran tres pestañas que corresponden a las tres fases de postprocesamiento con FSL.
- Se muestra la primera por defecto, en la que el usuario deberá escoger el proyecto y el conjunto de usuarios con en el que quiere trabajar.
- Una vez se haga clic en “Process!”, se lanzará un script con la conversión del formato de las imágenes y después el tratamiento de las imágenes para poder ser tratadas.

Flujo Alternativo:

- Si solo se quiere procesar un solo sujeto, o un grupo de sujetos que se ha añadido recientemente, se tiene un menú de selección con todos los sujetos para poder elegir cuales se quieren procesar.
- Si el usuario se dirige a la pestaña “Check DTIFIT” se ejecutará el flujo descrito en el caso de uso para comprobar las imágenes procedentes del proceso DTIFIT.
- Si el usuario se dirige a la pestaña “TBSS / Randomise” se ejecutará el flujo correspondiente a la creación de ficheros para el estudio estadístico.
- Si el usuario se dirige a la pestaña “Bedpostx”, se ejecutará el flujo que corresponde al proceso estadístico Bedpostx.
- Si el usuario se dirige a la pestaña “FSL Tools”, se ejecutará el flujo que corresponde a la ejecución de la aplicación gráfica de todas las herramientas de FSL.

Post-Condiciones:

#### **Check DTIFIT**

Descripción: Este apartado nos ofrece la posibilidad de comprobar si los sujetos se han procesado bien, y reprocesarlos con un nuevo valor de umbral por si fuera necesario.

Actores: Administrador y usuario con permisos.

Pre-Condiciones:

- Tener una sesión iniciada.
- Tener permisos sobre el proyecto.

Flujo Normal:

- Se muestra un visualizador de NX y se iniciará una sesión para abrir FSLView y poder comprobar si las imágenes han sido bien procesadas.
- Se crea una tabla con todos los sujetos procesados para que el usuario vaya marcando si se han procesado correctamente o no.
- Si todos los sujetos se han procesado correctamente se presionará sobre el botón "Finish".

Flujo Alternativo:

- En el caso que hayan sujetos que se quiere reprocesar con otro valor de umbral, al presionar sobre el botón "No OK", se nos abrirá un recuadro de texto para indicar el nuevo umbral.
- Si el usuario se dirige a la pestaña "TBSS / Randomise" se ejecutará el flujo correspondiente a la creación de ficheros para el estudio estadístico.
- Si el usuario se dirige a la pestaña "Bedpostx", se ejecutará el flujo que corresponde al proceso estadístico Bedpostx.
- Si el usuario se dirige a la pestaña "FSL Tools", se ejecutará el flujo que corresponde a la ejecución de la aplicación gráfica de todas las herramientas de FSL.

Post-Condiciones:

**TBSS / Randomise**

Descripción: En esta pestaña se realizará todo el estudio estadístico necesario para hacer la comparación de los sujetos con el proceso TBSS y Randomise.

Actores: Administrador y usuario con permisos.

Pre-Condiciones:

- Tener una sesión iniciada.
- Tener permisos sobre el proyecto.

Flujo Normal:

- Se muestra un desplegable con el proyecto a elegir, un espacio para especificar el nombre que se le quiere dar al estudio, la lista de sujetos disponibles de cada grupo y los dos grupos elegidos de los sujetos a procesar.
- Si el usuario hace clic en "Process!" se ejecutará un script con todos los pasos necesarios para la preparación de los ficheros estadísticos.

Flujo Alternativo:

- Si el usuario hace clic en el enlace al paso dos "Step 2", se dirigirá al flujo de ejecución del proceso estadístico Randomise.
- Si el usuario se dirige a la pestaña "Bedpostx", se ejecutará el flujo que corresponde al proceso estadístico Bedpostx.
- Si el usuario se dirige a la pestaña "FSL Tools", se ejecutará el flujo que corresponde a la ejecución de la aplicación gráfica de todas las herramientas de FSL.

Post-Condiciones:

**Bedpostx**

Descripción: En esta pestaña se realizará todo el estudio estadístico necesario para hacer la comparación de los sujetos con el proceso Bedpostx.

Actores: Administrador y usuario con permisos.

Pre-Condiciones:

- Tener una sesión iniciada.
- Tener permisos sobre el proyecto.

Flujo Normal:

- Se muestra una pantalla con el proyecto a elegir.
- Se da la opción de cambiar los parámetros por defecto del proceso Bedpostx.
- Si el usuario hace clic en “Process!” se ejecutará un script con todos los pasos necesarios para hacer el análisis estadístico correspondiente.

Flujo Alternativo:

- Si el usuario se dirige a la pestaña “FSL Tools”, se ejecutará el flujo que corresponde a la ejecución de la aplicación gráfica de todas las herramientas de FSL.

Post-Condiciones:

### **FSL Tools**

Descripción: Se muestra un visualizador NX para abrir el programa FSL.

Actores: Administrador y usuario con permisos.

Pre-Condiciones:

- Tener una sesión iniciada.
- Tener permisos sobre el proyecto.

Flujo Normal:

- Se inicia sesión en NX para abrir FSL.
- Se abrirá un menú con todas las funcionalidades de FSL para poder hacer una visualización, modificación o cualquier otra acción con FSL.

Flujo Alternativo:

Post-Condiciones:

### **SPM con máquinas virtuales**

#### **Batch Editor**

Descripción: Se muestra un visualizador NX para abrir el total de funciones de SPM.

Actores: Administrador y usuario con permisos.

Pre-Condiciones:

- Tener una sesión iniciada.
- Tener permisos sobre el proyecto.

Flujo Normal:

- Se tiene un desplegable para seleccionar el proyecto y un espacio en blanco para escribir el nombre que se le quiere dar al estudio.
- Al hacer click en “Create directory” se creará todo el árbol de carpetas y subcarpetas necesarias para realizar el estudio.
- Se mostrará un pop-up avisando de los paths en los que se deben guardar los Jobs.
- Se abrirá una sesión de NX con el Batch Editor de SPM donde poder crear el Job a ejecutar.

Flujo Alternativo:

- Si el usuario se dirige a la pestaña “Process Batch”, se ejecutará el flujo que corresponde al procesado con SPM.
- Si el usuario se dirige a la pestaña “SPM Tools”, se ejecutará el flujo que corresponde a la ejecución de la aplicación gráfica de todas las herramientas de SPM.

Post-Condiciones:

### **Process Batch**

Descripción: Se muestra un visualizador NX para abrir el total de funciones de SPM.

Actores: Administrador y usuario con permisos.

Pre-Condiciones:

- Tener una sesión iniciada.
- Tener permisos sobre el proyecto.

Flujo Normal:

- Se tiene un desplegable para seleccionar el proyecto y otro para escoger al estudio que se quiere procesar.
- Cuando el usuario haga click sobre el boton “Process”, se ejecutarán los scripts de edición del Job y se procesarán los datos.

Flujo Alternativo:

- Si el usuario se dirige a la pestaña “SPM Tools”, se ejecutará el flujo que corresponde a la ejecución de la aplicación gráfica de todas las herramientas de SPM.

Post-Condiciones:

### **SPM Tools**

Descripción: Se muestra un visualizador NX para abrir el total de funciones de SPM.

Actores: Administrador y usuario con permisos.

Pre-Condiciones:

- Tener una sesión iniciada.
- Tener permisos sobre el proyecto.

Flujo Normal:

- Se inicia sesión en NX para abrir SPM.
- Se abrirá un menú con todas las funcionalidades de SPM para poder acceder al total de opciones que ofrece SPM.

Flujo Alternativo:

Post-Condiciones:

# Índice de figuras

3.1. Planificación inicial . . . . .	14
4.1. Arquitectura de la plataforma . . . . .	17
4.2. Esquema interfaz post-procesamiento . . . . .	18
4.3. Diagrama de casos de uso . . . . .	19
5.1. Esquema SPHARM . . . . .	21
5.2. Esquema SPHARM.sh . . . . .	23
5.3. Extract Shape . . . . .	26
5.4. Post-Procesamiento con SPHARM . . . . .	26
5.5. Creación estudio estadístico . . . . .	26
5.6. Visualización con KWMeshVisu . . . . .	27
5.7. Esquema FSL . . . . .	31
5.8. DTIFIT . . . . .	35
5.9. Comprobar y reprocesar DTIFIT . . . . .	36
5.10. Procesamiento TBSS . . . . .	36
5.11. BedpostX . . . . .	37
5.12. FSL Tools . . . . .	37
5.13. Esquema SPM . . . . .	42
5.14. Estructura SPM . . . . .	47
5.15. Creación nuevo proyecto SPM . . . . .	49
5.16. Aviso SPM . . . . .	49
5.17. SPM Batch Editor . . . . .	50
5.18. Process Batch Jobs . . . . .	50



5.19. ITK-Snap . . . . . 55

# Índice de cuadros

5.1. Resultados SPHARM-PDM para 94 pacientes . . . . .	29
5.2. Resultados DTIFIT + TBSS para 53 pacientes . . . . .	39
5.3. Resultados Bedpostx para 123 pacientes . . . . .	40
5.4. Comparativa entre máquinas para un solo paciente . . . . .	53
5.5. Comparativa tiempos 1 CPU vs. 2 CPU . . . . .	54
5.6. Comparativa entre mejores versiones de cada modelo . . . . .	54

Signat:

**Jordi Casals Hernández**

## **RESUMEN**

Con la creciente generación de resonancias magnéticas, los servicios de radiología necesitan aplicaciones que les faciliten el trabajo de acceso remoto a los datos y a las herramientas que utilicen para la extracción de datos para realizar sus diagnósticos. El objetivo de este proyecto es el de estudiar e integrar en la plataforma web del grupo de Imagen Médica del PIC llamada PICNIC (*PIC NeuroImaging Center*) un conjunto de aplicaciones para el estudio y procesamiento de neuroimagen con la implementación de herramientas software en la plataforma grid del PIC.

## **RESUM**

Amb la creixent generació de resonàncies magnètiques, els serveis de radiologia necessiten aplicacions que els hi facilitin les tasques d'accés remot a dades i a les eines que utilitzen per a l'extracció de dades per realitzar els seus diagnòstics. L'objectiu d'aquest projecte es el d'estudiar i integrar en la plataforma web del grup d'Imatge Mèdica del PIC, anomenat PICNIC (*PIC NeuroImaging Center*), un conjunt d'aplicacions per a l'estudi i processament de neuroimatge amb l'implementació d'eines software en la plataforma grid del PIC.

## **ABSTRACT**

With the rising generation of magnetic resonance imaging, radiology services need applications that will facilitate the work of remote access to data and tools used for the extraction of data for diagnosis. The objective of this project is to study and integrate into the PIC web platform of the Medical Imaging group called PICNIC (*PIC NeuroImaging Center*) a set of applications for study and neuroimaging processing tools with the implementation of software tools into the PIC grid platform.