



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**“DISEÑO E IMPLEMENTACIÓN EN PLONE  
DE UN SISTEMA DE MANEJO DE  
SOLICITUDES DE PRESUPUESTO MEDIANTE  
FLUJOS DE TRABAJO”**

**T E S I S**

QUE PARA OBTENER EL GRADO DE:

**MAESTRO EN INGENIERÍA  
(COMPUTACIÓN)**

**P R E S E N T A:**

**ARTURO TLACAELEL CURIEL DÍAZ**

**DIRECTOR DE LA TESIS: DR. SERGIO RAJSBAUM GORODEZKY**

**MÉXICO, D.F.**

**2011.**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



# Índice general

<b>1. Introducción</b>	<b>5</b>
1.1. Descripción del Problema . . . . .	7
1.2. Objetivos . . . . .	9
1.2.1. Objetivo General . . . . .	9
1.2.2. Objetivos Particulares . . . . .	9
1.3. Descripción General del Sistema . . . . .	10
1.4. Estructura de la Tesis . . . . .	11
<b>2. Revisión de Literatura y Conceptos Básicos</b>	<b>13</b>
2.1. Contenidos en sistemas CMS . . . . .	13
2.1.1. Información <i>meta</i> e información cruda . . . . .	14
2.2. Workflows . . . . .	15
2.3. Python . . . . .	15
2.4. Frameworks . . . . .	16
2.4.1. Zope . . . . .	17
2.4.2. Plone . . . . .	18
<b>3. Descripción Técnica</b>	<b>29</b>
3.1. Tipos de Dato del Sistema . . . . .	29
3.1.1. Solicitud de Licencia (meta-tipo Solicitud) . . . . .	30
3.1.2. Solicitud de Becario (meta-tipo SolicitudBecario) . . . . .	37
3.1.3. Solicitud de Visitante (meta-tipo SolicitudVisitante) . . . . .	38
3.1.4. Folder de Solicitudes (meta-tipo SolicitudFolder) . . . . .	38
3.2. Operaciones básicas del sistema . . . . .	40
3.2.1. Crear nuevas solicitudes . . . . .	41
3.2.2. Revisión y Modificación de Solicitudes . . . . .	42
3.2.3. Cambio de Estado . . . . .	43
3.2.4. Importar Solicitudes . . . . .	43
3.2.5. Exportar Solicitudes . . . . .	44
3.2.6. Establecer Presupuesto . . . . .	44
3.3. Casos de Uso del Sistema . . . . .	45
3.4. Clases del Sistema . . . . .	46
3.4.1. La clase <i>Solicitante</i> . . . . .	47
3.4.2. La interfaz <i>ISolicitud</i> . . . . .	49
3.4.3. Estructura de las clases de solicitudes . . . . .	52

3.4.4.	La clase <i>SolicitudFolder</i> . . . . .	56
3.5.	Flujos de Información Del Sistema . . . . .	65
3.5.1.	Flujo de trabajo para solicitudes de Licencia (meta-tipo Solicitud) y Visitante (meta-tipo SolicitudVisitante) . . . . .	65
3.5.2.	Flujo de trabajo para Solicitudes de Becario (meta-tipo SolicitudBecario) . . . . .	72
3.5.3.	Flujo de trabajo para Folder de solicitudes . . . . .	79
3.6.	Relación entre las actividades del sistema . . . . .	84
3.7.	Requerimientos Mínimos . . . . .	85
<b>4.</b>	<b>Diseño Visual e Interfaz de Usuario</b>	<b>87</b>
4.1.	Crear nuevas solicitudes . . . . .	87
4.2.	Revisar Solicitudes Pendientes . . . . .	90
4.3.	Vista general y exportación de solicitudes . . . . .	98
4.4.	Importar Solicitudes . . . . .	99
4.4.1.	Importación Básica . . . . .	99
4.4.2.	El rol de <i>Importador de Solicitudes</i> . . . . .	102
4.5.	Manejo de Presupuesto . . . . .	106
4.5.1.	Presupuesto Total Local . . . . .	106
4.5.2.	Presupuesto Total Global . . . . .	108
4.6.	Folder de Solicitudes . . . . .	110
4.6.1.	Modificación de Intervalos y Archivado . . . . .	113
4.7.	Portlet Avisos . . . . .	115
4.7.1.	Contexto “Folder” . . . . .	115
4.7.2.	Contexto “Solicitud” . . . . .	116
<b>5.</b>	<b>Conclusiones</b>	<b>119</b>
5.1.	Sobre el proceso administrativo . . . . .	119
5.2.	Sobre la interacción con los usuarios . . . . .	120
5.3.	Problemas Resueltos Sobre el Diseño . . . . .	120
5.4.	Comunicación Con Otros Sistemas y Trabajo Futuro . . . . .	121

# Capítulo 1

## Introducción

La creciente capacidad de los sistemas de cómputo modernos para crear y distribuir información ha traído consigo diversas problemáticas sobre la forma en que éstos manejan sus datos; para empezar, ¿Cuál es la mejor manera de organizarlos? ¿Quién tiene acceso a ellos? ¿Cómo manejamos automáticamente la información introducida por cientos de usuarios? O, en otras palabras, ¿cómo hacemos eficientes y seguros nuestros sistemas de información?

En este contexto se creó el concepto de *contenido* que, por convención, se entiende como una unidad de información de alguna clase que se puede, efectivamente, manejar y transmitir electrónicamente como si fuera un bien u objeto físico [10]. Por extensión, se ha dado en llamar *Sistema de Gestión de Contenidos* (CMS por sus siglas en inglés) a los sistemas de cómputo que permiten manipular contenidos.

En general, el sistema CMS más simple lleva a cabo solo tres actividades principales sobre los contenidos [16]:

- Manejo de Recursos: Mueve y organiza contenidos existentes. Esta es la parte principal de todo CMS ya que requiere un análisis de contenidos sobre el cual se centran los procesos que modifican la información (los *workflows*). Esto incluye la estructura sobre la cual reposan los contenidos, la creación de nueva información y la definición de los actores del sistema.
- Transformación: Transforma el contenido de forma que sea presentable. ¿Cómo debe presentarse un contenido? ¿Se deben desplegar solo los metadatos? ¿Cómo se despliega la información? Esta actividad se centra en la definición externa de los contenidos, lo que incluye la presentación final en cada una de las etapas del flujo de trabajo.
- Publicación: Poner a disposición del público el contenido. Esta etapa se centra en atender las peticiones externas de los usuarios del sistema para acceder a la información. Los problemas más importantes a los que se enfrenta la actividad de publicación son vitalidad, disponibilidad, escalabilidad, interfaces, accesibilidad, tolerancia a fallos y seguridad.

Mediante estas actividades, un CMS normalmente crea y organiza información nueva, prepara la información para ser desplegable y posteriormente acepta o rechaza pe-

ticiones de modificación o despliegue que son regidas por una política de seguridad [16].

La popularidad de los CMS, ha permitido un desarrollo constante de éstos sobre varias plataformas. La mayoría tienen el único fin de facilitar la interacción entre un grupo de usuarios dentro de un ambiente común que les ofrece la capacidad de añadir y administrar contenidos de muchos tipos. De esta forma, la definición de contenidos particulares para casos específicos se ha vuelto una forma popular de modelar situaciones comunes en los negocios, tales que faciliten la interacción del sistema con los usuarios y el posterior manejo de la misma [12].

A principios de la década de los noventa, el costo por la construcción de nuevos CMS era de millones de dólares [12]. Por esta razón varias organizaciones tanto lucrativas como no lucrativas (entre ellas la comunidad de software libre), se han dado a la tarea de fabricar y proporcionar una serie de soluciones genéricas que cubren el enfoque principal de los CMS por un costo relativamente bajo, pero que a su vez faciliten la creación de instancias particulares de dichos sistemas para el manejo de problemas específicos.

Algunos de los principales sistemas CMS más populares que se conocen, según las empresas que los patrocinan, son:

- De código abierto; no comerciales:
  - Wordpress
  - Joomla
  - Drupal
  - Plone
  
- De código cerrado; comerciales:
  - Microsoft Office SharePoint Server
  - IBM Lotus Web Content Management
  - Vignette Content Management

Los CMS son constantemente utilizados para el manejo y administración de recursos humanos dentro de instituciones públicas y privadas. Sobre todo en el ámbito educativo, es importante contar con un sistema de manejo de información que no sea puramente transaccional y basado artículos, sino que permita a la audiencia del sitio manipular información no homogénea que se puede generar desde los diversos sectores que componen la estructura orgánica de los institutos [12].

Los sistemas de gestión de contenidos normalmente se dividen en categorías de acuerdo a su función principal [14]:

- Sistemas de Gestión de Contenido Empresarial (ECMS): son sistemas que hacen énfasis en la extensión que abarcan. Son usados para manejar todos los aspectos del proceso de publicación en organizaciones y generalmente poseen muchas características de otros tipos de CMS.

- Sistema de Gestión de Contenidos *Web* (WCMS): los sistemas WCMS se centran en el manejo de contenidos a través de la red.
- Sistemas de Gestión de Documentos (DMS): estos sistemas solo manejan documentos de texto y están diseñados para centrarse en el manejo interno de información y no tanto en la publicación vía internet.
- Sistema de Gestión de Derechos Digitales (DRMS): se encargan de gestionar contenidos protegidos por derechos de propiedad intelectual y toman la seguridad de dichos derechos como una prioridad.
- Sistemas de Gestión de Recursos Digitales (DAMS): manejan solamente contenidos de tipo binario, *i.e.* no manejan documentos sino imágenes, videos, programas, etc.

El sistema de interés para este trabajo es Plone. Plone es un sistema de gestión de contenidos desarrollado en Python. Dada su versatilidad es muy popular a nivel mundial y se puede clasificar en todas las categorías antes mencionadas, lo que lo hace ideal para administración de recursos.

Se eligió Plone por que su uso está muy extendido tanto dentro de empresas privadas como en entidades públicas e instituciones educativas. Se ha utilizado para construir portales, sitios web corporativos, sitios de noticias, servidores de *extranet* o *intranet*, sistemas de publicación, repositorios de documentos, como herramienta colaborativa (*groupware*) y para comercio en línea (*e-commerce*) [19]; podría llegar a ser el modelo genérico de CMS que rijan el funcionamiento de futuros sistemas. Otro factor importante en la elección de Plone es que es de código abierto, lo que permite un mejor análisis de las operaciones y funcionamiento del sistema. Adicionalmente, existe más de un grupo de desarrollo con experiencia en el sistema dentro de la UNAM, lo cual enriquece la investigación con la experiencia del trabajo realizado en diferentes proyectos locales de diversa índole.

Los CMS, y en particular Plone, trabajan de manera muy cercana con el concepto de *flujo de trabajo* o *workflow*. Un *workflow* es una serie de interacciones que deben ocurrir para completar una tarea. Normalmente se compone de un conjunto de estados y un conjunto de transiciones entre ellos. Es por medio de *workflows* que un CMS como Plone controla la interacción de los usuarios con los contenidos.

## 1.1. Descripción del Problema

El Instituto de Matemáticas de la UNAM (IMATE), es uno de los centros de investigación que actualmente utilizan Plone para sus tareas de administración de recursos humanos e información académica. Desde 2006 se inició el proyecto InfoMatem, una aplicación *web* basada en Plone que almacena la información de los miembros del Instituto, así como una gran cantidad de reportes generados automáticamente que permiten el manejo eficiente de los datos que se introducen año con año. El sitio se encuentra alojado dentro de las computadoras del Instituto y es manejado por miembros del IMATE [9].



Hasta el año 2010, InfoMatem no tenía forma de manejar solicitudes de presupuesto para viáticos y transporte de los miembros del Instituto. Se necesitaba un *software* que utilizara los datos suministrados por InfoMatem para permitir a los investigadores, técnicos académicos y estudiantes registrados crear solicitudes electrónicas mediante la interfaz de InfoMatem. La información recabada, debía emular el proceso básico que normalmente siguen las solicitudes de recursos dentro del instituto:

1. Un miembro del Instituto llena un formulario estándar de solicitud de licencia o recursos con sus datos.
2. La solicitud es entregada, junto con cualquier otro documento necesario, a la Secretaría Académica del Instituto, donde se revisa y se turna a la Comisión Especial del Consejo Interno.
3. La Comisión Especial del Consejo Interno recibe la solicitud, los miembros la evalúan y deciden si procede o no. Si procede, se envía al Consejo Interno para ser sancionada y debidamente registrada en la reunión semanal del mismo.
4. El Consejo Interno en sesión revisa y, en su caso, aprueba o rechaza la solicitud.
5. Se notifica a quien hizo la solicitud si ésta fue aprobada o rechazada.

En la tesis [3] se presenta la implementación de un producto de Plone destinado a resolver el problema descrito, sin embargo dicho *software* nunca fue desplegado como parte del sitio. La reciente migración de InfoMatem a una nueva versión de Plone provocó que el producto quedara obsoleto antes de poder instalarse, dificultando cualquier intento de futuras inclusiones en la nueva versión. Más aún, el *software* anterior había solo considerado el proceso de aprobación y no hacía ninguna operación con los datos recabados por la aplicación. Finalmente, el cambio en el diseño de las vistas del sitio principal afectó la estética general de InfoMatem, que ya no concordaba con la forma en que se habían implementado las vistas originales del producto (ni visualmente ni técnicamente). Por estas razones, se decidió rediseñar y reimplementar la aplicación. Las características que debía tener el nuevo producto, aunadas al proceso ya mencionado, son:

- Debe llevar un control automático de las cantidades aprobadas y ejercidas para cada investigador, técnico y becario del Instituto.
- Los encargados del presupuesto del Instituto deben ser capaces de revisar, en tiempo real, el presupuesto que ha sido utilizado y el presupuesto que resta para lo que queda del año. También deben tener la posibilidad de cambiar y establecer nuevos presupuestos.
- Las solicitudes se deben poder manejar por periodos independientes con su presupuesto individual, de forma que al terminar cada año fiscal se puedan archivar de forma conveniente y que no interfieran con la contabilidad de periodos futuros.

- Los miembros del Instituto deben ser capaces de revisar, en tiempo real, los totales de las cantidades que les han sido aprobadas, así como la información asentada en las solicitudes que han enviado y sus respectivos estados en el proceso.
- Debe tener funciones de importación y exportación de las solicitudes en formatos estándares.
- La interfaz y la implementación debe coincidir con la nueva arquitectura de InfoMatem.

## 1.2. Objetivos

En esta sección se enuncian los objetivos planteados para la creación del producto *Solicitudes*. El resultado final del *software* estuvo fuertemente ligado a la realización de los mismos.

### 1.2.1. Objetivo General

Diseñar e implementar un producto basado en Plone que permita manipular eficientemente la creación, revisión y aprobación o rechazo de solicitudes de recursos de viáticos y transporte para investigadores, becarios y técnicos académicos en el Instituto de Matemáticas en la UNAM, utilizando la información almacenada en el sistema InfoMatem.

### 1.2.2. Objetivos Particulares

Como objetivos adicionales:

- Facilitar la tarea de los administradores para llevar la contabilidad del Instituto de Matemáticas.
- Permitir la migración sencilla de las solicitudes de un sistema a otro.
- Auxiliar en la tarea de recabar datos para la generación de reportes mediante el sistema InfoMatem.
- Mantener a los usuarios informados sobre el estado de las solicitudes y sobre la cantidad de recursos que les ha otorgado el Instituto.
- Otorgar un mecanismo de organización de solicitudes que facilite la búsqueda y guardado de las mismas.
- Dotar a los administradores la documentación necesaria para expandir el sistema a otros procesos, no solo para solicitudes de viáticos.

### 1.3. Descripción General del Sistema

Para la implementación del sistema, se utilizó la API de Plone con el fin de crear varios tipos de dato específicos que llenaran los requisitos necesarios para manipular la información y generar los formularios desplegados a los usuarios finales. Se definieron tres tipos de solicitud básicos y un contenedor:

**Solicitud de Licencia (meta-tipo Solicitud)** Es el formato que se debe llenar en caso de que el investigador pida presupuesto de viaje (incluye viáticos, gastos de transporte y de inscripción a eventos).

**Solicitud de Visitante (meta-tipo SolicitudVisitante)** Un investigador lo debe llenar en caso de tener un invitado de otra institución académica (nacional o extranjera) para el cual se requiera presupuesto (incluye viáticos y gastos de transporte).

**Solicitud para Becarios (meta-tipo SolicitudBecario)** Es un formato para becarios que piden presupuesto de viaje (incluye viáticos, gastos de transporte y de inscripción a eventos).

**Folder de solicitudes (meta-tipo SolicitudFolder)** Es un tipo de contenedor que cumple la función de ser un receptor de solicitudes y, a la vez, representa conceptualmente un periodo de tiempo en el cual se puede asignar una cantidad máxima de presupuesto para los investigadores, becarios y técnicos académicos. Los folders contienen diferentes pestañas que corresponden a listas de solicitudes en determinado estado del proceso de aprobación o rechazo.

El producto Solicitudes, además, contiene los siguientes roles de usuario que se adaptan a la situación específica del IMATE:

**Investigador** Es un rol diseñado para ser asignado a los investigadores del Instituto. Éste rol tiene permisos para añadir solicitudes de Licencia y de Visitante, además de que permite la importación de solicitudes propias en texto simple hacia el sistema.

**Técnico Académico** Es un rol diseñado para los técnicos académicos del Instituto. Al igual que el de investigador, este rol tiene permisos para añadir solicitudes de Licencia y de Visitante. También permite la importación de solicitudes propias en texto simple hacia el sistema.

**Becario** Es un rol para los becarios del instituto. Éste rol tiene permisos para añadir solicitudes de Becario solamente. Puede importar hacia el sistema solicitudes de becario en texto simple.

**Comisionado** Es un rol que se debe asignar a los miembros de la Comisión Especial del Consejo Interno del Instituto que se encarga de revisar las solicitudes de presupuesto para viajes, pero no pueden enviarlas al Consejo Interno.

**Responsable de la Comisión** Es un rol que tiene las mismas atribuciones que “Comisionado” pero, a diferencia de éste, posee la capacidad de enviar las solicitudes revisadas al Consejo Interno o rechazarlas.

**Consejero** Éste rol se asigna a los miembros del Consejo Interno del Instituto. Pueden modificar la información del Consejo pero no puede aprobar ni rechazar solicitudes.

**Responsable del Consejo** Éste rol tiene las mismas atribuciones que “Consejero” pero, a diferencia de éste, puede aprobarlas, rechazarlas y, en su caso, eliminarlas.

**Programador de Presupuesto** Es un rol que permite controlar las funciones de contabilidad del producto. Se recomienda asignárselo al Secretario Académico, al Director y al encargado de presupuestos del Instituto.

**Importador de Solicitudes** Es un rol especial que permite hacer importaciones de solicitudes de texto simple hacia el sistema. A diferencia de las capacidades de importación básica, un *Importador de Solicitudes* puede importar solicitudes a nombre de cualquier investigador, técnico académico o becario e insertarlas en cualquier parte del proceso (borrador, revisión por la Comisión, revisión por el Consejo Interno o aprobada). Éste rol se recomienda asignarlo solo a petición de alguna autoridad a personal específico, ya que otorga la capacidad para aprobar las solicitudes importadas sin revisión previa. El objetivo de este rol es puramente administrativo: permitir la introducción tanto de solicitudes de último momento como de solicitudes que ya han sido sancionadas pero que aún no están en el sistema.

## 1.4. Estructura de la Tesis

En el capítulo 2 se introducen brevemente las tecnologías utilizadas y los conceptos básicos necesarios para comprender el diseño del producto. Aquí se otorga una descripción de Plone y el framework Zope, así como los cambios en la arquitectura de Plone 2 a Plone 3 y algunos detalles sobre la estructura de contenidos en este CMS.

La descripción técnica del producto se presenta en el capítulo 3. En este capítulo se da una explicación detallada del funcionamiento del sistema. Se presenta el diseño de la aplicación, la definición de tipos de dato, los flujos de trabajo, la descripción del proceso y los requerimientos técnicos del *software*.

El capítulo 4 presenta información sobre el diseño de vistas y la interfaz de usuario del sistema. Aquí se presentan tanto tomas de pantalla como detalles sobre la operación a nivel de usuario.

Finalmente, el capítulo 5 presenta las conclusiones obtenidas al llevar a cabo el despliegue del producto junto a la nueva versión de InfoMatem.



# Capítulo 2

## Revisión de Literatura y Conceptos Básicos

En este capítulo se introducen brevemente las tecnologías utilizadas y los conceptos básicos necesarios para comprender el diseño del producto.

Antes que nada, se dará una definición de contenidos en Plone, los conceptos de *workflow* y *framework* así como una breve introducción al Zope y Plone.

### 2.1. Contenidos en sistemas CMS

Los datos son pequeños pedazos de información que los sistemas recolectan y procesan. Un *contenido* puede ser definido simplemente como “un conjunto de datos”, aunque este conjunto debe ser caracterizado en un nivel superior donde la información tiene un significado específico como:

- Texto
- Sonido
- Imágenes
- Movimiento
- Archivos
- Etc.

Una definición más completa de contenido, entonces, puede ser la de *información aplicada a un propósito* [4]. Un contenido se puede ver como información cruda a la cual se le asignan datos adicionales que nos permiten caracterizarla y utilizarla para algún propósito concreto. El valor de un contenido entonces se basa en la combinación de su forma utilizable principal, junto con su aplicabilidad, accesibilidad, uso, utilidad e identidad única [4].

Un contenido normalmente se define en términos de las partes que posee en *metadatos* y *esencia* [10], donde esta última es la información cruda que define las características

principales del contenido mientras que los metadatos conforman la información adicional que describe de alguna forma a los datos crudos.

### 2.1.1. Información *meta* e información cruda

Para definir tipos de contenido dentro de un CMS, tenemos que definir los parámetros de la información *meta* para el tipo específico. El sistema utiliza esa información *meta* para dos tareas:

**Catalogar** Se lleva a cabo leyendo los *datos de indexado* para organizar, clasificar y diferenciar contenidos. Llamamos *datos de indexado* a los campos de información *meta* que son completamente independientes de la información cruda, *i.e.* que no describen la organización interna de ésta. Los datos de indexado que usa Plone son:

- Identificador (UID).- Es una clave única que Plone asigna a cada contenido.
- Creador.- Consiste en un nombre único del usuario que invocó la operación *add* para crear el contenido.
- Fecha de Creación.- Este campo guarda la fecha del sistema en la cual la operación *add* que dio origen al contenido fue exitosa.
- Título.- Es el nombre desplegable, no necesariamente único, del contenido.
- Tipo.- Contiene el nombre del esquema que se utiliza para interpretar la información cruda.

**Interpretar** Consiste en utilizar los *datos de esquema* para construir una descripción esquemática del contenido y su estado actual de acuerdo a la información cruda.

- Esquema.- Contiene una descripción para cada campo dentro del contenido:
  - Nombre del campo.- Es un identificador único para cada campo dentro del contenido.
  - Tipo de campo.- Especifica el tipo de dato que el campo recibe.
  - Orden relativo.- Indica en que posición se encuentra un campo con respecto a los otros.

La información cruda se compone de una cadena de bits que solo puede interpretarse utilizando la información *meta*. La estructura de esta cadena está dada por el esquema proporcionado en la información *meta*.

Mediante una descripción del tipo de dato, que llamaremos *esquema*, Plone puede definir automáticamente los datos *meta* de un tipo nuevo y, mediante formularios, permite almacenar la información cruda que introduce el usuario. Para la creación de los tipos adicionales, se definieron esquemas nuevos que describen los tipos usados por *Solicitudes*. Plone automáticamente genera vistas asociadas a cada instancia del nuevo tipo de dato, de forma que los usuarios tengan acceso inmediato a éstos y permitan introducir la información cruda de acuerdo a lo que marca el esquema.

## 2.2. Workflows

Un *flujo de trabajo* o *workflow* es una serie de interacciones que deben ocurrir para completar una tarea. Se compone de un conjunto de estados y un conjunto de transiciones entre dichos estados. Normalmente define el proceso que deben seguir los contenidos y se aplica directamente a éstos [1].

En un sistema de manejo de contenidos normalmente no se usa uno sino varios *workflows*, debido a que los procesos tienden a cambiar de contenido a contenido y de usuario a usuario. Gran parte de la complejidad del uso de sistemas CMS proviene del diseño e interacción con *workflows* [10].

El diseño de *workflows* está fuertemente ligado al proceso que siguen los contenidos. La descripción de un *workflow*, describe los estados por los cuales el contenido tiene que pasar dadas ciertas acciones.

Aunque no siempre es así, para el producto *Solicitudes* no se definieron cambios de estado automáticos en los *workflows* con el fin de que los usuarios tuvieran el máximo control posible sobre el proceso. Cada estado del *workflow* manda a llamar una serie de operaciones, específicas al estado, que permiten al sistema comunicar información automáticamente con InfoMatem y, en algunos casos, con los usuarios. Es mediante estas operaciones específicas denotadas por el cambio de estado que se mantiene el sistema actualizado en tiempo real. En general, las operaciones se pueden llamar manualmente sin necesidad de los *workflows*. Estas no son necesarias para su definición. Sin embargo, el *software* aquí presentado hace uso específico de *workflows* para llamar operaciones con el fin de mantener la seguridad del sitio y la integridad de los datos.

## 2.3. Python

Python es un lenguaje de programación interpretado de alto nivel y de propósito general. Originalmente, fue diseñado haciendo énfasis en la facilidad de lectura del código. Su biblioteca estándar contiene una amplia gama de estructuras de datos de alto nivel, combinadas con tipo dinámico, lo que lo hace muy atractivo para desarrollo rápido de aplicaciones. Python soporta varios módulos y paquetes, lo que fomenta la modularidad y reusabilidad de código [8].

Python soporta varios paradigmas de programación, siendo los principales de éstos la programación orientada a objetos y la programación imperativa. Además, a diferencia de muchos lenguajes de programación, Python utiliza la indentación para delimitar bloques de código.

Los tipos de dato principales en Python son [13]:

**str** Describe una secuencia de caracteres inmutable.

**bytes** Describe una secuencia inmutable de *bytes*.

**list** Es un tipo de dato que representa una lista ligada. Este tipo dinámico puede contener otros tipos de dato.



**tuple** Es un tipo de dato que representa una lista ligada estática. Puede contener varios tipos de dato dentro.

**set** Es un conjunto no ordenado que no contiene duplicados.

**dict** Es un grupo mutable de datos que tienen asociada una llave única dentro de cada instancia de la estructura.

**int** Es un número de precisión fija y de magnitud ilimitada.

**float** Es un número de punto flotante.

**complex** Representa un número complejo con una parte real y una parte imaginaria.

**bool** Es un tipo de dato que representa un valor binario de verdad: “True” o “False”.

Algunas de las palabras reservadas de Python e instrucciones para control de flujo son [13]:

**if...elif...else** Ejecuta condicionalmente un bloque de código.

**for** Itera sobre un objeto iterable, capturando cada elemento en una variable local para su uso sobre ese bloque de código.

**while** Ejecuta un bloque de código mientras una condición se mantenga como “True”.

**try** Permite que existan excepciones cuando ocurre algún imprevisto en algún bloque de código.

Una característica importante de Python que Plone y otros *frameworks* explotan son los paquetes redistribuibles de Python (o *eggs*). Los *eggs* son clases de Python encapsuladas que se pueden importar desde otras clases y sirven como un transporte de aplicaciones o bibliotecas escritas en este lenguaje. Varios *eggs* de Python pueden ser importados directamente desde Plone y utilizados a nivel global por todos los productos funcionando sobre el sitio. Es así como productos originalmente hechos solo para Python o solo para Plone, pueden salir de su esfera de acción y ser utilizados como aplicaciones independientes o intercambiados entre diferentes *frameworks* sin afectar el desarrollo [2].

## 2.4. Frameworks

Un *framework*, es una abstracción en la cual código común provee una funcionalidad genérica que puede ser selectivamente sobrescrita o especializada por el usuario de dicho código para ofrecer una funcionalidad específica [15]. Como tal podemos entender a Plone, más que como un CMS, como un *framework* el cual podemos adaptar a nuestras necesidades.

Los *frameworks* son una parte importante a la hora de desarrollar grandes sistemas orientados a contenidos. Nos permiten agilizar el desarrollo, aumentar la productividad

y disminuir el tiempo necesario para tener una versión funcional del sistema. Además de esto, los *frameworks* orientados a contenido nos dan mayor versatilidad que otros ya que nos permiten hacer una definición abstracta de objetos específicos para nuestra tarea y adaptar las operaciones que el *framework* mismo ya tiene definidas. De esta forma, nuestros tipos representan entidades conceptuales que se pueden extender e implementar para asemejar situaciones del mundo real [15].

Plone trabaja sobre Zope, un servidor *web* escrito en Python que actúa como *framework* de Plone. En el diseño de sistemas basados en Plone, normalmente se pueden adaptar componentes tanto de Zope como de Plone para realizar tareas específicas, por lo cual es importante conocer ambos y la forma en que se relacionan.

### 2.4.1. Zope

Zope es un servidor de aplicaciones web orientado a objetos que sirve como *framework* para la construcción de nuevos sistemas [5]. Actualmente existen dos versiones de Zope, la versión 2 y la versión 3, ambas firmemente ligadas a Plone. Plone trabaja sobre la versión 2.x de Zope, sin embargo incorpora algunos aspectos de Zope 3 en sus últimas versiones.

La estructura de objetos de Zope es jerárquica, lo que significa que un sitio Zope típico está compuesto por varios objetos que contienen otros objetos. Originalmente, Zope tenía la función única de publicar objetos creados.

Crear y trabajar con objetos en Zope se hace por medio de un navegador *web*, usando la interfaz de manejo de Zope (ZMI). La gestión y configuración básica del servidor se hace completamente por la red. Cualquier objeto en la jerarquía puede ser configurado desde la ZMI mediante vistas sencillas.

Una de las características importantes de Zope, es que se acopla con el modelo de desarrollo *web*, lo que permite separar las aplicaciones en partes y delegar su creación a diferentes áreas con el fin de establecer una mejor comunicación entre desarrolladores, diseñadores y usuarios, evitando que las tareas de uno intervengan en el trabajo de otros. Ésto lo hace una herramienta de trabajo colaborativo muy eficiente a la hora de trabajar en aplicaciones grandes [18].

Por defecto, los objetos de Zope son guardados en una base de objetos transaccional llamada “Base de Datos de Objetos Zope” (ZODB). Cada petición atendida por el servidor es tratada como una transacción aparte por la ZODB. Si un error ocurre mediante una acción *request*, los cambios se descartan de manera inmediata. La ZODB se puede, además, extender conectándose con bases de datos relacionales si es que se requieren durante la implementación.

Finalmente, Zope permite el desarrollo de *productos*, que son secciones de código adicionales que pueden interactuar con el núcleo de Zope, la ZODB y la jerarquía de objetos. Técnicamente, éstos son paquetes de Python comunes a los que se hace referencia desde el servidor Zope y que se pueden llamar desde cualquier parte como un objeto más del sistema [18].

### Zope 3

Zope 3 es una versión de Zope que extiende el concepto de un *framework* orientado a objetos y lo complementa mediante una arquitectura nueva. La diferencia más importante entre Zope 2 y Zope 3 es que Zope 2 es completamente orientado a objetos mientras que Zope 3 está orientado a una arquitectura de componentes [6]. El cambio de arquitectura se dio para corregir algunas desventajas de Zope 2, principalmente la necesidad de estructuras de *software* complejas que, al diseñarse en Zope 2, se hacían demasiado complicadas a causa del código necesario para su funcionamiento. Ésto hacía difícil la construcción de sistemas grandes. La arquitectura de componentes, por otra parte, permite estructurar el código en unidades pequeñas que se conectan mediante interfaces [17].

Ésta versión de Zope trabaja como una colección de componentes de software. Los componentes, son objetos que tienen un completo “entendimiento” de su funcionalidad y responsabilidad. Podemos entonces describir cada componente formalmente mediante el uso de interfaces. Usando interfaces, se pueden agrupar los componentes como una sola aplicación dentro del servidor, de manera que solo utilizando adaptadores se pueden llamar y organizar aplicaciones pequeñas como componentes de una aplicación más compleja, facilitando la interacción y el diseño [17].

Originalmente Zope 3 inició como una implementación de la base de Zope 2, y no es compatible con su predecesor. Mientras que muchos conceptos de Zope 2 pasaron a Zope 3, en realidad comparten poco código en común. Por esta razón se diseñó la tecnología *Five*, que permite llevar la arquitectura de componentes de Zope 3 a aplicaciones existentes de Zope 2. *Five* permite organizar objetos de Zope 2 como componentes y llamarlos como tales [2].

El producto implementado en [3] fue diseñado usando la estructura de desarrollo de Zope 2. Aunque la versión actual de Plone aún trabaja con Zope 2, cada versión nueva incorpora más y más conceptos de Zope 3, haciendo muchos productos como el mencionado incompatibles. El producto *Solicitudes* que aquí se presenta, se volvió a implementar usando la arquitectura de componentes de Zope 3 y *Five*. De esta forma se asegura que el producto permanecerá vigente por más tiempo.

#### 2.4.2. Plone

Plone es CMS basado en Zope que tiene miles de desarrolladores en todo el mundo. Está programado en Python y es un desarrollo basado en código abierto. Puede utilizarse como servidor intranet o extranet, como un sistema de publicación de documentos o como una herramienta de trabajo en grupo para colaborar entre entidades distantes.

El proyecto comenzó en 1999 por Alan Runyan, Alexander Limi, y Vidar Andersen. Rápidamente se convirtió en uno de los populares y poderosos CMS de código abierto. Sus características principales son [7]:

- Permite la producción rápida de sitios.
- Establece un enfoque particular sobre los metadatos de los contenidos, permitiéndonos cambiar las propiedades de los mismos con gran flexibilidad.

- Utiliza carpetas virtuales y “flujos de trabajo”, lo que le permite adaptarse a múltiples funciones (por ejemplo, como CRM).
- Posee un entorno gráfico basado en *web*, lo que permite acceder al sistema desde cualquier computadora con un navegador actualizado.
- Hace la gestión deslocalizada de los contenidos.
- Permite la edición de páginas en tiempo real.
- Facilita la colaboración entre usuarios mediante el uso de permisos.
- Por defecto permite manejar, de forma nativa, traducciones entre varias lenguas dentro del sitio.
- Nos deja desplegar imágenes de forma ilimitada (con la utilización masiva de CSS).
- Está enfocado a la reusabilidad.
- Fomenta la producción y el uso de contenidos al estar centrado en la experiencia del usuario.
- Gestiona historicos de cambios para cada instancia de contenido.
- Incluye varias plantillas (*templates*) estándares para el manejo sencillo de documentos y capital digital.
- Tiene un motor de búsqueda completo con indexación en tiempo real.
- Es modulable, evolutivo y fácil de personalizar.
- Nos permite hacer creación y gestión de *workflow* por defecto.

Además de todo esto, Plone introduce el concepto de *portlets*. Un *portlet* es una sección de código asociada a una vista, que se puede tratar como un dispositivo de *software* desplegable en pantalla. Consiste en [2]:

1. Una interfaz de Python. Por convención, normalmente hereda de la interfaz `IPortlet-DataProviderBy`.
2. Una clase de “contenido” persistente que guarda la información que se desplegará en el portlet y que implementa la interfaz.
3. Una forma de creación, utilizada para recabar información al momento de crear el portlet.
4. Una forma de edición, que sirve para editar la información guardada en portlets configurables.

5. Una forma de despliegue, que es la vista final que verán los usuarios con permiso de desplegar el portlet.

Plone puede ser extendido modularmente mediante la instalación de *productos*. Un *producto* en Plone es un módulo de *software* independiente que puede hacer llamadas a las bibliotecas que conforman el núcleo de Plone, de forma que extienda su funcionalidad. Plone permite registrar productos que proveen nuevos tipos de dato, procesos, portlets, vistas y funciones transparentes al usuario que puedan ser aprovechadas para el funcionamiento interno del servidor.

### Permisos en Plone

Tanto en Zope como en Plone, las vistas, acciones y atributos de los contenidos son protegidos por *permisos*. Los permisos son etiquetas que marcan a los usuarios para indicar que operaciones pueden llevar a cabo [17]. Zope es quien se encarga de verificar si el usuario actual tiene los permisos apropiados para llevar a cabo sus peticiones. En caso de no tenerlos, el servidor levanta una excepción *AccessControl.Unauthorized* que puede ser atrapada por los productos que trabajan sobre el mismo [2].

Los permisos no se asignan directamente a los usuarios, sino a roles. Cada usuario puede tener cualquier número de roles, tanto globales como en el contexto de un objeto específico. Los roles globales y locales se pueden también asignar a grupos, en cuyo caso todos los usuarios catalogados dentro de un grupo obtienen ese rol en particular. De esta forma Plone facilita el manejo de la seguridad al no tener que asignar roles individualmente [2].

Hay tres tipos principales de permiso en Plone:

- Permisos que se relacionan con las operaciones básicas sobre contenidos. En este grupo están incluidos *View* y *Modify portal content*, que permiten al usuario ver o modificar el objeto, respectivamente.
- Permisos que controlan la creación de tipos particulares de contenido. Por ejemplo, el permiso *ATContentTypes: Add Image* permite a un usuario añadir una imagen al sistema. Si está en la raíz del sitio, ésto se cumple a nivel global. Si está asignado solo en algunas carpetas particulares, solo lo podrá hacer en dichos lugares.
- Permisos que refuerzan las políticas del sitio. Por ejemplo, el permiso *Add portal member* que permite inscribir a usuarios como miembros. Normalmente, solo se dan a los administradores del sitio o a quienes tienen alguna tarea de administración global.

Plone descansa sobre una jerarquía de permisos compleja, que controla todos los aspectos de funcionalidad. La cantidad de permisos que cada sitio Plone contiene es variable, ya que los productos externos pueden generar y registrar sus propios permisos a medida que se introducen nuevas funcionalidades dentro del sistema [2].

- *Access contents information* (permiso de sistema): Es un permiso de bajo nivel heredado de Zope que permite controlar el acceso a uno o varios objetos.

- *View*: También se hereda de Zope y permite ver la vista principal de un objeto.
- *List Folder Contents*: Permite ver la lista de objetos que existen dentro de una carpeta. Es heredado de CMF.
- *Modify portal content*: Es heredado de CMF y permite hacer operaciones de edición sobre un objeto.
- *Manage portal*: Permite modificar la información global de un portal. Normalmente solo lo tienen administradores del sitio.
- *Add portal content*: Permiso de CMF que deja al usuario insertar objetos en el sitio. Es un permiso complementario que debe ir acompañado por los permisos particulares de cada objeto que se quiera insertar. Por ejemplo, para insertar una imagen un usuario debe contar con *ATContentTypes: Add Image* y, simultáneamente, el permiso de *Add portal content*.

Adicionalmente, algunos otros permisos del núcleo son necesarios para acceder a propiedades específicas de los usuarios o cambiar el estado de un flujo de trabajo:

- *Change portal events*: Permite cambiar el estado de un contenido. Es heredado del producto del núcleo DCWorkflow.
- *Manage properties*: Es un permiso de Zope que permite acceder a la información que contiene las propiedades de los usuarios, y modificarlas. Las propiedades incluyen los campos básicos como nombre, correo electrónico, etc, o campos no estándares que introduzcan productos de terceros.
- *Delete objects*: Permite borrar un contenido del sitio.

## Roles en Plone

Los *roles* en Plone se pueden definir como un conjunto de permisos [2] identificados por un nombre único, que es el nombre del rol. Usualmente es más fácil crear grupos de usuarios, a los cuales se les asigna un conjunto de roles una sola vez, que tener que manejar todos esos roles, para cada uno de los usuarios registrados, de forma individual. Por defecto Plone crea los grupos de usuarios *Administrators* y *Reviewers*, a los cuales se les asignan por omisión los roles *Manager* y *Reviewer* respectivamente. El rol *Manager* contiene permisos de administración global como *Add portal content* y *Reviewer* tiene permisos de visibilidad y edición sobre objetos, además de *Change portal events* [2].

Hay seis roles principales en una instalación por defecto de Plone:

- *Member*: Es el rol principal para un usuario que es miembro del portal y se asigna solo a usuarios que están conectados.
- *Manager*: Es un rol de super usuario. Como ya se dijo, se otorga por omisión al grupo *Administrators* (en donde se asignan los administradores del sitio).

- *Reviewer*: Es un rol que contiene los permisos necesarios para que un usuario pueda ver y aprobar contenido que ha sido enviado a revisión.
- *Reader*: Es un rol que normalmente tiene solo usos locales, y permite al usuario que lo tiene ver un contenido determinado (aún si los usuarios con el rol *Member* no lo pueden ver).
- *Editor*: Es el rol complementario que también es primordialmente local. A quien se le asigna, tiene la capacidad de editar la información de un contenido en ese ámbito en particular. Es la contraparte del rol *Reader*.
- *Contributor*: Es usado para permitir añadir contenidos nuevos en algún ámbito local.

Aunado a éstos, Zope define tres roles que se asignan automáticamente y que normalmente no pueden ser modificados [2]:

- *Owner*: Es el rol dado al dueño del objeto actual. Generalmente, lo tiene el usuario que lo crea. Contiene permisos para ver, modificar y cambiar el estado de un objeto.
- *Authenticated*: Es un rol que se les da a todos los usuarios autenticados. Éste es un rol de más bajo nivel que *Member* y no es modificable explícitamente. Por esta razón, normalmente se utiliza el rol *Member* en su lugar si se quiere uno referir a miembros autenticados dentro del sitio.
- *Anónimos*: Éste rol se le da a todos los usuarios no autenticados, efectivamente dividiendo a todos los usuarios que visitan el sitio mediante roles específicos. Sirve para establecer las políticas de seguridad de los usuarios que no son miembros del sitio pero que tienen un interés por visitarlo.

### **Workflows en Plone**

La herramienta de flujos de trabajo en Plone, le otorga ciertas capacidades y limitaciones que son clave para entender como trabajan los *workflows* en este CMS. El producto usado por Plone para manejar *workflows* es *DCWorkflow*, un producto del núcleo desarrollado por la corporación Zope.

Los *workflows* en Plone se componen de un conjunto de *estados* y *transiciones*. Un *estado* es la información de un objeto de contenido en un momento particular del tiempo. Todos los *workflows* tienen un estado inicial en el cual se encuentran los contenidos que lo utilizan al ser recién creados. A partir de este estado, el *workflow* moverá el contenido por una cadena de estados ya sea mediante la interacción con los usuarios o de forma automática. Para que un usuario pueda cambiar el estado de un objeto, requiere el permiso *Change portal events*.

Los disparadores que provocan el salto de un estado a otro, se llaman *transiciones*. Una *transición* conecta un estado inicial a un estado final. En Plone, en particular, las transiciones están definidas en términos del estado final al que llega la transición.

*DCWorkflow* asume que existe un objeto en el sistema que es el objetivo del *workflow*. Más aún, asume que todos los objetos del mismo tipo deben usar el mismo *workflow*. En Plone, todos los objetos tienen un *workflow* asociado. A continuación, se enumeran los tipos básicos de *workflow* que la herramienta dispone en una instalación estándar [11]:

- *Workflow* de Estado Único

Éste es un *workflow* simple que se utiliza para sitios que no necesitan un flujo de trabajo. Se compone de un solo estado sin transiciones que muestra inmediatamente todos los contenidos.

- Estados:

- Publicado: Todos los contenidos que usan este *workflow* tienen el permiso *View* asignado a todos los roles (incluyendo el rol *Anonymous*). El permiso *Modify portal content* es otorgado a los roles *Owner* y *Manager*.

En la figura 2.1, se muestra el diagrama del *workflow* de estado único: un estado sin transiciones.

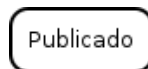


Figura 2.1: *Workflow* de Estado Único

- *Workflow* de publicación simple

El *workflow* de publicación simple es especialmente útil para pequeños sitios o intranets donde es necesario un revisor. El estado inicial es *Privado*, tal que el contenido no es accesible para todos.

- Estados:

- **Privado:** Asigna los permisos *Modify portal content*, *View* y *Access contents information* a los roles *Owner* y *Manager*.
- **Pendiente de revisión:** El permiso *Modify portal content* queda solo con el rol *Manager* y el rol *Reviewer* obtiene el permiso *View* y *Change portal events*. El rol *Owner* no puede editar el contenido en este estado.
- **Publicado:** Todos los contenidos en este estado del *workflow* tienen el permiso *View* asignado a todos los roles (incluyendo el rol *Anonymous*). El permiso *Modify portal content* está reservado para el rol *Manager*. El rol *Owner* no puede editar el contenido en este estado.

En la figura 2.2 se muestran las transiciones entre estados.



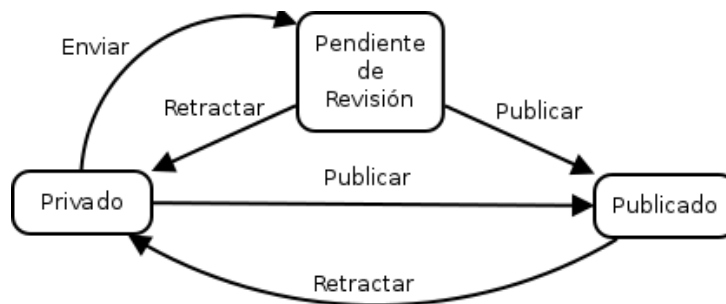


Figura 2.2: *Workflow* de Publicación Simple

■ *Workflow* de Intranet

En el *workflow* de intranet, normalmente solo el rol *Member* tiene el permiso *Access contents information*. Por defecto, todos los contenidos creados en la intranet son visibles para todos los miembros de la misma. A pesar de ello, solo contenidos publicados pueden aparecer en portlets, por ejemplo.

• Estados:

- **Privado:** Asigna los permisos *Modify portal content*, *View* y *Access contents information* a los roles *Owner* y *Manager*.
- **Borrador Interno:** El permiso *Modify portal content* queda solo con los roles *Manager* y *Owner* mientras que el rol *Member* obtiene los permisos *View* y *Change portal events*. El rol *Owner* no puede editar el contenido en este estado.
- **Pendiente de Revisión:** El permiso *Modify portal content* queda solo con el rol *Manager*, el rol *Member* obtiene el permiso *View* y el rol *Reviewer* obtiene los permisos *View* y *Change portal events*. El rol *Owner* no puede editar el contenido en este estado.
- **Publicado Internamente:** Todos los contenidos en este estado del *workflow* tienen el permiso *View* asignado al rol *Member*. El permiso *Modify portal content* está reservado para el rol *Manager*. El rol *Owner* no puede editar el contenido en este estado y el rol *Anonymous* no puede ver el contenido.
- **Visible Externamente:** Igual que “(Internamente) Publicado” pero el rol *Anonymous* tiene asignado el permiso *View*.

En la figura 2.3 se muestran las transiciones entre estos estados.

■ *Workflow* de la Comunidad

Éste es el *workflow* por defecto. Generalmente es suficiente para administrar cualquier tipo de portal o intranet.

• Estados:

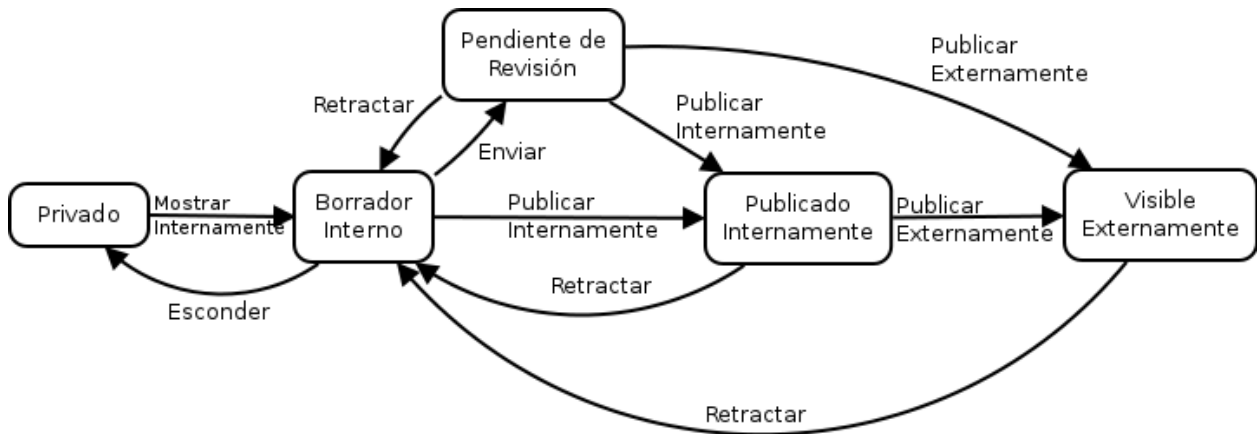


Figura 2.3: *Workflow* de Intranet

- **Privado:** Asigna los permisos *Modify portal content*, *View* y *Access contents information* a los roles *Owner* y *Manager*.
- **Borrador Interno:** El permiso *Modify portal content* queda con los roles *Manager* y *Owner*, mientras que el rol *Anonymous* obtiene los permisos *View* y *Access contents information*. Tanto *Owner* como *Manager* pueden cambiar el estado ya que tienen *Change portal events*.
- **Pendiente de revisión:** El permiso *Modify portal content* queda solo con el rol *Manager* y el rol *Reviewer* obtiene el permiso *View* y *Change portal events*. El rol *Owner* no puede editar el contenido en este estado.
- **Publicado:** Todos los contenidos en este estado del *workflow* tienen el permiso *View* asignado a todos los roles (incluyendo el rol *Anonymous*). El permiso *Modify portal content* está reservado para el rol *Manager*. El rol *Owner* no puede editar el contenido en este estado.

En la figura 2.4 se muestran las transiciones entre estados.

### El núcleo de Plone

El núcleo de Plone consiste en una serie de bibliotecas que vienen incluidas en la instalación por defecto de Plone y que le otorgan la funcionalidad básica como CMS. Las tres más importantes son:

- *Content Management Framework (CMF)*. El conjunto de bibliotecas CMF está diseñada para proveer la infraestructura básica para la creación de un CMS. Originalmente era un proyecto de Zope, completamente independiente de Plone, sin embargo con el tiempo se ha convertido en una parte importante ya que muchas secciones de CMF se utilizan en el núcleo de Plone; Plone está implementado sobre CMF, lo que indica que cada vez que CMF cambia aquel se beneficia de dichos cambios.

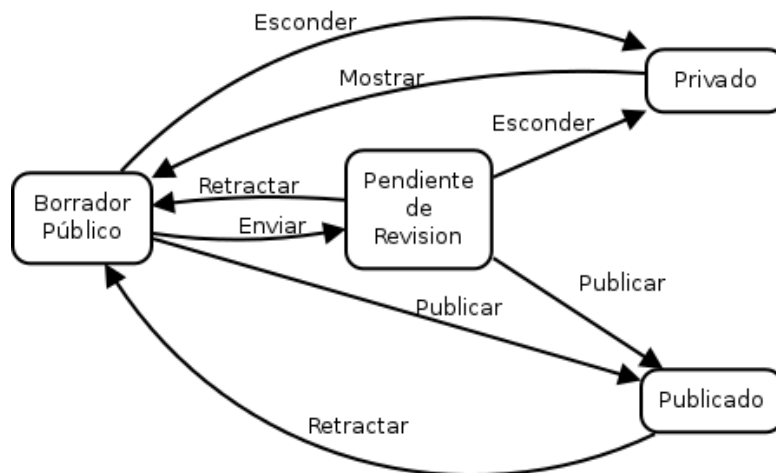


Figura 2.4: *Workflow* de la Comunidad

Dentro de Plone, CMF se encarga de manejar la infraestructura básica de membresía y workflows, envolviendo la estructura de seguridad de Zope en una nueva capa a nivel de usuario. Las siguientes herramientas de Plone son heredadas de CMF:

1. Herramienta de Membresía.- Permite crear y manipular miembros en un sitio Plone; ésto incluye la asignación de roles, búsqueda de usuarios, manejo de grupos, etc.
  2. Herramienta de Workflow.- Nos sirve para la administración simple de flujos de trabajo.
  3. Herramienta de Catálogo.- Ésta herramienta se encarga de registrar los contenidos de de un sitio Plone en un índice global y otorga las funciones necesarias para hacer ordenamiento y búsqueda dentro de dicho índice.
  4. Herramienta de Discusión.- Lleva el control del historial de cambios sobre cada contenido. Marca cada objeto con una interfaz que sirve para llevar un control de versiones básico.
  5. Herramienta de Acciones.- Auxilia en la administración y creación de llamadas a funciones del núcleo.
- *Archetypes (AT)*. Originalmente parte del núcleo CMF, Archetypes es la librería que se encarga de dotar a Plone con un método común para la construcción de objetos de contenido, basado en definición de esquemas. Los campos de un contenido pueden ser agrupados para edición, haciendo muy simple la creación de formas, por ejemplo.

Archetypes fue diseñado principalmente para desarrolladores, para aprovechar la funcionalidad básica de CMF al mismo tiempo que otorga una forma genérica de definir nuevos tipos de contenido sin tener que recurrir directamente a tareas

de programación complejas. Actualmente, Archetypes es la herramienta estándar para manejar la definición y creación de nuevos tipos de contenido, así como para realizar tareas básicas de administración sobre ellos. Los objetos de Archetypes trabajan muy cercanamente con CMF, de forma que entre ambos proyectos se completa el motor básico para la creación y administración de contenidos.

Cuenta con las siguientes características:

- Autogeneración de vistas de edición y presentación para los contenidos.
  - Registro de tipos de dato nuevos a través de las herramientas de CMF.
  - Configurabilidad de las acciones de CMF.
  - Transparencia en el guardado de datos.
  - Identificadores únicos
- *DCWorkflow*. Es la herramienta básica incluida en Plone para la definición sencilla y manejo de flujos de trabajo. Hace uso de las herramientas de las que dispone CMF para construirlos y registrarlos en el sistema. DCWorkflow nos permite describir flujos de trabajo y asignarlos a nuestros tipos de contenido personalizados de manera automática. Todos los contenidos del sitio están asociadas a un flujo de trabajo inicial. Cada instancia, comienza forzosamente en un estado de su flujo de trabajo.

Un estado de un flujo de trabajo se caracteriza mediante una configuración única de permisos activados en ese estado. Dentro de la descripción del flujo, es necesario establecer para cada estado:

- A que roles se le van a asignar permisos en ese estado.
- Que permisos están activos para dichos roles en ese estado.

Esta configuración, solo toma efecto en el contexto local de cada instancia de contenido. Puede haber, simultáneamente, otro contenido del mismo tipo en la misma carpeta pero con diferente configuración de permisos activos sobre el mismo.

Las transiciones de los flujos de trabajo están descritas en términos del estado al cual desembocan. En la definición de la transición, se indica el estado al cual desemboca. Para establecer una transición de un estado a otro, ésta se registra solo en el estado de salida. La definición de la misma se encarga del resto.

Finalmente, ya que DCWorkflow ha registrado la descripción de un nuevo flujo, CMF se da a la tarea de desplegar la información de transiciones y estados dentro del sitio.



# Capítulo 3

## Descripción Técnica

En este capítulo se hace una descripción detallada de los componentes del sistema, así como de los flujos de trabajo utilizados y las instrucciones básicas del mismo.

A continuación, se presenta la descripción de todos los tipos de dato nuevos que se tuvieron que crear para resolver el problema. Se muestra una descripción precisa de los campos que componen cada uno de estos tipos nuevos. Posteriormente, se presentan los permisos y restricciones de seguridad para cada una de las operaciones sobre los contenidos. Finalmente, el capítulo termina exponiendo mediante diagramas las consideraciones tomadas para del diseño del sistema.

### 3.1. Tipos de Dato del Sistema

El producto *Solicitudes* introduce varios tipos de dato personalizados a Plone. Éstos son:

**Solicitud de Licencia (meta-tipo Solicitud)** Es el formato que se debe llenar en caso de que el investigador pida presupuesto de viaje (incluye viáticos, gastos de transporte y de inscripción a eventos).

**Solicitud de Visitante (meta-tipo SolicitudVisitante)** Un investigador lo debe llenar en caso de tener un invitado de otra institución académica (nacional o extranjera) para el cual se requiera presupuesto (incluye viáticos y gastos de transporte).

**Solicitud para Becarios (meta-tipo SolicitudBecario)** Es un formato para becarios que piden presupuesto de viaje (incluye viáticos, gastos de transporte y de inscripción a eventos).

**Folder de solicitudes (meta-tipo SolicitudFolder)** Es un tipo de contenedor que cumple la función de ser un receptor de solicitudes y, a la vez, representa conceptualmente un periodo de tiempo en el cual se puede asignar una cantidad máxima de presupuesto para los Investigadores, Becarios y Técnicos Académicos. Los folders contienen diferentes pestañas que corresponden a listas de solicitudes en determinado estado del proceso de aprobación o rechazo.

Los campos utilizados para crear cada uno de estos tipos de dato, son los definidos por la API de Plone en el producto del núcleo llamado *Products.Archetypes*. Éste producto permite la creación de tipos de contenido nuevo utilizando los constructores estándar de Plone para tipos. De esta forma, garantizamos que los tipos de contenido nuevos fueran tratados de forma análoga a los tipos básicos del sistema.

Para la definición de tipos, se utilizaron los siguientes campos:

**LinesField** Guarda un objeto de tipo *Tuple* como está definido por la API de Python.

**StringField** Guarda un objeto de tipo *String* como está definido por la API de Python.

**DateTimeField** Guarda un objeto de tipo *Date* como está definido por la API de Python.

**FloatField** . Guarda un objeto de tipo *Float* como está definido por la API de Python.

### 3.1.1. Solicitud de Licencia (meta-tipo Solicitud)

Éste formato lo utilizan investigadores que piden presupuesto para viaje; *i.e.* viáticos, dinero para transporte y para pago de inscripciones. También con este formato se pueden justificar ausencias no remuneradas.

#### 1. pais

**Tipo:** LinesField

**Valor inicial:** MX (México)

**Necesario:** Si

**Descripción:** Nombre del país destino. El investigador debe llenar con el código del país al cual va a asistir a alguna conferencia, congreso o evento (incluido México).

#### 2. ciudad\_pais

**Tipo:** StringField

**Valor inicial:** -

**Necesario:** Si

**Descripción:** Nombre de la ciudad destino. El investigador llena a mano el nombre de la ciudad en la cual se llevará a cabo el evento al que asiste, dentro del país especificado en el campo “pais”.

#### 3. institucion

**Tipo:** StringField

**Valor inicial:** -

**Necesario:** Si

**Descripción:** Nombre de la institución que se visitará. Éste campo corresponde al código interno asignado para cada una de las instituciones con las cuales el Instituto de Matemáticas tiene colaboración.

## 4. fecha\_desde

**Tipo:** DateTimeField

**Valor inicial:** -

**Necesario:** Si

**Descripción:** Fecha de inicio del periodo de Licencia. Marca el inicio de la ausencia del investigador, ya sea remunerada o no. Es un campo *indexable* que posteriormente permite la búsqueda por intervalos de tiempo.

## 5. fecha\_hasta

**Tipo:** DateTimeField

**Valor inicial:** -

**Necesario:** Si

**Descripción:** Fecha de fin del periodo de Licencia. Marca el fin de la ausencia del investigador. Así como “fecha\_desde”, este campo es *indexable* y permite la búsqueda por intervalos.

## 6. objeto\_viaje

**Tipo:** StringField

**Valor inicial:** -

**Necesario:** Si

**Descripción:** Enuncia el objetivo del viaje, razón de la solicitud. Éste campo debe ser llenado con mucho detalle ya que presenta la razón de la ausencia y sirve como núcleo de la solicitud. En base a lo escrito aquí, el consejo decide si debe o no autorizar una solicitud de licencia.

## 7. investigacionarea

**Tipo:** LinesField

**Valor inicial:** -

**Necesario:** Si

**Descripción:** Área de investigación (según la clasificación de la AMS). Ésta asociada automáticamente a las áreas de investigación señaladas en el perfil del investigador que crea el objeto. El campo es *indexable* y permite la búsqueda por áreas de investigación.

## 8. trabajo



**Tipo:** StringField

**Valor inicial:** “No”

**Necesario:** No

**Descripción:** Describe si se va a presentar un trabajo durante el tiempo de la licencia. Solamente puede tener “Si” o “No”. Si el campo es marcado con “Si”, el investigador automáticamente tendrá acceso a campos adicionales para registrar el nombre de su artículo o conferencia, lo que posteriormente se verá reflejado en su currículum.

9. titulo\_trabajo

**Tipo:** StringField

**Valor inicial:** -

**Necesario:** No

**Descripción:** Aquí el investigador puede registrar el título de la conferencia o artículo que va a presentar.

10. pasaje

**Tipo:** StringField

**Valor inicial:** “No”

**Necesario:** No

**Descripción:** Éste campo describe si el investigador necesitará ayuda económica para pagar el medio de transporte. Solamente es posible contestar “Si” o “No”.

11. tipo\_pasaje

**Tipo:** LinesField

**Valor inicial:** -

**Necesario:** No

**Descripción:** Describe el tipo de transporte que el investigador va a utilizar. Sirve para justificar la cantidad de dinero solicitada para transportación. Vocabulario posible “Avión”, “Autobús” o “Automóvil”.

12. cantidad\_pasaje

**Tipo:** FloatField

**Valor inicial:** 0.0

**Necesario:** Si

**Descripción:** Contiene la cantidad solicitada para cubrir los costos de transportación durante el periodo establecido.

## 13. inscripcion

**Tipo:** StringField

**Valor inicial:** "No"

**Necesario:** No

**Descripción:** Describe si hay una cuota de inscripción. Vocabulario posible "Si" o "No".

## 14. cantidad\_inscripcion

**Tipo:** FloatField

**Valor inicial:** 0.0

**Necesario:** Si

**Descripción:** Contiene la cantidad necesaria para cubrir el costo de la inscripción.

## 15. viaticos

**Tipo:** StringField

**Valor inicial:** "No"

**Necesario:** No

**Descripción:** Describe si se van a necesitar viáticos. Vocabulario posible "Si" o "No".

## 16. cantidad\_viaticos

**Tipo:** FloatField

**Valor inicial:** 0.0

**Necesario:** Si

**Descripción:** Contiene la cantidad solicitada para cubrir gastos de viaje, de acuerdo al tabulador estándar del Instituto. El investigador tiene la facultad de poner cualquier cantidad que él considere.

## 17. displayAttachments

**Tipo:** FileField

**Valor inicial:** -

**Necesario:** No

**Descripción:** Contiene archivos adjuntos de cualquier tipo, que sirvan para justificar el dinero que se pide.

## 18. comentario\_owner

**Tipo:** StringField

**Valor inicial:** -

**Necesario:** No

**Descripción:** Guarda comentarios adicionales del solicitante que auxilien al Consejo Interno a tomar una decisión sobre la solicitud.

19. comentario\_ce

**Tipo:** StringField

**Valor inicial:** -

**Necesario:** No

**Descripción:** Guarda comentarios de los miembros de la Comisión Especial. Solo visible y modificable para usuarios con el permiso *Solicitud: Comisión Revisa Solicitud*.

20. comentario\_ci

**Tipo:** StringField

**Valor inicial:** -

**Necesario:** No

**Descripción:** Guarda comentarios de los miembros del Consejo Interno. Solo visible y modificable para usuarios con el permiso *Solicitud: Consejo Revisa Solicitud*.

21. fecha\_sesionci

**Tipo:** DateTimeField

**Valor inicial:** -

**Necesario:** No

**Descripción:** Fecha de revisión por el Consejo Interno. Solo visible y modificable para usuarios con el permiso *Solicitud: Consejo Revisa Solicitud*.

22. actaci

**Tipo:** StringField

**Valor inicial:** -

**Necesario:** No

**Descripción:** Número de acta en la cual figura la solicitud dentro de los registros del Consejo Interno. Éste número es para control administrativo y es asignado por el Consejo Interno. Solo visible y modificable para usuarios con el permiso *Solicitud: Consejo Revisa Solicitud*.

23. cantidad\_recomendada\_pasaje

**Tipo:** FloatField

**Valor inicial:** 0.0

**Necesario:** Si

**Descripción:** Contiene la cantidad recomendada por la Comisión Especial para cubrir gastos de transporte. Solo visible y modificable para usuarios con el permiso *Solicitud: Comisión Revisa Solicitud.*

24. cantidad\_recomendada\_inscripcion

**Tipo:** FloatField

**Valor inicial:** 0.0

**Necesario:** Si

**Descripción:** Contiene la cantidad recomendada por la Comisión Especial para cubrir gastos de inscripción. Solo visible y modificable para usuarios con el permiso *Solicitud: Comision Revisa Solicitud.*

25. cantidad\_recomendada\_viaticos

**Tipo:** FloatField

**Valor inicial:** 0.0

**Necesario:** Si

**Descripción:** Contiene la cantidad recomendada por la Comisión Especial para cubrir gastos de viaje. Solo visible y modificable para usuarios con el permiso *Solicitud: Comisión Revisa Solicitud.*

26. cantidad\_consejo\_pasaje

**Tipo:** FloatField

**Valor inicial:** 0.0

**Necesario:** Si

**Descripción:** Contiene la cantidad máxima asignada por el Consejo Interno para cubrir gastos de transporte. Solo visible y modificable para usuarios con el permiso *Solicitud: Consejo Revisa Solicitud.*

27. cantidad\_consejo\_inscripcion

**Tipo:** FloatField

**Valor inicial:** 0.0

**Necesario:** Si

**Descripción:** Contiene la cantidad máxima asignada por el Consejo Interno para cubrir gastos de inscripción. Solo visible y modificable para usuarios con el permiso *Solicitud: Consejo Revisa Solicitud.*

28. cantidad\_consejo\_viaticos

**Tipo:** FloatField

**Valor inicial:** 0.0

**Necesario:** Si

**Descripción:** Contiene la cantidad máxima asignada por el Consejo Interno para cubrir gastos de viaje. Solo visible y modificable para usuarios con el permiso *Solicitud: Consejo Revisa Solicitud*.

29. cantidad\_autorizada\_pasaje

**Tipo:** FloatField

**Valor inicial:** 0.0

**Necesario:** Si

**Descripción:** Contiene, de lo asignado por el Consejo Interno, la cantidad usada por el solicitante para cubrir gastos de transporte. Ésta es la cantidad que se refleja en la contabilidad restándose del total de presupuesto. Solo visible y modificable para usuarios con el permiso *Solicitud: Consejo Cambia Solicitud*.

30. cantidad\_autorizada\_inscripcion

**Tipo:** FloatField

**Valor inicial:** 0.0

**Necesario:** Si

**Descripción:** Contiene, de lo asignado por el Consejo Interno, la cantidad usada por el solicitante para cubrir gastos de inscripción. Ésta es la cantidad que se refleja en la contabilidad restándose del total de presupuesto. Solo visible y modificable para usuarios con el permiso *Solicitud: Consejo Cambia Solicitud*.

31. cantidad\_autorizada\_viaticos

**Tipo:** FloatField

**Valor inicial:** 0.0

**Necesario:** Si

**Descripción:** Contiene, de lo asignado por el Consejo Interno, la cantidad usada por el solicitante para cubrir gastos de viaje. Ésta es la cantidad que se refleja en la contabilidad restándose del total de presupuesto. Solo visible y modificable para usuarios con el permiso *Solicitud: Consejo Cambia Solicitud*.

### 3.1.2. Solicitud de Becario (meta-tipo SolicitudBecario)

La Solicitud de Becario, es una solicitud de licencia con datos adicionales. Además de los datos especificados en el tipo Solicitud de Licencia (meta-tipo Solicitud) de la sección anterior, se tienen los que se muestran a continuación.

1. nombrebecario

**Tipo:** ComputedField

**Valor inicial:** Dinámico.

**Necesario:** Si

**Descripción:** Es el nombre de usuario con el cual el becario entra al sistema, y que está asociado a su currículum y a un perfil guardado por InfoMatem.

2. nombrecompletobecario

**Tipo:** ComputedField

**Valor inicial:** Dinámico.

**Necesario:** Si

**Descripción:** Es el nombre completo del becario dentro del sistema. Éste nombre puede no ser único, pues es el nombre real y se utiliza con fines de despliegue.

3. grado

**Tipo:** StringField

**Valor inicial:** -

**Necesario:** Si

**Descripción:** Grado académico al cual el becario aspira o grado máximo de estudios (en su caso). Vocabulario posible “Licenciatura”, “Maestría” o “Doctorado”.

4. asesor

**Tipo:** DateTimeField

**Valor inicial:** -

**Necesario:** Si

**Descripción:** Nombre de usuario del Investigador que está encargado del becario que creó la solicitud. Éste dato se encuentra guardado en el currículum del becario que crea la solicitud, por lo cual se puede asignar automáticamente.

5. comentario\_asesor

**Tipo:** StringField

**Valor inicial:** -

**Necesario:** No

**Descripción:** Guarda comentarios adicionales que pone el asesor. Solo visible y modificable para usuarios con el permiso Solicitud: Investigador Revisa Solicitud.

### 3.1.3. Solicitud de Visitante (meta-tipo SolicitudVisitante)

La Solicitud de Visitante contiene, además de los datos especificados en el tipo Solicitud de Licencia (meta-tipo Solicitud), un campo para el nombre del visitante.

1. responsable

**Tipo:** ComputedField

**Valor inicial:** Dinámico.

**Necesario:** Si

**Descripción:** Nombre de usuario con el cual el solicitante entra al sistema. Puede ser solo un Investigador o un Técnico Académico, ya que solo éstos están autorizados a registrar visitantes.

2. nombrecompletoresponsable

**Tipo:** ComputedField

**Valor inicial:** Dinámico.

**Necesario:** Si

**Descripción:** Nombre completo del solicitante dentro del sistema. Se utiliza con fines de despliegue.

3. invitado

**Tipo:** StringField

**Valor inicial:** -

**Necesario:** Si

**Descripción:** Nombre completo del invitado para el cual el solicitante requiere presupuesto. Corresponde al nombre legal del invitado, el cual quedará asentado dentro de las actas del Consejo Interno.

### 3.1.4. Folder de Solicitudes (meta-tipo SolicitudFolder)

Es un tipo de contenedor que cumple la función de ser un receptor de solicitudes y, a la vez, representa conceptualmente un periodo de tiempo en el cual se puede asignar una cantidad máxima de presupuesto para los Investigadores, Becarios y Técnicos Académicos. Los folders contienen diferentes pestañas que corresponden a listas de solicitudes en determinado estado del proceso de aprobación o rechazo.

El Folder de Solicitudes contiene solo cuatro campos:

## 1. fecha\_desde

**Tipo:** DateTimeField

**Valor inicial:** -

**Necesario:** Si

**Descripción:** Fecha de inicio del periodo al cual corresponden las solicitudes que contiene. Sirve para mantener agrupadas todas las solicitudes el mismo periodo.

## 2. fecha\_hasta

**Tipo:** DateTimeField

**Valor inicial:** -

**Necesario:** Si

**Descripción:** Fecha de fin del periodo al cual corresponden las solicitudes que contiene. Al igual que el campo anterior, sirve para mantener agrupadas las solicitudes y permite facilitar la búsqueda por intervalos.

## 3. presupuesto\_inicial

**Tipo:** FloatField

**Valor inicial:** 0.0

**Necesario:** No

**Descripción:** Guarda la cantidad total máxima de presupuesto que se puede usar en ese periodo (contando todas las solicitudes aprobadas).

## 4. presupuesto\_asignado

**Tipo:** FloatField

**Valor inicial:** 0.0

**Necesario:** No

**Descripción:** Guarda la cantidad de presupuesto que ya ha sido utilizada en ese periodo (contando todas las solicitudes aprobadas).



## 3.2. Operaciones básicas del sistema

En esta sección se presentan las funciones básicas que operan sobre nuestros tipos de dato personalizados, desde el punto de vista de la implementación.

Utilizando la infraestructura de permisos de Plone, se puede controlar la seguridad mediante un manejo estricto y detallado de las operaciones implementadas.

El sistema introduce las siguientes operaciones básicas haciendo uso de las funciones implementadas sobre el núcleo de *CMF*:

**Crear nuevas solicitudes** Es la operación mediante la cual se instancia una solicitud nueva de alguno de los tipos de dato especificados anteriormente.

**Revisión y Modificación de Solicitudes** Consisten en leer y/o modificar los datos en una solicitud.

**Cambio de Estado** Es la operación que permite el cambio entre diversos estados del *workflow*, así como la asignación automática de permisos sobre el objeto que la llama.

**Importar Solicitudes** Consiste en introducir información al sistema y que éste, automáticamente, construya una nueva solicitud basada en esa información.

**Exportar Solicitudes** Consiste en descargar la información de un objeto solicitud en un formato amigable para otras plataformas.

**Establecer Presupuesto** Con esta operación se puede cambiar el presupuesto asignado para el Instituto de Matemáticas en el ámbito de viáticos y licencias.

El producto Solicitudes introduce varios permisos a Plone, cada uno de ellos ligado a las operaciones anteriores:

- *Solicitud: Add Solicitud.*- Éste permiso controla la llamada a `InvokeFactory`, la cual permite crear una instancia nueva de una solicitud de licencia.
- *SolicitudBecario: Add SolicitudBecario.*- Mediante este permiso, asignado solo a los Becarios, se puede llevar a cabo la creación de una instancia de la Solicitud para Becarios.
- *SolicitudVisitante: Add SolicitudVisitante.*- Los roles que contienen este permiso pueden crear una instancia de una Solicitud para Visitantes.
- *SolicitudFolder: Add SolicitudFolder.*- Éste permiso solo está asignado por defecto al administrador del sitio. Permite crear Folders de Solicitudes y, por ende, periodos fiscales dentro de los cuales se guardaran grupos de solicitudes.
- *Solicitud: Investigador Revisa Solicitud.*- Los Investigadores tienen, mediante este permiso, la capacidad de editar y acceder a los campos ocultos de una Solicitud de Becario que los tiene registrados como “Asesor”. Sirve para controlar el paso intermedio que existe entre la publicación por parte del Becario y la revisión por la Comisión Especial.

- *Solicitud: Revisar Solicitud.*- Éste permiso se otorga a todos los que pueden crear algún tipo de solicitud. Sirve para permitir ver los campos que la componen.
- *Solicitud: Modificar Solicitud.*- Con este permiso, se controla la modificación de los campos generales de la solicitud. Originalmente, solo lo tienen los usuarios con el rol *Owner*.
- *Solicitud: Comisión Revisa Solicitud.*- Los miembros de la Comisión Especial tienen este permiso, que les da acceso a los campos restringidos de información adicional exclusivos de la Comisión.
- *Solicitud: Consejo Revisa Solicitud.*- Los miembros del Consejo Interno tienen este permiso, el cual da acceso a los campos restringidos con de información adicional exclusivos del Consejo.
- *Solicitud: Consejo Cambia Solicitud.*- Éste permiso lo tienen solo algunos miembros del consejo. Permite que el rol que lo porta guarde datos nuevos en los campos restringidos para el consejo interno.
- *Importación de Solicitudes.*- Dicho permiso activa la capacidad de “Importación Global” que se describe más adelante.
- *Programar Presupuesto.*- Permite a quien lo tiene cambiar los datos de presupuesto para viáticos y licencias del instituto.

### 3.2.1. Crear nuevas solicitudes

La API de Plone, mediante el método *invokeFactory*, manda a crear una nueva solicitud si el usuario posee el permiso adecuado. Los identificadores únicos para cada objeto son generados y manejados por Plone.

#### 1. *Solicitud: Add Solicitud*

**Roles que asignan el permiso :**

- Investigador
- Técnico Académico

**Descripción :** Permite al usuario crear Solicitudes de Licencia.

#### 2. *SolicitudVisitante: Add SolicitudVisitante*

**Roles que asignan el permiso :**

- Investigador
- Técnico Académico

**Descripción :** Permite al usuario crear Solicitudes de Visitante.

#### 3. *SolicitudBecario: Add SolicitudBecario*

**Roles que asignan el permiso :**

- Becario

**Descripción :** Permite al usuario crear Solicitudes de Becario.

4. *SolicitudFolder: Add SolicitudFolder***Roles que asignan el permiso :**

- Manager (rol de sistema)

**Descripción :** Permite al usuario crear y modificar Folders de Solicitudes.

**3.2.2. Revisión y Modificación de Solicitudes**

Aquí se enlistan los permisos utilizados para la revisión y edición de solicitudes. Los flujos de trabajo definidos por el producto solicitudes está diseñados para permitir la edición y revisión intermitente de los datos en cada solicitud, por lo cual los permisos de visibilidad y edición están constantemente cambiando de rol a lo largo del proceso.

1. *Solicitud: Investigador Revisa Solicitud***Roles que asignan el permiso :**

- Investigador

**Descripción :** Permite a usuarios con el rol de Investigador revisar Solicitudes de Becario (meta-tipo SolicitudBecario).

2. *Solicitud: Modificar Solicitud***Roles que asignan el permiso :**

- Manager
- Owner

**Descripción :** Permite a usuarios que modifiquen las solicitudes que crearon asignándolo por *default* al rol de sistema *Owner*. Le da permiso a los usuarios con el rol de *Manager* a modificar cualquier solicitud.

3. *Solicitud: Comisión Revisa Solicitud***Roles que asignan el permiso :**

- Manager
- Comisionado
- Responsable de la Comisión

**Descripción :** Permite a los usuarios ver y modificar la información reservada para miembros de la Comisión Especial.

4. *Solicitud: Consejo Revisa Solicitud*

**Roles que asignan el permiso :**

- Manager
- Consejero
- Responsable del Consejo

**Descripción :** Permite a los usuarios ver y modificar la información reservada para miembros del Consejo Interno.

5. *Solicitud: Consejo Cambia Solicitud***Roles que asignan el permiso :**

- -

**Descripción :** Permite a los usuarios modificar los campos de cantidades ejercidas e información adicional. No está asignado por *default*, el *workflow* es el que introduce el permiso.

**3.2.3. Cambio de Estado**

El cambio de estado solo está disponible cuando el *workflow* otorga el permiso *Change portal events* al usuario. Solo los roles *Responsable del Consejo* y *Responsable de la Comisión* tendrán el permiso *Change portal events* en estados superiores a *borrador* en cada *workflow*.

Para cambiar el estado del Folder de Solicitudes (meta-tipo *SolicitudFolder*) solo se necesita tener el rol de sistema *Owner*, que se asigna automáticamente a los objetos creados por el usuario.

**3.2.4. Importar Solicitudes**

La importación de Solicitudes es una funcionalidad extra del sistema que solamente tienen los usuarios que crean solicitudes de algún tipo. Aunado a ésto, existe una función de importación avanzada para administradores que provoca la creación de un nuevo rol único de “Importador de Solicitudes”, que tiene la capacidad de importar solicitudes en bloque y de cualquier usuario sin importar que permiso de acceso tengan sobre dichos tipos de solicitudes.

1. *Importación de Solicitudes***Roles que asignan el permiso :**

- Importador de Solicitudes
- Investigador
- Técnico Académico
- Becario

**Descripción :** Permite a los usuarios importar solicitudes propias al sistema.

2. *Importación de Solicitudes Global*

**Roles que asignan el permiso :**

- Importador de Solicitudes

**Descripción :** Permite a los usuarios importar solicitudes de otros al sistema y ponerlas en cualquier estado del *workflow* (incluso, aprobadas).

### 3.2.5. Exportar Solicitudes

La exportación está disponible cuando el *workflow* de la solicitud o folder le asigna al usuario el permiso *View*. Todas las solicitudes visibles para un usuario, son exportables en formato de texto simple.

### 3.2.6. Establecer Presupuesto

*Solicitudes* establece un permiso que abre la capacidad de manejar el presupuesto total de los investigadores. Además, los roles con este permiso pueden ver todos los datos de contabilidad que genera el procesamiento de solicitudes de recursos. El uso de esta funcionalidad es delicado y, al igual que con la importación de solicitudes, debe ser una función solo destinada a administradores del sitio o personal administrativo de alto nivel.

#### 1. Programar Presupuesto

**Roles que asignan el permiso :**

- Programador de Presupuesto

**Descripción :** Permite a los usuarios modificar los datos de presupuesto máximo para cada investigador. También permite ver toda la información de contabilidad manejada por el producto.

### 3.3. Casos de Uso del Sistema

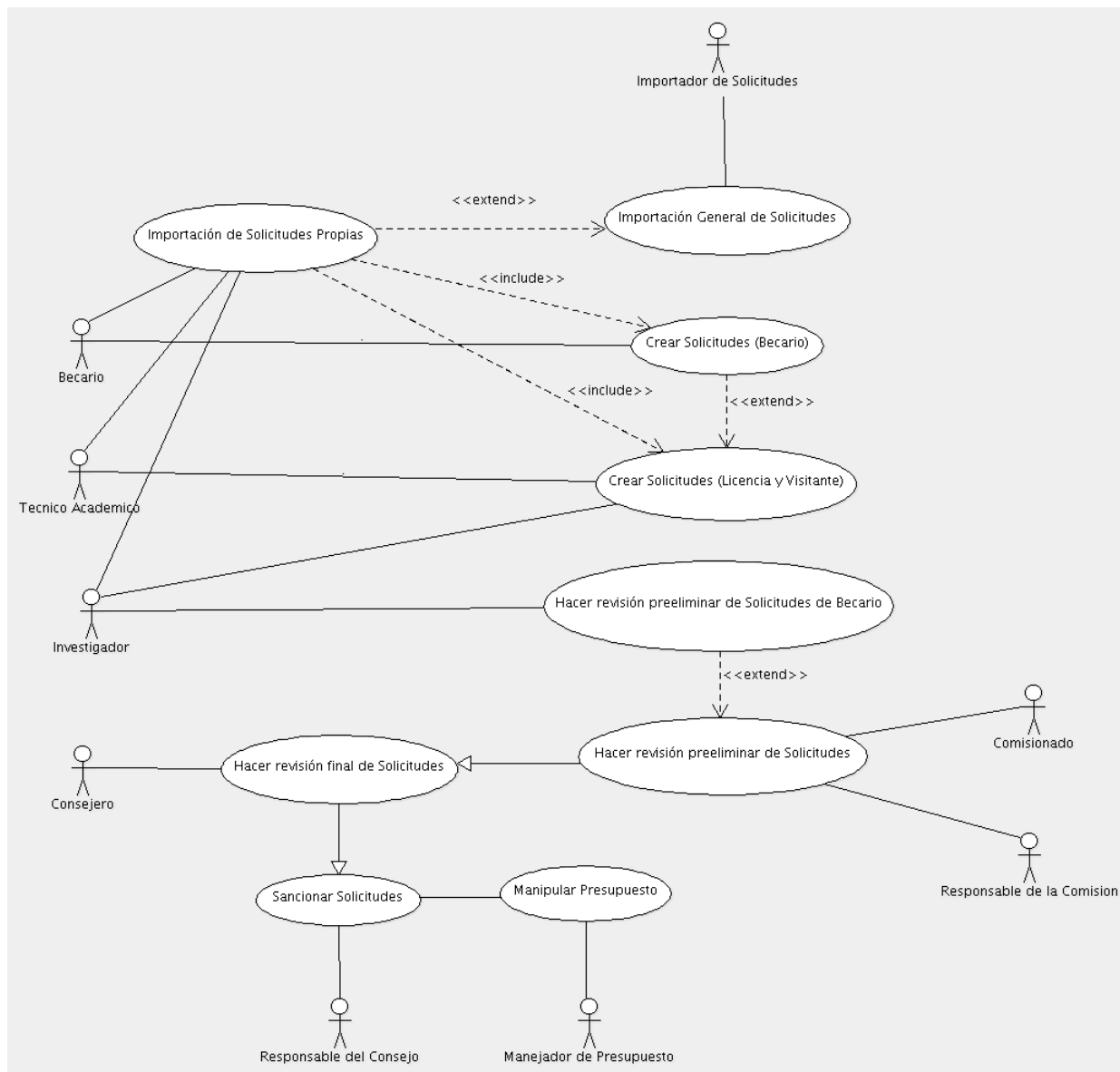


Figura 3.1: Casos de Uso del Sistema



### 3.4.1. La clase *Solicitante*

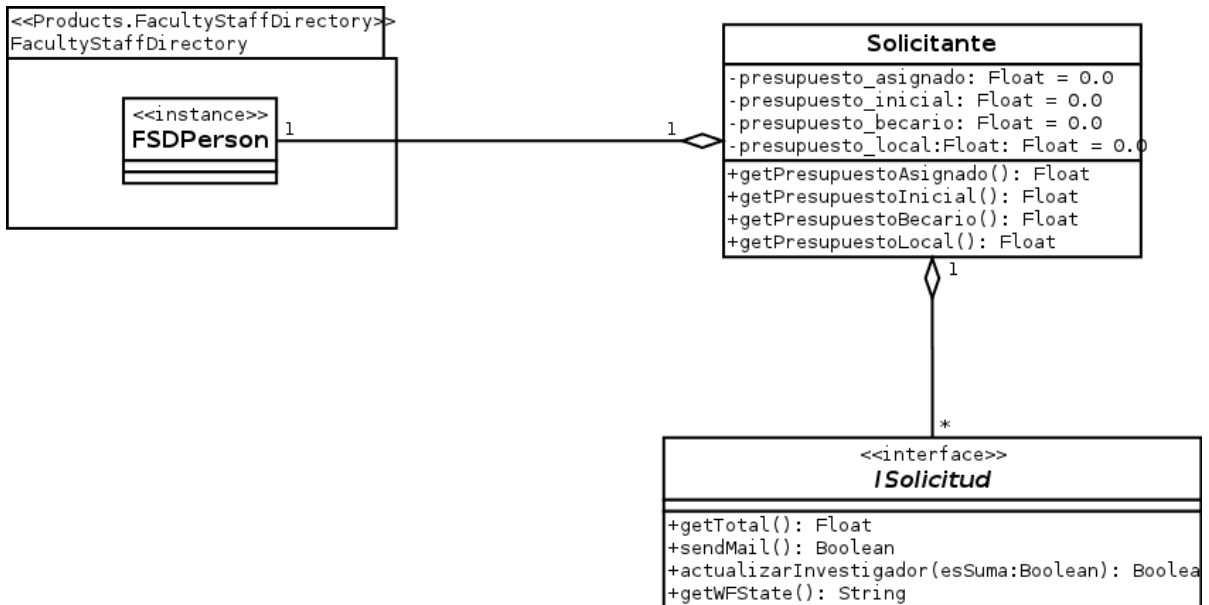


Figura 3.3: Clase *Solicitante*

La clase *Solicitante* se utiliza como una plantilla que representa la información presupuestal de un objeto *FSDPerson*, definido en el producto de Plone *Products.FacultyStaffDirectory*. El producto crea una instancia de esta clase cuando que llama a un objeto *FSDPerson* para la lectura y despliegue de datos financieros de datos, lo que a su vez permite que el constructor obtenga las propiedades del usuario registrado y acceso a la información de presupuesto en la base de datos. Consiste en cuatro campos:

1. presupuesto\_inicial

**Tipo:** Float

**Valor inicial:** 0.0

**Acceso:** Privado

**Descripción:** Guarda la cantidad total máxima de presupuesto que el solicitante puede usar.

2. presupuesto\_asignado

**Tipo:** Float

**Valor inicial:** 0.0

**Acceso:** Privado

**Descripción:** Guarda la cantidad de presupuesto que el solicitante tiene autorizado.



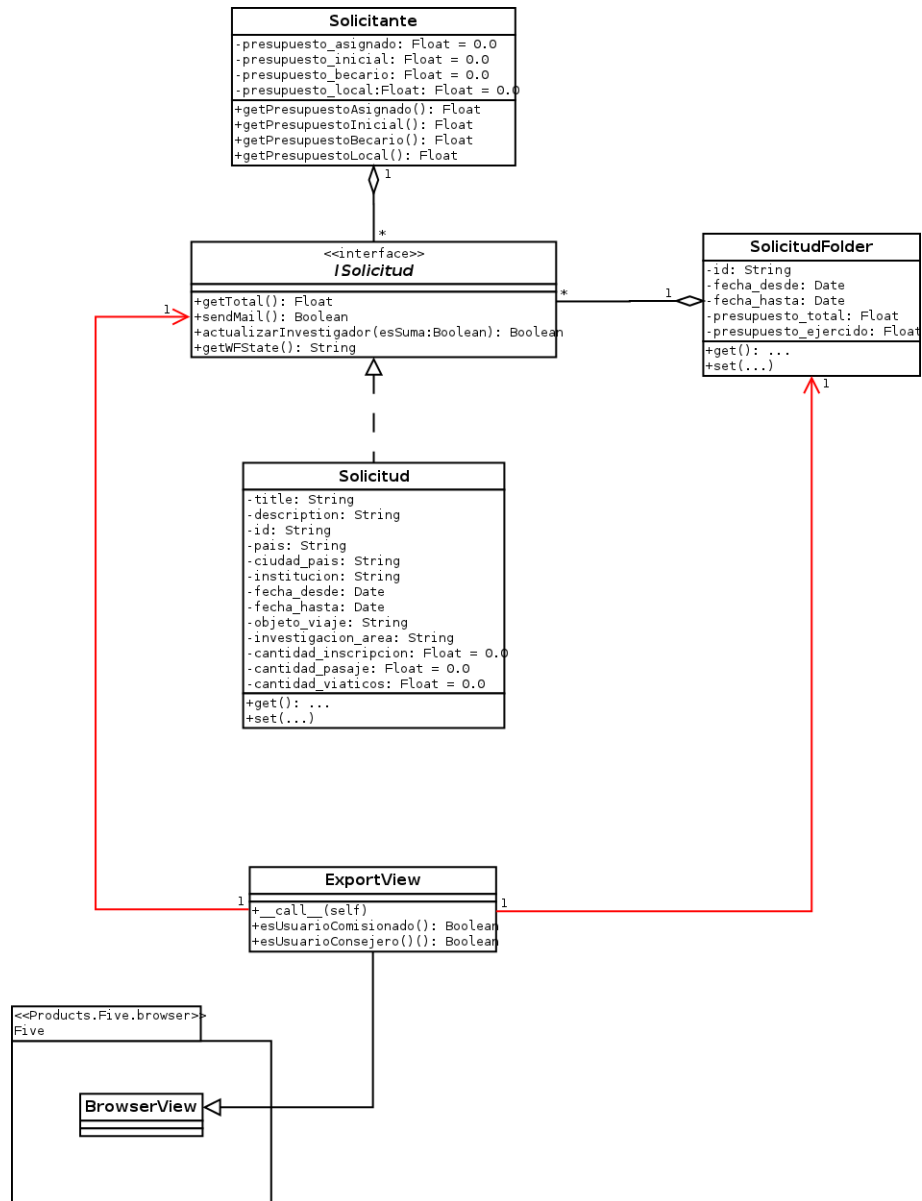
## 3. presupuesto\_becario

**Tipo:** Float**Valor inicial:** 0.0**Acceso:** Privado**Descripción:** Guarda la cantidad total máxima de presupuesto para becarios que el solicitante puede usar.

## 4. presupuesto\_local

**Tipo:** Float**Valor inicial:** 0.0**Acceso:** Privado**Descripción:** Guarda la cantidad total de presupuesto que el usuario ha ejercido (puede ser solo menor o igual al autorizado).

La clase, además, implementa los métodos *get* necesarios para acceder a la información guardada en los campos especificados en el cuadro anterior, sin embargo carece de métodos *set*; es decir, la clase no permite la modificación de éstos. El objetivo de esta clase es comunicar información financiera de cada usuario del sistema sin comprometer la integridad de los datos. El sistema implementa un adaptador para obtener los objetos *Solicitante* desde objetos *FSDPerson*, sin necesidad de modificar el código del producto *Products.FacultyStaffDirectory* ni instanciar clases adicionales.

3.4.2. La interfaz *ISolicitud*Figura 3.4: Interfaz *ISolicitud*

La interfaz marcadora *ISolicitud* es el estereotipo básico para los distintos tipos de solicitud dentro del sistema. Todas las solicitudes creadas son marcadas e identificadas como tales mediante ésta. La clase *Solicitud*, de la que heredan las demás solicitudes del sistema (como se verá en la sección 3.4.3), implementa la interfaz *ISolicitud*; con esto, se asegura que todas las instancias creadas quedarán marcadas de forma adecuada.

La clase *Solicitante* está relacionada con esta interfaz en el sentido de que cada solicitante puede tener y guardar un número ilimitado de solicitudes (hasta donde las limitaciones técnicas del almacenamiento lo permitan).

Obliga al desarrollador a implementar los siguientes métodos públicos:

1. `getTotal()`

**Descripción:** Calcula la cantidad total de dinero solicitada en esta solicitud.

**Regresa:** Float. Regresa la cantidad total que calculó.

2. `sendMail()`

**Descripción:** Manda un correo electrónico al propietario de la solicitud.

**Regresa:** Boolean. Regresa “True” si tiene éxito o “False” si no.

3. `actualizarInvestigador(Boolean esSuma)`

**Parámetros:** `esSuma: Boolean`

**Descripción:** Si `esSuma==“True”`, suma la cantidad regresada por `getTotal()` al presupuesto aprobado para un investigador. Si `esSuma==“False”`, resta la cantidad.

**Regresa:** Boolean. Regresa “True” si tiene éxito o “False” si no.

4. `getWFState()`

**Descripción:** Manda a llamar a una herramienta de Plone para determinar el nombre del estado actual dentro del flujo de trabajo de la solicitud.

**Regresa:** String. El identificador del estado actual del flujo de trabajo de ésta solicitud.

Las instancias marcadas con la interfaz *ISolicitud* pueden almacenarse dentro del tipo contenedor *SolicitudFolder*, descrito a detalle en la sección 3.4.4.

### Clase *ExportView*

La clase *ExportView* es una clase asociada a una vista (un archivo HTML) que solo puede ser llamada mediante los objetos marcados por *ISolicitud* o *SolicitudFolder*. Hereda del tipo *BrowserView*, una clase del paquete *Products.Five.Browser*. Contiene un *script* de exportación de solicitudes en texto simple; desde la vista asociada, se puede llamar a los métodos definidos por *ExportView* para calcular, en tiempo de visualización, las operaciones básicas y avanzadas de exportación, al tiempo que genera un archivo de texto simple que va hacia el usuario mediante un flujo en el objeto *response* creado al abrir sesión en Plone.

Tiene los siguientes métodos públicos:

1. `__call__(self)`

**Descripción:** Crea un documento de texto simple con la información de una *ISolicitud*, o un documento de texto simple con la información de todas las solicitudes dentro de un *SolicitudFolder*. El archivo creado se manda directamente al objeto *response* de la sesión abierta.

**Regresa:** Null.

2. `esComisionado()`

**Descripción:** Revisa si el usuario que manda a llamar `__call__()` tiene rol de Comisionado. Es un método auxiliar que se usa para determinar los permisos de importación a la hora de volcar información exclusiva para Comisionados en la solicitud.

**Regresa:** Boolean. Regresa “True” si es o “False” si no.

3. `esConsejero()`

**Descripción:** Revisa si el usuario que manda a llamar `__call__()` tiene rol de Consejero. Es un método auxiliar que se usa para determinar los permisos de importación a la hora de volcar información exclusiva para Consejeros en la solicitud.

**Regresa:** Boolean. Regresa “True” si es o “False” si no.

### 3.4.3. Estructura de las clases de solicitudes

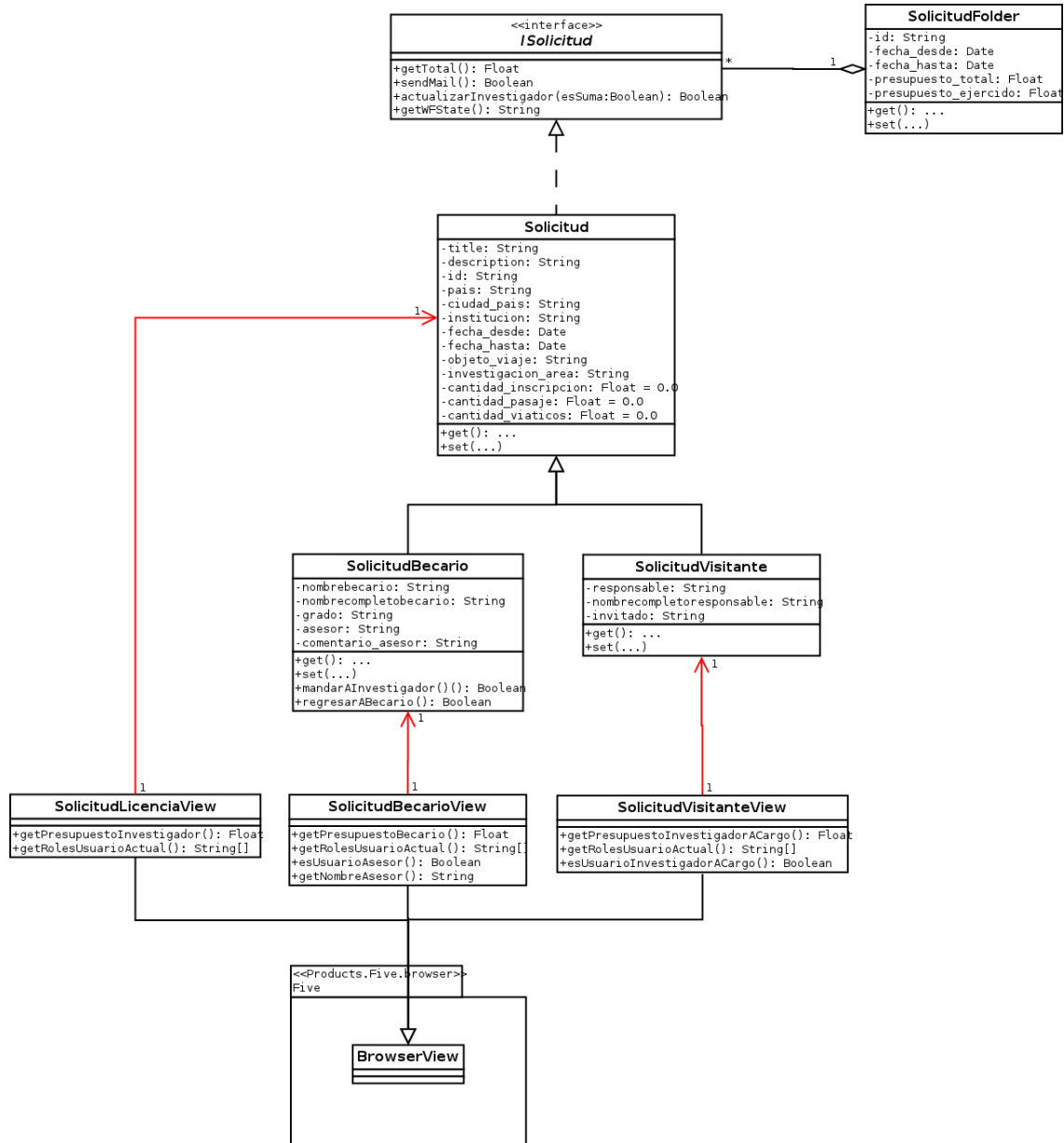


Figura 3.5: Clases de Solicitudes

La clase *Solicitud* es la clase de la que heredan todas las solicitudes del sistema. Ésta implementa la interfaz *ISolicitud* descrita en la sección 3.4.2 y contiene los campos especificados en la sección 3.1.1. Esta clase, además de implementar los métodos de *ISolicitud*, implementa los métodos *get* y *set* para cada uno de sus campos (todos son privados).

De *Solicitud*, heredan las clases *SolicitudBecario* y *SolicitudVisitante*, que añaden información y métodos adicionales para manipular solicitudes de Becario y de Visitante.

te, respectivamente. Los campos adicionales de éstas solicitudes se encuentran en las secciones 3.1.2 y 3.1.3 de este documento.

El tipo *SolicitudBecario* implementa dos métodos adicionales que son necesarios para una revisión previa por el asesor del estudiante que crea la solicitud:

1. `mandarAInvestigador()`

**Descripción:** Busca al asesor del becario en la base de datos, y cambia la propiedad del objeto del estudiante hacia el investigador para su revisión. Manda un correo avisando al asesor que existe una solicitud pendiente de revisión.

**Regresa:** Boolean. Regresa “True” si tiene éxito o “False” si no.

2. `regresarABecario()`

**Descripción:** Cambia la propiedad del objeto del asesor registrado en la solicitud al estudiante que la creó para su corrección. Manda un correo avisando al becario que existe una solicitud pendiente de revisión.

**Regresa:** Boolean. Regresa “True” si tiene éxito o “False” si no.

### **SolicitudLicenciaView**

La clase *SolicitudLicenciaView* es una clase asociada a una vista (un archivo HTML) que solo puede ser llamada mediante los objetos que son instancia de *Solicitud*. Hereda del tipo *BrowserView*, del paquete *Products.Five.Browser*. Contiene métodos necesarios para el despliegue de información correctamente. A diferencia de la clase *ExportView* de la sección 3.4.2, los métodos no realizan funciones complejas: solo restringen el acceso a los diferentes campos dependiendo de los permisos del usuario que manda a llamar la vista. Al igual que en la mayoría de las *BrowserViews*, los métodos obtienen información de un objeto de sesión *request* y la regresan a un objeto *response*, ambos creados al iniciar sesión en Plone.

Los métodos que define son:

1. `getPresupuestoInvestigador()`

**Descripción:** Instancia un objeto *Solicitante* asociado al usuario que creó la solicitud y regresa el presupuesto total asignado.

**Regresa:** Float. Regresa el presupuesto total asignado al usuario que llama la función.

2. `getRolesUsuarioActual()`

**Descripción:** Obtiene los roles del usuario que llama la función.

**Regresa:** `String[]`. Regresa un arreglo con los roles que tiene asociado el usuario.

### SolicitudBecarioView

La clase *SolicitudBecarioView* es una clase asociada a una vista (un archivo HTML) que solo puede ser llamada mediante los objetos que son instancia de *SolicitudBecario*. Hereda del tipo *BrowserView*, del paquete *Products.Five.Browser* y contiene métodos necesarios para el despliegue de información correctamente en solicitudes de becario.

Los métodos que define son:

1. `getPresupuestoBecario()`

**Descripción:** Instancia un objeto *Solicitante* asociado al usuario que creó la solicitud y regresa el presupuesto total asignado.

**Regresa:** Float. Regresa el presupuesto total asignado al usuario que llama la función.

2. `getRolesUsuarioActual()`

**Descripción:** Obtiene los roles del usuario que llama la función.

**Regresa:** `String[]`. Regresa un arreglo con los roles que tiene asociado el usuario.

3. `esUsuarioAsesor()`

**Descripción:** Revisa si el usuario actual es el asesor del estudiante que creó la solicitud.

**Regresa:** Boolean. Regresa “True” si el usuario es el asesor del estudiante que creó la solicitud o “False” si no.

4. `getNombreAsesor()`

**Descripción:** Busca el nombre completo del asesor del estudiante que creó la solicitud.

**Regresa:** String. Regresa el nombre del asesor.

### SolicitudVisitanteView

La clase *SolicitudVisitanteView* es una clase asociada a una vista (un archivo HTML) que solo puede ser llamada mediante los objetos que son instancia de *SolicitudVisitante*. Hereda del tipo *BrowserView*, del paquete *Products.Five.Browser* y contiene métodos necesarios para el despliegue de información correctamente en solicitudes de visitante.

Los métodos que define son:

1. `getPresupuestoInvestigadorACargo()`

**Descripción:** Instancia un objeto *Solicitante* asociado al usuario que creó la solicitud y regresa el presupuesto total asignado.

**Regresa:** Float. Regresa el presupuesto total asignado al usuario que llama la función.

2. `getRolesUsuarioActual()`

**Descripción:** Obtiene los roles del usuario que llama la función.

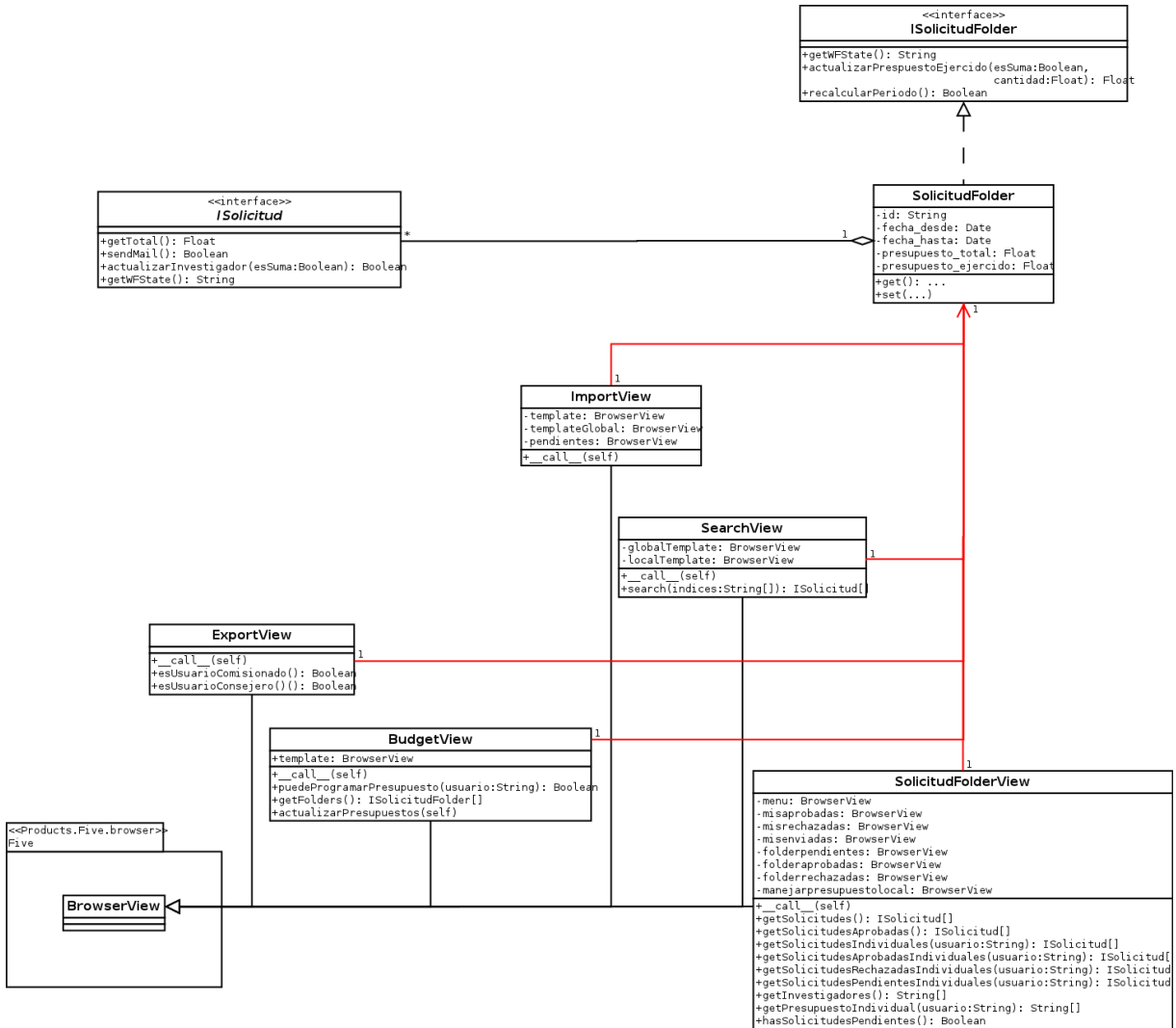
**Regresa:** `String[]`. Regresa un arreglo con los roles que tiene asociado el usuario.

3. `esUsuarioInvestigadorACargo()`

**Descripción:** Revisa si el usuario actual es el investigador que registró al invitado en su solicitud.

**Regresa:** Boolean. Regresa “True” si el usuario es el investigador encargado del visitante o “False” si no.



3.4.4. La clase *SolicitudFolder*Figura 3.6: Clase *SolicitudFolder*

La clase *SolicitudFolder* es una clase contenedora de objetos *ISolicitud* que representa un periodo del tiempo dado con un presupuesto tope. El conjunto de solicitudes aprobadas dentro de una instancia de *SolicitudFolder*, representan la cantidad de presupuesto asignado en dicho periodo. Los campos privados de este tipo de dato se encuentran listados en la sección 3.1.4 documento. La clase implementa los métodos *get* y *set* para cada uno de los campos.

La clase *SolicitudFolder* implementa la interfaz marcada *ISolicitudFolder* que define los siguientes métodos:

1. `getWFState()`

**Descripción:** Manda a llamar a una herramienta de Plone para determinar el nombre del estado actual dentro del flujo de trabajo de la solicitud.

**Regresa:** String. El identificador del estado actual del flujo de trabajo de ésta solicitud.

2. `actualizarPresupuestoEjercido(esSuma:Boolean,cantidad:Float)`

**Parámetros:** `esSuma:Boolean`, `cantidad:Float`

**Descripción:** Si `esSuma==“True”`, suma la cantidad ingresada en los parámetros al presupuesto ejercido, es decir, lo que ya ha sido usado. Si `esSuma==“False”`, resta la cantidad.

**Regresa:** Float. Regresa la cantidad total que calculó.

3. `recalcularPeriodo()`

**Descripción:** Revisa todas las solicitudes contenidas y usa el método *actualizarPresupuestoEjercido()* para actualizar los datos al tiempo que revisa cada solicitud.

**Regresa:** Boolean. Regresa “True” si tiene éxito o “False” si no.

### La clase *SolicitudFolderView*

La clase *SolicitudFolderView* es una clase asociada a una vista (un archivo HTML) que solo puede ser llamada mediante los objetos que son instancia de *SolicitudFolder*. Hereda del tipo *BrowserView*, del paquete *Products.Five.Browser* y contiene métodos necesarios para el despliegue de información de las solicitudes dentro de un folder.

En particular, *SolicitudFolderView* posee una serie de métodos que permiten obtener información sobre las solicitudes que se encuentran contenidas en el objeto *SolicitudFolder* asociado a la vista. Utilizando esos métodos, se construye la interfaz de usuario (descrita en el capítulo 4) y se despliegan los menús de navegación del producto.

La clase contiene en sus campos varios objetos de tipo *BrowserView* que se instancian automáticamente en el constructor y se ligan a archivos HTML que describen la distribución de elementos visuales del producto. Los objetos guardados en estos campos, permiten llamar archivos HTML dentro del cuerpo de la clase y desplegar vistas automáticamente mandando flujos de código HTML al objeto *response* de la sesión actual del sitio Plone. De esta forma, llamadas consecutivas al método `__call__` de la clase *SolicitudFolderView* puede resultar en el despliegue de información distinta dependiendo del estado guardado en los objetos de sesión *request* y *response*.

Los campos de la clase son:

1. menu

**Tipo:** BrowserView

**Acceso:** Privado

**Descripción:** Campo que genera automáticamente una instancia BrowserView asociada a una vista HTML que describe la distribución del menú general del producto solicitudes.

2. misaprobadas

**Tipo:** BrowserView

**Acceso:** Privado

**Descripción:** Campo que genera automáticamente una instancia BrowserView asociada a una vista HTML que describe la distribución en pantalla de una lista de solicitudes aprobadas.

3. misrechazadas

**Tipo:** BrowserView

**Acceso:** Privado

**Descripción:** Campo que genera automáticamente una instancia BrowserView asociada a una vista HTML que describe la distribución en pantalla de una lista de solicitudes rechazadas.

4. misenviadas

**Tipo:** BrowserView

**Acceso:** Privado

**Descripción:** Campo que genera automáticamente una instancia BrowserView asociada a una vista HTML que describe la distribución en pantalla de una lista de solicitudes en transición.

5. folderaprobadas

**Tipo:** BrowserView

**Acceso:** Privado

**Descripción:** Campo que genera automáticamente una instancia BrowserView asociada a una vista HTML que describe la distribución en pantalla de una lista de solicitudes aprobadas y que tienen permiso de acceso público.

## 6. folderrechazadas

**Tipo:** `BrowserView`**Acceso:** Privado**Descripción:** Campo que genera automáticamente una instancia `BrowserView` asociada a una vista HTML que describe la distribución en pantalla de una lista de solicitudes rechazadas y que tienen permiso de acceso público.

## 7. folderpendientes

**Tipo:** `BrowserView`**Acceso:** Privado**Descripción:** Campo que genera automáticamente una instancia `BrowserView` asociada a una vista HTML que describe la distribución en pantalla de una lista de solicitudes en transición y que tienen permiso de acceso público.

## 8. manejarpresupuestolocal

**Tipo:** `BrowserView`**Acceso:** Privado**Descripción:** Campo que genera automáticamente una instancia `BrowserView` asociada a una vista HTML que describe la distribución en pantalla de una pantalla para cambiar los datos de presupuesto del folder asociado a la instancia actual de *SolicitudFolderView*.

A continuación, se muestra una breve descripción de los métodos que *SolicitudFolderView* implementa para desplegar información. La mayoría son utilizados dentro del método `__call__` para validar información de seguridad antes de soltar el objeto *response* de la sesión actual. Desde el código HTML asociado a cada objeto *BrowserView* en los campos de *SolicitudFolderView* también se pueden llamar éstos métodos, sin embargo algunos regresan NULL como resultado si el usuario no posee los permisos de acceso adecuados.

1. `__call__(self)`**Descripción:** Manda a llamar a la vista “menu”, generada automáticamente desde un archivo HTML con la descripción del menú, y la manda al objeto *response* asociado a la sesión actual de Plone.**Regresa:** Null.2. `getSolicitudes()`**Descripción:** Regresa un arreglo con todos los objetos marcados como *ISolicitud* contenidos dentro del folder. El archivo HTML asociado se encarga de extraer la información de cada uno de los miembros del arreglo para su despliegue en forma de lista.

**Regresa:** ISolicitud[]. Un arreglo de objetos marcados por la interfaz *ISolicitud*.

3. getSolicitudesAprobadas()

**Descripción:** Regresa un arreglo con todos los objetos marcados como *ISolicitud* contenidos dentro del folder, cuyo estado de flujo de corresponda a una solicitud aprobada (*i.e.* su estado es “aprobada”). El archivo HTML asociado se encarga de extraer la información de cada uno de los miembros del arreglo para su despliegue en forma de lista.

**Regresa:** ISolicitud[]. Un arreglo de objetos marcados por la interfaz *ISolicitud*.

4. getSolicitudesIndividuales(usuario:String)

**Parámetros:** usuario:String, el nombre de usuario de quien queremos obtener las solicitudes.

**Descripción:** Regresa un arreglo con todos los objetos marcados como *ISolicitud* contenidos dentro del folder, con la condición de haber sido creados por el usuario cuyo nombre se pasa en los parámetros. El archivo HTML asociado se encarga de extraer la información de cada uno de los miembros del arreglo para su despliegue en forma de lista.

**Regresa:** ISolicitud[]. Un arreglo de objetos marcados por la interfaz *ISolicitud*.

5. getSolicitudesAprobadasIndividuales(usuario:String)

**Parámetros:** usuario:String, el nombre de usuario de quien queremos obtener las solicitudes.

**Descripción:** Regresa un arreglo con todos los objetos marcados como *ISolicitud* contenidos dentro del folder, con la condición de haber sido creados por el usuario cuyo nombre se pasa en los parámetros y cuyos estados correspondan al de una solicitud aprobada. El archivo HTML asociado se encarga de extraer la información de cada uno de los miembros del arreglo para su despliegue en forma de lista.

**Regresa:** ISolicitud[]. Un arreglo de objetos marcados por la interfaz *ISolicitud*.

6. getSolicitudesRechazadasIndividuales(usuario:String)

**Parámetros:** usuario:String, el nombre de usuario de quien queremos obtener las solicitudes.

**Descripción:** Regresa un arreglo con todos los objetos marcados como *ISolicitud* contenidos dentro del folder, con la condición de haber sido creados por el usuario cuyo nombre se pasa en los parámetros y cuyos estados correspondan al de una solicitud rechazada. El archivo HTML asociado se encarga de extraer la información de cada uno de los miembros del arreglo para su despliegue en forma de lista.

**Regresa:** ISolicitud[]. Un arreglo de objetos marcados por la interfaz *ISolicitud*.

7. `getSolicitudesPendientesIndividuales(usuario:String)`

**Parámetros:** `usuario:String`, el nombre de usuario de quien queremos obtener las solicitudes.

**Descripción:** Regresa un arreglo con todos los objetos marcados como *ISolicitud* contenidos dentro del folder, con la condición de haber sido creados por el usuario cuyo nombre se pasa en los parámetros y cuyos estados correspondan al de una solicitud que está en tránsito (*i.e.* no está ni aprobada ni rechazada). El archivo HTML asociado se encarga de extraer la información de cada uno de los miembros del arreglo para su despliegue en forma de lista.

**Regresa:** `ISolicitud[]`. Un arreglo de objetos marcados por la interfaz *ISolicitud*.

8. `getInvestigadores()`

**Descripción:** Regresa un arreglo con todos los nombres de usuario de los investigadores que tienen una solicitud creada dentro del folder asociado a ésta instancia de la clase.

**Regresa:** `String[]`. Un arreglo con los nombres de usuario de los investigadores que han creado solicitudes en el folder asociado a ésta instancia de la clase.

9. `getPresupuestoIndividual(usuario:String)`

**Parámetros:** `usuario:String`, el nombre de usuario de quien queremos obtener las solicitudes.

**Descripción:** Instancia un objeto de tipo *Solicitante* asociado al usuario cuyo nombre se pasa en los parámetros. Guarda los valores de presupuesto en un arreglo de tipo `String`. El archivo HTML asociado se encarga de procesar esa información para su correcto despliegue.

**Regresa:** `String[]`. La lista de valores que representa los campos de presupuesto de un objeto *Solicitante*.

10. `hasSolicitudesPendientes()`

**Descripción:** Determina si el usuario que llama la función tiene solicitudes creadas en tránsito (*i.e.* que no estén ni rechazadas ni aprobadas).

**Regresa:** `Boolean`. Regresa “True” si el usuario tiene solicitudes pendientes y “False” si no.

### La clase *ImportView*

La clase *ImportView* es una clase asociada a una vista (un archivo HTML) que solo puede ser llamada mediante los objetos que son instancia de *SolicitudFolder*. Hereda del tipo *BrowserView*, del paquete *Products.Five.Browser* y contiene métodos necesarios para desplegar el formulario y procesar las solicitudes de importación.

Esta clase posee tres campos asociados de tipo *BrowserView*. Estos campos son generados automáticamente al instanciar la clase, a partir de los archivos HTML que describen la distribución de información en los formularios de importación. Éstos son:

1. `template`**Tipo:** `BrowserView`**Acceso:** Privado**Descripción:** Campo que genera automáticamente una instancia `BrowserView` asociada a una vista HTML que describe la distribución de una pantalla de importación para usuarios sin permiso de importación global.2. `templateGlobal`**Tipo:** `BrowserView`**Acceso:** Privado**Descripción:** Campo que genera automáticamente una instancia `BrowserView` asociada a una vista HTML que describe la distribución de una pantalla de importación para usuarios con permiso de importación global.3. `pendientes`**Tipo:** `BrowserView`**Acceso:** Privado**Descripción:** Campo que genera automáticamente una instancia `BrowserView` asociada a una vista HTML que describe la distribución en pantalla de una lista de solicitudes en transición. Se despliega si la importación tuvo éxito.

Finalmente, el método `__call__` se encarga de procesar la información recabada de los objetos de sesión y hace o rechaza las solicitudes de importación en cada caso:

1. `__call__(self)`**Descripción:** Intenta recuperar, del objeto `request` asociado a la sesión actual de Plone, un archivo de texto. Si lo logra, lo intenta procesar para instanciar un objeto solicitud a partir de la información de texto.**Regresa:** Null.**La clase `SearchView`**

La clase `SearchView` es una clase asociada a una vista (un archivo HTML) que solo puede ser llamada mediante los objetos que son instancia de `SolicitudFolder`. Hereda del tipo `BrowserView`, del paquete `Products.Five.Browser` y contiene métodos necesarios para desplegar el formulario y procesar las solicitudes de búsqueda.

Esta clase posee dos campos asociados de tipo `BrowserView`. Estos campos son generados automáticamente al instanciar la clase, a partir de los archivos HTML que describen la distribución de información en los formularios de búsqueda:

## 1. globalTemplate

**Tipo:** BrowserView**Acceso:** Privado**Descripción:** Campo que genera automáticamente una instancia BrowserView asociada a una vista HTML que describe la distribución de una pantalla de búsqueda de solicitudes en todos los folders del sistema.

## 2. localTemplate

**Tipo:** BrowserView**Acceso:** Privado**Descripción:** Campo que genera automáticamente una instancia BrowserView asociada a una vista HTML que describe la distribución de una pantalla de búsqueda de solicitudes en el folder asociado a la instancia actual de *SearchView*.

En esta clase, el método `__call__` se encarga de verificar y acomodar la información a buscar, mientras que el método `search()` es el que efectivamente revisa la información del catálogo de Plone y regresa los objetos encontrados:

1. `__call__(self)`**Descripción:** Intenta recuperar, del objeto *request* asociado a la sesión actual de Plone, una serie de campos e identificadores indexables. Si hay información, la busca en el catálogo del sitio mediante la función `search()` y manda un arreglo con la lista de objetos *ISolicitud* encontrados al objeto *response* asociado a la sesión actual.**Regresa:** Null.2. `search()`**Parametros:** `indices:String[]`, un arreglo con la información a buscar en el catálogo.**Descripción:** Busca en el catálogo los objetos *ISolicitud* que cumplan con los criterios especificados por el campo “indices”.**Regresa:** `ISolicitud[]`. Un arreglo con todos los objetos *ISolicitud* encontrados.**La clase *BudgetView***

La clase *BudgetView* es una clase asociada a una vista (un archivo HTML) que solo puede ser llamada mediante los objetos que son instancia de *SolicitudFolder*. Hereda del tipo *BrowserView*, del paquete *Products.Five.Browser* y contiene métodos necesarios para desplegar el formulario de manejo de presupuesto. Tiene un campo asociado:



1. `template`**Tipo:** `BrowserView`**Acceso:** Privado**Descripción:** Campo que genera automáticamente una instancia `BrowserView` asociada a una vista HTML que describe la distribución de una pantalla de manejo de presupuesto.

Un método `__call__` se encarga de verificar los permisos del usuario que intenta acceder a esta vista y procesar todas las peticiones de cambios o actualización usando métodos auxiliares:

1. `__call__(self)`**Descripción:** Determina si el usuario tiene capacidades de manejo de presupuesto global o solo su propio presupuesto. Regresa la vista apropiada para cada caso al objeto *response* asociado a la sesión actual de Plone.**Regresa:** `Null`.2. `puedeProgramarPresupuesto(usuario:String)`**Parametros:** `usuario:String`, el nombre de un usuario del sitio.**Descripción:** Usa una herramienta de Plone para buscar un usuario y determinar si tiene el rol necesario para manejo global de presupuestos.**Regresa:** Boolean. Regresa “True” si el usuario puede manejar presupuesto globalmente o “False” si no puede hacerlo.3. `getFolders()`**Parametros:** `usuario:String`, el nombre de un usuario del sitio.**Descripción:** Busca en el catálogo todos los objetos marcados con la interfaz *ISolicitudFolder* y los regresa.**Regresa:** `ISolicitudFolder[]`. Regresa el conjunto de objetos dentro del sitio marcados como *ISolicitudFolder*.4. `actualizarPresupuestos(self)`**Descripción:** Busca en el catálogo todos los objetos marcados con la interfaz *ISolicitudFolder* y los recorre uno por uno haciendo operaciones aritméticas hasta actualizar las cuentas totales. Regresa una vista asociada a un objeto HTML al objeto *response* asociado a la sesión actual.**Regresa:** `Null`.

## 3.5. Flujos de Información Del Sistema

En la figura 3.7 se muestra mediante un diagrama de secuencia el flujo general de solicitudes dentro del sistema. A continuación, se describen a detalle los permisos que los *workflows* manejan por estado.

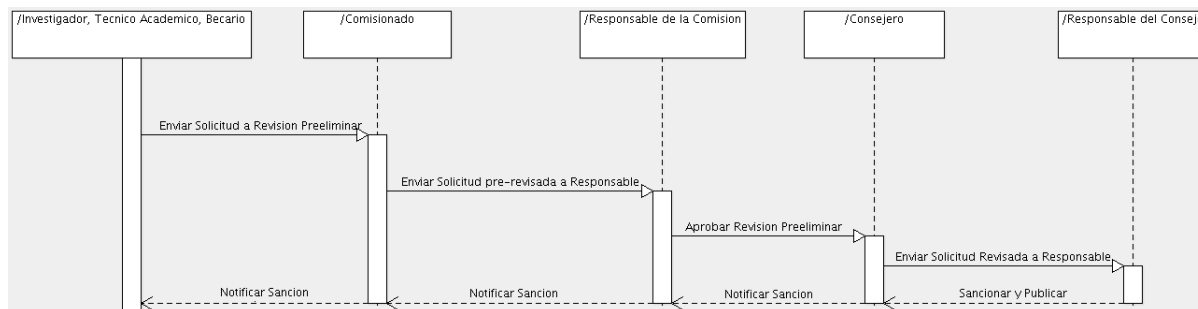


Figura 3.7: Flujo General de Solicitudes

### 3.5.1. Flujo de trabajo para solicitudes de Licencia (meta-tipo Solicitud) y Visitante (meta-tipo SolicitudVisitante)

Los permisos que maneja este *workflow* son los siguientes:

- *Change portal events* (permiso de sistema): Permite cambiar el estado de un contenido.
- *Modify portal content* (permiso de sistema): Permite modificar información de un contenido.
- *Manage properties* (permiso de sistema): Permite acceder información de las propiedades de usuario.
- *Delete objects* (permiso de sistema): Permite eliminar un contenido.
- *View*: Permite ver los datos de un contenido
- *Solicitud: Comision Revisa Solicitud*: ver sección “Permisos”.
- *Solicitud: Consejo Revisa Solicitud*: ver sección “Permisos”.
- *Solicitud: Consejo Cambia Solicitud*: ver sección “Permisos”.
- *Solicitud: Modificar Solicitud*: ver sección “Permisos”.

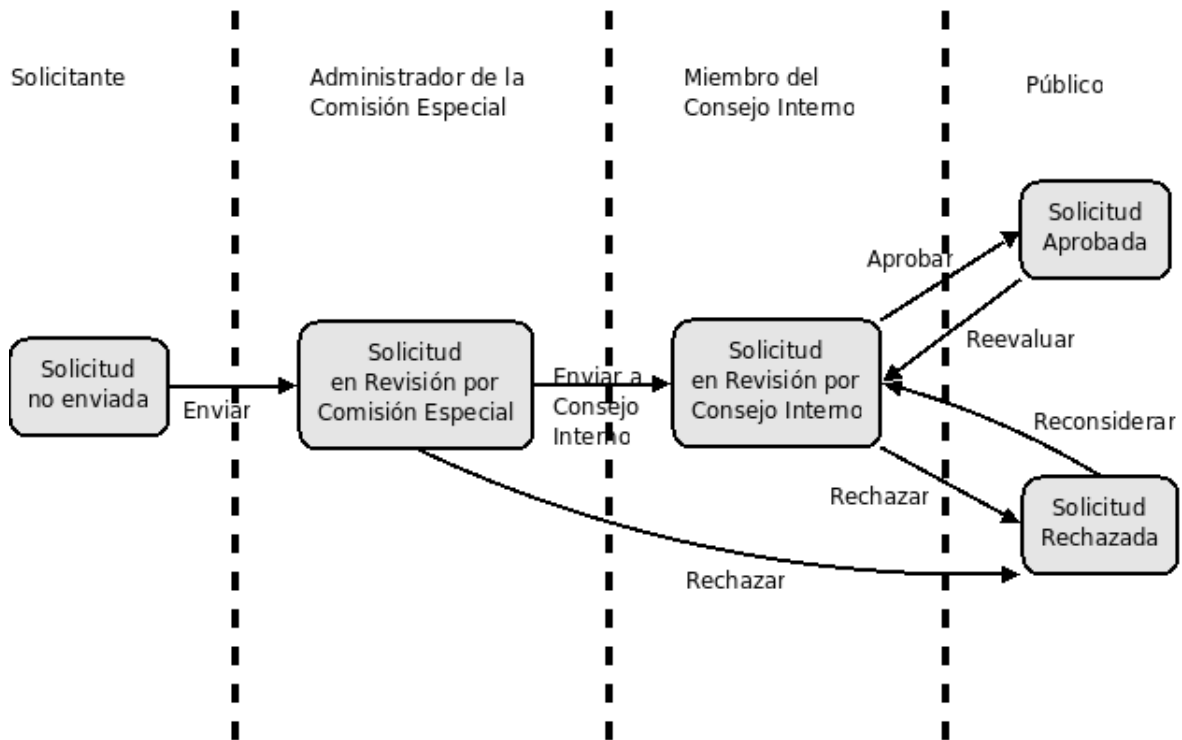


Figura 3.8: *Workflow* de Solicitudes de Licencia y Visitante

Los estados del *workflow* están definidos como sigue:

1. *Solicitud en borrador*: **id** = borrador

**Change portal events :**

- Owner

**Modify portal content :**

- Manager
- Owner

**Delete objects :**

- Manager
- Owner

**Manage properties :**

- Manager

**View :**

- Manager
- Owner

**Solicitud: Comision Revisa Solicitud :**

- -

**Solicitud: Consejo Revisa Solicitud :**

- -

**Solicitud: Consejo Cambia Solicitud :**

- -

**Solicitud: Modificar Solicitud :**

- Manager
- Owner

+ Transición de Salida:

- “Enviar solicitud” **id**=enviar, destino estado = revisioncomision

+ Descripción: Permite a usuarios con los roles de Investigador y Técnico Académico crear una nueva solicitud, modificarla y enviarla.

2. *Solicitud en revisión por Comisión Especial:* **id** = revisioncomision

**Change portal events :**

- Manager
- Responsable de la Comisión

**Modify portal content :**

- Manager
- Comisionado
- Responsable de la Comisión

**Delete objects :**

- Manager

**Manage properties :**

- Manager
- Comisionado
- Responsable de la Comisión

**View :**

- Manager
- Owner
- Comisionado
- Consejero
- Responsable de la Comisión
- Responsable del Consejo

**Solicitud: Comisión Revisa Solicitud :**

- Manager

- Comisionado
- Responsable de la Comision

**Solicitud: Consejo Revisa Solicitud :**

- -

**Solicitud: Consejo Cambia Solicitud :**

- -

**Solicitud: Modificar Solicitud :**

- -

+ Transición de Salida:

- “Aprobar y enviar a Consejo Interno” **id**=enviaraconsejo, destino estado = revisionconsejo
- “Solicitud Rechazada” **id**=rechazar, destino estado = rechazada

+ Descripción: Permite a usuarios con los roles de “Comisionado” y “Responsable de la Comisión”, modificar los campos reservados para la Comisión Especial en la solicitud. Además, permite al rol de “Responsable de la Comisión” enviar la solicitud al consejo o rechazar la solicitud.

3. *Solicitud en revisión por Consejo Interno:* **id** = revisionconsejo

**Change portal events :**

- Manager
- Responsable del Consejo

**Modify portal content :**

- Manager
- Consejero
- Responsable del Consejo

**Delete objects :**

- Manager

**Manage properties :**

- Manager
- Consejero
- Responsable del Consejo

**View :**

- Manager
- Owner
- Comisionado
- Consejero
- Responsable de la Comision

- Responsable del Consejo

**Solicitud: Comision Revisa Solicitud :**

- -

**Solicitud: Consejo Revisa Solicitud :**

- Manager
- Consejero
- Responsable del Consejo

**Solicitud: Consejo Cambia Solicitud :**

- -

**Solicitud: Modificar Solicitud :**

- -

+ Transición de Salida:

- “Solicitud Aprobada” **id**=aprobar, destino estado = aprobada
- “Solicitud Rechazada” **id**=rechazar, destino estado = rechazada

+ Descripción: Permite a usuarios con los roles de “Consejero” y “Responsable del Consejo”, modificar los campos reservados para el Consejo Interno en la solicitud. Además, permite al rol de “Responsable del Consejo” aprobar o rechazar la solicitud.

4. *Solicitud Aprobada*: **id** = aprobada

**Change portal events :**

- Manager
- Responsable del Consejo

**Modify portal content :**

- Manager
- Consejero
- Responsable del Consejo

**Delete objects :**

- Manager
- Responsable del Consejo

**Manage properties :**

- Manager
- Consejero
- Responsable del Consejo

**View :**

- Manager

- Owner
- Comisionado
- Consejero
- Investigador
- Técnico Académico
- Importador de Solicitudes
- Responsable de la Comision
- Responsable del Consejo

**Solicitud: Comision Revisa Solicitud :**

- -

**Solicitud: Consejo Revisa Solicitud :**

- -

**Solicitud: Consejo Cambia Solicitud :**

- Manager
- Consejero
- Responsable del Consejo

**Solicitud: Modificar Solicitud :**

- -

+ Transición de Salida:

- “Regresar solicitud al Consejo Interno” **id**=rechazararevisor, destino estado=revisionconsejo
- “Avisar al Investigador sobre modificaciones” **id**=avisarinvestigador, destino estado=aprobada

+ Descripción: La solicitud queda aprobada y cuenta dentro del presupuesto asignado. Éste estado permite a los usuarios con rol “Consejero” y “Responsable del Consejo” cambiar la cantidad ejercida en la solicitud (que debe ser menor o igual al monto aprobado). En este estado los usuarios con rol “Responsable del Consejo” pueden eliminar una solicitud.

5. *Solicitud Rechazada*: **id** = rechazada

**Change portal events :**

- Manager
- Responsable del Consejo

**Modify portal content :**

- Manager
- Consejero
- Responsable del Consejo

**Delete objects :**

- Manager
- Responsable del Consejo

**Manage properties :**

- Manager

**View :**

- Manager
- Owner
- Comisionado
- Consejero
- Investigador
- Técnico Académico
- Importador de Solicitudes
- Responsable de la Comision
- Responsable del Consejo

**Solicitud: Comision Revisa Solicitud :**

- Manager
- Comisionado
- Responsable de la Comision

**Solicitud: Consejo Revisa Solicitud :**

- Manager
- Consejero
- Responsable del Consejo

**Solicitud: Consejo Cambia Solicitud :**

- Manager
- Consejero
- Responsable del Consejo

**Solicitud: Modificar Solicitud :**

- -

**+ Transición de Salida:**

- “Reconsiderar Solicitud” **id**=reconsiderar, destino estado=revisiconconsejo

+ Descripción: La solicitud queda rechazada y **no** cuenta dentro del presupuesto asignado. En éste estado los usuarios con rol “Responsable del Consejo” pueden eliminar una solicitud.



### 3.5.2. Flujo de trabajo para Solicitudes de Becario (meta-tipo SolicitudBecario)

Los permisos que maneja este *workflow* son los siguientes:

- *Change portal events* (permiso de sistema): Permite cambiar el estado de un contenido.
- *Modify portal content* (permiso de sistema): Permite modificar información de un contenido.
- *Manage properties* (permiso de sistema): Permite acceder información de las propiedades de usuario.
- *Delete objects* (permiso de sistema): Permite eliminar un contenido.
- *View*: Permite ver los datos de un contenido
- *Solicitud: Comision Revisa Solicitud*: ver sección “Permisos”.
- *Solicitud: Consejo Revisa Solicitud*: ver sección “Permisos”.
- *Solicitud: Consejo Cambia Solicitud*: ver sección “Permisos”.
- *Solicitud: Modificar Solicitud*: ver sección “Permisos”.

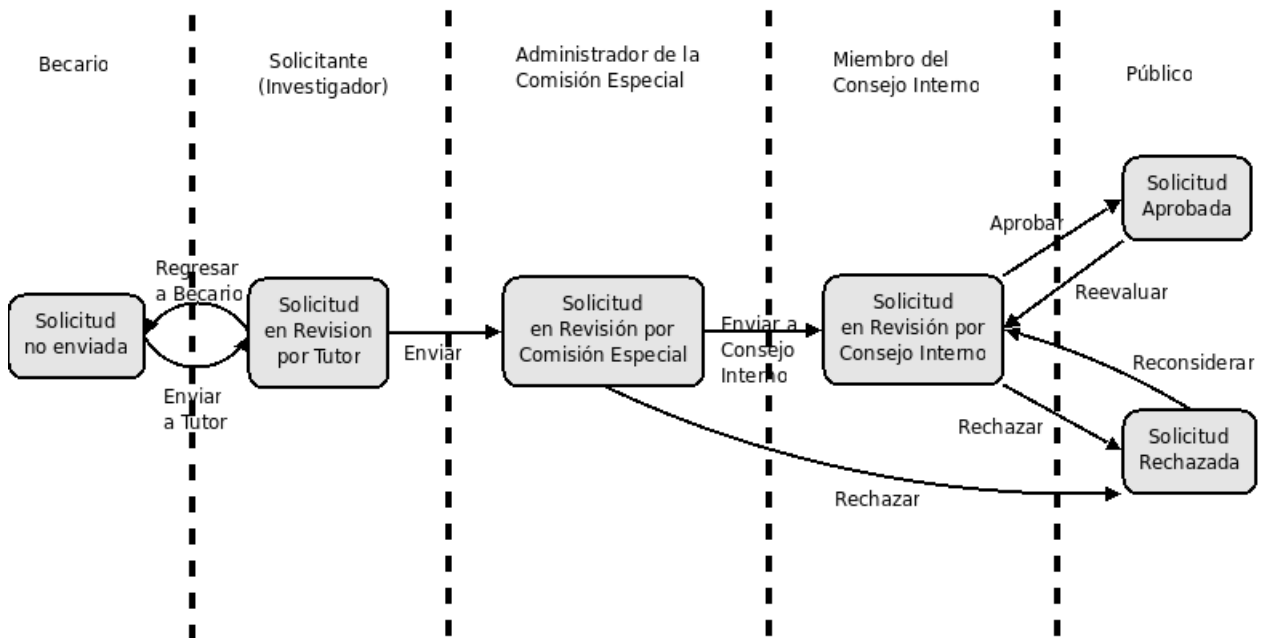


Figura 3.9: *Workflow* de Solicitud de Becario

Los estados del *workflow* están definidos como sigue:

1. *Solicitud en borrador*: **id** = borrador**Change portal events :**

- Owner

**Modify portal content :**

- Manager
- Owner

**Delete objects :**

- Manager
- Owner

**Manage properties :**

- Manager

**View :**

- Manager
- Owner

**Solicitud: Comision Revisa Solicitud :**

- -

**Solicitud: Consejo Revisa Solicitud :**

- -

**Solicitud: Consejo Cambia Solicitud :**

- -

**Solicitud: Modificar Solicitud :**

- Manager
- Owner

## + Transición de Salida:

- “Enviar solicitud” **id**=enviarainvestigador, destino estado = preeliminar

+ Descripción: Permite a usuarios con los roles de Becario crear una nueva solicitud, modificarla y enviarla.

2. *Solicitud en revisión preeliminar*: **id** = preeliminar**Change portal events :**

- Manager
- Investigador

**Modify portal content :**

- Manager
- Investigador

**Delete objects :**

- Manager
- Investigador

**Manage properties :**

- Manager

**View :**

- Manager
- Owner

**Solicitud: Comision Revisa Solicitud :**

- -

**Solicitud: Consejo Revisa Solicitud :**

- -

**Solicitud: Consejo Cambia Solicitud :**

- -

**Solicitud: Modificar Solicitud :**

- Manager
- Investigador

**+ Transición de Salida:**

- “Enviar solicitud” **id**=enviar, destino estado = revisioncomision
- “Regresar solicitud a becario” **id**=rechazarabecario, destino estado = borrador

+ Descripción: Permite a usuarios con los roles de “Investigador” y que son dueños de la solicitud, cambiar los datos y el estado de la misma. La solicitud puede tanto enviarse a la Comisión Especial como retornarla al Becario que la creó.

**3. Solicitud en revisión por Comision Especial: id = revisioncomision****Change portal events :**

- Manager
- Responsable de la Comision

**Modify portal content :**

- Manager
- Comisionado
- Responsable de la Comision

**Delete objects :**

- Manager

**Manage properties :**

- Manager
- Comisionado
- Responsable de la Comision

**View :**

- Manager
- Owner
- Comisionado
- Consejero
- Responsable de la Comision
- Responsable del Consejo

**Solicitud: Comision Revisa Solicitud :**

- Manager
- Comisionado
- Responsable de la Comision

**Solicitud: Consejo Revisa Solicitud :**

- -

**Solicitud: Consejo Cambia Solicitud :**

- -

**Solicitud: Modificar Solicitud :**

- -

**+ Transición de Salida:**

- “Aprobar y enviar a Consejo Interno” **id**=enviaraconsejo, destino estado = revisionconsejo
- “Solicitud Rechazada” **id**=rechazar, destino estado = rechazada

**+ Descripción:** Permite a usuarios con los roles de “Comisionado” y “Responsable de la Comisión”, modificar los campos reservados para la Comisión Especial en la solicitud. Además, permite al rol de “Responsable de la Comisión” enviar la solicitud al consejo o rechazar la solicitud.

**4. Solicitud en revisión por Consejo Interno: id = revisionconsejo****Change portal events :**

- Manager
- Responsable del Consejo

**Modify portal content :**

- Manager
- Consejero
- Responsable del Consejo

**Delete objects :**

- Manager

**Manage properties :**

- Manager
- Consejero
- Responsable del Consejo

**View :**

- Manager
- Owner
- Comisionado
- Consejero
- Responsable de la Comision
- Responsable del Consejo

**Solicitud: Comision Revisa Solicitud :**

- -

**Solicitud: Consejo Revisa Solicitud :**

- Manager
- Consejero
- Responsable del Consejo

**Solicitud: Consejo Cambia Solicitud :**

- -

**Solicitud: Modificar Solicitud :**

- -

**+ Transición de Salida:**

- “Solicitud Aprobada” **id**=aprobar, destino estado = aprobada
- “Solicitud Rechazada” **id**=rechazar, destino estado = rechazada

**+ Descripción:** Permite a usuarios con los roles de “Consejero” y “Responsable del Consejo”, modificar los campos reservados para el Consejo Interno en la solicitud. Además, permite al rol de “Responsable del Consejo” aprobar o rechazar la solicitud.

5. *Solicitud Aprobada:* **id** = aprobada**Change portal events :**

- Manager
- Responsable del Consejo

**Modify portal content :**

- Manager

- Consejero
- Responsable del Consejo

**Delete objects :**

- Manager
- Responsable del Consejo

**Manage properties :**

- Manager
- Consejero
- Responsable del Consejo

**View :**

- Manager
- Owner
- Comisionado
- Consejero
- Investigador
- Técnico Académico
- Importador de Solicitudes
- Responsable de la Comision
- Responsable del Consejo

**Solicitud: Comision Revisa Solicitud :**

- -

**Solicitud: Consejo Revisa Solicitud :**

- -

**Solicitud: Consejo Cambia Solicitud :**

- Manager
- Consejero
- Responsable del Consejo

**Solicitud: Modificar Solicitud :**

- -

**+ Transición de Salida:**

- “Regresar solicitud al Consejo Interno” **id**=rechazararevisor, destino estado=revisionconsejo
- “Avisar al Investigador sobre modificaciones” **id**=avisarinvestigador, destino estado=aprobada

**+ Descripción:** La solicitud queda aprobada y cuenta dentro del presupuesto asignado. Éste estado permite a los usuarios con rol “Consejero” y “Responsable del Consejo” cambiar la cantidad ejercida en la solicitud (que debe ser menor o igual al monto aprobado). En este estado los usuarios con rol “Responsable del Consejo” pueden eliminar una solicitud.

6. *Solicitud Rechazada*: **id** = rechazada

**Change portal events :**

- Manager
- Responsable del Consejo

**Modify portal content :**

- Manager
- Consejero
- Responsable del Consejo

**Delete objects :**

- Manager
- Responsable del Consejo

**Manage properties :**

- Manager

**View :**

- Manager
- Owner
- Comisionado
- Consejero
- Investigador
- Técnico Académico
- Importador de Solicitudes
- Responsable de la Comision
- Responsable del Consejo

**Solicitud: Comision Revisa Solicitud :**

- Manager
- Comisionado
- Responsable de la Comision

**Solicitud: Consejo Revisa Solicitud :**

- Manager
- Consejero
- Responsable del Consejo

**Solicitud: Consejo Cambia Solicitud :**

- Manager
- Consejero
- Responsable del Consejo

**Solicitud: Modificar Solicitud :**

- -
- + Transición de Salida:
  - “Reconsiderar Solicitud” **id**=reconsiderar, destino estado=revisiónconsejo
- + Descripción: La solicitud queda rechazada y **no** cuenta dentro del presupuesto asignado. En este estado los usuarios con rol “Responsable del Consejo” pueden eliminar una solicitud.

### 3.5.3. Flujo de trabajo para Folder de solicitudes

- *Access contents information* (permiso de sistema): Permite acceder a los contenidos dentro de un tipo de dato con la propiedad de “contenedor”.
- *Change portal events* (permiso de sistema): Permite cambiar el estado de un contenido.
- *Modify portal content* (permiso de sistema): Permite modificar información de un contenido.
- *Manage properties* (permiso de sistema): Permite acceder información de las propiedades de usuario.
- *Delete objects* (permiso de sistema): Permite eliminar un contenido.
- *Add portal content* (permiso de sistema): Permite insertar contenido nuevo dentro de un tipo de dato con la propiedad de “contenedor”.
- *View* (permiso de sistema): Permite ver los datos de un contenido

Los estados del *workflow* están definidos como sigue:

1. *No se reciben solicitudes*: **id** = noaceptando

#### **Access contents information :**

- Manager
- Owner
- Investigador
- Técnico Académico
- Becario
- Comisionado
- Consejero
- Responsable de la Comisión
- Responsable del Consejo

#### **Change portal events :**

- Manager



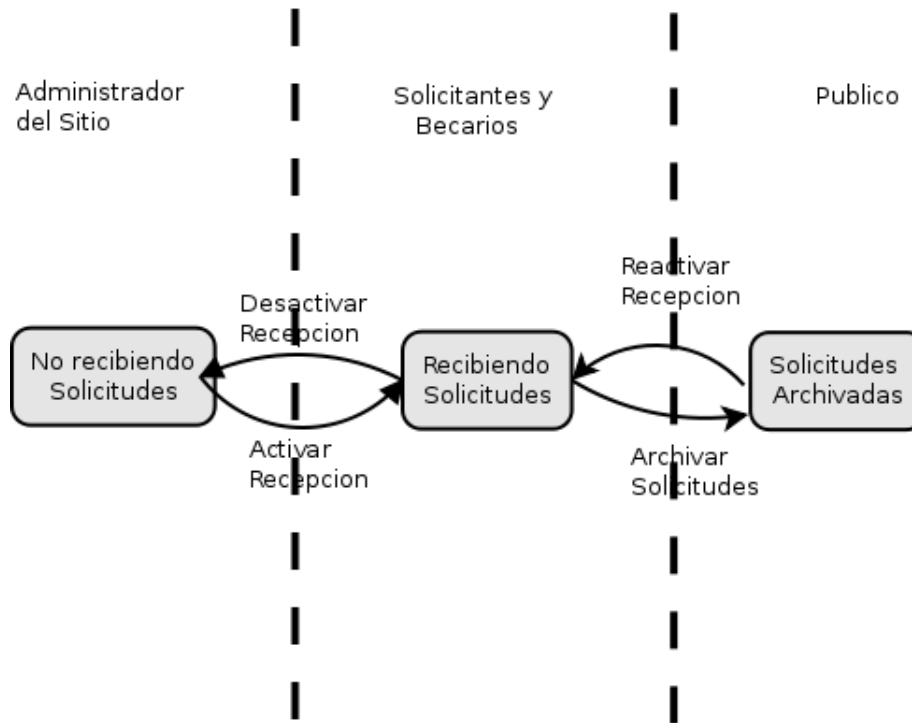


Figura 3.10: *Workflow* de Folder de Solicitudes

- Owner

**Modify portal content :**

- Manager
- Owner

**Delete objects :**

- Manager
- Owner

**Manage properties :**

- Manager
- Owner

**Add portal content :**

- Manager
- Owner

**View :**

- Manager
- Owner
- Investigador

- Técnico Académico
  - Becario
  - Comisionado
  - Consejero
  - Responsable de la Comision
  - Responsable del Consejo
- + Transición de Salida:
- “Comenzar a aceptar solicitudes” **id**=aceptar, destino estado=aceptando
  - “Mandar a Archivo” **id**=archivar, destino estado=historico
- + Descripción: El folder puede ser modificado pero no acepta solicitudes. Las solicitudes dentro del folder que ya han sido aprobadas cuentan como parte del presupuesto actual.

## 2. *Aceptando Solicitudes*: **id** = aceptando

### **Access contents information :**

- Manager
- Owner
- Investigador
- Técnico Académico
- Becario
- Comisionado
- Consejero
- Importador de Solicitudes
- Responsable de la Comision
- Responsable del Consejo

### **Change portal events :**

- Manager
- Owner

### **Modify portal content :**

- Manager
- Owner

### **Delete objects :**

- Manager
- Owner
- Investigador
- Técnico Académico
- Becario

### **Manage properties :**

- Manager
- Owner

**Add portal content :**

- Manager
- Owner
- Investigador
- Técnico Académico
- Becario

**View :**

- Manager
- Owner
- Investigador
- Técnico Académico
- Becario
- Comisionado
- Consejero
- Importador de Solicitudes
- Responsable de la Comision
- Responsable del Consejo

+ Transición de Salida:

- “Dejar de aceptar solicitudes” **id**=noaceptar, destino estado=noaceptando
- “Mandar a Archivo” **id**=archivar, destino estado=historico

+ Descripción: El folder **no** puede ser modificado y acepta solicitudes. Las solicitudes dentro del folder que ya han sido aprobadas cuentan como parte del presupuesto actual.

3. *Archivado (no se reciben solicitudes): id = historico*

**Access contents information :**

- Manager
- Owner
- Investigador
- Técnico Académico
- Becario
- Comisionado
- Consejero
- Responsable de la Comision
- Responsable del Consejo

**Change portal events :**

- Manager
- Owner

**Modify portal content :**

- Manager
- Owner

**Delete objects :**

- Manager
- Owner

**Manage properties :**

- Manager
- Owner

**Add portal content :**

- Manager
- Owner

**View :**

- Manager
- Owner
- Investigador
- Técnico Académico
- Becario
- Comisionado
- Consejero
- Responsable de la Comision
- Responsable del Consejo

**+ Transición de Salida:**

- “Aceptando Solicitudes” **id**=aceptar, destino estado=aceptando

**+ Descripción:** El folder **no** puede ser modificado y **no** acepta solicitudes. Las solicitudes dentro del folder que ya han sido aprobadas **no** cuentan como parte del presupuesto actual y se despliegan como archivo histórico.

### 3.6. Relación entre las actividades del sistema

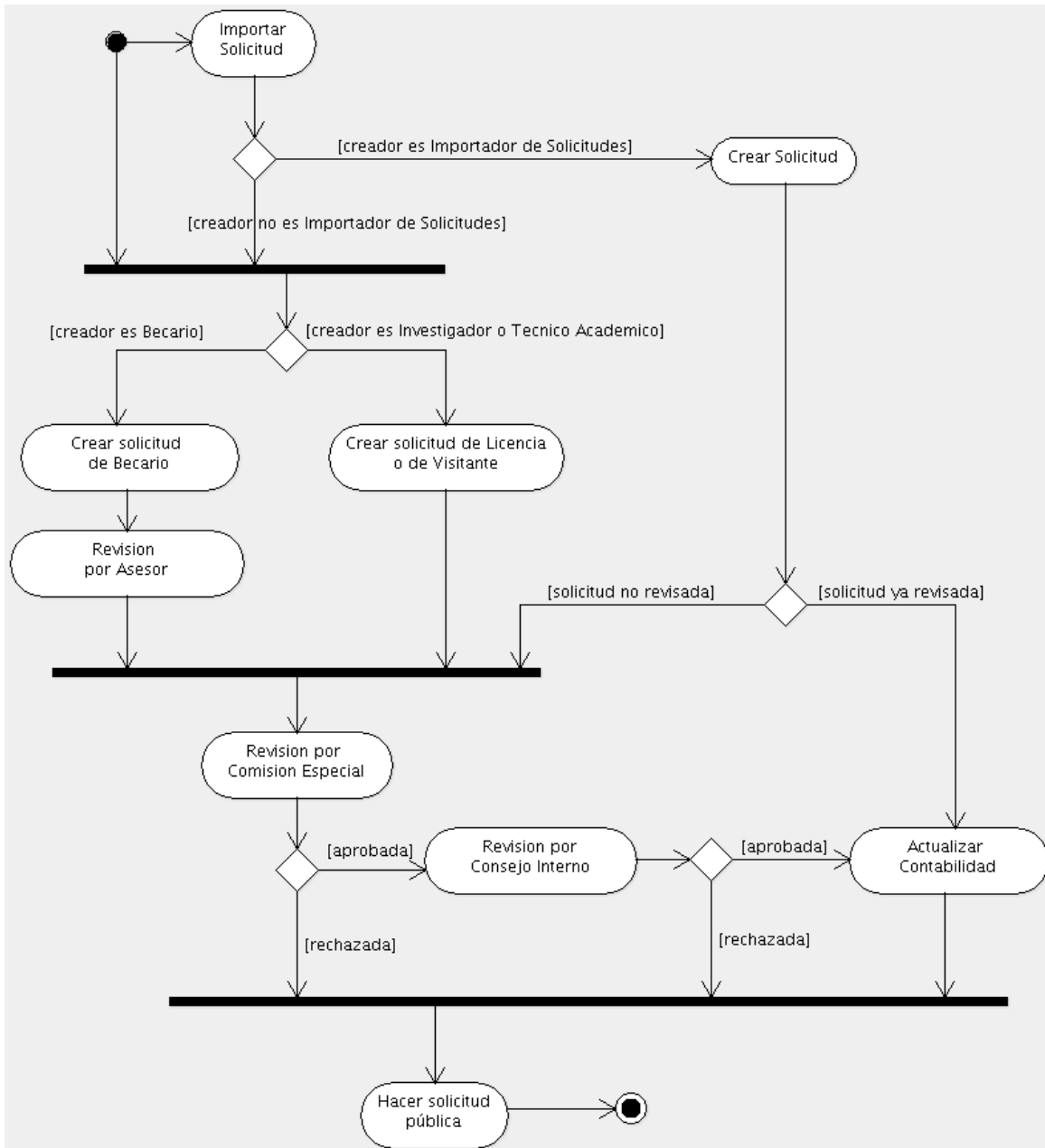


Figura 3.11: Actividades del Sistema

## 3.7. Requerimientos Mínimos

- MS Windows XP o superior.
- Apple OSX 10.4.x o superior.
- GNU/Linux 2.6.x o superior.
- Python 2.4
- Plone 3.x
- Mínimo 256 MB RAM.
- Mínimo 512 MB de espacio de disco duro.



# Capítulo 4

## Diseño Visual e Interfaz de Usuario

En este capítulo, se presenta el diseño visual del producto; se incluyen las tomas de pantalla, la explicación a nivel de usuario de las funciones básicas del sistema, el diseño de la herramienta de *portlet* y la forma en que la aplicación aparece desplegada en el sitio final de InfoMatem.

### 4.1. Crear nuevas solicitudes

Consiste en la instanciación de algún tipo de solicitud. Para los usuarios, el proceso para crear nuevas solicitudes es el siguiente:

1. Ir al contenedor de Solicitudes de elección, que es el contexto en el cual se creará la nueva solicitud. Se debe recordar que las solicitudes solo se pueden añadir en carpetas del tipo *Folder de Solicitudes*.

En la figura 4.1, está marcada con verde la zona de contenedores. En este caso, solo hay un contenedor llamado 2010 (que contiene las solicitudes para el año 2010).



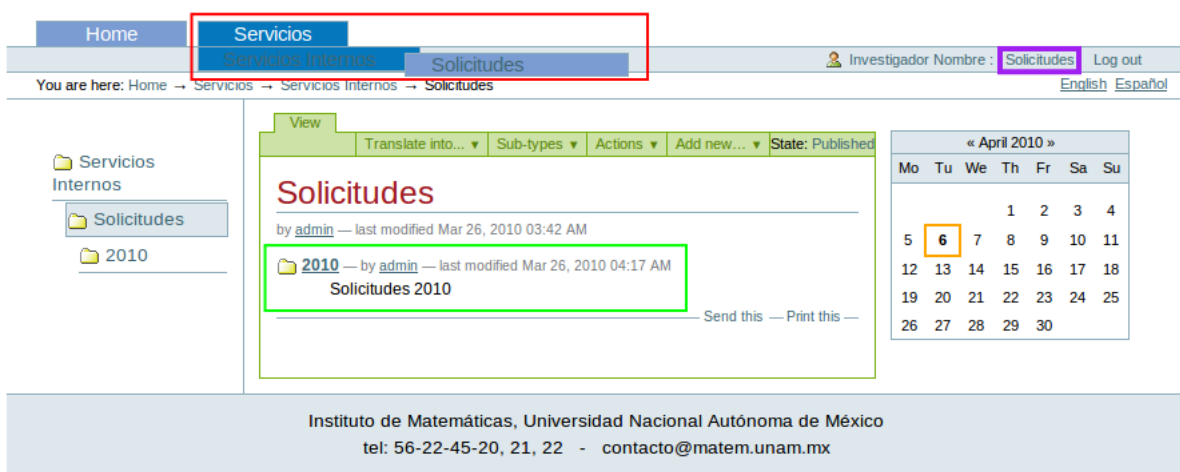


Figura 4.1: Ir a un contenedor

- Una vez en el contenedor, se despliega un menú con varias opciones como se muestra en la figura 4.2. En la parte superior derecha se encuentra un botón “Nueva Solicitud”, que manda a llamar una acción del portal que a su vez llama a la función que instancia tipos de contenido.



Figura 4.2: Menú principal del contenedor

Al hacer *click* en “Nueva Solicitud” se despliegan los tipos agregables tal y como se muestra en la figura 4.3. Las solicitudes solo se agregarán si en el menú principal tiene especificado “Estado del Periodo xxxx: Aceptando Solicitudes”, que constituye el estado del folder en el flujo de trabajo definido en la sección anterior para folders de solicitudes. En la figura 4.2, el texto está resaltado con rojo. Si en ese lugar aparece otra leyenda, significa que el periodo está cerrado y no se

pueden añadir solicitudes nuevas en ese contenedor.

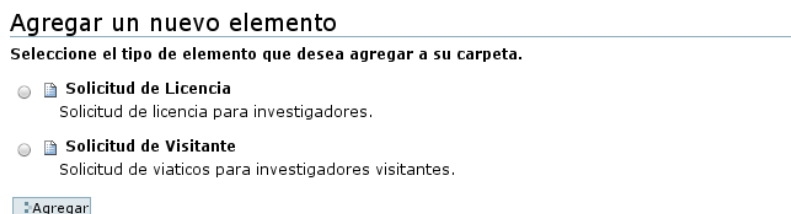


Figura 4.3: Agregar una solicitud al contenedor

3. Al añadir un nuevo elemento se crea una solicitud que se debe llenar y guardar en ese momento. Las formas de edición que se llenan al crear un nuevo tipo son las formas estándar que crea Archetypes, sin embargo en partes posteriores del proceso esta vista se sobrescribe y se utiliza una vista alterna definida por *Solicitudes*. Los campos marcados con un punto rojo son campos obligatorios. En la figura 4.4 se muestra una solicitud nueva en proceso de creación.

Una vez terminado el llenado de los campos, se procede dando en “Guardar” para terminar el proceso de creación. Toda solicitud nueva comienza en estado “Borrador”. El investigador o becario que la creó **debe enviar la solicitud para revisión** usando los botones que aparecen en la parte inferior de la misma como se muestra en la figura 4.5. El estado actual de la solicitud siempre aparece en la parte superior izquierda, resaltado con rojo.

### Agregar Solicitud de Licencia

Solicitud de licencia para investigadores.

Predeterminado Metadatos

**País** ■  
 País destino

**Ciudad** ■  
 Ciudad que visitará

**Institución** ■  
 Institucion que visitará

**Fecha de inicio** ■  
 Fecha en que iniciará la visita (puede ser aproximada)  
 /  /

**Fecha de término** ■  
 Fecha en que terminará la visita (puede ser aproximada)  
 /  /

**Objetivo de la visita** ■  
 Escriba el objetivo esperado de la visita

**Áreas de investigación** ■  
 Dudas acerca de la clasificación y como encontrar un área, acceda a la página oficial del [ams](#)

00-xx General	<input type="button" value="➤&gt;&gt;"/> <input type="button" value="⏪&lt;&lt;"/>
01-xx Historia y biografía	
03-xx Lógica matemática y fundamentos	
05-xx Combinatoria	
06-xx órdenes, látices, estructuras algebraicas ordenadas	
08-xx Sistemas algebraicos generales	

**Trabajo**  
 Especifique si presentará un trabajo

**Pasaje**  
 Especifique si solicita el pago de pasaje

**Viáticos**  
 Especifique si solicita viáticos. \$900 diarios (México) y \$1,200 diarios (Otros países), es necesario entregar comprobantes por el total

**Inscripción**  
 Especifique si solicita el pago de inscripción

**Archivos adjuntos**  
 Por favor adjunte la carta de invitación o cualquier otro documento relacionado  
 No se elig...n archivo

**Comentarios adicionales**  
 Agregue cualquier comentario que considere importante para su solicitud

Figura 4.4: Creando una solicitud

## 4.2. Revisar Solicitudes Pendientes

1. Ir al contenedor de Solicitudes de elección, que es el contexto en el cual se creará la nueva solicitud.

<b>Información</b>	Cambios guardados.		
<b>Información sobre el solicitante</b>	Arturo Curiel tiene un presupuesto total ejercido de 0.0 pesos, y puede tener máximo otros 3000.0 pesos.		
<b>¡Atención!</b>	Esta solicitud aún no ha sido enviada. Puede cambiar su información haciendo clic en la pestaña 'Hacer cambios' en la parte superior. Revise los datos y, si son correctos, envíela a revisión haciendo clic en el botón 'Enviar', ubicado en la parte inferior de la solicitud.		

**Solicitud (Licencia) de Arturo Curiel por 17800.0 (DF, UNAM, 2010/06/28)**

Estado: Solicitud en borrador

[Importar](#) | [Exportar](#)


Datos del Solicitante				
Nombre	Arturo Curiel			
Presupuesto:	Asignado	0.0		
	Disponible	3000.0		
Datos del Destino				
País	Mexico			
Ciudad	DF			
Institución	UNAM			
Fecha de inicio	2010/06/28	Fecha de término	2010/06/30	
Datos Adicionales				
Objetivo de la visita	Visita a la UNAM			
Áreas de investigación	03-xx Mathematical logic and foundations			
	06-xx Order, lattices, ordered algebraic structures			
Título del trabajo a presentar				
Archivos adjuntos	 tesisborrador.pdf — PDF document, 227Kb			
Comentarios adicionales	De paso me iré de vacaciones.			
Pasaje	si	Tipo de pasaje	Airplane	
Viáticos	si	Inscripción	no	
Cantidades				
Concepto	Cantidad Solicitada	Cantidad Recomendada por Comision Especial	Cantidad Autorizada por el Consejo Interno	Cantidad Utilizada
Cantidad para pasajes	7800.0	0.0	0.0	0.0
Cantidad para viáticos	10000.0	0.0	0.0	0.0
Cantidad para inscripción	0.0	0.0	0.0	0.0
<b>Total</b>	<b>17800.0</b>	<b>0.0</b>	<b>0.0</b>	<b>0.0</b>

Figura 4.5: Envío o eliminación con botones en la parte inferior de la imagen.

- En el menú principal del contenedor de la figura 4.6, al hacer *clic* en “Solicitudes pendientes de revisión” aparece un listado de las solicitudes que el usuario puede revisar, hacer alguna modificación o enviar a la Comisión o al Consejo. En la figura 4.7 se muestra la pantalla para un investigador que tiene varios roles en el sitio.

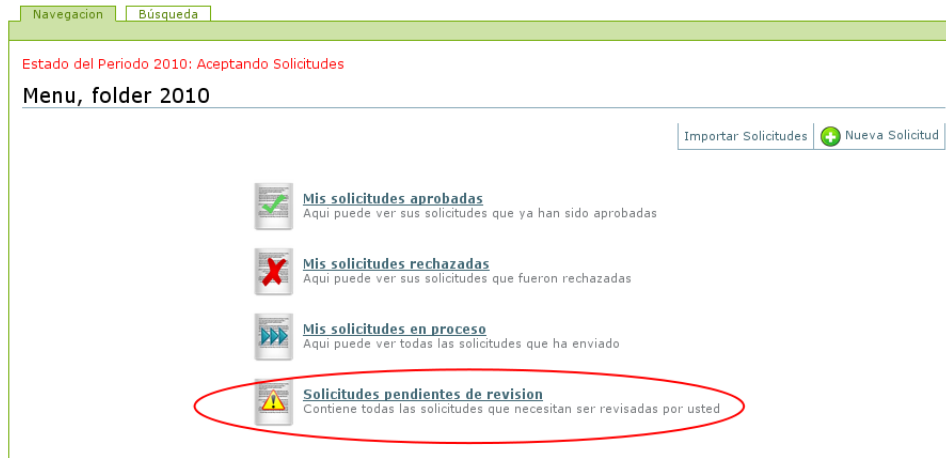


Figura 4.6: Menú Principal

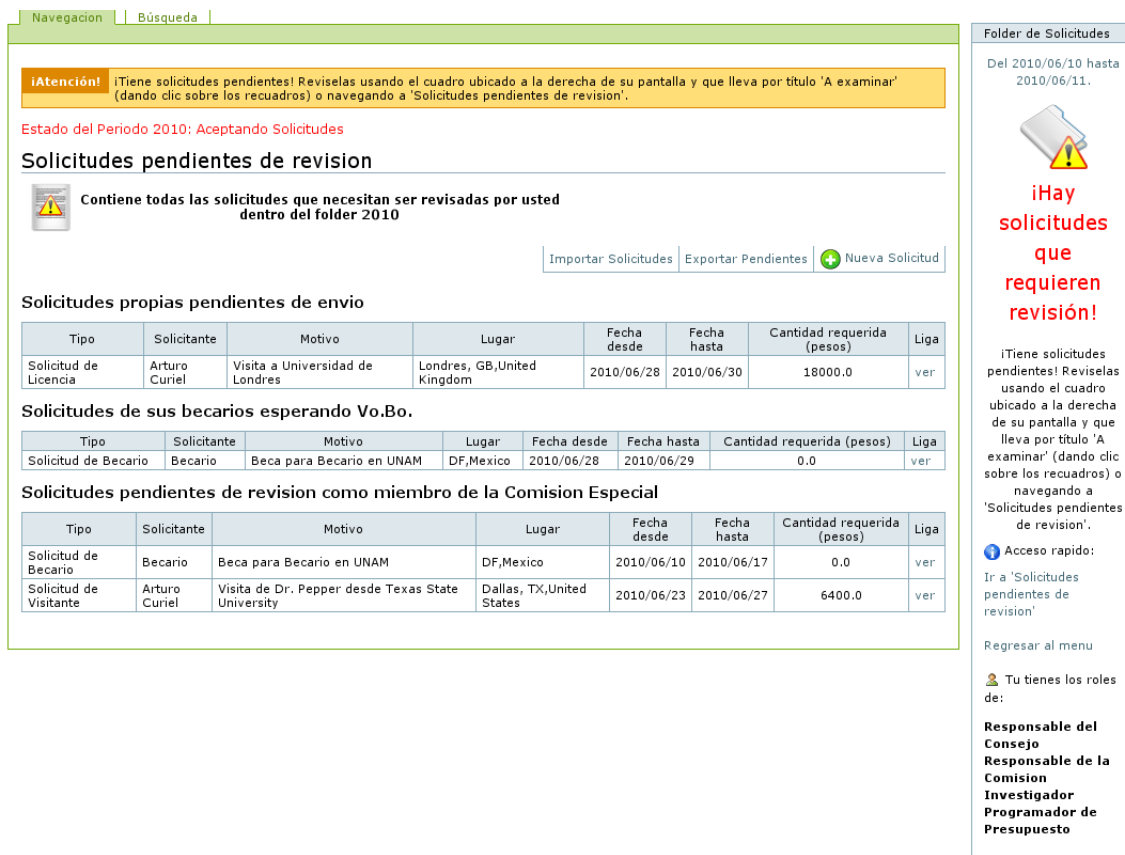


Figura 4.7: Vista de Solicitudes Pendientes

- Al abrir una solicitud pendiente, el encabezado se despliega como se muestra en la figura 4.8. Los Investigadores solo tienen pendientes sus solicitudes no enviadas, mientras que los Comisionados y Consejeros tienen como pendientes las solicitudes

no revisadas.

**Información** Cambios guardados.

**Información sobre el solicitante** Arturo Curiel tiene un presupuesto total ejercido de 0.0 pesos, y puede tener máximo otros 3000.0 pesos.

**¡Atención!** Esta solicitud aún no ha sido enviada. Puede cambiar su información haciendo clic en la pestaña 'Hacer cambios' en la parte superior. Revise los datos y, si son correctos, envíela a revisión haciendo clic en el botón 'Enviar', ubicado en la parte inferior de la solicitud.

**Solicitud (Licencia) de Arturo Curiel por 17800.0 (DF, UNAM, 2010/06/28)**

Estado: Solicitud en borrador

[Importar](#) | [Exportar](#)

Datos del Solicitante				
Nombre	Arturo Curiel			
Presupuesto:	Asignado	0.0		
	Disponible	3000.0		
Datos del Destino				
Pais	Mexico			
Ciudad	DF			
Institución	UNAM			
Fecha de inicio	2010/06/28	Fecha de término	2010/06/30	
Datos Adicionales				
Objetivo de la visita	Visita a la UNAM			
Áreas de investigación	03-xx Mathematical logic and foundations			
	06-xx Order, lattices, ordered algebraic structures			
Título del trabajo a presentar				
Archivos adjuntos	tesisborrador.pdf – PDF document, 227Kb			
Comentarios adicionales	De paso me iré de vacaciones.			
Pasaje	si	Tipo de pasaje	Airplane	
Viáticos	si	Inscripción	no	
Cantidades				
Concepto	Cantidad Solicitada	Cantidad Recomendada por Comision Especial	Cantidad Autorizada por el Consejo Interno	Cantidad Utilizada
Cantidad para pasajes	7800.0	0.0	0.0	0.0
Cantidad para viáticos	10000.0	0.0	0.0	0.0
Cantidad para inscripción	0.0	0.0	0.0	0.0
<b>Total</b>	<b>17800.0</b>	<b>0.0</b>	<b>0.0</b>	<b>0.0</b>

[Enviar](#) | [Eliminar solicitud.](#)

Tu Solicitud de Licencia

iDebe enviar la solicitud para su revisión!

Esta solicitud aún no ha sido enviada. Puede cambiar su información haciendo clic en la pestaña 'Hacer cambios' en la parte superior. Revise los datos y, si son correctos, envíela a revisión haciendo clic en el botón 'Enviar', ubicado en la parte inferior de la solicitud.

**Acceso rapido:**  
Ir al menu

**Tu tienes los roles de:**  
**Investigador**

---

**A examinar**

Solicitud (Licencia) de Arturo Curiel por 0.0 (asdasd, asdasd, 2010/06/10) id1 – 10/06/2010

Solicitud (Licencia) de Arturo Curiel por

Figura 4.8: Solicitud Pendiente

4. Para llevar a cabo el envío, aprobación o rechazo (dependiendo si el usuario es Investigador, Responsable del Consejo o Responsable de la Comisión) se utilizan los botones de cambio de estado en la parte inferior de la solicitud, mostrado en las figuras 4.9, 4.10 y 4.11. La información del estado siempre se desplegará en la parte superior izquierda. Los botones de envío o rechazo de la solicitud siempre aparecerán en la parte inferior de la misma:

Comentarios adicionales		De paso me ire de vacaciones.		
Pasaje		si	Tipo de pasaje	Airplane
Viáticos		si	Inscripción	no
Cantidades				
Concepto	Cantidad Solicitada	Cantidad Recomendada por Comision Especial	Cantidad Autorizada por el Consejo Interno	Cantidad Utilizada
Cantidad para pasajes	7800.0	0.0	0.0	0.0
Cantidad para viáticos	10000.0	0.0	0.0	0.0
Cantidad para inscripción	0.0	0.0	0.0	0.0
<b>Total</b>	<b>17800.0</b>	<b>0.0</b>	<b>0.0</b>	<b>0.0</b>

Figura 4.9: Cambio de estado para Investigador

**Información** Cambios guardados.

**Información sobre el solicitante** Arturo Curiel tiene un presupuesto total ejercido de 0.0 pesos, y puede tener máximo otros 3000.0 pesos.

**Información** Esta solicitud está siendo revisada por la comisión especial. Espere noticias pronto.

**Solicitud (Licencia) de Arturo Curiel por 17800.0 (DF, UNAM, 2010/06/28)**  
Estado: Solicitud en revision por Comision Especial

Datos del Solicitante				
Nombre: Arturo Curiel				
Presupuesto:	Asignado	0.0		
	Disponible	3000.0		
Datos del Destino				
País		Mexico		
Ciudad		DF		
Institución		UNAM		
Fecha de inicio		2010/06/28	Fecha de término	2010/06/30
Datos Adicionales				
Objetivo de la visita: Visita a la UNAM				
Áreas de investigación: 15-xx Linear and multilinear algebra; matrix theory 35-xx Partial differential equations				
Título del trabajo a presentar:				
Archivos adjuntos:				
Comentarios adicionales: De paso me iré de vacaciones.				
Pasaje		si	Tipo de pasaje	Airplane
Viáticos		si	Inscripción	no
Cantidades				
Concepto	Cantidad Solicitada	Cantidad Recomendada por Comision Especial	Cantidad Autorizada por el Consejo Interno	Cantidad Utilizada
Cantidad para pasajes	7800.0	7800.0	0.0	0.0
Cantidad para viáticos	10000.0	10000.0	0.0	0.0
Cantidad para inscripción	0.0	0.0	0.0	0.0
<b>Total</b>	<b>17800.0</b>	<b>17800.0</b>	<b>0.0</b>	<b>0.0</b>
Información Interna				
Recomendación de la comisión especial			Que se apruebe.	

Rechazar
Aprobar y enviar a Consejo Interno

**Licencia**

Esta solicitud está siendo revisada por la comisión especial. Espere noticias pronto.

**Acceso rapido:**

Ir al menu

Tu tienes los roles de:

**Responsable del Consejo**

**Responsable de la Comision Investigador**

---

**A examinar**

- Solicitud (Licencia) de Arturo Curiel por 0.0 (DF, Institucion, 2010/06/10) id1 - 10/06/2010
- Solicitud (Licencia) de Arturo Curiel por 0.0 (asdasd, asdasd, 2010/06/10) id1 - 10/06/2010
- Solicitud (Becario) de Becario por 0.0 (DF, UNAM, 2010/06/10) id2 - 10/06/2010
- Solicitud (Visitante) de Arturo

Figura 4.10: Cambio de estado para Responsable de la Comision

**Información** Esta solicitud está siendo revisada por el Consejo Interno. Espere noticias pronto.

**Solicitud (Licencia) de Arturo Curiel por 17800.0 (DF, UNAM, 2010/06/28)**  
 Estado: Solicitud en revision por Consejo Interno

Importar | Exportar

Datos del Solicitante				
Nombre		Arturo Curiel		
Presupuesto:	Asignado	0.0		
	Disponibile	3000.0		
Datos del Destino				
País		Mexico		
Ciudad		DF		
Institución		UNAM		
Fecha de inicio		2010/06/28	Fecha de término	2010/06/30
Datos Adicionales				
Objetivo de la visita		Visita a la UNAM		
Áreas de investigación		15-xx Linear and multilinear algebra; matrix theory 35-xx Partial differential equations		
Título del trabajo a presentar				
Archivos adjuntos				
Comentarios adicionales		De paso me iré de vacaciones.		
Pasaje	si	Tipo de pasaje	Airplane	
Viáticos	si	Inscripción	no	
Cantidades				
Concepto	Cantidad Solicitada	Cantidad Recomendada por Comision Especial	Cantidad Autorizada por el Consejo Interno	Cantidad Utilizada
Cantidad para pasajes	7800.0	7800.0	7800.0	0.0
Cantidad para viáticos	10000.0	10000.0	10000.0	0.0
Cantidad para inscripción	0.0	0.0	0.0	0.0
<b>Total</b>	<b>17800.0</b>	<b>17800.0</b>	<b>17800.0</b>	<b>0.0</b>
Información Interna				
Recomendación de la comisión especial		Que se apruebe.		
Consejo Interno				
Fecha de revisión por el CI		2010/06/28		
Número de acta de Consejo Interno		234-2010		
Comentarios del Consejo Interno		Revisada y Aprobada.		

Rechazar | Aprobar

Enviar — Imprimir —

Esta solicitud está siendo revisada por el Consejo Interno. Espere noticias pronto.

**Acceso rapido:**  
Ir al menu  
Tu tienes los roles de:  
**Responsable del Consejo**  
**Responsable de la Comision Investigador**

**A examinar**

- Solicitud (Licencia) de Arturo Curiel por 0.0 (DF, Institucion, 2010/06/10) id1 — 10/06/2010
- Solicitud (Licencia) de Arturo Curiel por 0.0 (asdasd, asdasd, 2010/06/10) id1 — 10/06/2010
- Solicitud (Becario) de Becario por 0.0 (DF, UNAM, 2010/06/10) id2 — 10/06/2010
- Solicitud (Visitante) de Arturo Curiel por 0.0 (DF, Instituto, 2010/06/11) id1 — 11/06/2010
- Solicitud (Licencia)

Figura 4.11: Cambio de estado para Responsable del Consejo



5. Una vez enviada, aprobada o rechazada una solicitud, ésta dejará de mostrarse pendiente para el usuario. Ejemplo de ello se ve en la figura 4.12, donde un investigador observa que una solicitud suya está siendo revisada por la Comisión Especial.

Visualizar

**Información** Ha cambiado el estado del elemento

**Información sobre el solicitante** Arturo Curiel tiene un presupuesto total ejercido de 0.0 pesos, y puede tener máximo otros 3000.0 pesos.

**Información** Esta solicitud está siendo revisada por la comisión especial. Espere noticias pronto.


**Solicitud (Licencia) de Arturo Curiel por 17800.0 (DF, UNAM, 2010/06/28)**  
Estado: Solicitud en revision por Comision Especial

[Importar](#) | [Exportar](#)

Datos del Solicitante				
Nombre		Arturo Curiel		
Presupuesto:	Asignado	0.0		
	Disponibile	3000.0		
Datos del Destino				
Pais		Mexico		
Ciudad		DF		
Institución		UNAM		
Fecha de inicio		2010/06/28	Fecha de término	2010/06/30
Datos Adicionales				
Objetivo de la visita		Visita a la UNAM		
Áreas de investigación		15-xx Linear and multilinear algebra; matrix theory 35-xx Partial differential equations		
Título del trabajo a presentar				
Archivos adjuntos				
Comentarios adicionales		De paso me iré de vacaciones.		
Pasaje		si	Tipo de pasaje	Airplane
Viáticos		si	Inscripción	no
Cantidades				
Concepto	Cantidad Solicitada	Cantidad Recomendada por Comision Especial	Cantidad Autorizada por el Consejo Interno	Cantidad Utilizada
Cantidad para pasajes	7800.0	7800.0	0.0	0.0
Cantidad para viáticos	10000.0	10000.0	0.0	0.0
Cantidad para inscripción	0.0	0.0	0.0	0.0
<b>Total</b>	<b>17800.0</b>	<b>17800.0</b>	<b>0.0</b>	<b>0.0</b>

[Enviar](#) — [Imprimir](#) —

**Tu Solicitud de Licencia**



Esta solicitud está siendo revisada por la comisión especial. Espere noticias pronto.

**Acceso rapido:**  
[Ir al menu](#)  
 Tu tienes los roles de:  
**Investigador**

---

**A examinar**

Solicitud (Licencia) de Arturo Curiel por 0.0 (asdasd, asdasd, 2010/06/10)  
id1 — 10/06/2010

[Lista de revisión completa...](#)

---

« Junio 2010 »

Lu	Ma	Mi	Ju	Vi	Sá	Do
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Figura 4.12: Solicitud Enviada

6. Cuando el usuario ya no tiene solicitudes pendientes, la vista de “Solicitudes pendientes de revisión” muestra un mensaje que lo indica (figura 4.13)

The screenshot displays a web interface for reviewing pending requests. At the top, there are tabs for 'Navegacion' and 'Búsqueda'. Below this, a red header indicates the current period: 'Estado del Periodo 2010: Aceptando Solicitudes'. The main heading is 'Solicitudes pendientes de revision'. A warning icon is followed by the text: 'Contiene todas las solicitudes que necesitan ser revisadas por usted dentro del folder 2010'. Below this, there are three buttons: 'Importar Solicitudes', 'Exportar Pendientes', and 'Nueva Solicitud'. A message states: 'No hay solicitudes pendientes. Puede revisar sus solicitudes aprobadas y rechazadas navegando a 'Mis solicitudes aprobadas' y 'Mis solicitudes rechazadas', respectivamente.' Below this, it says: 'No hay ninguna solicitud para mostrar.' On the right side, there is a sidebar titled 'Folder de Solicitudes' showing a date range 'Del 2010/06/10 hasta 2010/06/11.' and a folder icon. Below the icon, it says: 'No hay solicitudes pendientes. Puede revisar sus solicitudes aprobadas y rechazadas navegando a 'Mis solicitudes aprobadas' y 'Mis solicitudes rechazadas', respectivamente.' There is also a section for 'Acceso rapido:' with links to 'Ir a 'Mis solicitudes aprobadas'' and 'Ir a 'Mis solicitudes rechazadas''. At the bottom of the sidebar, it says 'Regresar al menu' and 'Tu tienes los roles de:' followed by a list of roles: 'Responsable del Consejo', 'Responsable de la Comision', 'Investigador', 'Programador de Presupuesto'.

Figura 4.13: Vista de Solicitudes Pendientes (sin pendientes)

### 4.3. Vista general y exportación de solicitudes

En la figura 4.14 se muestra una vista de una solicitud. En la parte derecha, debajo del título, se puede observar el botón “**Exportar**” que permite sacar la información de la solicitud en forma de texto simple. El formato generado por el *script* de exportación es compatible con la importación, por lo cual una solicitud se puede copiar mediante exportaciones e importaciones en diferentes instancias de Plone.

**Visualizar**

**Información** Ha cambiado el estado del elemento

**Información sobre el solicitante** Arturo Curiel tiene un presupuesto total ejercido de 0.0 pesos, y puede tener máximo otros 3000.0 pesos.

**Información** Esta solicitud está siendo revisada por la comisión especial. Espere noticias pronto.

**Solicitud (Licencia) de Arturo Curiel por 17800.0 (DF, UNAM, 2010/06/28)**  
 Estado: Solicitud en revision por Comision Especial

Importar **Exportar**

Datos del Solicitante				
Nombre	Arturo Curiel			
Presupuesto:	Asignado	0.0		
	Disponible	3000.0		
Datos del Destino				
Pais	Mexico			
Ciudad	DF			
Institución	UNAM			
Fecha de inicio	2010/06/28	Fecha de término	2010/06/30	
Datos Adicionales				
Objetivo de la visita	Visita a la UNAM			
Áreas de investigación	15-xx Linear and multilinear algebra; matrix theory 35-xx Partial differential equations			
Título del trabajo a presentar				
Archivos adjuntos				
Comentarios adicionales	De paso me iré de vacaciones.			
Pasaje	si	Tipo de pasaje	Airplane	
Viáticos	si	Inscripción	no	
Cantidades				
Concepto	Cantidad Solicitada	Cantidad Recomendada por Comision Especial	Cantidad Autorizada por el Consejo Interno	Cantidad Utilizada
Cantidad para pasajes	7800.0	7800.0	0.0	0.0
Cantidad para viáticos	10000.0	10000.0	0.0	0.0
Cantidad para inscripción	0.0	0.0	0.0	0.0
<b>Total</b>	<b>17800.0</b>	<b>17800.0</b>	<b>0.0</b>	<b>0.0</b>

Enviar — Imprimir —

Tu Solicitud de Licencia

Esta solicitud está siendo revisada por la comisión especial. Espere noticias pronto.

Acceso rapido:  
Ir al menu

Tu tienes los roles de:  
**Investigador**

A examinar

Solicitud (Licencia) de Arturo Curiel por 0.0 (asdasd, asdasd, 2010/06/10) id1 — 10/06/2010

Lista de revisión completa...

« Junio 2010 »

Lu	Ma	Mi	Ju	Vi	Sá	Do
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
<b>28</b>	29	30				

Figura 4.14: Solicitud en proceso de revisión. El botón “Exportar” genera un archivo de texto simple.

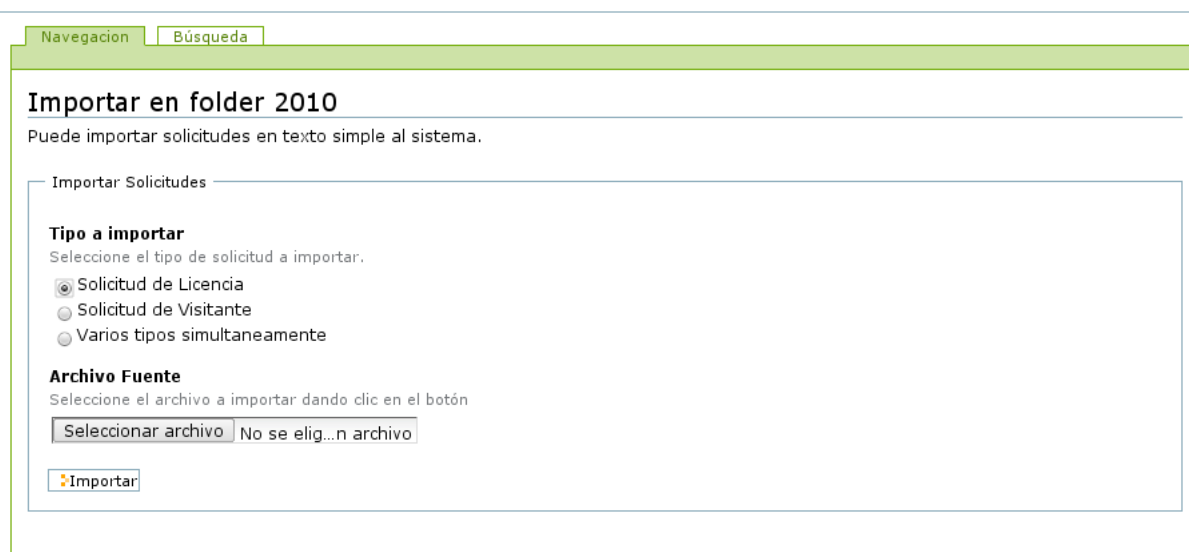
La solicitud despliega los datos del solicitante, entre los cuales destaca el **presupuesto ya asignado** (marcado con rojo en la figura 4.14). Es de notar que el cálculo de este valor no contendrá a la solicitud actual si ésta no está aprobada. La función de estos datos es la de auxiliar a los revisores en su decisión de la cantidad a aprobar o rechazar, conociendo de antemano cuanto dinero ha usado el investigador que solicita presupuesto.

## 4.4. Importar Solicitudes

Aquí se describen los dos tipos de importación que tiene el sitio, la importación básica y la importación avanzada. Como ya se vio en la sección anterior, la activación de una y otra depende de si entre los roles del usuario se encuentra “Importador de Solicitudes”.

### 4.4.1. Importación Básica

En esta vista se puede importar hacia el sistema un formato en modo texto (necesariamente con extensión .txt), siempre y cuando se tengan permisos para ello. Para ir a la vista de importación, solo se tiene que oprimir el botón “Importar” en cualquier solicitud. Un investigador solo puede importar solicitudes de Licencia y de Visitante. Un becario puede importar solo solicitudes de Becario. Se pueden importar solicitudes en lote o individualmente. En la figura 4.15 se muestra la vista de importación. En las figuras 4.16 y 4.17 se muestran los formatos para importación individual.



The screenshot shows a web interface for importing data. At the top, there are two tabs: 'Navegacion' and 'Búsqueda'. Below the tabs, the main heading is 'Importar en folder 2010'. Underneath the heading, there is a sub-heading 'Importar Solicitudes' and a brief instruction: 'Puede importar solicitudes en texto simple al sistema.' The form contains two main sections: 'Tipo a importar' and 'Archivo Fuente'. The 'Tipo a importar' section has three radio button options: 'Solicitud de Licencia' (which is selected), 'Solicitud de Visitante', and 'Varios tipos simultaneamente'. The 'Archivo Fuente' section has a text input field with the placeholder 'Seleccionar archivo' and a button labeled 'No se elig...n archivo'. At the bottom of the form, there is a prominent 'Importar' button.

Figura 4.15: Vista “Importar”

```

:: Datos del Destino ::

    Pais: CODIGO_DE_PAIS
    Ciudad: CIUDAD
    Institucion: INSTITUCION
    Desde: AAAA/MM/DD
    Hasta: AAAA/MM/DD

:: Datos Adicionales ::

    Objetivo de la visita: OBJETIVO
    Grado: LICENCIATURA o MAESTRIA o DOCTORADO (solo uno, en blanco si no es becario)
    Area de Investigacion: CODIGO_DE_AREA_1,CODIGO_DE_AREA_2,... (varias, separado por comas)
    Titulo del trabajo a presentar: TITULO
    Comentarios Adicionales: COMENTARIOS
    Medio de Transporte: AVION,AUTOBUS,CARRO (separado por comas)

:: Cantidades ::

Transporte:                0.0

Viaticos:                  0.0

Inscripcion:              0.0

```

Figura 4.16: Formato importación individual (Licencia y Becario)

```

:: Datos del Visitante ::

    Nombre: NOMBRE DEL VISITANTE
    Pais: CODIGO_DE_PAIS
    Ciudad De Origen: CIUDAD
    Institucion De Origen: INSTITUCION
    Desde: AAAA/MM/DD
    Hasta: AAAA/MM/DD

:: Datos Adicionales ::

    Objetivo de la visita: OBJETIVO
    Area de Investigacion: CODIGO_DE_AREA_AMS_1,CODIGO_DE_AREA_AMS_2,... (uno o varios separados por comas)
    Titulo del trabajo a presentar: TITULO
    Comentarios Adicionales: COMENTARIOS
    Medio de Transporte: AVION,AUTOBUS,CARRO (uno o varios separados por comas)

:: Cantidades ::

Transporte:                0.0

Viaticos:                  0.0

```

Figura 4.17: Formato importación individual (Visitante)

En los apéndices se encuentra la información correspondiente a los códigos de Area de la Sociedad Matemática Americana y los códigos de país que se pueden introducir en los formatos de importación.

En la figura 4.18 está el formato de importación en lote. En la importación en lote, cada línea del texto corresponde a una solicitud cuyos campos están separados por el símbolo “;” (punto y coma). Es primordial que se respete el orden de los campos para el correcto funcionamiento de la importación en lote.

La importación en lote permite importar varios tipos de solicitud en un mismo archi-

vo de texto, sin embargo el sistema solo permitirá importará los tipos permitidos para el usuario; *i.e.*, para Investigadores y Técnicos Académicos solo importará solicitudes de Licencia y de Visitante, mientras que para Becarios solo podrá importar solicitudes de Becario.

Importación en lote

Solicitud de Licencia;NOMBRE\_DE\_INVESTIGADOR o TECNICO; INSTITUCION; CODIGO\_DE\_PAIS; CIUDAD; OBJETIVO; AAAA/MM/DD; aaaa/mm/dd; CODIGOS\_DE\_AREA\_AMS; TITULO DE TRABAJO A PRESENTAR; TIPO DE TRANSPORTE; CANTIDAD DE TRANSPORTE; CANTIDAD DE VIATICOS; CANTIDAD DE INSCRIPCION;COMENTARIOS;

Solicitud de Visitante;NOMBRE\_DE\_INVESTIGADOR o TECNICO; NOMBRE DE VISITANTE; INSTITUCION;CODIGO\_DE\_PAIS; CIUDAD; OBJETIVO; AAAA/MM/DD; aaaa/mm/dd; CODIGOS\_DE\_AREA\_AMS; TITULO DE TRABAJO A PRESENTAR; TIPO DE TRANSPORTE; CANTIDAD DE TRANSPORTE; CANTIDAD DE VIATICOS; COMENTARIOS;

Solicitud de Becario;NOMBRE\_DE\_BECARIO; GRADO; INSTITUCION; CODIGO\_DE\_PAIS; CIUDAD; OBJETIVO; AAAA/MM/DD; aaaa/mm/dd; CODIGOS\_DE\_AREA\_AMS; TITULO DE TRABAJO A PRESENTAR; TIPO DE TRANSPORTE; CANTIDAD DE TRANSPORTE; CANTIDAD DE VIATICOS; CANTIDAD DE INSCRIPCION; COMENTARIOS;

Figura 4.18: Formato importación en lote

En la importación por lote, AAAA/MM/DD es la fecha de inicio de la visita y aaaa/mm/dd es la fecha de término de la visita. En CODIGOS\_DE\_AREA\_AMS se pueden poner varios códigos de área separados por comas (CODIGO\_DE\_AREA\_AMS\_1, CODIGO\_DE\_AREA\_AMS\_2, ...). Lo mismo ocurre con TIPO DE TRANSPORTE. El GRADO de la solicitud de Becario tiene que ser solo uno entre Licenciatura, Maestría y Doctorado. Cada línea debe terminar, forzosamente, con punto y coma (“;”). Se debe respetar la cantidad de puntos y coma de cada línea, es decir, no se pueden poner ni más ni menos caracteres “;” de los que tiene el formato en la figura 4.18, ya que son primordiales para el procesamiento. Si, por ejemplo donde dice “;TITULO DE TRABAJO A PRESENTAR;” no se quiere especificar nada, el usuario deberá entonces sustituir TITULO DE TRABAJO A PRESENTAR por un espacio en blanco, de forma que quede “. ;”.

Tanto en la importación en lote como en la individual, se deben conservar los saltos de línea y los símbolos “:”, “::”, “;”. Todas las fechas deben tener formato “aaaa/mm/dd”. En cada caso, el orden especificado de los elementos de la solicitud debe ser respetado.

#### 4.4.2. El rol de *Importador de Solicitudes*

El rol de Importador de Solicitudes otorga un usuario la posibilidad de importar solicitudes para otros usuarios.

Un usuario que es Importador de Solicitudes no tendrá acceso a la importación básica, sino que su formulario de importación le permitirá importar cualquier tipo de solicitud ya sea individualmente o en bloque. En la figura 4.19 se muestra el formulario de importación para un Importador de Solicitudes:

The screenshot shows a web interface for importing requests. The main content area is titled '2010 Importar' and contains the following elements:

- Navigation:** 'Navegacion' and 'Búsqueda' tabs.
- Header:** '2010' and 'Importar'.
- Description:** 'Puede importar solicitudes en texto simple al sistema.'
- Form Section:**
  - Importar Solicitudes:** A large text area for input.
  - Tipo a importar:** 'Seleccione el tipo de solicitud a importar.' with radio buttons for:
    - Solicitud de Licencia
    - Solicitud de Visitante
    - Solicitud de Becario
    - Varios tipos simultaneamente (selected)
  - Archivo Fuente:** 'Seleccione el archivo a importar dando clic en el botón' with a 'Seleccionar archivo' button and a 'No se elig...n archivo' message.
  - Importar:** A button to submit the form.

The right sidebar, titled 'Folder de Solicitudes', shows a folder icon and the text: 'Del 2010/06/10 hasta 2010/06/11.' Below this, it states: 'No hay solicitudes pendientes. Puede revisar sus solicitudes aprobadas y rechazadas navegando a 'Mis solicitudes aprobadas' y 'Mis solicitudes rechazadas', respectivamente.' It also includes 'Acceso rapido:' with links to 'Mis solicitudes aprobadas' and 'Mis solicitudes rechazadas', and a 'Regresar al menu' link. At the bottom, it lists roles: 'Tu tienes los roles de: Responsable del Consejo, Responsable de la Comision, Investigador, **Importador de Solicitudes** (circled in red), Programador de Presupuesto'.

Figura 4.19: Formulario de importación para Importadores de Solicitudes

La diferencia primordial con la importación básica es que los formatos de importación necesarios requieren tener, forzosamente, el nombre de usuario en el sitio de Infomatem del investigador para el cual se quiere importar una solicitud (aún si el usuario desea importar una solicitud propia). Los formatos de importación de solicitud de Licencia, solicitud de Becario y solicitud de Visitante para Importadores de Solicitudes se muestran en las figuras 4.20 y 4.21.

```
::Datos del Solicitante::

  Id de Solicitante: NOMBRE_DE_USUARIO_EN_INFOMATEM,NOMBRE_DE_ASESOR_EN_INFOMATEM (solo uno si no es becario)
  Estado de Solicitud: ESTADO_DE_LA_SOLICITUD (en blanco si no es Importador de Solicitudes)

:: Datos del Destino ::

  Pais: CODIGO_DE_PAIS
  Ciudad: CIUDAD
  Institucion: INSTITUCION
  Desde: AAAA/MM/DD
  Hasta: AAAA/MM/DD

:: Datos Adicionales ::

  Objetivo de la visita: OBJETIVO
  Grado: LICENCIATURA o MAESTRIA o DOCTORADO (solo uno, en blanco si no es becario)
  Area de Investigacion: CODIGO_DE_AREA_1,CODIGO_DE_AREA_2,... (varias, separado por comas)
  Titulo del trabajo a presentar: TITULO
  Comentarios Adicionales: COMENTARIOS
  Medio de Transporte: AVION,AUTOBUS,CARRO (separado por comas)

:: Cantidades ::

  Transporte:          0.0
  Viaticos:           0.0
  Inscripcion:        0.0
```

Figura 4.20: Formato importación individual (Licencia y Becario)



```

::Datos del Solicitante::

  Id de Solicitante: NOMBRE_DE_USUARIO_DE_INFOMATEM
  Estado de Solicitud: ESTADO_DE_LA_SOLICITUD (en blanco si no es Importador de Solicitudes)

:: Datos del Visitante ::

  Nombre: NOMBRE DEL VISITANTE
  Pais: CODIGO_DE_PAIS
  Ciudad De Origen: CIUDAD
  Institucion De Origen: INSTITUCION
  Desde: AAAA/MM/DD
  Hasta: AAAA/MM/DD

:: Datos Adicionales ::

  Objetivo de la visita: OBJETIVO
  Area de Investigacion: CODIGO_DE_AREA_AMS_1,CODIGO_DE_AREA_AMS_2,... (uno o varios separados por comas)
  Titulo del trabajo a presentar: TITULO
  Comentarios Adicionales: COMENTARIOS
  Medio de Transporte: AVION,AUTOBUS,CARRO (uno o varios separados por comas)

:: Cantidades ::

  Transporte:          0.0
  Viaticos:           0.0

```

Figura 4.21: Formato importación individual (Visitante)

En estos casos, los formatos son iguales a los de importación básica excepto por dos campos nuevos. En donde dice NOMBRE\_DE\_USUARIO\_DE\_INFOMATEM, el usuario debe introducir el nombre de usuario del Investigador para el cual se está subiendo la solicitud. Por ejemplo, si el usuario quiere importar una solicitud para “Jorge Perez” y el nombre de usuario de Jorge Perez es jperez86, entonces la línea sería:

```
Id de Solicitante: jperez86
```

Si “Jorge Perez” es Becario y tiene como asesor al investigador “Manuel Rodríguez”, cuyo nombre de usuario es mrodriguez56, entonces la línea diría:

```
Id de Solicitante: jperez86,mrodriguez56
```

En el campo ESTADO\_DE\_LA\_SOLICITUD, el usuario puede poner en que estado de los siguientes quiere que la solicitud se introduzca:

- Borrador: Es el estado inicial de cada Solicitud.
- En revisión por Investigador A Cargo: **Solo para solicitudes de Becario.** En este estado, la solicitud puede ser revisada por el asesor del becario.
- En revisión por Comision Especial: Es este estado, algún miembro de la Comisión Especial puede revisar y aprobar o rechazar una solicitud.
- En revisión por Consejo Interno: Es este estado, algún miembro del Consejo Interno del Instituto puede revisar y aprobar o rechazar una solicitud.

- Aprobada: En este estado, la solicitud es aprobada automáticamente durante el proceso de importación.

En los apéndices se encuentra la información correspondiente a los códigos de Área de la Sociedad Matemática Americana y los códigos de país que se pueden introducir en los formatos de importación.

En la figura 4.22 está el formato de importación en lote para Importadores de Solicitudes. En la importación en lote, cada línea del texto corresponde a una solicitud cuyos campos están separados por el símbolo “;” (punto y coma). Es primordial que se respete el orden de los campos para el correcto funcionamiento de la importación en lote.

Importacion en lote

```
Solicitud de Licencia;NOMBRE_DE_USUARIO_DE_INFOMATEM; INSTITUCION; CODIGO_DE_PAIS; CIUDAD; OBJETIVO; AAAA/MM/DD; aaaa/mm/dd; CODIGOS_DE AREA_AMS; TITULO; TIPO DE TRANSPORTE; CANTIDAD DE TRANSPORTE; CANTIDAD DE VIATICOS; CANTIDAD DE INSCRIPCION; COMENTARIOS; ESTADO_DE_LA_SOLICITUD
```

```
Solicitud de Visitante;NOMBRE_DE_USUARIO_DE_INFOMATEM; NOMBRE DE VISITANTE; INSTITUCION; CODIGO_DE_PAIS; CIUDAD; OBJETIVO; AAAA/MM/DD; aaaa/mm/dd; CODIGOS_DE AREA_AMS; TITULO; TIPO DE TRANSPORTE; CANTIDAD DE TRANSPORTE; CANTIDAD DE VIATICOS; COMENTARIOS;ESTADO_DE_LA_SOLICITUD
```

```
Solicitud de Becario;NOMBRE_DE_USUARIO_DE_INFOMATEM,NOMBRE_DE_ASESOR_EN_INFOMATEM; GRADO; INSTITUCION; CODIGO_DE_PAIS; CIUDAD; OBJETIVO; AAAA/MM/DD; aaaa/mm/dd; CODIGOS_DE AREA_AMS; TITULO; TIPO DE TRANSPORTE; CANTIDAD DE TRANSPORTE; CANTIDAD DE VIATICOS; CANTIDAD DE INSCRIPCION; COMENTARIOS; ESTADO_DE_LA_SOLICITUD
```

Figura 4.22: Formato importación en lote

Al igual en en los formatos individuales, NOMBRE\_DE\_USUARIO\_DE\_INFOMATEM es el nombre de usuario en el sistema del investigador para el cual se está importando la solicitud y ESTADO\_DE\_LA\_SOLICITUD es el estado en el cual se inserta la solicitud al sistema, tal y como se especificó arriba.

## 4.5. Manejo de Presupuesto

Un usuario con el rol de “Manejador de Presupuesto” tendrá acceso a las vistas de “Presupuesto Total Local” y “Manejar presupuesto”. El presupuesto total local consiste en el presupuesto asignado en un determinado periodo, mientras que la vista de manejar presupuesto corresponde al resumen global de todos los presupuesto para todos los investigadores, técnicos académicos y becarios. En la figura 4.23 se aprecia una vista de un investigador que es “Manejador de Presupuesto”. Con un rectángulo rojo está marcado el acceso directo a la vista de “Manejar presupuesto”, con uno azul está marcado el acceso al “Presupuesto Total Local” (en el menú principal del contenedor) y con un recuadro morado se encuentra marcado el rol del usuario.

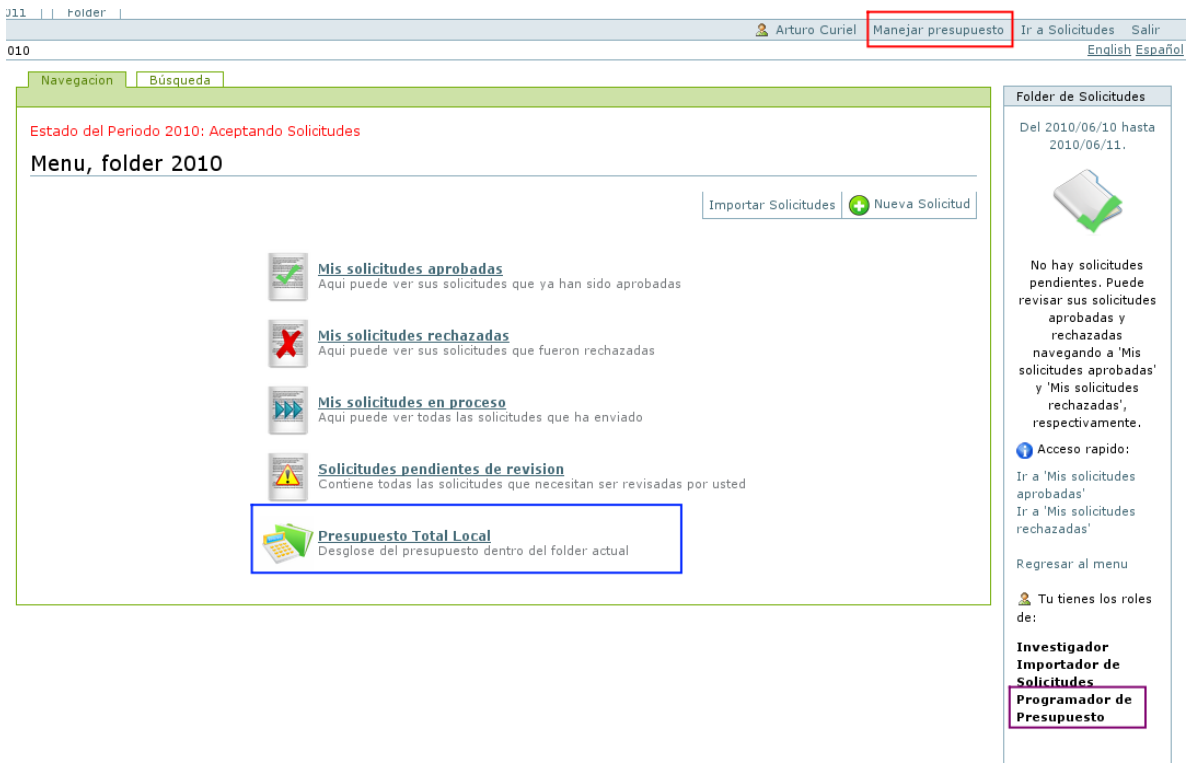


Figura 4.23: Vistas para el rol de “Manejador de Presupuesto”

### 4.5.1. Presupuesto Total Local

Esta vista es exclusiva del “Manejador de Presupuesto” y podemos ver un ejemplo en la figura 4.24. Aquí se muestra un balance total de todas las cantidades aprobadas para las solicitudes **dentro del folder de solicitudes actual** (no de todo el sistema). La vista muestra los topes máximos de presupuesto para los investigadores y becarios, *i.e.* las cantidades máximas que un investigador y un becario pueden recibir, respectivamente. Éstos valores se imponen en la vista de “Presupuesto Total Global” descrita en la siguiente sección, y aparecen en la vista local solo como referencia.

Marcado con rojo en la figura 4.24 se encuentra el campo llamado “Máximo presupuesto para este Periodo”, el cual contiene el presupuesto máximo que se puede asignar a todos los investigadores (en conjunto) durante el periodo especificado por el folder. El valor del campo puede ajustarse a juicio del manejador de presupuesto. Ésta cantidad es un tope máximo, es decir, la suma de todas las cantidades asignadas a investigadores y becarios en ese folder no deberá rebasar la cantidad impuesta en el campo.

Marcado en azul se encuentra la lista de todos los investigadores con las cantidades que les han sido asignadas dentro del folder actual. Al final hay unas filas etiquetadas con los valores “Presupuesto para este periodo”, “Presupuesto asignado localmente” y “Cantidad restante para este periodo”.

“**Presupuesto para este periodo**” es el presupuesto máximo marcado con el recuadro rojo.

“**Presupuesto asignado localmente**” es la suma total de todas las cantidades asignadas a Investigadores, Becarios y Técnicos Académicos **solo en este periodo**.

“**Cantidad restante para este periodo**” es la resta dada por “Presupuesto para este periodo” - “Presupuesto asignado localmente” y contiene la cantidad del presupuesto máximo que falta por asignar.

Navegacion
Búsqueda

### Presupuesto Total Local

**Desglose del presupuesto dentro del folder actual**  
**Folder 2010**

**Presupuesto máximo por investigador:**

	Quantity
Investigadores	\$ 3000.0
Becarios	\$ 3000.0

**Maximo Presupuesto para este Periodo: 2010/06/10 hasta 2010/06/11**

▶ Poner nuevo presupuesto

**Cantidades Aprobadas (Totales)**

Tipo	Solicitante	Cantidad aprobada para este periodo	Cantidad restante para este investigador
Investigador	Becario	0.0	0.0
Investigador	Arturo Curiel	17800.0	-14800.0
1			
Presupuesto para este periodo		80000.0	
Presupuesto ejercido localmente		17800.0	
Cantidad restante para este periodo		62200.0	

▶ Volver a calcular cantidades

Figura 4.24: Vista “Presupuesto Total Local”

Finalmente, con morado se muestra (también en la figura 4.24) un botón con la leyenda “Volver a calcular cantidades”. Este botón al ser presionado revisa que no haya

errores en los cálculos y, si los hay, recalcula los valores correctos. Sirve para asegurarnos de siempre tener las cantidades correctas.

#### 4.5.2. Presupuesto Total Global

Esta vista es exclusiva del “Manejador de Presupuesto” y podemos ver un ejemplo en la figura 4.25. Aquí se muestra un balance total de todas las cantidades aprobadas para las solicitudes dentro de **todos los folders de solicitudes en el sistema** que están abiertos (que aceptan nuevas solicitudes).

La vista muestra en un rectángulo rojo los topes máximos de presupuesto para los investigadores y becarios, *i.e.*, las cantidades máximas que un investigador y un becario pueden recibir, respectivamente. Los valores que aquí se elijan serán los usados como tope en cada folder de solicitudes.

Marcado con azul en la misma figura 4.25 se encuentra una tabla titulada “Cantidades aprobadas por periodo”. Aquí se enlistan todos los folders de solicitudes que se encuentran aceptando solicitudes (los periodos) junto a las cantidades que han sido aprobadas en cada uno de ellos. De los campos en las columnas, “Presupuesto para este periodo” es el presupuesto máximo especificado para el folder correspondiente. “Presupuesto asignado localmente” es la suma total de todas las cantidades asignadas a Investigadores, Becarios y Técnicos Académicos **en el folder correspondiente**. Finalmente, “Cantidad restante para este periodo” es la resta dada por “Presupuesto para este periodo” - “Presupuesto asignado localmente” y contiene la cantidad del presupuesto máximo que falta por asignar en el folder.

##### Presupuesto Total

##### Presupuesto máximo por investigador:

	Cantidad
Investigadores	\$ 405.0
Becarios	\$ 3000.0

[Poner nuevo presupuesto](#) [Ir asignación de presupuesto individual...](#)

##### Cantidades aprobadas por periodo

Título	Periodo	Presupuesto para este periodo	Presupuesto ejercido localmente	Cantidad restante para este periodo
2010	Del 2010/06/10 al 2010/06/11	80000.0	0.0	80000.0
2011	Del 2010/06/10 al 2010/06/18	0.0	0.0	0.0
Presupuesto maximo total (todos los periodos)			80000.0	
Presupuesto ejercido en todos los periodos			0.0	
Presupuesto restante			80000.0	

[Volver a calcular cantidades](#)

Figura 4.25: Vista “Manejar presupuesto”

Con verde se encuentra marcado un vínculo que dice “Ir a asignación de presupuesto individual...”. Ésta liga despliega, en lugar de la vista normal, una lista de investigadores y miembros del instituto con sus topes máximos de presupuesto asignados individualmente (marcado con rojo en la figura 4.26).

Abajo, en la tabla siguiente, se encuentran marcados en azul los valores totales “Presupuesto máximo total”, “Presupuesto asignado en todos los periodos” y “Presupuesto

restante”. El “Presupuesto máximo total” corresponde a la suma de los topes máximos de todos los periodos abiertos. Es decir, si tenemos un solo folder titulado **2010** con tope máximo de 10000 pesos, el “Presupuesto máximo total” será de 10000 pesos. Si tenemos un folder llamado **2010 becarios** y otro llamado **2010 investigadores**, con 5000 y 10000 pesos como tope máximo, respectivamente, entonces el “Presupuesto máximo total” sería de 15000 pesos. **Ésta suma solo contempla folders abiertos para la recepción de solicitudes.** Si, por ejemplo, en el folder **2010** de la vista 4.26 se cerrara, entonces el “Presupuesto máximo total” sería cero (porque ya no habría folders abiertos).

#### Presupuesto Total

##### Presupuesto máximo por investigador:

Persona	Cantidad	
Arturo Curiel	\$ 405.0	<a href="#">Cambiar</a>
Aureliano Buendía	\$ 45.0	<a href="#">Cambiar</a>

[Ir asignacion de presupuesto total...](#)

##### Cantidades aprobadas por periodo

Título	Periodo	Presupuesto para este periodo	Presupuesto ejercido localmente	Cantidad restante para este periodo
2010	Del 2010/06/10 al 2010/06/11	80000.0	0.0	80000.0
2011	Del 2010/06/10 al 2010/06/18	0.0	0.0	0.0
Presupuesto maximo total (todos los periodos)			80000.0	
Presupuesto ejercido en todos los periodos			0.0	
Presupuesto restante			80000.0	

[Volver a calcular cantidades](#)

Figura 4.26: Vista “Manejar presupuesto” (lista de personas)

Finalmente, con morado se muestra (también en la figura 4.26) un botón con la leyenda “Volver a calcular cantidades”. Este botón al ser presionado revisa que no haya errores en los cálculos y, si los hay, recalcula los valores correctos. Sirve para asegurarnos de siempre tener las cantidades correctas.

## 4.6. Folder de Solicitudes

La vista principal del folder de solicitudes corresponde a un menú con el cual se puede acceder fácilmente a las distintas vistas. En la figura 4.27 se puede ver un ejemplo con las vistas de un “Consejero”.

Figura 4.27: Pestañas, el camino a las vistas de un folder de solicitudes

Cada vista cumple una función específica mostrando solicitudes **dentro del mismo folder**:

**Contenidos** Es la vista principal de cualquier folder. Contiene una lista de los elementos visibles. **Solo le aparece a administradores.**

**Mis solicitudes aprobadas** Muestra todas las solicitudes que han sido aprobadas para el usuario que observa la vista y presenta la suma de todas las solicitudes que le han aprobado restada al tope de dinero que puede ser asignado a un solo investigador. En la figura 4.28 se puede apreciar que el investigador ha recibido \$ 1200 pesos para sus gastos mientras que su tope está en \$ 20000. El tope es general: para todos los investigadores se usa el mismo. Abajo de esa tabla, aparece la lista de las solicitudes aprobadas (en este caso, solo una).

Navegacion Búsqueda

Estado del Periodo 2010: Aceptando Solicitudes

### Mis solicitudes aprobadas

 Aquí puede ver sus solicitudes que ya han sido aprobadas dentro del folder 2010

Importar Solicitudes Exportar Mis Aprobadas  Nueva Solicitud

#### Cantidades Aprobadas (Totales)

Importar Exportar

Solicitante	Suma de cantidades aprobadas (pesos)
Investigador1 Apellido1	1200.0
Total Asignado	1200.0
Presupuesto Inicial	20000.0
Restante	18800.0

#### Cantidades Aprobadas (por Solicitud)

Tipo	Solicitante	Motivo	Lugar	Fecha desde	Fecha hasta	Cantidad aprobada (pesos)	Ver
Solicitud de Licencia	Investigador1 Apellido1	Visita a Institucion	Ciudad,Andorra	2010/02/09	2010/02/04	1000.0	html


Figura 4.28: Vista “Mis solicitudes aprobadas”


Navegacion Búsqueda

**¡Atención!** ¡Tiene solicitudes pendientes! Reviselas usando el cuadro ubicado a la derecha de su pantalla y que lleva por titulo 'A examinar' (dando clic sobre los recuadros) o navegando a 'Solicitudes pendientes de revision'.

Estado del Periodo 2010: Aceptando Solicitudes

### Mis solicitudes en proceso

 Aquí puede ver todas las solicitudes que ha enviado dentro del folder 2010

Importar Solicitudes Exportar Mis Solicitudes Enviadas  Nueva Solicitud

#### Solicitudes en proceso de revisión

Tipo	Solicitante	Motivo	Lugar	Fecha desde	Fecha hasta	Cantidad utilizada (pesos)	Estado actual	Liga
Solicitud de Becario	Becario	Beca para Becario en UNAM	DF,Mexico	2010/06/10	2010/06/17	0.0	Revision por Comision Especial	ver
Solicitud de Becario	Becario	Beca para Becario en UNAM	DF,Mexico	2010/06/28	2010/06/29	0.0	Revision por Tutor	ver
Solicitud de Visitante	Arturo Curiel	Visita de Dr. Pepper desde Texas State University	Dallas, TX,United States	2010/06/23	2010/06/27	0.0	Revision por Comision Especial	ver

Figura 4.29: Vista “Mis solicitudes enviadas”. Ésta vista es idéntica a la de “Mis solicitudes rechazadas” y “Mis solicitudes pendientes”

**Mis solicitudes rechazadas** En esta vista se muestran las solicitudes que fueron rechazadas por el Consejo Interno en forma de una lista.



**Mis solicitudes en proceso** Aquí se muestran, también en una lista, todas las solicitudes que ha enviado el investigador y que no han sido ni aprobadas ni rechazadas. Sirve para que el investigador pueda ver el panorama de todas las solicitudes que introducido al sistema y que no han sido resueltas.

**Solicitudes pendientes de revisión** Muestra las solicitudes que el usuario tiene que revisar. Aquí se muestran todas las solicitudes que requieren alguna acción por parte del usuario actual para poder seguir adelante con su proceso por lo cual son las de mayor urgencia.

En la figura 4.29 se muestra el formato de la vista “Mis solicitudes enviadas”. Éste formato es idéntico para las tres vistas anteriores; solo cambia la información desplegada dependiendo del estado de las solicitudes.

Las siguientes vistas, a diferencia de las anteriores, son globales; sin embargo, conservan el formato de la figura 4.29.

**Solicitudes publicas aprobadas** En esta vista se muestran **todas** las solicitudes aprobadas. Una vez que una solicitud está aprobada es pública, por lo cual cualquier usuario puede verlas.

### Búsqueda avanzada por contenido

Este formulario de búsqueda le permitirá encontrar contenido en el site especificando uno o más términos de búsqueda. Recuerde que puede usar la búsqueda rápida en cualquier momento, la cual es normalmente suficiente. Este formulario de búsqueda es sólo por si desea ser más específico.

Términos a Buscar \_\_\_\_\_

**Autor**  
Devuelve los elementos creados por un usuario.  
Solicitudes enviadas por... ▾

**País**  
País especificado en la solicitud. Elija un país... ▾

**Ciudad**  
Ciudad especificada en la solicitud. \_\_\_\_\_

**Institución**  
Institución especificada en la solicitud. \_\_\_\_\_

**Fecha de inicio**  
Fecha de inicio especificada en la solicitud. -- ▾ / -- ▾ / -- ▾ [img] -- ▾ : -- ▾

**Fecha de término**  
Fecha de finalización especificada en la solicitud. -- ▾ / -- ▾ / -- ▾ [img] -- ▾ : -- ▾

**Áreas de Investigación**  
Solicitudes del Area... ▾

**Nombre del becario**  
Seleccione un becario (si aplica)... ▾

**Asesor**  
Seleccione un asesor (si aplica)... ▾

**Objetivo de la visita**  
\_\_\_\_\_

**Título del trabajo a presentar**  
\_\_\_\_\_

**Cantidad total**  
Cantidad total aprobada o requerida en pesos (MXN) según sea el caso. 0.0

**Mayores que cantidad**  
Si la cantidad es mayor a cero, regresará las solicitudes que requieren (o tienen aprobada) una cantidad mayor a la especificada. 0.0

**Menores que cantidad**  
Si la cantidad es mayor a cero, regresará las solicitudes que requieren (o tienen aprobada) una cantidad menor a la especificada. 0.0

Figura 4.30: Vista “Búsqueda”

**Solicitudes publicas rechazadas** En esta vista se muestran **todas** las solicitudes rechazadas. Una vez que una solicitud es rechazada se hace pública, por lo cual cualquier usuario puede verlas.

**Búsqueda** Es un formato que permite buscar información dentro de los campos de las solicitudes registradas en el catálogo de contenidos del sistema, como se puede ver en la figura 4.30.

#### 4.6.1. Modificación de Intervalos y Archivado

Como se mencionó en la sección anterior, el *folder de solicitudes* es un contenedor de objetos de tipo *Solicitud de Licencia*, *Solicitud de Visitante* y *Solicitud de Becario*. Se compone de cuatro datos principales:

**Título** Es el identificador del folder, su nombre dentro del sitio.

**Descripción** Es un campo que describe los contenidos del folder.

**Fecha de Inicio** Es una fecha que marca el inicio del periodo al que corresponde el folder.

**Fecha de Cierre** Es una fecha que marca el fin del periodo al que corresponde el folder.

Con las fechas de inicio y fin se enmarca el intervalo del tiempo en el cual se hicieron las solicitudes contenidas en el folder. Ambos campos están registrados en el catálogo y pueden ser utilizados con, por ejemplo, colecciones para mejorar la organización. En la figura 4.31 se puede observar la pantalla de creación de un folder de solicitudes.

Folder de solicitudes.

Predeterminado ■ Categorización Fechas Propietario Configuración

**Título ■**  
Folder de Solicitudes

**Descripción**  
Una breve descripción del contenido.  
Folder de Solicitudes

**Start date ■**  
Date on which this period starts  
2010 / marzo / 5

**End date ■**  
Date on which this period ends  
2010 / marzo / 12

Guardar Cancelar

Figura 4.31: Campos de un folder de solicitudes

El folder de solicitudes tiene, además, tres estados: “Aceptar solicitudes”, “Dejar de aceptar solicitudes” y “Archivado”. En la figura 4.32 se pueden apreciar éstos. Cuando un folder acepta solicitudes, significa que cualquier investigador puede introducir una nueva solicitud. Por otro lado, cuando un folder deja de aceptar solicitudes significa que ninguna solicitud nueva puede ser ingresada; sin embargo, las solicitudes que ya se están procesando o ya han sido procesadas pueden ser accedidas por cualquier usuario autenticado. Cuando un folder se “manda a archivo”, significa que ya no puede ser modificado y que ya no cuenta para el presupuesto actual del Instituto. Esto sirve para que las solicitudes dentro del folder no sean consideradas en el nuevo presupuesto total. Por ejemplo, supongamos que termina el periodo 2010 y empieza el 2011. Si no mandamos el folder 2010 a archivo, las cantidades aprobadas se descontarán del presupuesto del 2011.

Los folders archivados conservan la información de las solicitudes y el total de las cantidades aprobadas para el periodo que representa, pero solo para fines de consulta. Los valores no influyen en el presupuesto de los folders aún activos.

Solo un administrador puede crear y modificar folders de solicitudes (incluyendo el cambio de “aceptar” a “no aceptar” o “archivar”).

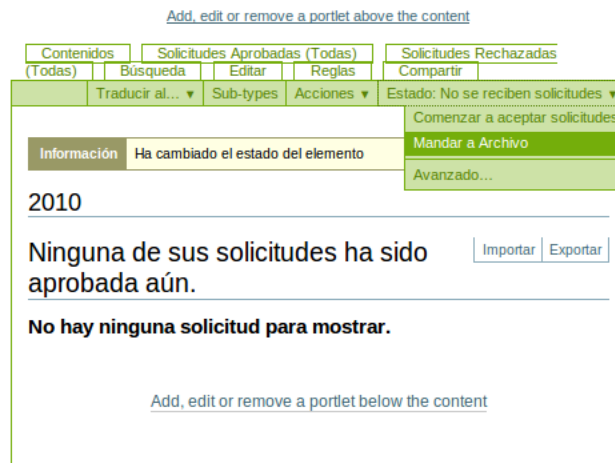


Figura 4.32: Estados en un folder de solicitudes

## 4.7. Portlet Avisos

El portlet Avisos es una herramienta sencilla que despliega mensajes distintos dependiendo del contexto en el que se encuentre. Solo puede diferenciar, hasta el momento, dos de éstos: el contexto “Folder” y el contexto “Solicitud”. Para añadirlo a la navegación basta con entrar a “Agregar Portlets” y seleccionarlo de la lista.

### 4.7.1. Contexto “Folder”

En el contexto folder el portlet desplegará dos posibles estados. Si hay solicitudes pendientes (solicitudes que necesitan revisión por parte del usuario actual) se mostrará el estado pendiente como en la figura 4.33.

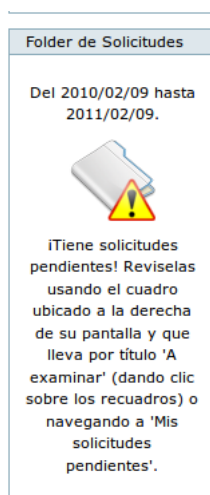


Figura 4.33: Portlet en contexto “Folder”, modo “Pendientes”

Si, por otro lado, no hay solicitudes pendientes se mostrará el estado de la figura 4.34.

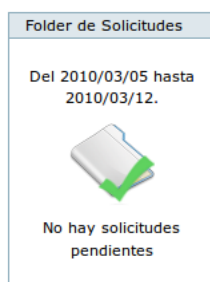


Figura 4.34: Portlet en contexto “Folder”, modo “Sin pendientes”

Es en el portlet donde se muestran las fechas establecidas para el folder de solicitudes.

### 4.7.2. Contexto “Solicitud”

En el contexto “Solicitud” el portlet desplegará cinco posibles estados. Si hay solicitudes pendientes (solicitudes que necesitan revisión por parte del usuario actual) se mostrará el estado pendiente como en la figura 4.35.

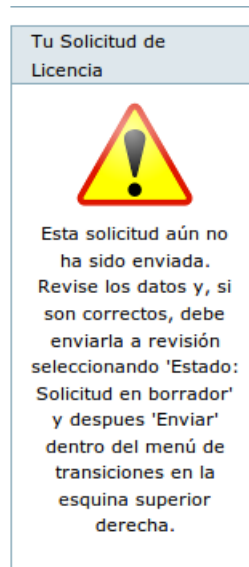


Figura 4.35: Portlet en contexto “Solicitud”, modo “Pendientes”

Si la solicitud es enviada y aún no la revisa la Comisión Especial, se mostrará la figura 4.36.

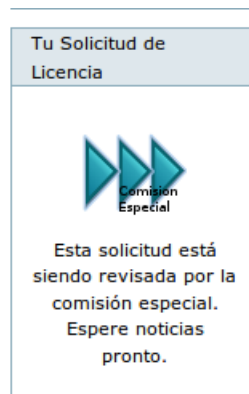


Figura 4.36: Portlet en contexto “Solicitud”, modo “Revisión por Comisión Especial”

Si la solicitud está en revisión por el Consejo Interno, se mostrará la figura 4.37.

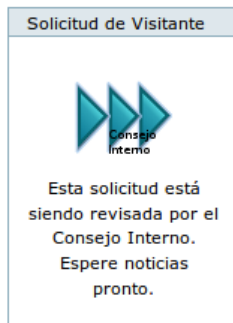


Figura 4.37: Portlet en contexto “Solicitud”, modo “Revisión por Consejo Interno”

El Consejo Interno tiene la capacidad de sancionar tanto para aprobar como para rechazar una solicitud. Toda solicitud sancionada se hará visible para todos los usuarios. En este caso la figura mostrada sería 4.38 o 4.39 según sea el caso.

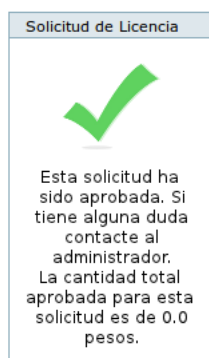


Figura 4.38: Portlet en contexto “Solicitud”, modo “Aprobada”



Figura 4.39: Portlet en contexto “Solicitud”, modo “Rechazada”



# Capítulo 5

## Conclusiones

La aplicación del sistema dentro del Instituto de Matemáticas tuvo varios resultados, principalmente basados en la experiencia de los usuarios y el contacto de los mismos con la administración del Instituto. En este capítulo se muestran las observaciones más relevantes que se hicieron, una vez desplegado el sitio.

### 5.1. Sobre el proceso administrativo

La ventaja más evidente de la automatización completa del proceso de envío de solicitudes, está en la contabilidad. Previo a la introducción del software aquí descrito, la contabilidad dentro del Instituto de Matemáticas se hacía completamente a mano, auxiliada por hojas de cálculo o bases de datos que se debían llenar a mano a medida que las solicitudes eran revisadas. El proceso, al ser completamente dependiente del personal administrativo, estaba sujeto a errores y no reportaba datos automáticamente. Mediante *Solicitudes*, se logro que:

- InfoMatem calcule en tiempo real el presupuesto total del Instituto, descartando los errores que pueden surgir en la contabilidad tradicional.
- El Consejo Interno tenga información detallada sobre el presupuesto disponible para cada investigador, becario y técnico académico; de la misma forma, los consejeros pueden revisar las solicitudes asignadas tanto por periodo como por investigador para determinar los montos exactos asignados a cada miembro del Instituto.
- El proceso de entrega y revisión de solicitudes se agilice, ya que en el momento que el Investigador termina de introducir la información, ésta ya se encuentra disponible para su revisión por el siguiente usuario en la cadena de *workflow*.
- La migración de la información del sitio se pueda llevar a cabo muy rápidamente, ya que la importación y exportación en bloque permiten copiar todas las solicitudes de en un sitio Plone a otro sin mayor complicación y conservando la integridad de los datos.



Además, la creación de los roles descritos dentro del sistema garantiza que los permisos se puedan asignar y despojar de usuarios específicos en caso de haber cambios en los miembros del consejo o de las autoridades administrativas del Instituto. Al reasignar roles, los nuevos propietarios pueden de inmediato revisar la información disponible y ejecutar las mismas operaciones que sus antecesores, haciendo este tipo de transiciones más amigables.

## 5.2. Sobre la interacción con los usuarios

En cuanto a la interacción con los usuarios de InfoMatem (Investigadores, Becarios y Técnicos Académicos), se lograron las siguientes ventajas mediante la implementación del producto:

- Los usuarios ven en tiempo real todas las opciones que tienen para crear, modificar o retirar solicitudes. Mediante la ayuda de elementos como el *portlet* “Avisos” y la ayuda contextual, los usuarios tienen más claro como llenar sus solicitudes.
- Cada usuario puede ver cuanto dinero ha gastado, lo que le permite formarse un criterio a la hora de crear una nueva solicitud y elegir cantidades más apropiadas basadas en los registros a los que tiene acceso.
- El tiempo que existía anteriormente entre la aprobación de una solicitud y el aviso al solicitante se reduce a cero, ya que en el instante en que el Consejo Interno aprueba o rechaza alguna de éstas, un aviso aparece en la cuenta de InfoMatem del solicitante que le permite comprobar el estado actual de la solicitud.

De esta forma, *Solicitudes* reduce la distancia y los tiempos de respuesta entre la creación de la solicitud y la sanción por parte del Consejo Interno, siempre dando información clara y transparente al solicitante.

## 5.3. Problemas Resueltos Sobre el Diseño

Durante el diseño y la implementación surgieron algunos problemas que fueron tratados en su momento. El primero de éstos, fue sobre la disponibilidad de información para los usuarios.

Aunque a los usuarios les gusta tener muchas opciones, las quejas principales durante las pruebas del sistema fueron que había demasiada información en pantalla. Se tuvieron que implementar vistas claras, con iconos fácilmente distinguibles, que confundieran lo mínimo posible al usuario. Además, cada opción dentro de los menús tuvo que ser explicada con la mayor claridad posible y algunas de las opciones originales debieron ser eliminadas para dar paso a una interfaz más ligera.

También, durante las pruebas fue evidente que los usuarios necesitan ver la información relevante en todo momento; ésto es, la información sobre su presupuesto, el estado de las solicitudes y las instrucciones básicas fueron datos que se integraron a las vistas

principales para que siempre estuvieran a la vista. De esta forma, el usuario no tiene que buscar demasiado a la hora de revisar sus solicitudes.

Finalmente, la construcción de *workflows* tuvo que ser simplificada al máximo para dejar libertad a los administradores del sitio para manipular información añadida por usuarios. Un *workflow* demasiado complejo o estricto, podía ocasionar problemas a la hora de introducir, por ejemplo, solicitudes atrasadas o ya aprobadas de antemano. Un *workflow* sencillo permite minimizar la automatización, de forma que siempre queda espacio para la intervención humana en caso de existir problemas técnicos con alguna solicitud.

## 5.4. Comunicación Con Otros Sistemas y Trabajo Futuro

El producto aún no se comunica con bases de datos externas, sin embargo se implementaron los métodos necesarios para obtener la información de presupuesto individual siempre y cuando se tengan los permisos adecuados. En la sección 3.4 se introdujo la clase *Solicitante*, la cual que posee métodos que permiten acceder desde otros productos a la información contable de un investigador o un becario sin perjudicar la integridad de los datos. Como la clase *Solicitante* está sujeta a la clase *FSDPerson* que, a su vez, está sujeta a la herramienta de *plone\_membership*, solo los usuarios con permiso de consultar el catálogo de miembros del sistema pueden instanciarla y obtener la información recabada por el producto (para cada miembro). Un adaptador simple que recorra toda la lista de miembros puede importar la definición de clase y calcular los totales sin necesidad de modificar el producto.

Actualmente está en desarrollo una *script* basado en *ZopeSkel* que permitirá la creación de un producto con las mismas pautas de funcionamiento, pero personalizado para su utilización generalizada. Dicho *script* será liberado bajo licencia GPL a principios de 2011 y permitirá facilitar la actualización y mantenimiento del producto en caso de que existan cambios importantes ,tanto en el sitio principal del IMATE como en el proceso de creación y registro de solicitudes de viáticos; con el *script*, se podrá generar automáticamente otro producto que cumpla con los nuevos requisitos pero que, a su vez, conserve el mismo diseño básico descritos en este documento. Así, se espera que los desarrolladores encargados de darle mantenimiento al sistema puedan darle continuidad al producto y modificarlo a su conveniencia sin mayor problema (ademas de que puede ser de utilidad para otras instituciones).



# Bibliografía

- [1] Wil Van Der Aalst, Kees Van Hee, Prof. Dr. Kees, Max Hee, Remmert Remmerts De Vries, Jaap Rigter, Eric Verbeek, and Marc Voorhoeve. *Workflow Management: Models, Methods, and Systems*. The MIT Press, Cooperative Information Systems series, 2004.
- [2] Martin Aspeli. *Professional Plone Development*. Packt Publishing, Birmingham, UK., 1 edition, September 2007.
- [3] Eduardo Espinosa Avila. Desarrollo de un sistema de administración de procesos en plone. Master's thesis, Universidad Nacional Autónoma de México, Mexico, D.F., 2009.
- [4] Bob Boiko. *Content Management Bible, 2nd Edition*. Wiley Publishing Inc., Indianapolis, 2 edition, 2005.
- [5] Zope Corporation. What is zope? Online, 2010.
- [6] Zope Corporation. What is zope 3? Online, 2010.
- [7] Plone Foundation. What is plone? Online, 2010.
- [8] Python Foundation. What is python? executive summary. Online, 2010.
- [9] UNAM Instituto de Matemáticas. Infomatem: Sistema de información del instituto de matemáticas. Online, 2010.
- [10] Andreas Ulrich Mauthe and Peter Thomas. *Professional Content Management Systems: Handling Digital Media Assets*. John Wiley & Sons, 2004.
- [11] Andy McKay. *The Definitive Guide to Plone*. Apress, New York, USA., 1 edition, 2004.
- [12] Lance Merker. Best practices for using a cms. *University Business, Solutions for Higher Education Management*, July 2007.
- [13] Mark Pilgrim. *Dive into Python*. Free Software Foudation, May 2004.
- [14] Rob Prideaux. Knowing when you need a cms. CompuMentor (Online), 2004.
- [15] Dirk Riehle. *Framework Design: A Role Modeling Approach*. Swiss Federal Institute of Technology, 2000.

- [16] Phil Suh, Dave Addey, and James Ellis. *Content Management Systems*. Glasshaus, 2003.
- [17] Philipp von Weitershausen. *Web Component Development with Zope 3*. Springer, January 2007.
- [18] Amos Latteier y Michel Pelletier. *The Zope Book*. Sams, July 2001.
- [19] Chuanhong Zhou and Huilan Zeng. Enterprise knowledge management based on plone content management system. In *PROLAMAT*, pages 115–120, 2006.