



---

## ADMINISTRADOR DE TAREAS PARA APLICACIÓN FISCAL

*Rismary Cabrera Estévez*

*Havanatur (Cuba)*

*risitos1987@gmail.com*

*Calle 8 No.30 Reparto La Gloria, Camagüey, Cuba, 535-3847393*

*Fecha de envío: 20/Marzo/2015*

*Fecha de aceptación: 09/Abril/2015*

---

### **Resumen / Abstract**

El sector jurídico ha sido beneficiado como parte de la informatización de la sociedad. Formando parte de esta iniciativa se encuentra el proyecto Sistema de Gestión Fiscal (*SGF*) que tiene como objetivo informatizar los procesos fiscales para agilizar la toma de decisiones, la eficiencia en el trabajo y elevar los niveles de conformidad por parte de los ciudadanos. Diariamente en las fiscalías se recibe un gran número de casos que tienen términos que no pueden vencerse. Controlarlos es un proceso complicado que requiere de una buena gestión de la información entre las partes involucradas. Esta investigación tiene como objetivo ofrecer una aplicación libre para el proyecto *SGF*, que automatice los procesos de control y seguimiento de los casos y que permita mantener actualizados a los fiscales de todos los procesos y que la misma pueda integrarse fácilmente a proyectos que lo requieran.

The legal sector has been benefited as part of the computerization of society. The Fiscal Management System (*FMS*) project is included in this initiative, which aims to computerize the fiscal processes to streamline decision-making, work efficiency and raise levels of compliance by citizens. Daily, in each of the procurator a large number of cases with terms which cannot expire are received. The control of these cases is a very complicated process that requires good management information among the parts involved. This research aims to offer a free application for *FMS* project that automates the process of controlling and monitoring of cases and allows prosecutors to be updated on all processes and that it can be easily integrated into projects that require it.

**Palabras Claves / Key Word:** aplicación web, tarea(s) automática(s), administrador de tareas, notificaciones / automatic task, notifications, task manager, web application.

## **Introducción**

El progreso de las Tecnologías de la Informática y las Comunicaciones (TIC) ha llevado consigo un cambio drástico a nivel mundial. Actualmente estas tecnologías están sufriendo un desarrollo vertiginoso que beneficia prácticamente a todas las esferas de nuestra sociedad, convirtiéndose en una herramienta fundamental en el desarrollo económico y social de cualquier nación. Estos cambios han tenido un gran impacto en la forma de revolucionarnos, comunicarnos y estructurar el pensamiento, generando una gran influencia en la toma de decisiones de disímiles instituciones.

Entre los diferentes programas de informatización de la sociedad cubana se encuentra el desarrollo de una aplicación web en la Universidad de las Ciencias Informáticas (UCI) para los procesos en las fiscalías, con el objetivo de agilizar la toma de decisiones y elevar el control de los casos que se procesan.

La asignación del trabajo en las fiscalías se realiza de jefe a subordinados. Por lo general cada jefe tiene un grupo numeroso de subordinados y estos a su vez un gran número de casos a procesar, debiendo estar pendiente en todo momento de cada uno de ellos, ya que tienen plazos definidos e inquebrantables para su cumplimiento y/o entrega. Todo este proceso requiere un alto volumen de información a tramitar, generando gran cantidad de documentos que precisan de un considerable empleo de tiempo y esfuerzo, lo que hace más complejo el control que debe llevarse sobre todos los documentos y generando gran carga de trabajo para los fiscales, lo que trae consigo que existan deficiencias en la ejecución del trabajo.

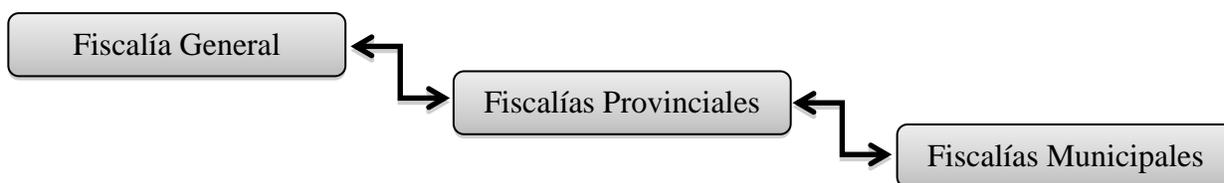
El proyecto, cuyos resultados aquí se exponen, surgió precisamente como una forma de dar solución a la problemática existente, teniendo como Objetivo desarrollar un administrador de tareas que automatice el trabajo de la administración y control de las tareas existentes en el sistema SGF y que pueda ser integrado a otros proyectos que lo requieran.

## Marco Teórico

Un Fiscal, agente fiscal, agente del ministerio público, procurador fiscal o promotor fiscal es el funcionario público o magistrado en algunos países, integrante del Ministerio Público, que lleva materialmente la dirección de la investigación criminal y el ejercicio de acción penal pública; es decir, es a quien corresponde desempeñar directa y concretamente las funciones y atribuciones de este, en los casos que conoce. Su misión institucional es proteger el orden político y jurídico del Estado y la sociedad, procurando el restablecimiento de la legalidad quebrantada promoviendo acciones y medidas contra los infractores. (Colaboradores de Wikipedia, 2015)

Estos funcionarios, en los casos que tienen a su cargo, ejercen sus funciones con independencia y autonomía, bajo los principios de legalidad y objetividad, debiendo practicar todas las diligencias que fueren conducentes al éxito de la investigación, siendo responsables, en el ejercicio de sus funciones, en el ámbito civil, penal y disciplinario. Las autoridades y jefaturas del Ministerio Público, en los distintos sistemas, suelen ejercer un control jerárquico más o menos permanente de los agentes de su dependencia, que se extiende tanto a la eficacia y eficiencia en el cumplimiento de sus objetivos, como a la legalidad y oportunidad de sus actuaciones. (Colaboradores de Wikipedia, 2015)

En la “Figura 1” se muestran los niveles jerárquicos existentes en la entidad en cuestión y el flujo de información entre estos.



*Figura 6. Estructura Jerárquica en la Fiscalía*

El control de las tareas se realiza actualmente de la siguiente manera:

**Autocontrol:** El Fiscal que está llevando un caso debe ser capaz de mantener un control del mismo, estar bien actualizado en todo momento con los límites de tiempo que dispone para terminar el proceso.

**Seguimiento:** Luego de la asignación de un caso por un Fiscal Superior a alguno de sus subordinados el mismo debe dar seguimiento a dicho caso, mantenerse actualizado en todo lo referente a la situación procesal del mismo.

Mundialmente existen diversos sistemas para la ayuda a la toma de decisiones de los fiscales, entre los que se pueden citar:

**SGF (Sistema de Gestión Fiscal) – Perú:** es un software que establece plazos de registro en la investigación de una carpeta fiscal supervisado por una Unidad de Seguimiento y Gestión, implementado con el nuevo sistema procesal penal.

**SAF (Sistema de Apoyo a los Fiscales) - Chile:** El SAF es un Sistema Computacional que apoya la labor de investigación y posterior desarrollo del proceso penal, a través de las distintas opciones y funciones contenidas en los módulos que lo componen. Permite administrar con agilidad un mayor número de causas penales, manejo eficaz de información y trabajo en equipo entre los involucrados.

**SIG – Fiscalía (Sistema de Información y Gestión de Fiscalía) – Uruguay.** Sistema informático para la modernización de la operatividad de las fiscalías.

En todos estos sistemas en determinado momento se podrá acceder a los procesos que se estén ejecutando, así como del estado en que se encuentren, pero siempre con accesos a reportes del sistema que deben consultar los usuarios.

En el caso de la aplicación web (**SGF** en Cuba) es similar, los fiscales manejan todos sus procesos y actividades mediante el sistema según la esfera de trabajo a la cual pertenecen y luego al darle alta en el sistema al expediente, se podrá hacer consultas del estado de los casos mediante los reportes del sistema siempre solicitados por el usuario ya que no dispone de ningún módulo que supervise los procesos.

Para la administración automática de procesos se cuenta con diversas aplicaciones, las cuales son programas informáticos diseñados para ejecutar procesos definidos a intervalos regulares. Los procesos que deben ejecutarse se especifican en una base de datos o fichero. Además proporcionan información sobre el estado de dichos procesos permitiendo la modificación de los mismos. Partiendo de una aplicación el administrador se encarga de la ejecución, gestión y control de los procesos que se generan desde la aplicación, verificando a cada minuto para ver si existe alguna tarea a realizarse. Todo esto se efectúa en segundo plano.

Los sistemas operativos gestionan los programas mediante el concepto de proceso. En un instante dado, en el ordenador pueden existir diversos procesos listos para ser ejecutados. Sin embargo, solamente uno de ellos puede ser ejecutado (en cada microprocesador). De ahí la necesidad de que una parte del sistema operativo gestione, de una manera equitativa, qué proceso debe ejecutarse en cada momento. Es por ello que surgen los primeros administradores de tareas, que con su evolución permitió administrar otras tareas que no fuesen propias del sistema.

Con el transcurso del tiempo se crean nuevas herramientas para gestionar y/o administrar procesos propios de diversos programas con una independencia parcial de la que brinda el sistema operativo, aparecieron disímiles software para satisfacer requerimientos de otros programas, permitiendo además, que usuarios interactuaran con ellos y ejercieran un rol para

gestionar los diferentes procesos. Con la aparición de complejas aplicaciones web para gestionar el negocio concerniente a empresas, surge la necesidad de que existieran sistemas de administración para sus tareas, que garantizaran el correcto funcionamiento de estas y permitieran su monitoreo y gestión.

En la actualidad es frecuente que la responsabilidad de la ejecución de los crones (tareas a ejecutar) de las aplicaciones web sea el Cron del sistema operativo, ejecutando los scripts que satisfacen las necesidades del sitio web. Algunos CMS (Sistema de Administración de Contenido) traen unido su propio sistema para estas gestiones, como es el caso de Joomla y Drupal.

Los sistemas de administración automáticos de tareas se caracterizan por poseer interfaz liviana, de fácil accesibilidad y con algoritmos que permitan su actualización automática y constante, permitiendo mostrar el contenido real en cada momento, se construyen según la necesidad y los requerimientos, deben permanecer corriendo de forma permanente para posibilitar la ejecución de sus trabajos en tiempo y forma, suelen incluir la coordinación de tareas humanas con tareas automáticas y suponen un constante monitoreo del estado de los procesos que tramitan, así como la gestión de los mismos.

Dentro de las ventajas de los sistemas para administrar tareas se citan la automatización de tareas, la posible ejecución de procesos, la centralización de la información de los procesos, la reducción de costos en cuanto a recursos humanos destinados con anterioridad para estas funciones y la garantía de reportes fiables de los procesos gestionados.

Al igual que cualquier programa informático, los administradores de tareas cuentan con desventajas a pesar de las disímiles ventajas que presentan, entre ellas se encuentran que una

falla o falta de energía eléctrica puede provocar que queden procesos sin ejecutarse en el momento señalado, al igual que un excesivo número de procedimientos a ejecutarse automáticamente puede colgar el ordenador, ya que su trabajo se basa en hilos del sistema.

Como ejemplos de este tipo de aplicaciones existentes se pueden citar:

**Quartz:** Este es un framework (herramienta) de scheduler (planificador) de nivel empresarial, en base Java J2EE, que posee dentro de sus características principales, niveles de administración de clúster, y tolerancia a fallas, manejo transaccional y lo mejor de todo es una solución de código abierto, basada en licencia de tipo Apache 2.0. Quartz opera embebido en una aplicación Java, en un servidor de aplicaciones y de otras formas similares, calendarizando, ejecutando y notificando los procesos que se implementen con el mismo. (Urrutia, 2006)

**QuelIt:** Es una poderosa utilidad de planificación que permite planificar programas, tareas o incluso ficheros batch MS-DOS para cargar automáticamente a las horas específicas que se elijan. Con QuelIt se puede planificar un programa para cargar diariamente, semanalmente, mensualmente o a cada hora. Admite incluso ser configurado para que ejecute una tarea o programa cada minuto. QuelIt también soporta ejecuciones condicionales (por ejemplo, sólo ejecutar un programa si un fichero existe). Permite enviar argumentos mediante líneas de comandos al programa que esté lanzando. Incluso consigue enviar un correo electrónico después que una tarea ha comenzado o se ha completado. Es capaz de ejecutar sus tareas planificadas en ordenadores remotos.

**Cron Jobs del CPanel:** CPanel (acrónimo de Panel de Control) es una herramienta de administración basada en tecnologías web para administrar sitios que incluye el módulo Cron Jobs. Los trabajos de cron permiten automatizar ciertos comandos y scripts en un sitio y

básicamente ejecutan tareas programadas por el usuario. Puede configurar un comando o script para correr a un tiempo específico cada día, semana, etc. Por ejemplo, consigue poner un trabajo de cron que borre archivos temporales para que el espacio de disco no sea usado por esos archivos.

Estas aplicaciones analizadas sirven de apoyo para la administración de procesos, pero no pueden ser utilizados como solución ya que todas tienen la desventaja de depender completamente del Cron (para Linux) o el Sheduler (para Windows) lo que traería consigo una dependencia total de un servicio del sistema operativo atentando contra la portabilidad del sistema requerido. Además se requiere de una propuesta capaz de administrar tareas muy propias de aplicaciones existentes en las fiscalías y estos sistemas sólo proporcionarían pequeñas funcionalidades.

## 7. Método

Para el desarrollo de esta investigación se utilizaron diferentes métodos empíricos y teóricos.

### **Métodos Teóricos:**

Entre los métodos teóricos el **Análisis Histórico Lógico** que permitió estudiar de forma analítica la trayectoria histórica real de los fenómenos, su evolución y desarrollo, el cual se aplicó en la elaboración del estado del arte posibilitando realizar un análisis de la evolución de los sistemas y herramientas para la gestión y administración de tareas.

Otro método teórico utilizado es el **Analítico – Sintético** que facilita el entendimiento del fenómeno en el que se trabaja, es más útil la división de este en diferentes fases, y de esta forma descubrir sus características generales, lo que ayuda a seguir una correcta investigación. Su uso

permitió realizar una investigación y estudio detallado de documentos y del sistema para así identificar el problema concreto existente, sus causas y lograr un profundo análisis y diseño.

### Métodos Empíricos:

Entre los métodos empíricos a utilizar se empleó la **Entrevista**. Se entrevistaron diferentes fiscales con el objetivo de profundizar sobre las distintas tareas que realizan los mismos. También se entrevistó al jefe de desarrollo del proyecto para conocer las tareas que son generadas por el sistema ya implementado.

Después de realizar un estudio de diferentes metodologías, lenguajes y herramientas se realizó una selección basada en el uso de tecnologías libres en aras de obtener resultados profesionales y medibles a corto plazo, adecuadas al sistema SGF y al logro de los objetivos de la investigación.

Como metodología de desarrollo se seleccionó **RUP** (del inglés Rational Unified Process), la cual dividiendo el proyecto en cuatro fases, las que se muestran en la “Figura 2”, es una guía para los equipos de proyectos en cómo administrar el desarrollo iterativo de un modo controlado, mientras se balancean los requisitos del negocio, el tiempo al mercado y los riesgos del proyecto.

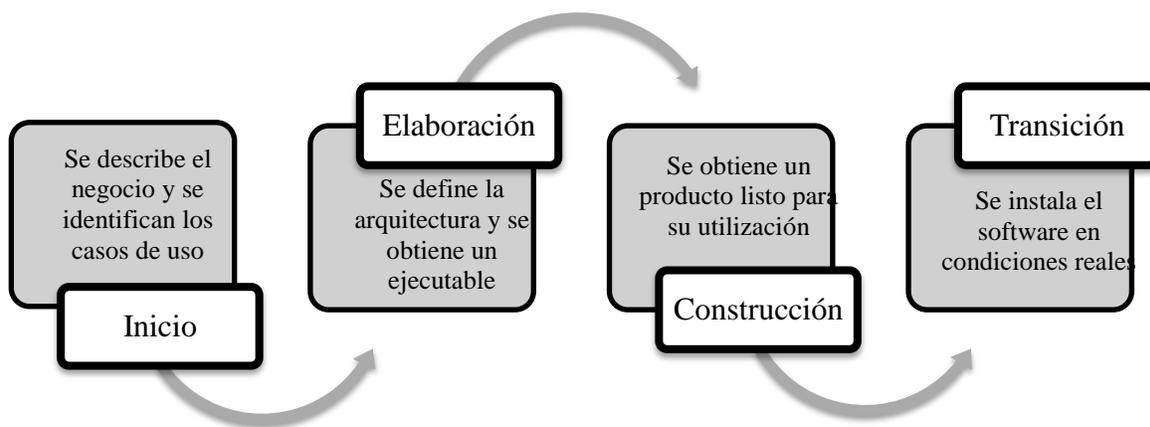


Figura 7. Fases de Desarrollo del Software

El proceso describe los diversos pasos involucrados en la captura de los requisitos, detalla qué entregables producir, cómo desarrollarlos y también provee patrones. Haciendo uso de esta metodología se crearán los artefactos correspondientes a los flujos de trabajos de Modelación del Negocio y Levantamiento de Requisitos, con la ayuda del **Lenguaje de Modelado Unificado (UML)**.

El Lenguaje de Modelado Unificado **UML** es un lenguaje estándar para escribir planos de software. Se utiliza para definir un sistema, detallar los artefactos en el mismo y para documentar y construir. A su vez ayuda a interpretar grandes sistemas mediante gráficos o texto, obteniendo modelos explícitos que contribuyen a la comunicación durante el desarrollo, ya que al ser estándar, pueden ser interpretados por personas que no participaron en su diseño. En este contexto, UML sirve para especificar modelos concretos, no ambiguos y completos.

Como herramienta CASE (computer aided software engineering, ingeniería de software asistida por computadora) se seleccionó el **Visual Paradigm** el cual es una herramienta multiplataforma de modelado visual UML y una herramienta CASE muy potente y fácil de utilizar. Aporta a los desarrolladores de software una plataforma de desarrollo puntera para construir aplicaciones de calidad. Aporta una excelente interoperabilidad con otras herramientas CASE y muchos de los entornos IDE líderes del mercado.

Con la selección descrita y aplicando la **Ingeniería de Requisitos** y sus disímiles técnicas que apoyan la ejecución de las actividades que se realizan durante las etapas que se desarrollarán, al finalizar este proceso, se tendrán identificados los requisitos funcionales y no funcionales del software y concluyendo con la validación que permite comprobar el cumplimiento a las necesidades de los clientes.

La implementación estará dividida en dos fases, un módulo web y un módulo desktop.

Como IDE (entornos integrados de desarrollo) para el desarrollo Web se seleccionó **Eclipse** el cual es una plataforma de programación utilizada para crear entornos integrados de desarrollo. Es como una armazón sobre la que se pueden montar herramientas de desarrollo para cualquier lenguaje, mediante la implementación de los plugins (del inglés «[un] enchufable o inserción») adecuados.

La arquitectura de plugins de Eclipse permite, además de integrar diversos lenguajes sobre un mismo IDE, introducir otras aplicaciones accesorias que pueden resultar útiles durante el proceso de desarrollo como: herramientas UML, editores visuales de interfaces y ayuda en línea para librerías. (The Eclipse Foundation, 2015)

Como lenguaje de programación se seleccionó **PHP** (Hypertext Pre-processor), este es un lenguaje de programación libre para la creación de páginas web dinámicas. Permite la creación de aplicaciones con interfaz gráfica, conexión a servidores de base de datos y puede ser ejecutado en sistemas Unix, Windows, Linux y Mac OSX. Es un lenguaje de alto nivel, sirve principalmente para proporcionar características dinámicas a una página web.

En el caso del lado del cliente se eligió **JavaScript**, este es un lenguaje de programación utilizado para crear pequeños programas, encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Su uso se basa, fundamentalmente, en la creación de efectos especiales en las páginas y la definición de interactividades con el usuario. Es un lenguaje muy eficaz para operaciones rápidas y con requerimientos gráficos que no dependen totalmente del servidor.

Se seleccionó el Framework **Symfony**, este es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web mediante algunas de sus principales características, entre las que se encuentran que separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona además varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Como servidor Web **Apache Web Server**. Es un servidor de aplicaciones web, Multiplataforma, Es un servidor de web conforme al protocolo HTTP/1.1. Es modular ya que puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API (*Application Programming Interface*) de programación de módulos, para el desarrollo de módulos específicos, gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP.

Servidor de base de datos **PostgreSQL** es un sistema de gestión de base de datos relacional orientada a objetos y libre.

Para la solución desktop se selecciona el lenguaje de programación **Python** ya que es un lenguaje de programación interpretado, lo que ahorra un tiempo considerable en el desarrollo del programa, pues no es necesario compilar ni enlazar. El intérprete se puede utilizar de modo interactivo, lo que facilita experimentar con características del lenguaje, escribir programas desechables o probar funciones durante el desarrollo del programa. Es un lenguaje multiplataforma, pues el mismo código funciona en cualquier arquitectura, la única condición es que disponga del intérprete del lenguaje. No es necesario compilar el código una vez para cada arquitectura. Es muy útil en el desarrollo de aplicaciones que trabajen con hilos del sistema o

demonios. “El intérprete de Python (**Python Interpreter**) estándar incluye un modo interactivo, en el cual se escriben las instrucciones en una especie de shell: las expresiones pueden ser introducidas una a una, pudiendo verse el resultado de su evaluación inmediatamente. Esto resulta útil tanto para las personas que se están familiarizando con el lenguaje como también para los programadores más avanzados: se pueden probar porciones de código en el modo interactivo antes de integrarlo como parte de un programa.” (Ecured, 2015)

Como IDE de desarrollo (Entorno de Desarrollo Integrado, por sus siglas en inglés) para este lenguaje se seleccionó **PyScripter**, ya que es una herramienta libre para el diseño de aplicaciones en el lenguaje Python. Tiene la ventaja de ser bastante ágil con respecto a otros y proporciona una amplia mezcla de características que lo convierten en un entorno de desarrollo productivo. Posee un alto grado de completamiento de código, plantillas de parámetros de código, ventana de relojes, explorador de archivo de fácil configuración, explorador de código, editor de ventanas, etc.

Para la creación del ejecutable de la aplicación se selecciona **Py2exe** el cual es una extensión que convierte los scripts de Python en programas ejecutables incluyendo en su interior las librerías necesarias para la ejecución del distributable, de manera que puedan ejecutarse sin necesidad de instalar el intérprete de Python. Ofrece la ventaja de la distribución al usuario final sólo del ejecutable de la aplicación desarrollada, siendo transparente para el lenguaje en el cual fue desarrollado, librerías y técnicas utilizadas para su desarrollo.

Para el diseño de la interfaz gráfica se prefiere **PyQt GPL** el cual es un complemento de la biblioteca gráfica Qt para el lenguaje de programación Python. “Qt es una biblioteca

multiplataforma para desarrollar interfaces gráficas de usuario y también para el desarrollo de programas sin interfaz gráfica como herramientas de la consola y servidores”. (Custom 2014)

## 8. Resultados

Con el desarrollo de la presente investigación, se obtuvo una herramienta administradora de tareas que enlazada a un módulo para la web del sistema de gestión fiscal es capaz de mantener informados mediante un sistema de avisos a los fiscales de los términos de sus procesos.

Una vez los procesos se encuentran en el sistema de gestión fiscal para el administrador de tareas es una tarea a ejecutar su seguimiento y se encontrará ejecutándose en segundo plano chequeando a intervalos de tiempos (definidos) si existe alguna tarea a procesar. En la Figura “3” puede verse su integración entre el administrador y el sistema SGF.

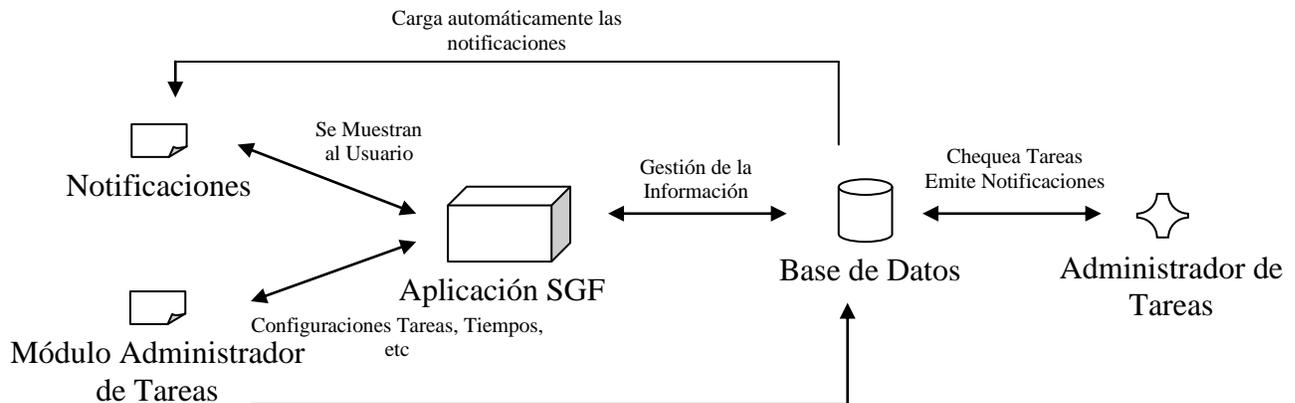


Figura 8. Integración del Administrador a la Aplicación

Tipos de Tareas:

- Notificación de avisos: Existen diferentes tipos de notificaciones dependiendo del estado en que se encuentren los casos, una vez el caso se encuentre cerrado el sistema da por cumplida la tarea de seguimiento a dicho caso. El administrador genera un aviso

(definido) al fiscal que procesa un caso a los días (definidos según la gravedad por tipos de casos) antes de su vencimiento. Esta tarea se mantiene viva por días (definido según la gravedad del tipo de caso) hasta que el estado del caso no se modifica. En caso de vencerse el período para el proceso se realizarán notificaciones a los fiscales superiores del que está ejecutando el caso.

- Envíos de Correos: Existe una tarea a administrar que enviará correos (tiempo definido) con un reporte del estado de los casos a los fiscales superiores.

#### Funcionalidades desde el Módulo en la Web:

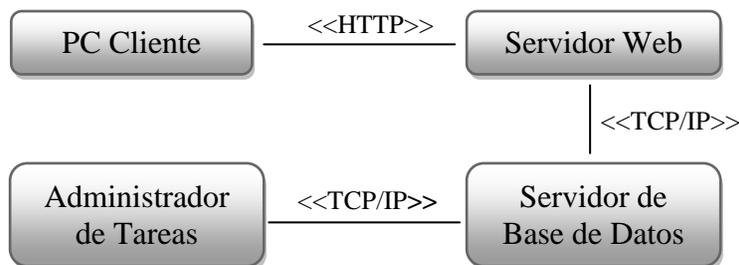
- Crear Tipo de Notificación: El usuario deberá definir los tipos de notificaciones.
- Crear Tarea: El usuario podrá definir independientemente del chequeo que realiza el sistema si desea dar un seguimiento diferente a un caso en específico.
- Mostrar Tareas: Se muestran las tareas definidas por el usuario.
- Modificar Tarea: Se permite modificar la tarea creada por el usuario mientras no haya sido ejecutada.
- Eliminar Tareas: El sistema debe permitir al usuario eliminar las tareas que ha creado mientras no hayan sido ejecutadas.
- Buscar Tareas: El sistema debe permitir buscar las tareas según un criterio de búsqueda definido por el usuario.
- Mostrar Notificaciones: El sistema muestra al usuario todas las notificaciones que le llegan mediante un ícono de avisos que se muestra desde cualquier sitio que se encuentra en el sistema.

- Eliminar Notificaciones: El sistema permite al usuario eliminar sus notificaciones. En este caso la notificación cambia su estado pero no se elimina de la base de datos, quedando una traza de las mismas como método de control.

Funcionalidades desde la Aplicación Administradora:

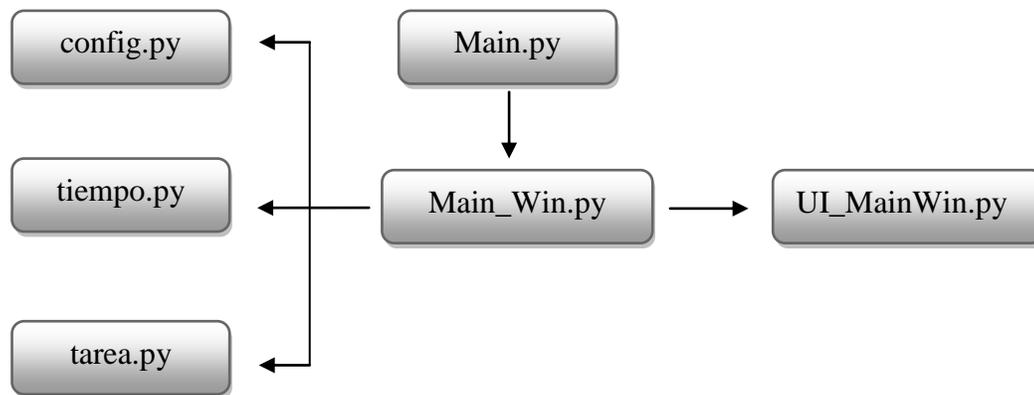
- Ejecutar Tarea: El sistema chequea las tareas y las ejecuta según sea el caso en el momento necesario.
- Actualizar estado de la tarea: Según sea el caso el sistema actualiza el estado de las tareas. (Ejecutadas, Pendientes)
- Controlar ejecución de tareas: El sistema chequea según el intervalo definido si existen tareas por ejecutar.
- Configurar sistema: Permite configurar cuenta de correo, intervalo de chequeo de tareas, servidor de base de datos.
- Iniciar servicio de chequeo y ejecución de tareas. El administrador del sistema tiene la posibilidad de iniciar el servicio de chequeo de las tareas.
- Detener servicio de chequeo y ejecución de tareas. El administrador tiene la posibilidad de detener el servicio de chequeo de las tareas.

En la solución se generó un diagrama de despliegue “Figura 4” el cual indica la situación física de los componentes lógicos desarrollados. Es decir se sitúa el software en el hardware que lo contiene.



*Figura 9. Fases de Desarrollo del Software*

Se generó además el diagrama de componentes “Figura 5” que representa cómo el sistema es dividido en componentes y muestra las dependencias entre estos. Los componentes físicos incluyen archivos, cabeceras, librerías compartidas, módulos, ejecutables, o paquetes. Pueden ser usados para modelar y documentar cualquier arquitectura de sistema. Uno de los usos principales es que puede servir para ver qué componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema.

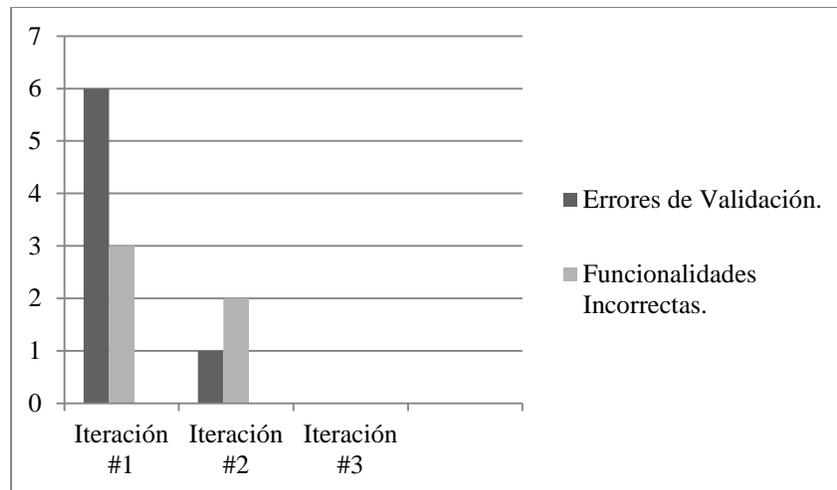


*Figura 10. Diagrama de Despliegue*

Se desarr

alidad

del sistema “Figura 6”.



*Figura 11. Resultados de las Pruebas de Caja Negra*

### **Beneficios y Aportes:**

- ✓ Conversión de tareas humanas en maquinales.
- ✓ Mejor control y seguimiento de los procesos.
- ✓ Funcionalidades para la ayuda a la toma de decisiones.
- ✓ Ahorro de recursos.
- ✓ Fácil Integración a otras aplicaciones.

### **9. Conclusiones**

A continuación se relacionan las principales conclusiones a las que se arribó:

1. Se ha dado cumplimiento al objetivo de esta investigación, pues como resultado se obtuvo una propuesta informática con altas expectativas para la gestión de los procesos en el sistema SGF.
2. La metodología propuesta resultó eficiente y quedará disponible para la utilización en sistemas similares.

3. Mediante pruebas realizadas se constató que la solución podrá ser usada para administrar procesos en otras aplicaciones.
  
4. El costo de inversión es bajo ya que las tecnologías propuestas son libres, los resultados son profesionales y medibles a corto plazo.

## **Bibliografía**

AGESIC. Sistema de Información y Gestión de Fiscalía (SIG - Fiscalía) Recolectado el (2015, enero 23) de

[http://www.agesic.gub.uy/innovaportal/v/3201/1/agesic/sistema\\_de\\_informacion\\_y\\_gestion\\_de\\_fiscalia\\_sig\\_-\\_fiscalia.html](http://www.agesic.gub.uy/innovaportal/v/3201/1/agesic/sistema_de_informacion_y_gestion_de_fiscalia_sig_-_fiscalia.html)

Bahit, E. (2011) POO y MVC en PHP. Recolectado de <http://www.etnassoft.com/biblioteca/poo-y-mvc-en-php/>

Colaboradores de Wikipedia. (2015). CPanel Recolectado el (2015, enero 23) de

<http://es.wikipedia.org/wiki/CPanel>

Colaboradores de Wikipedia. (2015). Fiscal (funcionario). Wikipedia, La enciclopedia libre

Recolectado el (2015, enero 23) de

[http://es.wikipedia.org/w/index.php?title=Fiscal\\_\(funcionario\)&oldid=80856793](http://es.wikipedia.org/w/index.php?title=Fiscal_(funcionario)&oldid=80856793)

Custom (2014) Desarrollo de integraciones: multiplataforma en QT. Recolectado de

<https://www.customsolutions.es/es/blog/2012/06/06/desarrollo-multiplataforma-en-qt>

Ecured, (2015). Python. Recolectado de <http://www.ecured.cu/index.php/Python>

Fiscalía Ministerio Público de Chile (2013) SAF SISTEMA DE APOYO A LOS FISCALES.

Recolectado el (2015, enero 23) de [www.oas.org/juridico/ppt/mesicic4\\_chl\\_sist.ppt](http://www.oas.org/juridico/ppt/mesicic4_chl_sist.ppt)

Machaca, T (2012). Implementación del Sistema de Gestión Fiscal en el Nuevo Sistema Procesal Penal y su Repercusión en los Usuarios. Recolectado el (2015, enero 23) de

[http://es.slideshare.net/tony\\_machaca?utm\\_campaign=profiletracking&utm\\_medium=sssite&utm\\_source=ssslideview](http://es.slideshare.net/tony_machaca?utm_campaign=profiletracking&utm_medium=sssite&utm_source=ssslideview)

Potencier, F y Weaver, R. Symfony 2.1, el libro oficial. Recolectado de

[http://librosweb.es/libro/symfony\\_2\\_1](http://librosweb.es/libro/symfony_2_1)

Pressman, R. S. (2005). Ingeniería del Software: un enfoque práctico. La Habana: Editorial Félix Varela, 2005

Sitio oficial en Argentina de PythonCard. (2011) “About PythonCard”. Recolectado de

<http://python.org.ar/pyar/PythonCard>

SONDA S.A (2013) Sistema de Apoyo a los Fiscales, herramienta clave de la Reforma Procesal

Penal en Chile. Recolectado el (2015, enero 23) de <http://www.sonda.com/caso/12>

The Eclipse Foundation (2015). Eclipse.org. Recolectado de <http://www.eclipse.org>

Urrutia, J (2006) Herramientas y Framework de Scheduler a nivel empresarial (Quartz O.Source y Flux) Recolectado el (2015, enero 23) de

<http://www.misbytes.com/wp/2006/09/30/herramientas-y-framework-de-scheduler-a-nivel-empresarial-quartz-osource-y-flux>