

Automatización de Apercibimientos para I.E.S Fernando Aguilar Quignon

Emilio Diego Tenorio Lopez.

1. Estudio del problema y análisis del sistema.

Con esta aplicación se pretende automatizar la creación de los apercibimientos para los alumnos de I.E.S Fernando Aquilar, el principal problema es la obtención de la información del portal de Seneca al no disponer de una API donde poder extraer los datos de faltas de asistencia.

Para afrontar este problema se tendrá que automatizar la lectura de los informes generados por Seneca en formato PDF creando un programa que extraiga la información que necesitamos para crear los apercibimientos.

1.1 Introducción.

Actualmente se dispone de una base de datos en de Microsoft Acces en la que se recoge la información que genera dichos PDF de la plataforma Seneca.

Esta información hay que pasarla a la base de datos a mano y una vez echo esto se pueden generar dos tipos de informes, uno a los alumnos para notificarles el apercibimientos y un segundo que usan los profesores en sus reuniones con las asignaturas y alumnos que superen las 2 apercibimientos.

Para hacer esto se ha desarrollado bajo una aplicación web (Django) la cual sera mas practica a la hora de implementase en el instituto.

1.2 Objetivos.

Con esta aplicación web se pretende automatizar en la medida de lo posible la creacción de apercibimientos y la generación de los informes en PDF. Estas dos ultimas funcionalidades las tenia ya el antiguo programa en Acces.

1.3 Funciones y Requisitos.

Una vez implementado el servicio proporcionara una marera cómoda de generar los apercibimientos del mes a los alumnos en pocos minutos. El programa a crear tendrá que tener como mínimo las mismas funcionalidades que el anterior.

Para hacer mas cómoda la carga de PDF se podrán comprimir todos estos en un ZIP y la aplicación los descomprimirá y analizará.

1.4 Planteamiento y Evaluación de las diversas soluciones.

La aplicación podría disponer de una interfaz la cual la limitaría al tenerla que diseñar para un Sistema Operativo concreto, puesto que en el departamento se usa tanto Windows como Linux la opción mas cómoda es la de crear una aplicación web. De tal manera que no solo se pueda acceder desde un equipo si no que la aplicación este disponible a todos aquellos que estén en la misma red.

Solución a la falta de Integridad. Como he mencionado antes al no disponer de una API se tendrá que sacar la información de los PDF, al sacarlos de un PDF carece de sentido el crear una base de datos compleja, con relaciones ya que no hay manera de asegurar una integridad referencial con la información sacada desde un PDF.

La base de datos dispondrá de una sola tabla la cual contendrá por cada línea los datos de aperebimientos de cada PDF. A posteriori se analizará que de esos registros cumplen los requisitos para formar un aperebimiento y los que cumplan dicha condición serán marcados como “Apercibimiento” en la base de datos, eliminando el resto que no lo son al carecer de utilidad.

La solución final es que no es necesaria esa integridad por lo que se usa la base de datos para almacenar los aperebimientos, de tal manera que se puedan generar los informes que necesitamos.

Solución a los errores. La aplicación a pesar de funcionar correctamente usa dos programas con los que pasar los datos del formato PDF a Texto, se ha usado dos programas diferentes para asegurarnos que la información que sea erróneamente extraída por uno, pueda ser corregida por el otro, de tal manera que se crean dos conjuntos de datos los cuales deben de contener la misma información pero de diferentes programas, cuando la inserción del primer falla se captura la excepción producida y se busca la misma información en el segundo conjunto de datos.

A pesar de esta manera de mitigar los errores cabe la posibilidad de que ninguno de los dos programas tenga la información de manera correcta, por lo que se marca el (Alumno, Curso y Materia) en estado de error, permitiendo desde una interfaz web la inserción de estos datos de manera manual. Y se bloquea la generación de aperebimiento para el curso que tenga un dato en estado de “Error”, de esta manera se intenta que no se generen informes con información errónea.

Solución a las Asignaturas con 1 hora o menos semanal. Dentro del PDF se muestra la información “Horas Lectivas” y “Total del Periodo” al dividir estos números obtenemos un número que al ser menor o igual que 0,2 podemos saber que dicha asignatura tiene menos de una hora semanal por lo que en vez de aplicarle el criterio del 25% para considerarlo un aperebimiento se le aplicará el porcentaje del 50%.

Solución a la carga masiva de PDF. Para hacer más cómoda la carga de PDF la aplicación permitirá cargar estos desde un comprimido en formato ZIP, este no debe de tener caracteres raros y no contener carpetas. Una vez subido el ZIP al servidor este es descomprimido en una carpeta temporal y cada PDF es añadido a la base de datos. Posteriormente se elimina el fichero ZIP tanto de la base de datos como de la carpeta contenedora. A continuación se lanza la clase encargada de analizar los PDF almacenados. Una vez extraída la información los PDF son eliminados.

Solución a la duplicidad de la información. Durante la inserción de las faltas de asistencia se busca la información en la base de datos, si la información se encuentra no se vuelve a insertar, si embargo al buscar no se encuentra ninguna fila con la información a buscar se produce una excepción que desemboca en la inserción de la información que se está procesando, así con cada alumno de cada PDF.

1.5 Justificación de la Solución elegida.

Los problemas y las soluciones que arriba se detallan buscan solucionar de una manera “simple” las necesidades del departamento. Siendo estas la manera mas lógica de llevarlo a cabo.

1.6 Modelado de la Solución.

1.6.1 Recursos Humanos.

Para sacar adelante el proyecto es necesario la colaboración de los profesores y del personal que actualmente realiza esta operación a mano, ya que sera el que usara esta aplicación en el futuro. De esta manera se adecuara la aplicación a las necesidades de la escuela.

1.6.2 Recursos Software.

Al tratarse de una aplicación Web sera necesario una base de datos en Mysql o Postgres y un servicio Web como Nginx con el modulo WSGI de Python instalado para hacer funcionar el proyecto en Django.

1.6.3 Recursos Hardware.

Debido al numero limitado numero de clientes que tendrá la aplicación no es necesario un servidor muy potente por lo que el servidor que ya funciona el el departamento puede soportar fácilmente la aplicación.

1.7 Planificación temporal.

Los dos primeros meses se ha estado trabajando en hacer que la clase encargada del análisis de los PDF pueda recopilar la información de una manera correcta.

Una vez conseguido esto este ultimo mes se ha estado trabajando en la aplicación web y y las funcionalidades que esta brindara a quien la va a usar.

2. Ejecución del proyecto.

El proyecto Django tiene sus archivos, los mas importantes son dos archivos encargados del análisis de las faltas de asistencia y el encargado de crear los informes, estos son.

ProcesarPDF.py este fichero contiene la clase encargada de procesar los PDF y como ultimo paso analiza si los datos estriados cumple los requisitos para crear un apercibimiento.

GenInforme.py este fichero contiene dos clases encargadas de la creación de los apercibimientos para los alumnos y el informe para la sesion de evaluación de los profesores.

3. Fase de Pruebas.

Para buscar errores en el procesamiento de los PDF se ha cargado la aplicación con los pdf de años anteriores y se han cotejado la información de los informes de manera manual para colaborar de que los apercibimientos y los informes para la sesión de evaluación son correctos.

El viernes día 20 de abril se hablara con el director para verificar unos apercibimientos que la aplicación detecta pero que no se ha reflejado en los informes de otros años todo para saber si el comportamiento actual es el correcto.

Durante la fase de pruebas se detecto un problema al contar los apercibimientos y otro a la hora de analizar los pdf, estos problemas quedaron solucionados.