

Módulo de seguimiento para el corrector web de prácticas de Compiladors

José María Carbonell López

Resumem— El módulo de seguimiento para el corrector web de prácticas de Compiladors es una extensión para la actual aplicación que permite corregir de forma automática las prácticas de los alumnos de la asignatura de Compiladors, desarrollada por Christian García Miñano [1] y Marc Morales Castany [2].

Sin embargo, este proyecto nace de la necesidad por parte del profesor Javier Sánchez Pujadas (profesor de la asignatura de Compiladors) de recopilar información de las entregas de prácticas realizadas por parte de los alumnos. Lo que se ha desarrollado es un módulo que permite recolectar los datos de varios archivos de texto pre-formateado, para almacenar toda la información en la base de datos y así luego poder generar informes. Así logramos, a partir de la información generada, generar el conocimiento requerido, para mejorar aspectos de la docencia en los que los alumnos muestren más carencias.

Palabras clave— módulo estadístico, estadística, compiladors, model-view-controller, Spring framework, extreme programming informes.

Abstract— The tracking module for the automatic correction of practices of Compiler subject is actually an extension for the actual application that allows to automatically correct practices that are uploaded from students, developed by Christian García Miñano [1] and Marc Morales Castany [2].

However, this project is born because of the need of teacher Javier Sánchez Pujadas (teacher of the subject of Compiladors) to collect information on deliveries made by students. What has been developed is a module that allows to collect the data of several pre-formatted text files, to store all the information in the database and this way be able to generate reports. Thus, based on the generated information, we could generate the knowledge required to improve aspects of teaching in that students show more deficiencies.

Index Terms— Statistical module, statistics, compilers, model-view-controller, Spring framework, extreme programming, reporting.



1 INTRODUCCIÓN

ACTUALMENTE, vivimos en un mundo donde las tecnologías han evolucionado hasta el punto de que muchas de las cosas que antes se hacían de forma manual, ahora ya están automatizadas. Esto se ha aplicado en muchos campos, incluido también en la enseñanza.

Es por ese motivo, que entre dos alumnos desarrollaron una aplicación que permitiese corregir las prácticas de la asignatura de Compiladors de forma automática. A pesar de ser útil, se encontró la carencia de que la aplicación no guardaba ningún tipo dato de la entrega realizada por el alumno, para que luego nos sirviese para hacer un seguimiento sobre la docencia.

Este proyecto, trata concretamente en realizar un módulo o extensión a la aplicación que ya existía anteriormente, con el fin de poder guardar, recopilar y tratar la información para guardarla en la base de datos. Posteriormente esta información nos va a permitir generar conocimiento en forma de informes en formato HTML, que sirva al profesor para analizar en qué aspectos de la docencia, se deberían realizar acciones de refuerzo.

En este documento se van a encontrar las siguientes secciones:

- **Introducción:** sección introductoria donde se expone la identificación de la oportunidad, la justificación de la necesidad y se hace un breve resumen de en qué van a consistir las demás secciones.
- **Objetivos:** se especifican detalladamente cuales son los objetivos globales del proyecto, así como su nivel de prioridad.
- **Estado del arte:** se exponen qué herramientas de mercado actualmente existen relacionadas con nuestro proyecto, así como el motivo que nos diferencia de las demás.
- **Metodología:** se describe qué metodología o metodologías se han utilizado para llevar a cabo este proyecto y la justificación de su elección.
- **Planificación:** en esta sección se detalla toda la planificación del proyecto, las tareas a realizar en cada iteración, dividido por semanas. También se describen qué problemas han ocurrido durante la

ejecución del proyecto y a qué grado de completud se ha llegado en los objetivos.

- **Desarrollo y herramientas:** describe como ha sido el desarrollo global del proyecto, así como las diferentes etapas de la ingeniería del software que se han utilizado. También se hace mención a distintas herramientas que se han utilizado para llevarlo a cabo.
- **Resultados:** se exponen cuales han sido los resultados del proyecto y cual es el grado de satisfacción.
- **Conclusiones:** se describen cuales han sido las conclusiones sacadas al realizar este proyecto.
- **Agradecimientos:** una pequeñas sección de agradecimientos a la gente que me ha apoyado y ha formado parte de este proyecto.
- **Bibliografía:** se lista todo el material y documentación utilizado para la realización de este proyecto.

2 OBJETIVOS

En este proyecto se ha optado por clasificar los objetivos en función de un orden de prioridad. De esta forma conseguimos centrarnos en objetivos que son más críticos y poder desarrollar un módulo en condiciones. Estas prioridades son las siguientes: crítico, prioritario y secundario.

- **Crítico:** es el nivel máximo de importancia, con el indicamos cual es el o los objetivos principales. Los objetivos clasificados como críticos son los que afectan directamente al cumplimiento de la justificación de la necesidad que ha originado el proyecto.
- **Prioritario:** es un nivel de importancia alto. A pesar de no ser crítico, un objetivo marcado como prioritario, generará insatisfacción en el caso de no verse cumplido.
- **Secundario:** es el nivel más bajo de importancia, con el indicamos los objetivos que cubren necesidades no tan básicas de cara a satisfacer la motivación que origino el proyecto. Los objetivos marcados como secundarios, generaran satisfacción en caso de cumplirse, pero en caso de no estar no generaran insatisfacción.

A continuación se van a especificar los objetivos marcados para el proyecto y su grado de prioridad. Están ordenados mayoritariamente en el orden en el que van a ser ejecutados.

-
- E-mail de contacte: josemaria.carbonell@e-campus.uab.cat
 - Menció realitzada: Enginyeria del Software.
 - Treball tutoritzat per: Javier Sánchez Pujadas (DCC)
 - Curs 2016/17

Nº	Objetivo	Prioridad
1	Asegurarnos del correcto funcionamiento de la aplicación actual antes del 10 de Octubre	Crítico
2	Extracción, clasificación y guardado de datos para uso estadístico	Crítico
3	Generación de informes en formato HTML	Crítico
3.1	Porcentaje de alumnos que superan cada una de las pruebas de una práctica	Crítico
3.2	Media aritmética de correcciones necesarias para superar una prueba	Crítico
3.3	Medidor uso de la aplicación entre dos fechas concretas (cuantas veces ha hecho una corrección)	Prioritario
3.4	Posibilidad de consultar estadísticas sobre un alumno concreto	Secundario
4	Elaborar y ejecutar con éxito tests de integridad de datos	Prioritario
5	Generación de informes en formato CSV	Secundario
6	Asociar material a consultar por práctica / prueba	Secundario
7	Posibilidad de clasificar las pruebas por palabra clave	Secundario

Tabla 1 - Objetivos del proyecto y su prioridad

Debido a la naturaleza del objetivo número 3, los objetivos han podido variar ligeramente. Esto es debido a que con la información recopilada por el módulo, podíamos generar información variada y de muchas formas. Finalmente se han optado por fijar esos sub-objetivos, porque se han considerado los más oportunos.

2.1 Detalles de los objetivos

A continuación se van a detallar los objetivos.

Objetivo 1: Debido a que el proyecto está realizado por un tercero con una tecnología en la cual el implementador no ha tenido experiencia, se consideró crítico el hecho de conseguir hacer funcionar el portal

antes del 10 de octubre. De esta forma, nos aseguramos de mantener un buen ritmo en el proyecto y acostumbrarnos a la tecnología.

Objetivo 2: Este objetivo es la base del proyecto y quizás el más importante. Se ha requerido hacer un análisis completo de que datos son necesarios extraer, para luego clasificar y guardar en la base de datos.

Objetivo 3: El grado de impacto de este objetivo es variable (crítico, prioritario, o secundario) dependiendo del tipo de estadística que queramos reportar. Esto es debido a que hemos visto que podíamos definir muchos tipos de información estadística diferente, pero que obviamente toda no podemos abarcar. Por eso el contenido de este objetivo ha tenido una naturaleza más flexible y ha variado a lo largo de la realización del proyecto.

Objetivo 4: Se ha considerado de importancia alta, poder realizar comprobaciones de la integridad de los datos almacenados. De esta forma nos hemos asegurado que los datos y las operaciones realizadas con ellos, son correctas. Por ello se ha realizado el diseño y la ejecución de una serie de tests para cumplir este objetivo.

Objetivo 5: Este objetivo consiste principalmente en poder exportar los datos generados por el informe HTML, en formato excel, o más concretamente CSV.

Objetivo 6: Este objetivo sugiere poder asociar o adjuntar documentos (PDF's, Word, Imágenes...), a cada una de las prácticas o pruebas disponibles a corregir. De esta forma el profesor puede adjuntar material docente a las prácticas o pruebas, para reforzar su actividad docente.

Objetivo 7: Posibilidad de poder clasificar las pruebas por palabra clave. De esta forma podemos agrupar pruebas por categorías y usarlo como una variable de entrada más para realizar los informes estadísticos.

3 ESTADO DEL ARTE

Uno de los pilares de nuestro proyecto, es el de recolectar información. Actualmente existen algunas aplicaciones en el mercado que permiten realizar esta recolección de datos de archivos de texto.

Tanto Template-Parser [26] como Altova Text / Flat File Conversion Tool [27], son herramientas de escritorio, ambas de pago, que nos permite la extracción de datos de diversos formatos como: TXT, CSV, XML, XLS e incluso de base de datos access. Estas aplicaciones permiten formatear estos datos de entrada y darle una salida. Por ejemplo podríamos extraer la información de un archivo de texto y darle salida en otro formato diferente. Esto puede resultar útil en algunos casos. Sin embargo, en

nuestro caso, ya disponemos del archivo de texto pre-formateado por defecto y lo que necesitamos es analizar y extraer la información para insertarlo en la base de datos.

El otro pilar fundamental de nuestro proyecto, es el de poder generar informes con los datos almacenados en la base de datos. También existen herramientas en el mercado que realizan esta tarea.

La herramienta Moddle mismamente, que es una herramienta educativa, tiene en su repositorio de extensiones varios módulos como Configurable Reports [28], que permiten a los administradores generar informes sobre entregas de los alumnos, entre otras cosas. Encontramos que tiene similitud con nuestro proyecto, puesto que también es una extensión sobre una aplicación base. Evidentemente en este proyecto, no tiene utilidad, pues la aplicación correctora de prácticas es a medida y necesita que su módulo de seguimiento también lo sea.

Existen otras herramientas independientes, basadas en Business Intelligence como Pentaho [29], que permiten hacer informes sobre datos almacenados. Pero estas herramientas, son demasiado potentes para lo que buscamos realizar y probablemente su potencial quede desaprovechado.

También existen herramientas OreReports [30] en java o Reportico [31], que permiten la generación de informes. Pero no tienen la posibilidad de extraer los datos de archivos de texto.

En resumen, la aplicación correctora de prácticas cubre unos aspectos muy específicos que en otras plataformas son más difíciles de cubrir. Por lo tanto el módulo de seguimiento, también necesita desarrollarse a medida, para que se pueda desmarcar de otras herramientas existentes.

4 METODOLOGÍA

La metodología que se ha utilizado en gran parte de este proyecto es una metodología ágil para el desarrollo de proyectos de software, llamada "extreme programming", más conocida como XP [4].

La razón principal por la cual nos hemos decantado por esta metodología, es principalmente por la naturaleza del proyecto, más concretamente del objetivo de generar informes. Se intuyó, por las primeras entrevistas, que el cliente podría requerir cambios en los requisitos de la aplicación. Precisamente por eso, esta metodología de desarrollo se adaptó perfectamente a nuestras necesidades en primeras instancias. Es por ello que se estableció un modelo iterativo e incremental de dos semanas, en el que a cada periodo le llamamos iteración.

En cada iteración se definieron unos requisitos por parte del cliente y el grupo de trabajo definió las tareas a realizar con tal de cumplirlos.. De esta forma también con esta metodología, se consiguió que el cliente se viese involucrado de forma activa en el proyecto y viese los progresos en periodos muchos más cortos de tiempo. Por otro lado otra de las características del XP es la simplicidad en el diseño. Partiendo de la base que es un proyecto en evolución (la aplicación actual proviene de dos TFG's), es recomendable mantener una simplicidad en el diseño, con tal de mejorar el mantenimiento del código y facilitar las posibles ampliaciones que vengan en el futuro.

No obstante, durante la segunda mitad del desarrollo del proyecto, y sobretodo, una vez que ya se definieron muy bien los requisitos con el cliente, se optó por tomar una metodología un poco más tradicional. Se siguió usando la metodología XP, pero mezclada con la de Cascada [24]. Esto lo hicimos para realizar las fases de implementación y testeo principalmente, donde el cliente ya no se veía tan involucrado. De esta forma optimizamos mucho más el tiempo de desarrollo y ha ido bastante bien.

5 PLANIFICACIÓN

En los inicios del proyecto, se estableció una planificación sobre los objetivos y requisitos críticos/prioritarios que se han ido cumpliendo hasta el día de hoy. Por eso motivo, se realizó la planificación sin tener en cuenta los objetivos secundarios, queríamos centrarnos exclusivamente en los objetivos importantes y si daba tiempo, entonces sí, desarrollar los secundarios.

5.1 Iteraciones y WBS

La planificación del proyecto se ha desglosado en varias iteraciones, de las cuales hemos obtenido ciertas tareas y un resultado cada dos semanas aproximadamente. Con esta planificación también tenemos el WBS necesario para desarrollar el proyecto.

Iteración 1 (16/09/2016 - 29/09/2016)

- Reunión inicial con el cliente.
- Leer documentación del proyecto del TFG anterior.
- Descargar e instalar todo el software necesario para el desarrollo del proyecto.
- Leer el manual de instalación e importar el proyecto al servidor web local.
- Investigación y formación sobre funcionamiento del Framework Spring.
- Investigación y formación sobre funcionamiento del servidor web y mysql.

- Repaso manual de Java.
- Elaboración y entrega del informe inicial.
- Output iteración: informe inicial y planificación

Iteración 2 (30/09/2016 - 13/10/2016)

- Corregir errores conexión a la base de datos y hacer volcado de los datos SQL.
- Hacer uso exhaustivo de la aplicación para comprender su funcionamiento.
- Documentar fallos encontrados.
- Output iteración: aplicación funcionando (objetivo 1)

Iteración 3 (14/10/2016 - 27/10/2016)

- Corregir posibles fallos en la base de datos actual.
- Establecer qué datos deben almacenarse.
- Diseñar mediante diagrama Entidad-Relación la nueva propuesta de base de datos e implementar las tablas.
- Documentar diseños y modificaciones sobre la base de datos.
- Output iteración: base de datos implementada

Iteración 4 (28/10/2016 - 10/11/2016)

- Inspeccionar el archivo de texto resultante de una corrección.
- Desarrollar un parser para la extracción de la información
- Volcado del parser a la base de datos.
 - Requisitos: determinar que información es valiosa del archivo.
 - Diseñar el parser.
 - Implementación.
 - Testeo intensivo (el parser es el pilar en el que se sujeta todo el proyecto).
- Documentación de funcionamiento del parser.
- Elaboración y entrega del informe de progreso I (4 de noviembre).
- Output iteración: parser implementado (objetivo 2)

Iteración 5 (11/11/2016 - 24/11/2016)

- Diseño plantilla HTML.
- Maquetación plantilla HTML.
- Construcción de consultas SQL e integración

- del HTML con los datos
- dinámicos.
- Testeo output (datos estadísticos) correcto (objetivo 4).
- Output iteración: generar informe HTML (objetivo 3.1)

Iteración 6 (25/11/2016 - 08/12/2016)

- Diseño plantilla HTML.
- Maquetación plantilla HTML.
- Construcción de consultas SQL e integración del HTML con los datos
- dinámicos.
- Testeo output (datos estadísticos) correcto. (objetivo 4)
- Diseñar e implementar sistema para generar el informe también en CSV
- (objetivo 5)
- Output iteración: generar informe HTML y/o CSV (objetivo 3.2).

Iteración 7 (09/12/2016 - 22/12/2016)

- Diseño plantilla HTML.
- Maquetación plantilla HTML.
- Construcción de consultas SQL e integración del HTML con los datos
- dinámicos.
- Testeo output (datos estadísticos) correcto (objetivo 4).
- Diseñar e implementar sistema para adjuntar documentación a las pruebas de corrección (objetivo 6).
- Elaboración y entrega del informe de progreso II (16 de diciembre).
- Output iteración: generar informe HTML (objetivo 3.3)

Iteración 8 (23/12/2016 - 12/01/2017)

- Diseño plantilla HTML.
- Maquetación plantilla HTML.
- Construcción de consultas SQL e integración del HTML con los datos
- dinámicos.
- Diseñar e implementar sistema para poder clasificar las pruebas (objetivo 7).
- Testeo output (datos estadísticos) correcto

- (objetivo 4).
- Output iteración: generar informe HTML (objetivo 3.4).

Iteración 9 (13/01/2017 - 26/01/2017)

- Elaboración propuesta del informe final (20 de Enero 2017).

Iteración 10 (27/01/2017 - 13/02/2017)

- Elaboración propuesta de la presentación (3 de Febrero 2017).
- Entrega informe final. (7 de Febrero 2017)
- Elaboración y entrega dossier TFG.
- Defensa tribunal.

5.2 Problemas encontrados

Como ya se ha especificado en reiteradas ocasiones, este proyecto es una ampliación de otro proyecto anterior. Es por eso que la puesta en marcha, era algo imprescindible y que debía ser considerado con especial atención. Por ese motivo se ha realizado una examinación intensiva de la estructura de la aplicación y del código. Con el objetivo también de entender mejor el funcionamiento de SpringFramework.

Es importante remarcar, que el cumplimiento del objetivo número 1, se retrasó un poco debido a que la aplicación presentaba muchísimos problemas por incompatibilidades en las dependencias del proyecto de Eclipse. Se tuvo que realizar un análisis intensivo sobre compatibilidades de los diferentes JAR (tanto de SpringFramework como de otros que habían), la versión de eclipse que se usa para el desarrollo y el proyecto en cuestión.

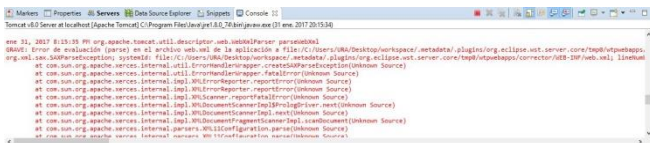


Figura 1 - Problemas de entorno

Diseñar, implementar y testear el parser, fue sin duda la tarea clave de este proyecto. Es por ello, que nos llevó un poco más tiempo realizarla. Aún así, es importante remarcar que una vez terminado, se tuvieron que hacer algunos cambios.

Esto se ha debido a que se ha desarrollado de forma paralela, un proyecto para generar ampliaciones de prácticas y por consiguiente generar también un archivo

de texto resultante diferente al inicial. Por ese motivo, nos tuvimos que reunir con el cliente, de forma urgente. Para aclarar y concretar cuales habían sido los cambios en el archivo de texto resultante y de esta forma llevar a cabo los cambios necesarios en el parser para que funcionase correctamente.

Estos problemas se consiguieron resolver de forma satisfactoria, aunque acarrió que los objetivos secundarios no se pudiesen realizar por falta de tiempo.

Aún así, la mayor dificultad de este proyecto, ha sido sin duda el hecho de tener que entender la arquitectura y código realizado por otro desarrollador. A pesar de estar relativamente bien estructurado, nos hemos encontrado con el problema de no poder contactar con la persona que realizó la aplicación principal y eso siempre complica el desarrollo en algún momento de atasco.

5.3 Grado de realización del proyecto

En la siguiente tabla se van a especificar los objetivos, así como el grado que han sido realizados. Los objetivos secundarios no se han podido desarrollar, principalmente porque el tiempo que tenemos es limitado y hemos preferido centrarnos en lo importante.

Nº	Objetivo	Completado
1	Asegurarnos del correcto funcionamiento de la aplicación actual antes del 10 de Octubre	100%
2	Extracción, clasificación y guardado de datos para uso estadístico	100%
3	Generación de informes en formato HTML	100%
3.1	Porcentaje de alumnos que superan cada una de las pruebas de una práctica	100%
3.2	Media aritmética de correcciones necesarias para superar una prueba	100%
3.3	Medidor uso de la aplicación entre dos fechas concretas (cuantas veces ha hecho una corrección)	100%
3.4	Posibilidad de consultar estadísticas sobre un alumno concreto	0%
4	Elaborar y ejecutar con éxito tests de integridad de datos	70%
5	Generación de informes en formato CSV	0%

6	Asociar material a consultar por práctica / prueba	0%
7	Posibilidad de clasificar las pruebas por palabra clave Secundario	50%

En el objetivo 4, no lo marcamos como completo porque consideramos que se podrían haber realizado más pruebas. Sin embargo, teniendo en cuenta que la fase de test requiere un tiempo bastante elevado y debido a la ajustada fecha de entrega, se han diseñado e implementado los test que hemos considerado más necesarios.

En cuanto a los objetivos 3.1, 3.2, y 3.3, consideramos que hemos extendido los datos generados por los informes.

6 DESARROLLO Y HERRAMIENTAS

Se han seguido todas las fases que componen la ingeniería del software. Al usar un método de desarrollo ágil en primera instancia, no siempre han sido en orden. De hecho la fase de testeo se ha ido efectuando a medida que se iba avanzando en la implementación del código.

6.1 Captura de requisitos

Esta fase tuvo lugar en las primeras semanas del proyecto y consistió principalmente en reuniones con el cliente, con tal de comprender el funcionamiento de la aplicación correctora de prácticas. De ese modo, en entrevistas posteriores se establecieron los requisitos, con el objetivo de extraer todos los objetivos del proyecto.

Las herramientas utilizadas para recolectar requisitos, fueron las entrevistas y las reuniones con el cliente.

Para documentarlos se ha hecho uso de una herramienta llamada ReqView [22], ya que nos permite generar una trazabilidad. Es una herramienta similar al Rational Doors, pero mucho más sencilla.

Contents	Description
1 Recolecta, tratamiento y exposición de datos	1 Recolecta de datos
1.1 Números redondeados a dos decimales	1.1 Números decimales
1.2 Diseño del parser en función del TXT	1.2 Diseño
1.2.1 Datos a recoger:	
1.2.2 Tiene que ser ágil	
1.2.3 Guardado en única tabla	

Figura 2 - Requisitos en ReqView

6.2 Análisi y diseño

El principal trabajo realizado en la parte de diseño del módulo ha sido principalmente el tener que rediseñar algunos aspectos de la base de datos que ya existía en la aplicación.

Para ello se ha realizado un diagrama de tipo E-R

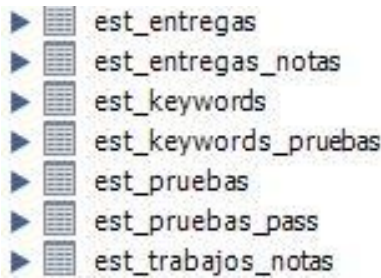


Figura 3 - Nuevas tablas SQL

(Entidad Relación), que posteriormente ha sido convertido a tablas SQL, o se han modificado campos de tablas ya existentes.

Por otro lado también se han creado diagramas UML de clases, para ordenar y relacionar las clases nuevas que incorporaría el módulo.

Se han utilizado herramientas online para la creación y documentación de los diferentes diagramas mencionados [17].

6.3 Implementación

Para poder llevar a cabo el desarrollo del proyecto, se han instalado todas las herramientas necesarias en una máquina local. De esta forma agilizamos cualquier cambio y podemos encontrar los problemas con mayor rapidez. Por contra, nos pudimos encontrar que cuando se tuviese que hacer la puesta en marcha de la aplicación, el servidor del cliente donde va a estar alojado presentase algún problema de incompatibilidad con la aplicación.

Es por este motivo, que se solicitó acceso a dicho servidor, para poder realizar pruebas durante el desarrollo del proyecto.

En cuanto a la implementación en sí, se ha usado la misma tecnología, es decir, Spring framework de java y se ha intentado respetar al máximo la arquitectura de directorios existente.



Figura 4 - Arquitectura directorios

Para la realización de gráficas, se ha hecho uso de la API de Google Chart [32].

6.4 Test

Como bien es sabido, la etapa de test, es una de las más importantes y en las que más esfuerzo hay que poner. Es por eso, que a lo largo del proyecto, se ha ido realizando el test paulatinamente mientras se

desarrollaba el módulo. Se han diseñado y ejecutado 3 tipos de tests diferentes.

6.4.1 Test unitario

A medida que se han ido desarrollando clases, también se han ido diseñando e implementando los tests unitarios que corroborarían que la clase funciona correctamente. La tecnología que se ha usado para realizar estos tests ha sido JUnit [12].

6.4.2 Test de caja negra en el parser

El objetivo principal de este test, es el de asegurar que el resultado que devuelve el parser (escaner del archivo de texto resultante) es el esperado. Para ello se ha diseñado un script, que simula las entregas de los alumnos a través de la aplicación del corrector de prácticas, y comprueba si el resultado es el que debería. Este test, nos ha resultado de gran utilidad, ya que en alguna ocasión ha detectado algún error que a simple vista parecía indetectable y posteriormente podría acarrear serios problemas estadísticos.

6.4.3 Pruebas de rendimiento

La naturaleza de la aplicación, sugiere al alumno realizar entrega tras entrega para auto-avaluarse y así poder sacar buena nota en las prácticas. Debido a eso, cada vez que un alumno realiza una entrega genera datos en la base de datos, por lo que hay que tener un control a la hora de tener muchos registros en las tablas. A priori, podía parecer no ser un problema, pero nos encontramos con problemas de lentitud que tuvimos que analizar.

Para encontrar estos pequeños cuellos de botella, hicimos uso de una herramienta llamada Hibernate JConsole [25], que no es más que un módulo de hibernate que nos permite hacer un seguimiento de las consultas SQL, que ocasionan más lentitud en el sistema. También nos permite monitorizar parámetros como tiempo de ejecución, gráficas, entre otras cosas.

7 RESULTADOS

Los resultados que se han logrado a lo largo del desarrollo de este proyecto, son bastante satisfactorios. Se ha logrado:

- Adaptarnos y conocer a una nueva tecnología que desconocíamos hasta el inicio del proyecto. Gracias a esto hemos logrado instalar y hacer funcionar la aplicación existente en un tiempo más que aceptable.
- Crear un parser, o escaneador de archivo de texto, para poder extraer la información de una forma cómoda para su tratamiento posterior.
- Mostrar la información filtrada y tratada en forma de estadística, con el fin que el profesor sepa interpretar los resultados y así realizar las acciones docentes oportunas.

- Realizar el módulo estadístico con un rendimiento en cuanto a velocidad, que no ocasiona ningún problema en el sistema.
- Separar en la mayor parte y con comentarios, el código realizado para el módulo estadístico del código existente. Esto puede permitir realizar ampliaciones en el futuro por posibles desarrolladores.

Estadísticas de Trabajo 1

Entre 01-02-2017 y 28-02-2017 para el tipo de nota SINTACTICO



Figura 5 - Parte del informe html

7.1 Integración con el corrector

Actualmente, la extracción de datos, se realiza sobre archivos de texto generados manualmente y no sobre la entrega realizada por los alumnos. Esto es debido a que el archivo de texto a analizar es el resultado de un programa desarrollado en Cosel, que se tiene que subir a la aplicación. Actualmente no se ha dispuesto de ese programa, pues se está desarrollando paralelamente un generador de prácticas aleatorio por parte de otro alumno.

8 CONCLUSIONES

Como conclusiones, podríamos comentar que a pesar de que este proyecto pudiera parecer trivial en un principio, se ha complicado debido al hecho de no poder elegir la tecnología de desarrollo y al venir impuesto como requisito restrictivo. Personalmente tengo experiencia en otros lenguajes de programación orientados a la realización de aplicaciones web, sin embargo nunca había trabajado con Java para hacer web ni mucho menos con Spring Framework.

También es un punto en contra, el hecho de tener que coger un proyecto ya realizado anteriormente y desarrollar ampliaciones. Siempre es preferible, tener todo el código picado desde cero por uno mismo, o por el equipo de trabajo, porque se siguen los mismos estilos de programación y hay cierta homogeneidad en el código. Aún así cabe destacar, que la aplicación existente estaba dotada en gran parte de comentarios, que facilita la comprensión del código.

De todos modos, siempre es positivo salir de la zona de confort para conocer otras tecnologías, lenguajes o arquitecturas del software, o incluso realizar extensiones del código de otros, con el fin de profundizar en el

aprendizaje como profesional del sector.

Finalmente deseamos, que la aplicación tenga éxito, haga su función y que ayude al profesor ni que sea mínimamente en realizar su tarea docente.

AGRADECIMIENTOS

Me gustaría agradecer a todas esas personas que han participado de una forma u otra en este proyecto. En primer lugar, quiero agradecer a mi tutor del proyecto Javier Sánchez Pujadas, por darme la posibilidad de participar en el desarrollo de este módulo de seguimiento y también por la paciencia que ha tenido a la hora de explicar el funcionamiento de la aplicación, así como la definición de requisitos. También quiero agradecer a al tècnico de sistemas Jorge Ramírez Espejo, que es el tècnico que se encarga del mantenimiento del servidor donde va a ser alojado este proyecto.

Finalmente me gustaría agradecer, a todas esas personas, que en lo personal siempre me han apoyado en las situaciones complicadas. Hay momentos en los que las cosas no salen, cuestan o se tuercen. Es agradable tener gente a tu lado cuando eso ocurre.

BIBLIOGRAFÍA

- [1] «Estudio, diseño e implementación de una web de corrección automática de prácticas y trabajos la asignatura Compiladores» [Online]. Available: https://ddd.uab.cat/pub/tfg/2015/tfg_27911/Articulo.pdf [Accessed: 19-september-2016]
- [2] «Test i prova pilot d'una web de correcció automàtica de pràctiques de Compiladors » [Online]. Available: https://ddd.uab.cat/pub/tfg/2016/tfg_47509/Article.pdf [Accessed: 20-september-2016]
- [3] «Metodologías ágiles de gestión de proyectos» [Online]. Available: <https://www.marblestation.com/?p=661> [Accessed: 22-september-2016]
- [4] «Extreme programming» [Online]. Available: https://en.wikipedia.org/wiki/Extreme_programming [Accessed: 23-September-2016]
- [5] «Manual d'instal·lació TFG corrector pràctiques compiladors» [Archivo PDF]
- [6] «Article TFG corrector pràctiques compiladors» [Archivo PDF]
- [7] Wrike Planner [Online]. Available: <https://www.wrike.com> [Accessed: 23-February-2016]
- [8] «Download MySQL Community Server,» [En línea]. Available: <http://dev.mysql.com/downloads/mysql/>. [Accessed: 05 Octubre 2016].
- [9] «Java SE Development Kit 8 Downloads,» [En línea]. Available: <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>. [Accessed: 05 Octubre 2016].
- [10] «Download MySQL Community Server,» [En línea]. Available: <http://dev.mysql.com/downloads/mysql/>. [Accessed: 05 Octubre 2016]
- [11] «Tomcat 8 Downloads,» [En línea]. Available: <http://tomcat.apache.org/download-80.cgi>. [Accessed: 05 Octubre 2016].
- [12] «JUnit,» [En línea]. Available: <http://junit.org/>. [Accessed: 02 Noviembre 2016].
- [13] «MySQL Workbench,» [En línea]. Available: <http://dev.mysql.com/downloads/workbench/>. [Accessed: 05 Octubre 2016].
- [14] «Spring Tutorial,» [En línea]. Available:

- <http://www.tutorialspoint.com/spring/>. [Accessed: 20 Octubre 2016].
- [15] «Casos de prueba: JUnit,» [En línea]. Available: <http://www.jtech.ua.es/j2ee/publico/lja-2012-13/sesion04-apuntes.html/>. [Accessed: 26 Noviembre 2016].
- [16] «Java Practices -> Parse text,» [En línea]. Available: <http://www.javapractices.com/topic/TopicAction.do?Id=87>. [Accessed: 02 Noviembre 2016].
- [17] «Flowchart Maker & Online Diagram Software,» [En línea]. Available: <https://www.draw.io/>. [Accessed: 10 Diciembre 2016].
- [18] «HTML Tutorial,» [En línea]. Available: <http://www.w3schools.com/html/>. [Accessed: 01 Diciembre 2016].
- [19] «Reporting in Java using Dynamic Reports» [En línea]. Available: <https://www.mkkyong.com/java/reporting-in-java-using-dynamicreports-and-jasperreports/>. [Accessed: 16 Noviembre 2016].
- [20] «How to activate Hibernate Statistics to analyze performance issues» [En línea]. Available: <http://www.thoughts-on-java.org/how-to-activate-hibernate-statistics-to-analyze-performance-issues/>. [Accessed: 28 Noviembre 2016].
- [21] «Java Scanner Class - java.util.Scanner» [En línea]. Available: <http://www.journaldev.com/872/java-scanner-class-java-util-scanner>. [Accessed: 3 Noviembre 2016].
- [22] «Get ReqView Free» [En línea]. Available: <https://www.reqview.com/download.html>. [Accessed: 25 Octubre 2016].
- [23] «Hibernate» [En línea]. Available: <http://hibernate.org/>. [Accessed: 12 Octubre 2016].
- [24] «Waterfall Model» [En línea]. Available: https://en.wikipedia.org/wiki/Waterfall_model. [Accessed: 10 Diciembre 2016].
- [25] «Hibernate Jconsole» [En línea]. Available: <http://hibernate-jcons.sourceforge.net/>. [Accessed: 10 Enero 2016].
- [26] «Template Parser» [En línea]. Available: <http://www.template-parser.com/>. [Accessed: 10 Octubre 2016].
- [27] «Altova Text / Flat File Conversion Tool» [En línea]. Available: <https://www.altova.com/download-text-conversion.html/>. [Accessed: 10 Octubre 2016].
- [28] «Moodle blocks: Configurable Reports» [En línea]. Available: https://moodle.org/plugins/block_configurable_reports/. [Accessed: 12 Octubre 2016].
- [29] «Pentaho» [En línea]. Available: <http://www.pentaho.com/>. [Accessed: 13 Octubre 2016].
- [30] «OreReports» [En línea]. Available: <http://oreports.com/>. [Accessed: 13 Octubre 2016].
- [31] «Reportico PHP Open Source Report Designer» [En línea]. Available: <http://www.reportico.org/site/index.php>. [Accessed: 13 Octubre 2016].
- [32] «Google Charts» [En línea]. Available: <https://developers.google.com/chart/>. [Accessed: 12 Noviembre 2016].