

Ministerio del Poder Popular para la Educación Superior
Universidad Simón Bolívar
Departamento de Computación
Laboratorio de Ingeniería de Software



Sprint 2

Jose Palma(13-11044)
Rafael Blanco(13-10156)
Arnaldo Quintero (13-11150)
Luigi Di Martino (10-10200)
Carlos Vazquez (12-10958)
Jhon Torres(12-11135)
Emilio Blanco (11-10106)
Adolfo Jeritson (12-10523)

INTRODUCCIÓN

En este informe se va a desarrollar los aspectos relacionados al segundo Sprint del proyecto de Ingeniería de Software I, en el cual se busca realizar un proyecto de apoyo a la Biblioteca de la Universidad a través del sistema SIGPAE. Utilizando las técnicas de programación ágil vistas en clase, se irá desarrollando este proyecto en diferentes Sprints a lo largo del trimestre. Para ello nos vamos a apoyar en diferentes herramientas que nos facilitarán el trabajo. Entre ellas, tenemos a Eclipse, PyDec, PyUnit, Django y GitHub.

Mediante la implementación del TDD y la planificación del trabajo mediante historias, resulta más factible la realización de las actividades en conjunto, de forma organizada y sistemática. Dándonos así una herramienta para elevar la tasa de éxitos en proyectos como el que se desarrolla a continuación.

A continuación se presentarán diferentes puntos que explican cómo fue nuestro trabajo en este segundo Sprint, entre estas cosas se encuentran las historias de usuario que desarrollamos, las horas que trabajamos los diferentes miembros del equipo y cómo fue nuestra experiencia usando TDD y otras herramientas como GitHub.

DESARROLLO

HISTORIAS DE USUARIO

6. Convertir PDF a HTML.
7. Extraer texto o HTML de PDF de imágenes.
8. Mostrar todos los campos estándares asociados a programas analíticos.
9. Consultar programas analíticos almacenados en SIGPAE.
10. Salvar estado de transcripción.
11. Seleccionar instancia responsable del programa.

Cabe destacar que del sprint pasado quedaba por corregir y se corrigieron los siguientes aspectos:

- Vista previa del PDF.
- Lectura de comas y espacios.

MODELO ENTIDAD - RELACIÓN

Para esta altura del trabajo aún no hemos necesitado del uso de una base de datos, ya que todo se trabaja del lado del cliente en los campos de texto. Esperamos que para entregas futuras donde se tenga que guardar la información de la transcripción sí tengamos que realizar uno.

DISTRIBUCIÓN DEL TRABAJO

En principio, el trabajo fue distribuido de la siguiente manera: 1 persona asignada a la historia 6, 2 personas a las historias 7 y 10 respectivamente, 2 personas a la historia 8, 1 personas a la historia 9, 1 persona a la historia 11 y 1 persona para la documentación. Se investigó por separado sobre el framework a utilizar e información relevante para cubrir las historias asignadas.

Mientras completamos el trabajo los equipos o personas que lograban terminar su trabajo sirvieron de apoyo a los que presentaban dificultades, comprendemos que algunas historias representaban mayor trabajo que otras. Se emplearon aproximadamente 4 horas de trabajo individual y 6 horas de trabajo grupal.

ANÁLISIS DEL DOMINIO DE DATOS

HISTORIAS DE USUARIO	DESCRIPCIÓN	DATOS
6. Convertir PDF a HTML.	Facilitar la identificación de las secciones del documento.	Texto (HTML)
7. Extraer texto o HTML de PDF de imágenes.	Evitar que la transcripción necesaria para incluir programas existentes en un repositorio tenga que hacerse manualmente.	Texto
8. Mostrar todos los campos estándares asociados a programas analíticos.	Poder visualizar los campos que puede llenar a partir de la información que le presenta el archivo PDF que transcribe.	Texto (seleccionable)
9. Consultar programas analíticos almacenados en SIGPAE.	Poder leer y chequear si un programa analítico en PDF ya ha sido transcrito, además de poder usarlo como punto de partida para la transcripción de un programa similar.	Texto
10. Salvar estado de transcripción.	Evitar tener que reiniciar la transcripción cada vez que reinicie la sesión y para retomar la transcripción con el mínimo retrabajo posible.	Texto
11. Seleccionar instancia responsable del programa.	Poder registrar tal responsabilidad del programa.	Texto (seleccionable)

MANEJO DE GITHUB DE INTERACCIÓN

El método que se empleó para utilizar esta herramienta fue la creación de una rama maestra, y dos ramas secundarias, pues solo disponíamos de dos computadoras; luego de que cada vez que se lograba implementar, probar y verificar una historia, ésta fue subida a una rama secundaria del repositorio de GitHub como avance. Cuando todas las historias fueron implementadas y probadas, se hizo merge a la rama maestra.

EXPERIENCIA TDD

El TDD nos ayuda muchísimo a encontrar y prevenir fallas en el programa, es por eso que su uso a la hora de desarrollar ahorra mucho trabajo e incluso hace que ciertas tareas sean más fáciles. Las ventajas que nos da el TDD hacen que podamos desarrollar un software de buena calidad y que sea bastante robusto frente a los desafíos que enfrentará con los usuarios.

CONCLUSIÓN

El objetivo de este Sprint era realizar las 6 historias (complementando las realizadas en el primer sprint) que lo conforman. Recapitulando el primer sprint, tenemos un sistema que es capaz de leer documentos de texto en PDF que son enviados al sistema, para luego devolver el texto en un gran campo, donde se puede copiar la información para pegarla en otros campos descriptivos. Estos campos son igualmente editables. También es posible escoger desde una lista pre- definida cual es el trimestre y año correspondiente de la entrada en vigencia del programa. Ahora con lo realizado en este sprint garantizamos facilitar la identificación de las secciones del documento, incluir programas existentes en un repositorio, visualizar los campos que puede llenar a partir de la información que le presenta el archivo PDF, leer y chequear si un programa analítico en PDF ya ha sido transcrito, salvar estado de transcripción y poder registrar tal responsabilidad del programa.

Sin lugar a dudas el TDD representa un modelo de programación bastante eficaz para poder desarrollar softwares que cumplan con los requerimientos exigidos por el cliente, lo cual es el principal fin al desarrollar un software, satisfacer las necesidades que el cliente requiera.

Como recomendación, podemos decir que como el empleo continuo de estas técnicas mejoran los resultados obtenidos, debemos seguir aplicándolas a lo largo del proyecto para poder lograr las metas que nos vamos a ir proponiendo.