



Ministerio del Poder Popular para la Educación Superior
Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI-3715 - Ingeniería de Software - Laboratorio

Sprint 4

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)
Stephanie Espinoza (09-11100)

INTRODUCCIÓN	2
DESARROLLO	3
DISTRIBUCIÓN DEL TRABAJO	3
DIAGRAMA DE CLASES	4
DIAGRAMA DE INTERACCIÓN	5
MODELO ENTIDAD RELACIÓN	6
ANÁLISIS DEL DOMINIO DE DATOS	7
HISTORIAS DE USUARIO	7
DESCRIPCIÓN	7
MANEJO DE GITHUB DE INTERACCIÓN	9
EXPERIENCIA TDD	10
CONCLUSIÓN	11

INTRODUCCIÓN

En este informe se va a desarrollar los aspectos relacionados al cuarto 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, PyDev, 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 en esta ocasión.

A continuación se presentarán diferentes puntos que explican cómo fue nuestro trabajo en este último, entre estas cosas se encuentran todas las historias de usuario que desarrollamos a lo largo del curso, 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

- 1 - Convertir PDF a texto
- 2 - Recortar y pegar textstrings en áreas editables
- 3 - Editar textstrings en áreas editables de la pantalla
- 5 - Registrar período en que entra en vigencia un programa
- 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
- 12 - Extraer código de asignatura y departamento.
- 13 - Manejar campos adicionales.
- 14 - Separar fuentes de información recomendadas.
- 16 - Solicitar convertir una transcripción en programa analítico propuesto.
- 18 - Reportar transcripciones y programas por dependencia.

Cabe destacar que por ser el último sprint se corrigieron errores de las historias ya mencionadas para presentarlas en la entrega final.

DISTRIBUCIÓN DEL TRABAJO

Nuestro equipo se comprometió a realizar las historias ya mencionadas. Para este sprint nos comprometimos con la historia 18 en pro a corregir errores de historias pasadas y pulir el trabajo final.

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.

DIAGRAMA DE CLASES

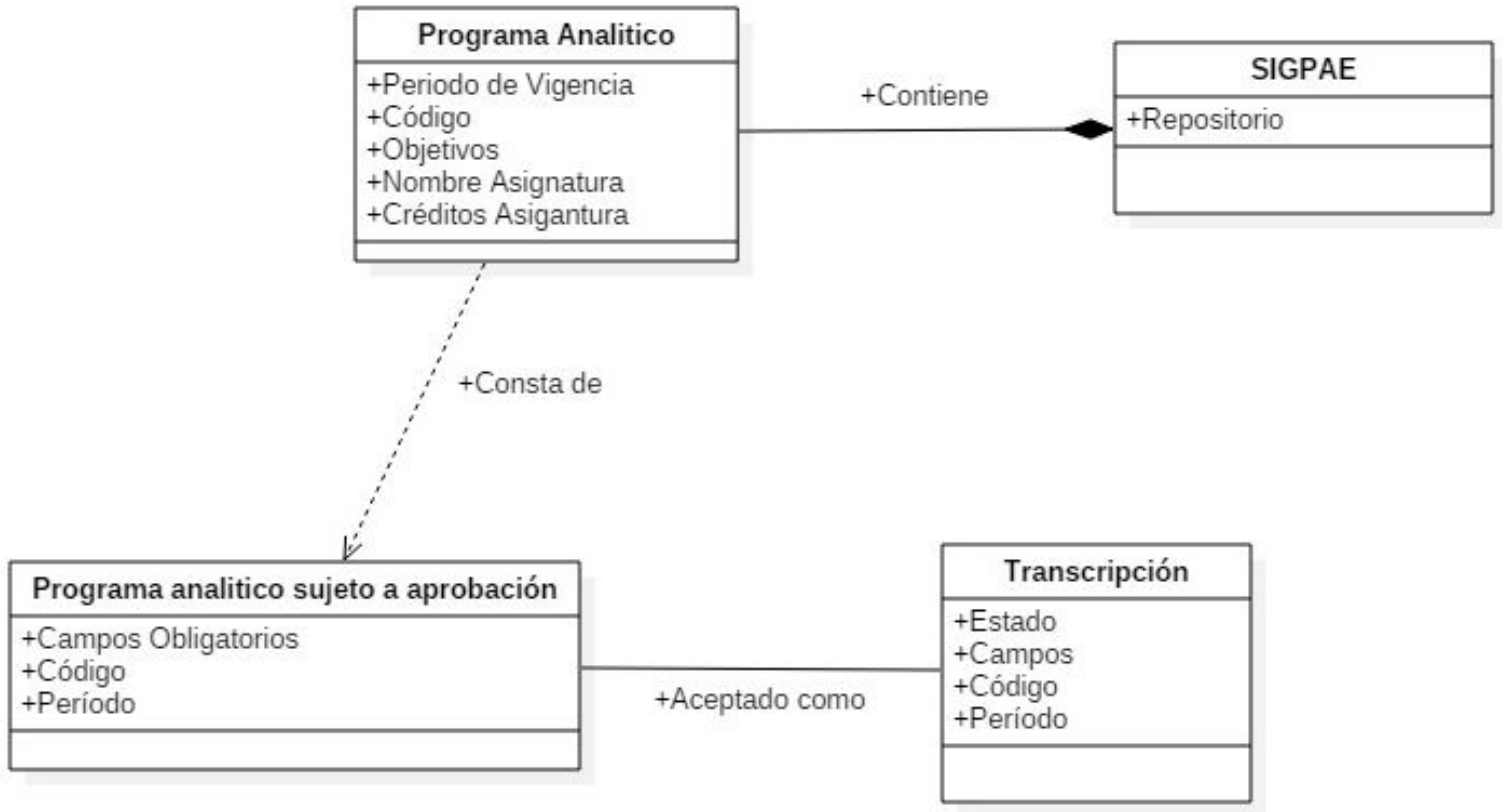
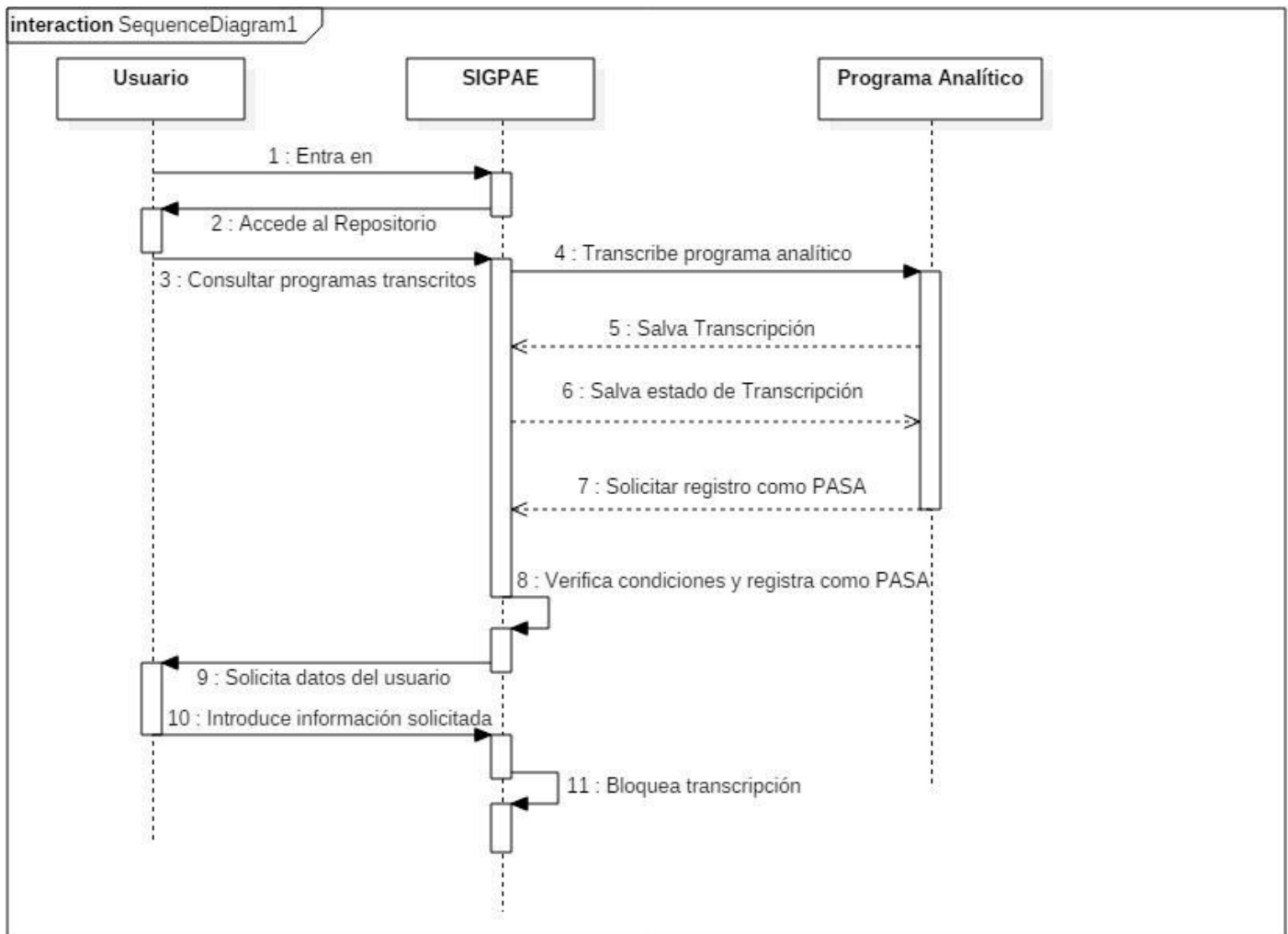
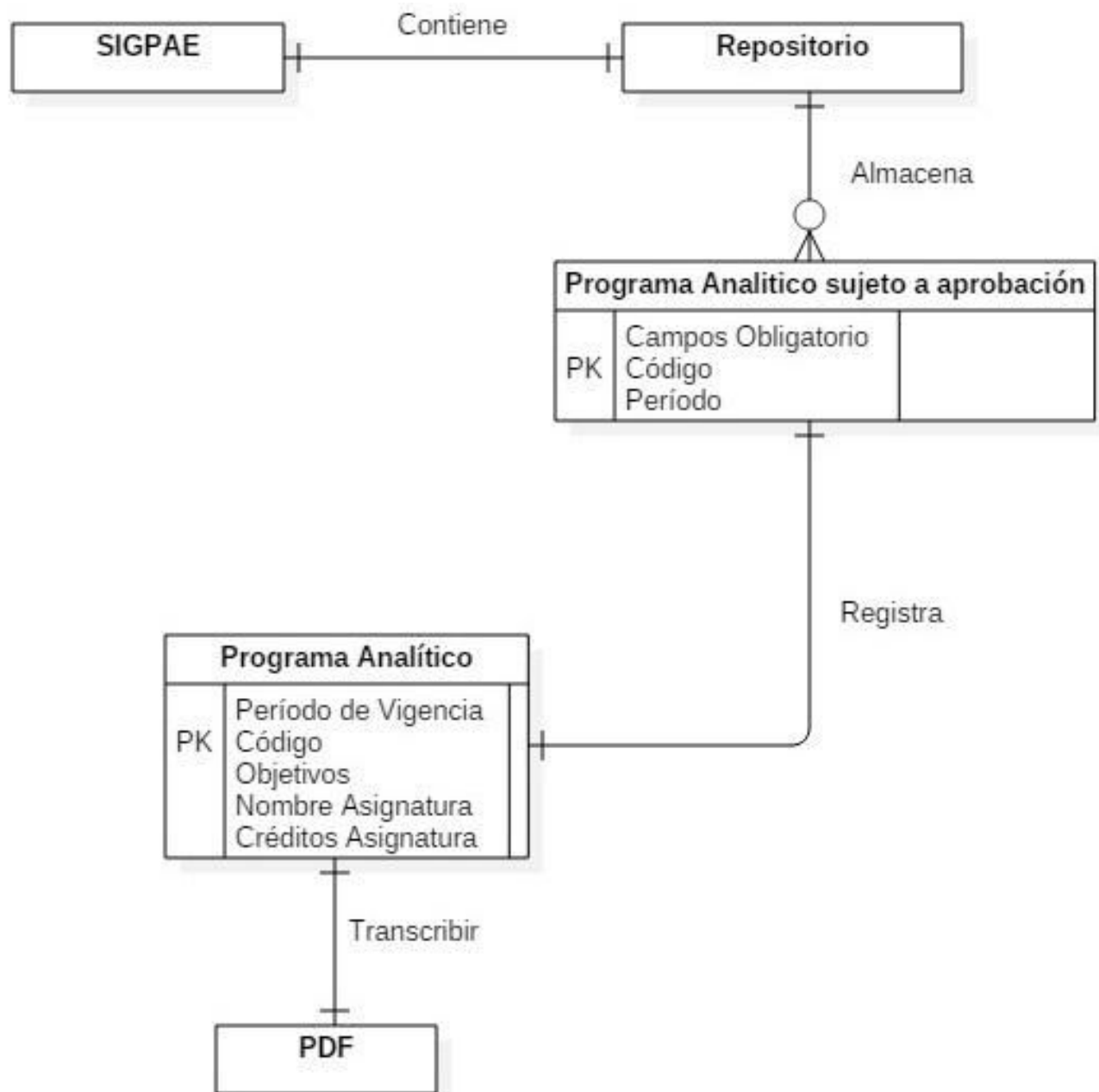


DIAGRAMA DE INTERACCIÓN



MODELO ENTIDAD RELACIÓN



ANÁLISIS DEL DOMINIO DE DATOS

HISTORIAS DE USUARIO	DESCRIPCIÓN	DATOS
1. Convertir PDF a texto	Convertir un programa analítico que se encuentra a partir de una página web en un archivo PDF, en un texto (textString) para evitar que la transcripción necesaria para incluir programas existentes en un repositorio tenga que hacerse manualmente.	Texto
2. Recortar y pegar textstrings en áreas editable.	Recortar la subcadena que escoja de un textstring en un área editable de la pantalla y pegarla en cualquier otra área editable de la pantalla que contenga textstrings.	Texto
3. Editar textstrings en áreas editables de la pantalla	Desplegar, ver y editar un textstring en un área de la pantalla para poder corregir manualmente cualquier posible error en la conversión o su procesamiento posterior.	Texto
5. Registrar período en que entra en vigencia un programa	Registrar el período a partir del cual es vigente el programa. El código de una asignatura y el período en que entró en vigencia identifican unívocamente cualquier programa analítico en la Universidad Simón Bolívar por lo que registrar tal período es crucial en la construcción de un repositorio de programas analíticos.	Texto
8. Mostrar todos los campos estándares asociados a programas analíticos	Mostrare todos los campos estándar asociados a programas analíticos para que el transcriptor pueda visualizar los campos que puede llenar a partir de la información que le presenta el archivo PDF que transcribe.	Texto

9. Consultar programas analíticos almacenados en SIGPAE	Consultar los programas analíticos de estudios contenidos en el repositorio de SIGPAE para leerlos, chequear si un programa analítico en PDF ya ha sido transcrito y, eventualmente, usarlo como punto de partida para la transcripción de un programa similar.	Texto
10. Salvar estado de Transcripción	Salvar el estado de una transcripción para 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	Seleccionar de un menú el nombre del ente responsable de dictar el programa analítico para que quede registrada tal responsabilidad del programa.	Texto
12. Extraer código de asignatura y departamento.	Activar la opción que el sistema proponga un código de asignatura y un departamento (o instancia) y me permita cualquier corrección al respecto para reducir la tasa de errores que podrían presentarse en una transcripción netamente manual.	Texto (seleccionable)
13. Manejar campos adicionales.	Capturar información que aparece en un programa analítico cuya estructura no se corresponde con lo indicado en el reglamento para no perder información que puede resultar útil. Para ello necesito manejar de manera más flexible campos adicionales.	Texto
14. Separar fuentes de información recomendadas.	Tener la opción de que el sistema ayude a extraer y separar las fuentes de información en forma estructurada para que,	Texto

	<p>eventualmente, se pueda revisar si son correctas, si se encuentran en la Biblioteca o si son de acceso abierto, información que va a ser clave para que la</p> <p>Biblioteca pueda dar el soporte apropiado sobre todo respecto a programas nuevos.</p>	
16. Solicitar convertir una transcripción en programa analítico propuesto.	Solicitar que una transcripción se registre en el SIGPAE como un programa analítico sujeto a aprobación por la instancia responsable para que los programas analíticos viejos que se transcriben puedan enriquecer el repositorio SIGPAE de programas analíticos.	Texto
18. Reportar transcripciones y programas por dependencia.	Solicitar un reporte de las transcripciones y/o programas contenidos en los repositorios SIGPAE y SIGPAE-Históricos para conocer el estado de los procesos de transcripción.	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 eran realizar las últimas historias y concluir con la entrega del software. Por experiencia de los anteriores Sprint hemos decidido comprometernos con menos historias con la finalidad de no defraudar al cliente y con la intención de realizarlas todas. Realizamos la historia 18 y pulimos el trabajo para la presentación final, realizamos nuevas pruebas unitarias y corregimos errores que nos mostraban las mismas. Esperamos un buen feedback de nuestro desempeño en este sprint y de nuestro trabajo final. A petición de nuestros profesores en la presentación mostraremos la totalidad de las historias mencionadas con nuestra expositora Stephanie Espinoza, que se integró a nuestro grupo para el 3er Sprint y desde entonces ha participado en nuestro grupo activamente.

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, se deben seguir aplicando para lograr resultados fructíferos con los grupos de desarrollo de software nuevos por continuar trabajando con este proyecto.

Fue muy grato trabajar en equipo y esperamos que nuestro proyecto sirva de apoyo a nuevos equipos de desarrollo de software y la universidad Simón Bolívar.

Atentamente

Equipo Lambda.