



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 3

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)

TABLA DE CONTENIDO

TABLA DE CONTENIDO	1
INTRODUCCIÓN.....	2
2.CONTENIDO.....	3
2.1 HISTORIAS DE USUARIO	3
2.2 DIAGRAMA DE CLASES	4
2.3 DIAGRAMA DE INTERACCIÓN	4
2.4 MODELO ENTIDAD-RELACIÓN.....	5
2.5 ANÁLISIS DEL DOMINIO	6
2.6 MANEJO DEL REPOSITORIO GITHUB.....	7
2.7 DISTRIBUCIÓN DEL TRABAJO	7
2.8 EXPERIENCIA TDD.....	7
3. CONCLUSIÓN.....	8

INTRODUCCIÓN

En este informe se va a desarrollar los aspectos relacionados al tercer 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 tercer 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.

2.CONTENIDO

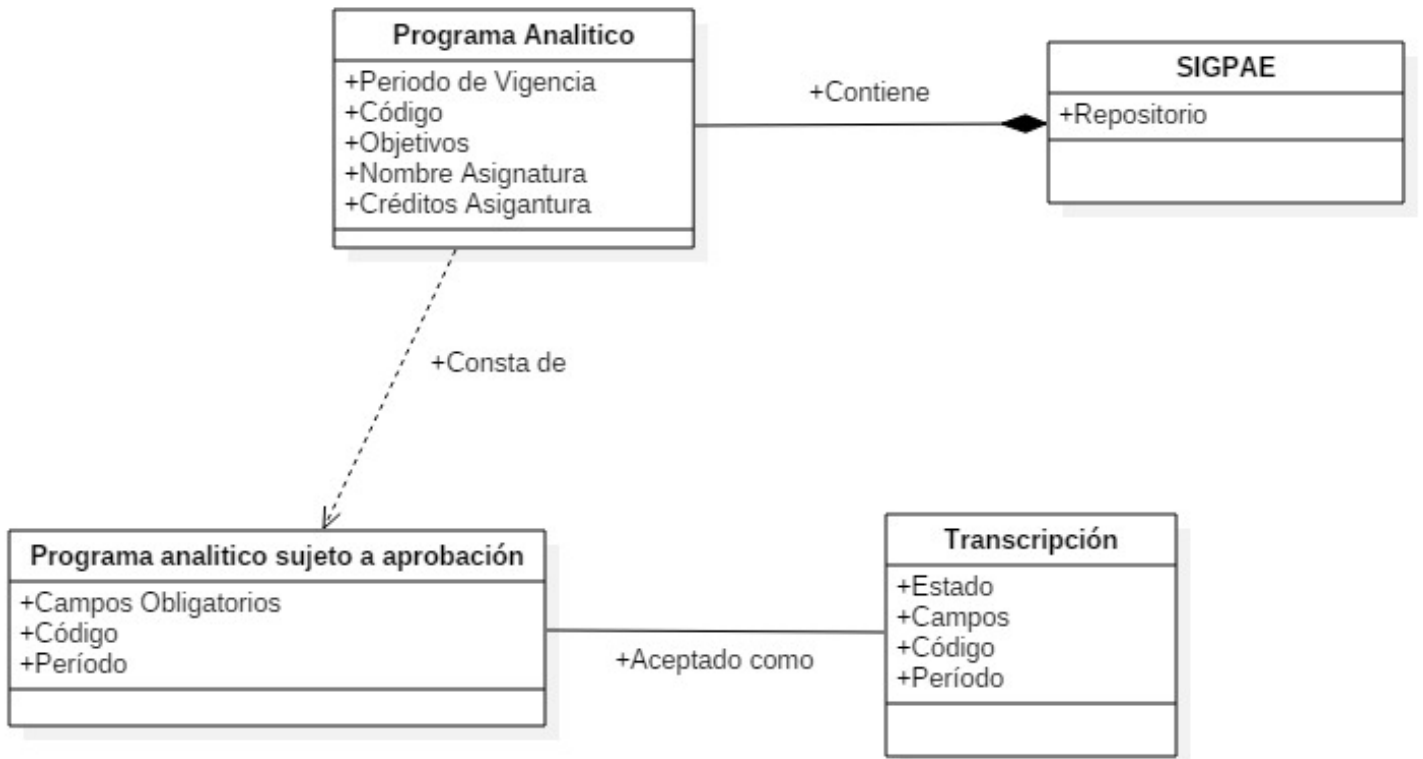
2.1 HISTORIAS DE USUARIO

12. Extraer código de asignatura y departamento.
13. Manejar campos adicionales.
14. Separar fuentes de información recomendadas.
15. Optar por alternativa a *Fecha de entrada en vigencia*.
16. Solicitar convertir una transcripción en programa analítico propuesto.
17. Optar por alternativa a campo *Objetivos*.

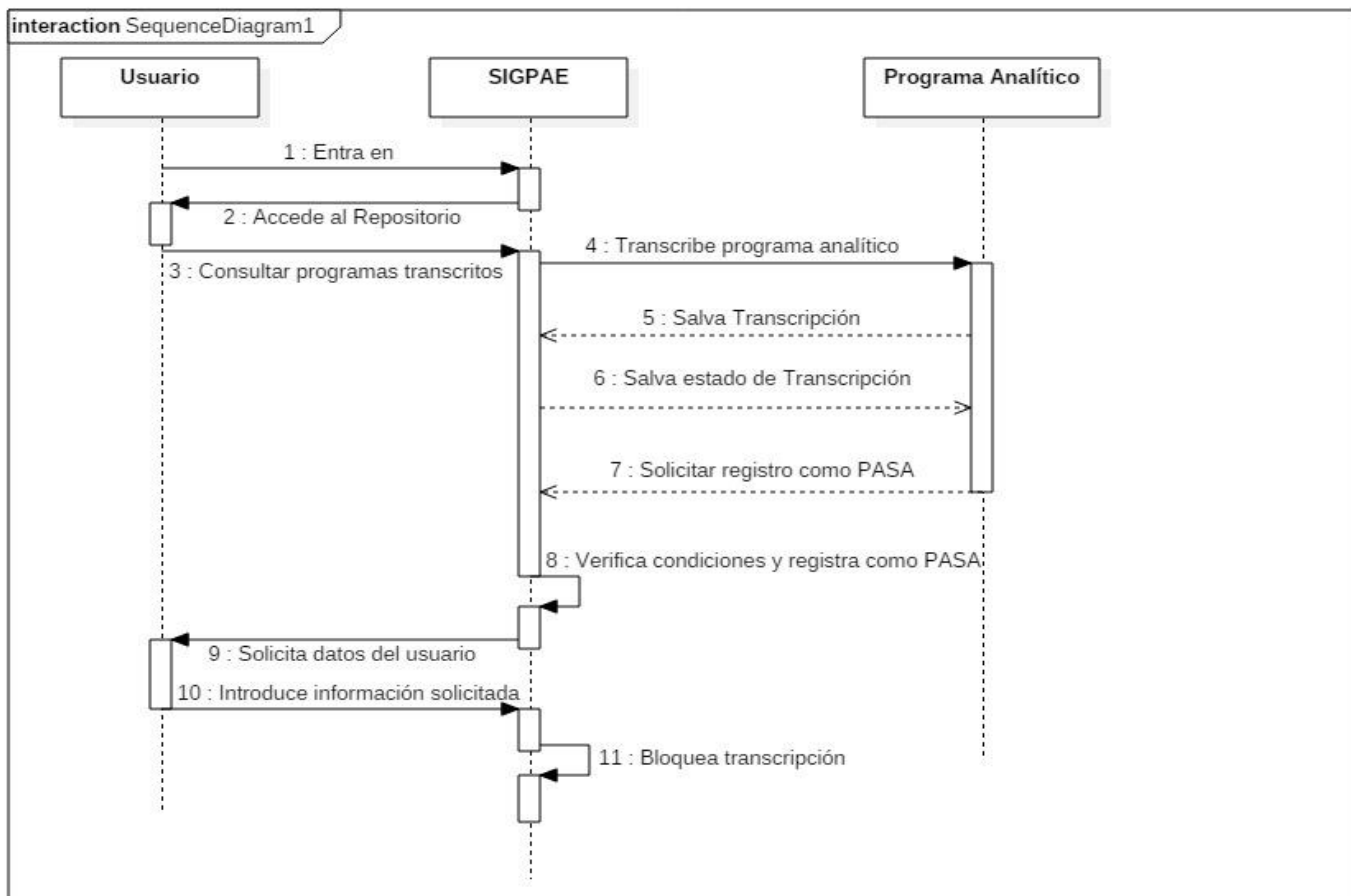
Cabe destacar que en el sprint pasado se acordó corregir y en efecto, se corrigieron los siguientes aspectos:

- No permitir al usuario seleccionar más de 40 horas.
- No permitir al usuario seleccionar más de 16 créditos.
- Se sustituyó "referencias bibliográficas" a "Fuentes recomendadas de información"
- Corrección de consultas: se distinguieron las consultas entre PDF, Transcripción y Programas Autorizados.

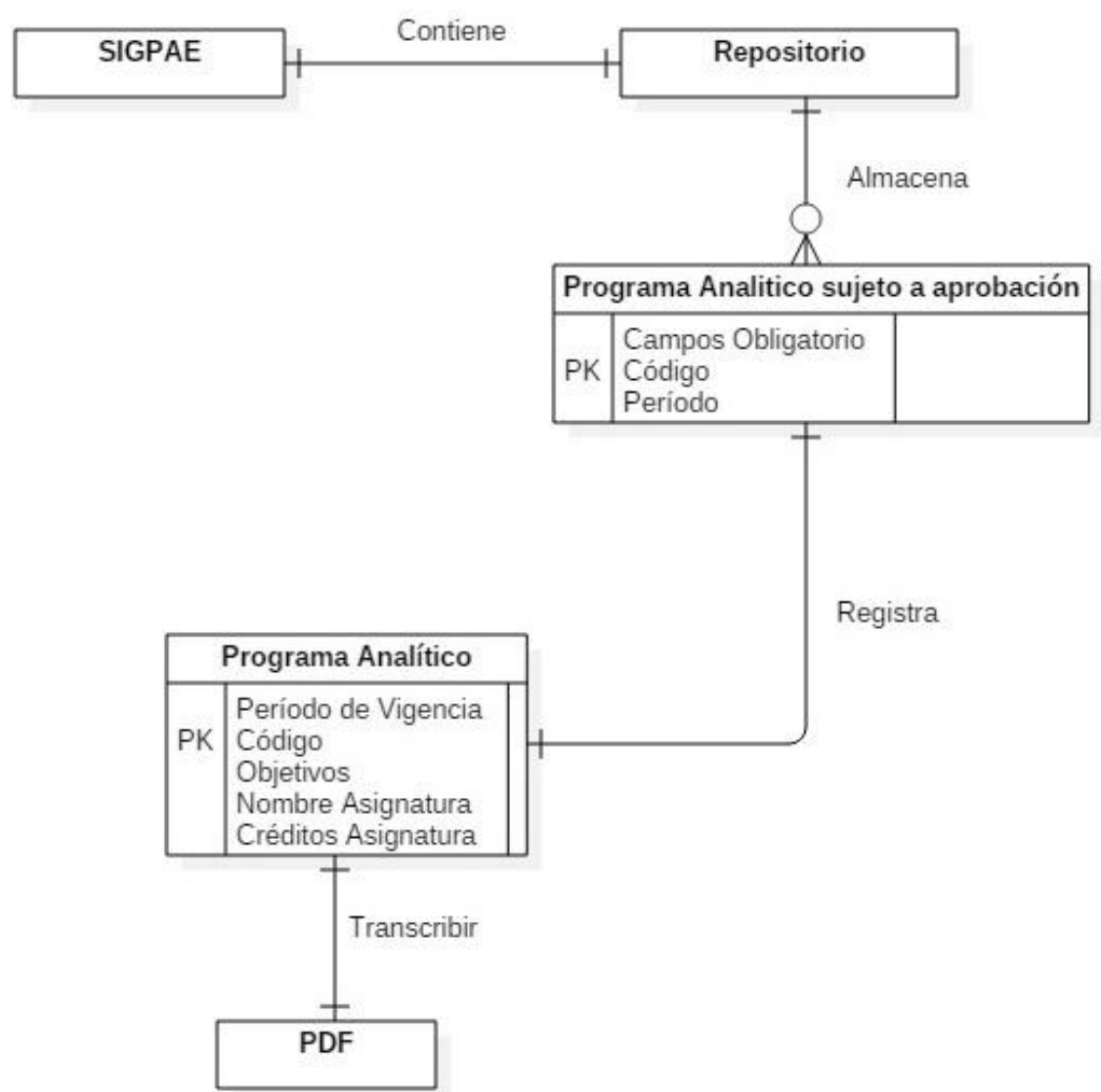
2.2 DIAGRAMA DE CLASES



2.3 DIAGRAMA DE INTERACCIÓN



2.4 MODELO ENTIDAD-RELACIÓN



2.5 ANÁLISIS DEL DOMINIO

DESCRIPCIÓN	HISTORIAS DE USUARIO
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.	12. Extraer código de asignatura y departamento.
<p>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.</p> <p>Para ello necesito manejar de manera más flexible campos adicionales.</p>	13. Manejar campos adicionales.
<p>Tener la opción de que el sistema ayude a extraer y separar las fuentes de información en forma estructurada para que, 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>	14. Separar fuentes de información recomendadas.
Escoger entre manejar la fecha de entrada en vigencia como un período y un año, o como una fecha para no tener que tomar la decisión de a qué período académico corresponde una fecha indicada en un programa analítico viejo.	15. Optar por alternativa a <i>Fecha de entrada en vigencia</i> .
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.	16. Solicitar convertir una transcripción en programa analítico propuesto.

<p>Seleccionar reemplazar el campo <i>Objetivos</i> por dos campos consecutivos <i>Objetivos Generales</i> y <i>Objetivos Específicos</i> para tener una estructura más flexible que se corresponde con dos modalidades de especificar los objetivos en los programas analíticos.</p>	<p>17. Optar por alternativa a campo <i>Objetivos</i>.</p>
---	--

2.6 MANEJO DEL REPOSITORIO GITHUB

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.

2.7 DISTRIBUCIÓN DEL TRABAJO

Nuestro equipo se comprometió a realizar las historias 13, 14 y 16 respectivamente, dejando como opciones tentativas realizar las historias 12, 15 y 17. Hemos tomado esa decisión con el fin de no defraudar al cliente y con la meta de cumplir con nuestro compromiso en su totalidad.

En principio, el trabajo fue distribuido de la siguiente manera: 2 persona asignadas a la historia 13, 2 personas a las historias 14, 2 personas a la historia 16 y 2 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.

2.8 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.

3. CONCLUSIÓN

El objetivo de este Sprint era realizar las 6 historias que lo conforman. 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 3 historias completas y buena parte de dos adicionales (historia 12 y 15 específicamente). Esperamos un buen feedback de nuestro desempeño en este sprint, en la presentación mostraremos más que las historias realizadas. Para este Sprint integramos en nuestro equipo a la compañera Stephanie Espinoza y se integró en nuestra dinámica sin problemas.

Sin lugar a dudas el TDD representa un modelo de programación bastante eficaz para poder desarrollar software 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.