

TRABAJO FINAL DE GRADUACIÓN

UNIVERSIDAD EMPRESARIAL SIGLO 21
FORMANDO LÍDERES

*Proyecto de desarrollo de un servicio de red social
Universitaria*

Alumno: Diaz Molina, Jorge Ariel
Carrera: Licenciatura en Informática
Año: 2009
Tutor: Prof. Calixto Maldonado

*Para esas dos almas, una allá y la otra acá. Cuyo
valor e infinito amor me obsequiaron el mayor de los
dones; el poder pensar en libertad. La posibilidad de
cazar sueños, solo mirando al cielo.*

Para ella y ella, que lo soportaron todo.

Índice

1	Título.....	1
2	Introducción.....	2
3	Objetivos.....	4
3.1	General.....	4
3.2	Particulares.....	4
4	Limites y Alcances.....	5
4.1	Limites.....	5
4.2	Alcance.....	7
5	¿Que son las redes sociales?.....	9
5.1	Facebook.....	10
5.2	My Space.....	11
5.3	LinkedIn.....	12
6	Los proyectos Informáticos.....	13
7	Arquitectura del Sistema.....	14
8	Caso de Uso (Use Cases).....	15
9	Metodologías de Desarrollo.....	16
9.1	RUP (Rational Unified Process)	17
9.1.1	Características Principales.....	18
9.2	SCRUM.....	21
9.2.1	SCRUM es un framework simple.....	22
10	Lenguaje de Modelado Unificado	27
10.1	Consideraciones.....	27
10.2	Diagramas de Estructura Estática.....	28
10.2.1	Asociación:.....	29
10.2.2	Agregación.....	29
10.2.3	Composición.....	30
10.2.4	Clase de Asociación.....	30
10.2.5	Diagramas de Caso de Uso.....	31
10.2.6	Diagrama de Secuencia.....	32
10.2.7	Diagrama de Colaboración.....	33
10.2.8	Diagrama de Estado.....	34
10.2.9	Diagrama de Paquetes.....	34
10.2.10	Conclusión.....	35
11	Acta inicialización del proyecto (Project Charter).....	36
12	Declaración de Alcance (Scope Statement).....	38
12.1	1. Sumario Ejecutivo.....	38
12.2	Objetivos.....	38
12.2.1	Objetivo General (Solución):.....	38
12.2.2	Objetivos Particulares.....	38
12.2.3	Historia Objetivo.....	39
12.3	Descripción del Proyecto.....	39
12.3.1	Alcance del Proyecto.....	39
12.3.2	Dependencias Externas:.....	39
12.3.3	Se asume que:.....	39
12.3.4	Limitaciones:.....	40

12.4	Hitos del Proyecto.....	40
2	Reuniones del Proyecto.....	40
2.1	Reuniones de Reportes.....	40
2.2	Administración de Publicaciones.....	41
2.3	Administración de Cambios.....	41
2.4	Administración de Comunicaciones.....	41
2.5	Administración de Logros.....	42
2.6	Administración de Recursos.....	42
12.5	El Scope Statement, WBS, Calendario del Proyecto, Plan de Riesgos son aprobados por el/los:	42
12.6	Los cambios en las líneas bases sobre el rendimiento del proyecto son aprobados por el/los:.....	42
12.7	Los entregables del Proyecto son aprobados/ aceptados por el/los:.....	42
13	Requerimientos del sistema.....	43
14	Decisiones Estratégicas.....	45
14.1	Metodología de Desarrollo.....	45
14.2	Arquitectura Inicial.....	45
14.2.1	Arquitecturas posibles.....	45
14.2.2	Basada en MVC	46
14.2.3	Basada en Servicios.....	46
14.2.4	Selección de la arquitectura.....	47
14.2.5	Especificaciones de la arquitectura.....	47
14.3	Lenguaje de Programación.....	49
15	Estimaciones.....	50
15.1	Introducción.....	50
15.2	Estimación de alcance.....	50
15.3	Estimación del tiempo.....	50
15.3.1	Relaciones Entre los Entregables.....	51
16	Plan de Fases.....	53
17	Cronograma.....	55
18	Factores susceptibles de afectar la fecha de entrega (Riesgos / Oportunidades).....	56
19	Estimaciones de Fecha de Entrega.....	57
20	Roles y Responsabilidades.....	58
21	Primera Fase.....	60
21.1	Listado de Casos de Uso.....	60
21.2	Subscribirse a la red social.....	61
21.2.1	Ficha de Caso de Uso.....	61
21.2.2	Diagrama de Clases.....	62
21.2.3	Pantallas.....	62
21.2.4	Diagrama de Secuencia.....	63
21.3	Login de Usuario.....	64
21.3.1	Ficha de Caso de Uso.....	64
21.3.2	Diagrama de Clases.....	64
21.3.3	Pantallas.....	65
21.4	Actualizar Perfil.....	66
21.4.1	Ficha de Caso de Uso.....	66
21.4.2	Diagrama de Clases.....	66
21.4.3	Pantallas.....	67

21.5	Actualizar Más Información.....	68
21.5.1	Ficha de Caso de Uso.....	68
21.5.2	Diagrama de Clases.....	68
21.5.3	Pantallas.....	69
21.6	Actualizar Información de Contacto.....	70
21.6.1	Ficha de Caso de Uso.....	70
21.6.2	Diagrama de Clases.....	70
21.6.3	Pantalla.....	71
21.7	Enviar E-mail de Invitación.....	72
21.7.1	Ficha de Caso de Uso.....	72
21.7.2	Diagrama de Clases.....	72
21.8	Invitar Contactos a Formar Parte de la Red.....	73
21.8.1	Ficha de Caso de Uso.....	73
21.8.2	Diagrama de Clases.....	73
22	Segunda Fase.....	74
22.1	Listado de Casos de Uso.....	74
22.2	Enviar Invitación de Amistad.....	75
22.2.1	Ficha de Caso de Uso.....	75
22.2.2	Pantallas.....	75
22.2.3	Diagrama de Secuencia.....	76
22.3	Ver Invitaciones Recibidas.....	77
22.3.1	Ficha de Caso de Uso.....	77
22.3.2	Diagrama de Clases.....	77
22.3.3	Pantallas.....	78
22.4	Cambiar Estado de las Invitaciones de Amistad.....	79
22.4.1	Ficha de Caso de Uso.....	79
22.4.2	Diagrama de Clases.....	80
22.5	Listar Amigos.....	80
22.5.1	Ficha de Caso de Uso.....	80
22.5.2	Diagrama de Clases.....	80
22.5.3	Pantallas.....	81
22.6	Cambiar Estado de los Amigos.....	82
22.6.1	Ficha de Caso de Uso.....	82
22.6.2	Diagrama de Clases.....	82
22.6.3	Pantallas.....	83
22.7	Redactar Mensaje.....	84
22.7.1	Ficha de Caso de Uso.....	84
22.7.2	Diagrama de Clase.....	85
22.7.3	Pantallas.....	85
22.7.4	Diagrama de Secuencia.....	86
22.8	Ver Bandeja de Entrada.....	87
22.8.1	Ficha de Caso de Uso.....	87
22.8.2	Diagrama de Clases.....	87
22.8.3	Pantallas.....	88
22.9	Leer Mensaje.....	89
22.9.1	Ficha de Caso de Uso.....	89
22.9.2	Diagrama de Clase.....	89
22.9.3	Pantalla.....	90

22.10	Contestar Mensaje.....	91
22.10.1	Ficha de Caso de Uso.....	91
22.11	Cambiar Estado de Mensaje.....	92
22.11.1	Ficha de Caso de Uso.....	92
23	Tercera Fase.....	93
23.1	Listado de Casos de Uso.....	93
23.2	Agregar Evaluador.....	94
23.2.1	Ficha de Caso de Uso.....	94
23.2.2	Diagrama de Clases.....	95
23.2.3	Pantallas.....	95
23.3	Registrar Examen.....	96
23.3.1	Ficha de Caso de Uso.....	96
23.3.2	Diagrama de Clases.....	96
23.3.3	Pantalla.....	97
23.3.4	Diagrama de Actividad.....	98
23.4	Agregar Tema a una Materia.....	98
23.4.1	Ficha de Caso de Uso.....	98
23.5	Listar Temas.....	100
23.5.1	Ficha de Caso de Uso.....	100
23.5.2	Pantallas.....	100
23.6	Seleccionar Preguntas de Examen.....	101
23.6.1	Ficha de Caso de Uso.....	101
23.6.2	Diagrama de Clase.....	102
23.6.3	Pantallas.....	102
23.7	Agregar Pregunta de Examen.....	103
23.7.1	Ficha de Caso de Uso.....	103
23.7.2	Diagrama de Clase.....	103
23.7.3	Pantallas.....	104
23.8	Cambiar Estado Materias.....	105
23.8.1	Ficha de Caso de Uso.....	105
23.8.2	Diagrama de Clase.....	106
23.8.3	Pantallas.....	106
23.9	Escribir Mi Pregunta.....	107
23.9.1	Ficha de Caso de Uso.....	107
23.9.2	Diagrama de Clase.....	108
23.9.3	Pantallas.....	108
23.9.4	Diagrama de Actividad.....	109
23.10	Listar Mis Preguntas.....	110
23.10.1	Ficha de Caso de Uso.....	110
23.10.2	Pantallas.....	110
23.11	Ver Respuestas.....	111
23.11.1	Ficha de Caso de Uso.....	111
23.11.2	Diagrama de Clase.....	112
23.11.3	Pantallas.....	112
23.12	Listar Preguntas a Responder.....	112
23.12.1	Ficha de Caso de Uso.....	112
23.12.2	Diagrama de Clase.....	112
23.12.3	Pantallas.....	113

23.13 Responder Preguntas.....	114
23.13.1 Ficha de Caso de Uso.....	114
23.13.2 Diagrama de Clase.....	114
23.13.3 Pantallas.....	115
23.14 Publicar Noticia.....	116
23.14.1 Ficha de Caso de Uso.....	116
23.14.2 Diagrama de Clase.....	116
23.15 Ver Mis Noticias.....	117
23.15.1 Ficha de Caso de Uso.....	117
23.15.2 Pantallas.....	117
24 Cuarta Fase.....	118
24.1 Listado de Casos de Uso.....	118
24.2 Subir Archivo.....	120
24.2.1 Ficha de Caso de Uso.....	120
24.2.2 Diagrama de Clases.....	121
24.2.3 Pantallas.....	121
24.3 Listar Archivos.....	123
24.3.1 Ficha de Casos de Uso.....	123
24.3.2 Diagrama de Clases.....	123
24.3.3 Pantallas.....	123
24.4 Listar Archivos de Mis Amigos.....	125
24.4.1 Ficha de Casos de Uso.....	125
24.4.2 Diagrama de Clases.....	125
24.4.3 Pantallas.....	126
24.5 Bajar Archivos.....	127
24.5.1 Ficha de Casos de Uso.....	127
24.5.2 Diagrama de Clases.....	127
24.5.3 Diagrama de Secuencia.....	128
24.5.4 Pantallas.....	128
24.6 Ver Comentarios de Una Noticia.....	129
24.6.1 Ficha de Caso de Uso.....	129
24.6.2 Diagrama de Clases.....	130
24.6.3 Pantallas.....	130
24.7 Escribir Comentario.....	131
24.7.1 Ficha de Caso de Uso.....	131
24.7.2 Pantallas.....	131
24.8 Listar Personas que me Pueden Ayudar.....	132
24.8.1 Ficha de Caso de Uso.....	132
24.8.2 Diagrama de Clases.....	132
24.8.3 Pantallas.....	133
25 Conclusiones Sobre el Proyecto.....	134
25.1 Herramientas que contribuyen directamente al desarrollo de Software.....	134
25.1.1 Arquitectura de una Aplicación.....	134
25.1.2 Buenas Prácticas en la Programación y Patrones utilizados.....	135
25.1.3 Framework de Desarrollo.....	135
26 Herramientas que contribuyen Indirectamente al desarrollo de software.....	136
26.1.1 Metodología de Desarrollo.....	136
26.1.2 Lenguaje de Modelado.....	137

26.1.3 Herramientas de Soporte.....	137
27 Conclusiones Personales.....	138
27.1 Conclusiones sobre la Industria.....	138
27.2 Conclusiones sobre el Rol de Informático.....	138
28 Anexo Glosario.....	141
29 Anexo Administración de Proyectos.....	147



PRESENTACIÓN DEL PROYECTO

- Título
- Introducción
- Objetivos
- Límites y alcances



1 Titulo

*Proyecto de desarrollo de un servicio de red social
Universitaria*



2 Introducción

“Es la visión primordial de MiFacu.com encontrar soluciones simples a las 3 preguntas canónicas que todo estudiante se realiza; ¿Quién me puede ayudar?, ¿De dónde puedo estudiar? Y ¿Qué te tomaron en el examen?”

Las redes sociales son una de las partes fundamentales de lo que se conoce actualmente como Web 2.0. Este último término hace referencia a la Web basada en comunidades de usuarios, siendo estas, Web blogs, foros, wikis, folcsonomias¹ y redes sociales.

La Web 2.0 tiene por finalidad realzar la creatividad, compartir información, incentivar la colaboración entre los distintos miembros, y compartir funcionalidades de la Web.

Dentro de éste entorno, es que Wikipedia² define una red social como:

“una estructura social que se puede representar en forma de uno o varios grafos en el cual los nodos representan individuos (a veces denominados actores) y las aristas relaciones entre ellos. Las relaciones pueden ser de distinto tipo, como intercambios financieros, amistad, relaciones, o rutas aéreas. También es el medio de interacción de distintas personas como por ejemplo juegos en línea, chats, foros, spaces, etc”

El sitio Social networkList³ declara un listado de más de 80 redes sociales estableciendo una clasificación por rubros a los cuales están dedicadas. Estos van desde los meramente culturales hasta aquellos temas más triviales tales como boites, tragos y moda. Con rangos de cantidad de usuarios que van desde unos pocos, hasta los millones, parecería que estos servicios de redes sociales dan soporte Web a todos aquellos intereses comunes que poseen los usuarios de Internet.

Son ejemplos de la masiva concurrencia de usuarios con los que cuentan estos servicios en el año 2008, los sitios; Facebook⁴ y Myspace⁵. El primero de ellos cuenta con más de 120.000.000 de usuarios alrededor del mundo y es seguido de cerca por el segundo, con más de 115.000.000 de usuarios. Ambos, figuran en el listado de los 10 más populares de Alexa.com, compañía dedicada a medir las estadísticas de usuarios que navegan por la Web. Además, de lo mencionado anteriormente, es importante destacar que cinco de los diez primeros sitios, forman parte de lo que llamamos *La WEB 2.0*.

1 Es la creación de tags, clasificaciones e indexaciones compartidas. Ejem: del.icio.us

2 http://es.wikipedia.org/wiki/Redes_sociales

3 http://socialnetworklist.com/component/option,com_bookmarks/Itemid,26/

4 <http://www.facebook.com>

5 <http://www.myspace.com>



¿Pero acaso es un requisito que las redes sociales sean de masiva concurrencia? No. Una red social puede contar con unos pocos usuarios, si la misma es especializada, o se focaliza en un público determinado. Sin embargo, no es la cantidad de usuarios ni su especialización lo que caracteriza a las redes sociales, sino más bien la explotación que ellas realizan de *Las 3CS*⁶; *Comunicación, Comunidad, Cooperación*. La diferencia entre unas y otras, son las herramientas que utilizan y la creatividad para fomentar la interacción entre sus usuarios. Es así como podemos encontrar herramientas tales como “*gentes que tal vez conoces*”, de facebook, que permite visualizar los amigos que comparten varios de tus amigos, o el buscador de “*personas que quieren aprender tu idioma*” de Livemocha⁷.

Y es en éste contexto, es decir en el de las relaciones sociales, como las que ocurren en las comunidades universitarias, en donde la WEB 2.0, y mas precisamente las redes sociales, puede prestar colaboración.

De este proyecto se debe esperar la realización de una herramienta de software para dar soporte informático para la comunicación, cooperación y creación de vínculos entre los distintos actores de una universidad. Para ello, se tomará de ejemplo La Universidad Siglo 21, en el primer semestre del año 2009.

Bajo ningún punto de vista, este trabajo, y la herramienta resultante, deben ser considerados como una visión final y absoluta, ni tampoco se pretende realizar un trabajo de la magnitud de una red social como Facebook, o MySpace, o alguna de las redes sociales mas conocidas de la actualidad, ya que estas compañías millonarias cuentan con miles de programadores y personal dedicados a la creación y mantenimiento de estos sistemas. Si se pretende brindar un puntapié de partida tanto en fundamentos teóricos como en aquellos que hacen a su implementación practica para el desarrollo de una red social.

6 Fuente: revista *Users* – N# 203 – Editorial Tectimes -2008

7 Red social para aprender idiomas - <http://www.livemocha.com/>



3 Objetivos

3.1 General

El siguiente trabajo busca establecer algunas bases teóricas, prácticas y metodológicas para analizar, diseñar, desarrollar e implementar una herramienta que permita dar soporte a las tareas y actividades, que desarrollan los distintos actores de una universidad al interactuar entre sí. Como ejemplo de desarrollo se seguirá las actividades de los principales miembros de la universidad siglo 21 durante el primer semestre del año 2009.

Al concluir el proyecto se espera contar con una herramienta informática o de software, para ser implementada y probada, capaz de dar soporte a las actividades de mensajería interna entre los distintos miembros, compartir material de estudio, realizar preguntas a los amigos o compañeros relacionadas a las materias de la carrera que cursan. Además de lo mencionado anteriormente, se espera contar con la documentación de procedimientos que permita la puesta en marcha, su utilización, el posterior mantenimiento y escalabilidad de la herramienta.

3.2 Particulares

- Permitir a un alumno conocer a otros alumnos que asisten a su universidad.
- Permitir a un alumno compartir material de estudio.
- Permitir a un alumno conocer novedades sobre una materia en particular.
- Permitir al alumno conocer novedades de las actividades de sus compañeros.
- Permitir al alumno declarar su perfil y ver el perfil de los demás compañeros.



4 Limites y Alcances

Tanto los limites como los alcances serán descriptos para la aplicación resultante, ya que la documentación del proyecto, la presentación final o todas aquellas actividades relacionadas a la administración del proyecto, se consideran supuestas, para cumplir con los objetivos de la materia.

4.1 Limites

Importante:

Este trabajo, bajo ninguna circunstancia debe ser comparado en magnitud y despliegue con el producto de un servicio de red social comercializado por alguna compañía millonaria, ya que por razones obvias, los recursos encomendados al desarrollo y mantenimiento de estas aplicaciones exceden ampliamente a los propósitos de un trabajo final de graduación.

Queda **dentro** de los límites de la aplicación, el desarrollo de los siguientes módulos considerando los mismos como circuitos de funcionalidades de negocio:

1. El desarrollo de un módulo que permita la administración del perfil de cada uno de los usuarios que intervienen.
2. El desarrollo como la implementación de un módulo que permita el ingreso a la aplicación y la suscripción al sistema (*Log in*).
3. El desarrollo e implementación de un módulo que permita el envío de mensajería interna dentro del sistema.
4. El desarrollo e implementación de un módulo que permita a un alumno realizarle preguntas a sus compañeros.
5. El desarrollo e implementación de un módulo que permita la creación de relaciones entre los miembros de la comunidad
6. El desarrollo e implementación de un módulo que permita compartir material de estudio entre los miembros de la comunidad.
7. El desarrollo e implementación de componentes que permitan el envío de mensajería externa a e-mails



8. El desarrollo e implementación de un componente que permita visualizar las noticias o las novedades de un usuario.
9. La creación de filtros de sanitización de los datos.
10. Protección de ataques XSS y CSRF

Quedará **fuera** de los límites del proyecto:

- El desarrollo de un chat, o cualquier medio de mensajería instantáneo, no excluyendo la posibilidad de la utilización de herramientas de terceros para dicha finalidad. De no ser así, se dejará sentado, la forma en la cual se podrá agregar dicha funcionalidad.
- El streaming de audio y video, o transmisión en tiempo real.
- El desarrollo de herramientas que permitan la graficación de las personas como nodos y las relaciones interpersonales o cualquier tipo de herramienta de graficación dinámica de relaciones, pudiendo incluir la posibilidad de herramientas estadísticas.
- La creación de juegos, u otras aplicaciones para compartir entre los miembros.
- El desarrollo de interfases para dispositivos móviles. Pudiendo dejar sentadas las bases para su desarrollo posterior a través de servicios prestados por la aplicación.
- Queda fuera de la aplicación, el desarrollo de un módulo que permita, la realización o *feeds* o noticias personales aptas para ser interpretadas por otras aplicaciones o sitios personales.
- Quedan fuera del alcance del proyecto, cuestiones relevantes al *look and feel* de la aplicación, o relacionados a la estética del diseño grafico de la aplicación. Sin descuidar cuestiones relacionadas a la buena presentación y navegación de la aplicación. Además de aspectos relacionados a la facilidad de uso de la aplicación.
- La codificación de un foro, no excluyendo la posibilidad de la utilización de componentes de terceros *open source*, o gratis para tal finalidad.
- El envío de SMS o aquellos tipos de mensajería externa que requiera algún medio de comunicación en el que intermedien proveedores pagos de conectividad.
- La creación de alguna herramienta de seguimiento de los usuarios, que permita sacar estadísticas personales, u obtener información que excedan los límites y la finalidad de este sistema.
- La codificación de módulos que se encuentran fuera del alcance de la herramienta a desarrollar. salvo por la codificación considerada vital tanto para la implementación



como el diseño de la arquitectura del sistema.

4.2 Alcance

Se establece el alcance del proyecto en relación a aquello que se encuentra dentro de los límites del sistema (concordancia con la numeración de los límites).

Queda dentro del alcance del proyecto:

1. **Administración de perfiles:** el desarrollo de componentes que permitan la creación, obtención, actualización y borrado de los datos personales de un usuario.
2. **Log in:** el mismo incluirá la asignación de permisos a los actores que interviene, los accesos a la aplicación, la encriptación en la serialización de las contraseñas, y la suscripción de los miembros.
3. **Mensajería interna:** la creación de componentes para permitir la edición y visualización de mensajes y el cambio de estado de los mismos entre leídos y pendientes de leer. Además de ello permitirá ver las notificaciones de las acciones realizados por otros miembros de la red social y las aceptaciones de de las solicitudes que se realizan. Toda la mensajería, será almacenada en la base de datos.
4. **Preguntas:** El modulo a desarrollar deberá permitir enviar preguntas sobre una materia determinada a las personas o compañeros que pueden resolver dicha pregunta. Además de ello el estudiante deberá ser capaz de leer las respuestas de sus compañeros y determinar cual de ellas es la mejor o la que ha resuelto su inquietud.
5. **Relaciones entre los miembros de la comunidad:** incluirá poder establecer relaciones de amistad entre los miembros y modificar el estado de dichas amistades. Este módulo deberá incluir también la capacidad de enviar invitaciones a los conocidos y amigos, las mismas tendrán una difusión tanto interna como externa a las casillas de e-mail de los estudiantes que las reciban.
6. **Compartir material:** incluirá tanto el subir archivos como el descargar archivos que un usuario haya puesto como disponibles entre sus amigos. Además de lo mencionado, los usuarios deben poder compartir links a las URL que consideren (folcsonomías).
7. **Envío de mensajería externa:** incluirá el envío de mails, para invitar a nuevos usuarios a formar parte de la comunidad. Además de lo anterior, las noticias más destacadas de sus amigos se enviar a la casilla que el estudiante registró como nombre de usuario.
8. **Estadísticas básicas:** incluirá una página de inicio, o *dashboard* con las estadísticas o



noticias básicas de cada uno de los usuarios y los links para acceder a las distintas secciones.

9. ***filtros de sanitización de datos***: incluirá la creación de filtros de sanitización de datos que permitan prevenir ataques de *sql injection*⁸ o XSS⁹ (*Cross Site Scripting*). Además de lo anterior, la aplicación resultante debe incluir mecanismos de protección contra ataques CSRF o *Cross Site Request Forgery*.¹⁰

8 Técnica utilizada para insertar comandos sql en las bases de datos con alguna finalidad maliciosa

9 Técnica utilizada para insertar tags de html con alguna finalidad maliciosa.

10 Código Malicioso que tiene por finalidad aprovechar la confianza que un sitio tiene de un usuario conocido.



FUNDAMENTACIÓN TEÓRICA

- Que son las redes sociales
- Los Proyectos Informáticos
- Arquitectura del Sistema
- Metodologías de Desarrollo
- UML



5 ¿Que son las redes sociales?

Existen diferentes conceptos sobre las redes sociales, pero todos ellos convergen en lo estructural de su composición, y la existencia de nodos interrelacionados entre si.

La revista User¹¹ define a una Red social propiamente dichas como:

“Un sistema que agrupa a usuarios bajo diferentes criterios, y permiten que las personas se conozcan y establezcan un contacto frecuente. En muchas ocasiones, estas redes son abiertas, es decir que cualquier persona puede acceder a ellas y contactarse con otros integrantes de la comunidad. En otras, se dice que son cerradas, por que cada usuario decide a quien desea admitir y rechazar.”

Por otro lado podemos encontrar la definición de Wikipedia en español¹², que sostiene que:

“Una red social es una estructura social que se puede representar en forma de uno o varios grafos en el cual los nodos representan individuos (a veces denominados actores) y las aristas relaciones entre ellos. Las relaciones pueden ser de distinto tipo, como intercambios financieros, amistad, relaciones sexuales, o rutas aéreas. También es el medio de interacción de distintas personas como por ejemplo juegos en línea, chats, foros, spaces, etc.”

Pero la mayor enciclopedia del mundo on-line difiere en su versión en español con la versión en inglés. La primera de ellas no hace una diferencia entre los conceptos de *Redes Sociales* y el de *Servicios de Redes Sociales*. Wikipedia en inglés define el concepto de *Red social* como:

“Una estructura social conformada por nodos (los cuales generalmente son individuos u organizaciones) que están relacionados por uno o más tipos específicos de interdependencia, tales como valores, visiones, ideas, intercambio financiero, amistades, parentesco, gustos contrapuestos, conflictos, o comercio. El resultado de las estructuras basadas en gráficos son siempre muy complejos”¹³

11 Revista especializada en informática – Edición N# 203

12 http://es.wikipedia.org/wiki/Red_social

13 Interpretada al castellano de http://en.wikipedia.org/wiki/Social_network



Mientras que define a un Servicio de Red Social como:

*“Aquellos concentrados en construir comunidades online de personas que comparten intereses y actividades, o que están interesados en explorar los intereses y actividades de los otros. La mayoría de los servicios sociales están basados en tecnología Web, y proveen una variedad de medios para que los usuarios interactúen entre sí, tales como e-mails, y servicios de mensajería instantánea”.*¹⁴

En este trabajo, utilizaré el término *Red Social* indistintamente del de *Servicio de Redes Sociales*, por considerar que es éste el sentido gestáltico que asumimos cuando se nos habla de redes sociales. Además de que Wikipedia en español no realiza semejante distinción.

Pero este concepto, además, suele hacer referencia no solo a las redes sociales propiamente dichas, si no que también se utiliza para hacer referencia a todo aquello tipificado bajo lo denominado Web 2.0. La misma revista *User*, sostiene que tanto los blogs, como los foros son redes sociales, por sus características de agrupar a los usuarios por gustos, conocimientos, e intereses en comunes.

A continuación y a manera de ejemplo procederé a mencionar tres redes sociales que se hicieron famosas en el transcurso de año 2008.

5.1 Facebook

Existen muchas redes sociales en la actualidad pero, sin duda la más importante en cuanto a cantidad de miembros es Facebook. Fundada en el 4 de febrero del 2004, originalmente para que los alumnos de Harvard se conozcan entre sí. Con el paso del tiempo la red social privada se fue extendiendo a las demás universidades de EE.UU. Después siguieron los estudiantes de secundaria. En la actualidad cualquier persona mayor de 13 años puede ser miembro de Facebook.

El éxito de facebook radica en que es fácil de usar por cualquier persona, aunque también ha recibido críticas por sus pantallas sobrecargadas de información. Pero aún así, cualquier usuario común de Internet con un poco de practica, termina sintiéndose cómodo con el. Otro aspecto que colabora en este sentido, es que todos los usuarios tienen sus páginas personales o perfil, con el mismo aspecto, lo que facilita navegar entre las actividades de todos tus contactos, ver sus fotos y videos como si fuera tu propia página personal.

Otra característica de Facebook, es que es una red privada, por lo que se necesita permiso

¹⁴ Interpretada al castellano de http://en.wikipedia.org/wiki/Social_network_services



para que el resto de las personas pueda visitar tu página personal, por lo que no nos vamos a dar con la sorpresa de algún comentario inesperado, y nuestra información personal solo va a ser vista por personas autorizadas.

Pero sin duda, lo que hace a esta red social muy particular, es la cantidad de servicios que brinda. Esto es posible, por que la misma permite a cualquier persona desarrollar una aplicación para poder ser usada por millones de personas¹⁵. Solo debemos tener algunos conocimientos sobre programación de aplicaciones Web y suscribirnos para recibir una llave que nos permitirá formar parte de la comunidad de desarrolladores de la red social. Actualmente, Facebook cuenta con más de 2000 aplicaciones realizadas por desarrolladores independientes, lo que le permite ir creciendo en sus funcionalidades día a día. Entre las más utilizadas podemos encontrar, aplicaciones referentes a juegos, álbumes de fotografía, y videotecas.

5.2 My Space

My Space es una de las redes más populares del momento, disputándose directamente con Facebook el trono de la mayor cantidad de usuarios.

Cuando realizamos una suscripción al sistema, lo que realmente estamos realizando es crear una página personal con una URL propia a la cual podrá acceder cualquier usuario. Podemos subir fotos, música, video, y si contamos con algún conocimiento de lenguajes de desarrollo Web, podemos personalizar el aspecto de nuestra página personal. A diferencia de Facebook, el perfil de los usuarios de esta red social, esta orientado a los adolescentes en lugar de orientarse a universitarios.

Algunas de las cosas que MySpace nos brinda en su perfil son:

- *Moods*: son pequeños emoticones, que son usados para describir el humor con el que siente el usuario.
- *Blurbs*: es un pequeño resumen o algunas palabras o frases que describen en forma general al usuario.
- *Blogs*: el usuario en su perfil, tiene algunos campos estándares, para ingresar contenido actualizado.
- *Multimedia*: Además de lo mencionado, en MySpace el usuario puede subir sus propios archivos fotográficos, de música y de video. Y posteriormente definir cuales se mostrarán en su perfil.
- *Comentarios*: en la sección de comentarios, los amigos del usuario pueden dejar

¹⁵ <http://developers.facebook.com/?ref=pf>



cualquier comentario sobre todas aquellas cosas sobre las cuales leen en la página personal.

- *Personalización del Perfil*: El servicio permite a los usuarios, personalizar su perfil a través de la codificación html. Son ejemplos de secciones personalizables: “*a cerca de mi*”, “*me gustaría conocer a alguien*”, e “*intereses*”.

5.3 LinkedIn

Es una red social orientada a los negocios fundada en diciembre del 2002 y lanzada al mercado en mayo del 2003. Su principal utilidad es hacer las veces de una red de contactos profesionales detallada de las personas en las cuales confiamos para hacer negocios.

Una de las cuestiones más interesantes con las que cuenta LinkedIn, es estar basada en la teoría de los seis grados del sociólogo Duncan Watts, la cual sostiene que es posible acceder a cualquier persona del planeta en solo seis saltos. Dicha teoría continua afirmando que cada persona, entre familiares y amigos, se relaciona con al menos 100 personas. Esto significa que cualquier persona potencialmente puede acceder a 10.000 personas estableciendo contacto con los amigos de sus amigos (segundo salto) y así sucesivamente multiplicándose cada vez por 100.

En LinkedIn los contactos están clasificados en diferentes grados, siendo los primeros los de un contacto más directo.



6 Los proyectos Informáticos

Para comenzar a definir lo que es un proyecto informático, debemos partir por definir lo que es un proyecto en sí. Es en este sentido es que adhiero a la definición de proyecto como:

Un esfuerzo temporal emprendido para crear un producto o un servicio único. Así, el resultado final buscado puede diferir con la misión de la organización que la emprende, ya que el proyecto tiene determinado, específicamente, un plazo y el esfuerzo es temporal.

Wikipedia en español¹⁶, establece que existen distintos tipos de proyectos, estableciendo una primera clasificación basada en los *fines* de los mismos, diferenciado los proyectos *productivos* , de los *públicos* o sociales. Los primeros buscan generar rentabilidad económica y obtener ganancias en dinero, mientras que los segundos buscan generar un impacto sobre la calidad de vida de la población objetivo, los cuales no necesariamente se expresan en dinero.

La misma fuente establece otras clasificaciones basadas en *el contenido del proyecto* , la *organización participante* , y la *complejidad* . Y es dentro de la primera de éstas tres clasificaciones donde podemos encontrar a los *proyectos informáticos* .

16 <http://es.wikipedia.org/wiki/Proyecto> - Octubre 2008



7 Arquitectura del Sistema

La selección de la arquitectura del sistema, es un trabajo que no debe ser tratado a la ligera. Es lo que determinará el éxito o el fracaso del sistema. Se debe realizar con suma conciencia, y previendo las actividades de programación a llevar a cabo. De no ser así, cuando los sistemas crecen, se hacen imposibles de mantener y estabilizar, llegando al punto de no ser implementables.

Pero, ¿que se entiende por arquitectura de un sistema?

Desde ya adelante que es una pregunta difícil de contestar. Existen tantas explicaciones para definir que es una arquitectura, como arquitectos de software existen. Para darnos una idea de lo que el concepto plantea, la IEEE en sus FAQs responde a esta misma pregunta como: “No estamos seguros, pero una arquitectura se reconoce cuando se ve¹⁷”.

El mismo artículo, realiza un esfuerzo para definir la arquitectura como:

La arquitectura de un sistema es la organización fundamental de un sistema, plasmado en sus componentes, sus relaciones, el ambiente que lo rodea, y los principales lineamientos de su diseño y evolución.

Es decir, de ahora en más nos referiremos a la arquitectura del sistema, como la forma en la que agrupamos los paquetes de código, los mensajes que existen entre las distintas capas del sistema, lineamientos de diseño, buenas prácticas de programación, y todo aquello referente al sistema y su evolución.

¿Podemos realizar un sistema sin una arquitectura visible? Sí.

Pero estaremos librados al azar, disminuirémos nuestras posibilidades de éxito, el mantenimiento del sistema se tornará complicado, programaremos por intuición, y además de lo anterior, no estaremos siguiendo los lineamientos de una de las metodologías de desarrollo que he seleccionado para éste trabajo, ya que la misma sostiene como principio fundamental centrarse en la arquitectura.

17 <http://www.iso-architecture.org/ieee-1471/ieee-1471-faq.html> (Inglés).



8 Caso de Uso (Use Cases)

Normalmente se hace referencia a un caso de uso (UC) como un requerimiento del sistema. Es decir que existirá por lo menos un UC por requerimiento del sistema. Nos permite desglosar el comportamiento del sistema y de sus usuarios, de tal forma que tanto los responsables del negocio como el personal técnico puedan entenderlo.

En la interpretación del libro “Writing Effective Use Cases”¹⁸ el autor sostiene que:

“Un UC captura un contrato entre los interesados en el sistema y el comportamiento del mismo. El UC describe el comportamiento del sistema bajo distintas condiciones tales como las respuestas de una solicitud realizada por uno de los interesados, llamado actor principal (AP). El AP inicia una interacción con el sistema para lograr un objetivo. Y el sistema responde protegiendo los intereses de todos los interesados...”

“Un UC es fundamentalmente un formulario de texto, aunque también puede ser realizado como un digrama UML, un diagrama de flujo, un diagrama de secuencia, o lenguaje de programación. Normalmente sirve para la comunicación de una persona con otra y normalmente estas personas no tienen entrenamiento especial alguno”

Este formulario puede contener la información que se consideré valiosa de transmitir de una persona a otra. No existe un formulario estándar, cada uno los proyectos requiere su propio estereotipo, lo importante es que el mismo sea fácil de leer, y se transforme en un lenguaje común entre los interesados en el proyecto.

En este proyecto seleccione un formulario basado, en el mismo libro en el cual hago referencia anteriormente.

18 Alistar Cockburn – Addison Wesley - Inglés



9 Metodologías de Desarrollo

Una metodología de desarrollo, consiste en una serie de procesos, procedimientos, y artefactos, destinados a guiar a los miembros de un equipo para alcanzar sus objetivos en el contexto de un producto de software.

Existen distintas metodologías actualmente en el mercado, algunas con más renombre que otras, o con más fama, debido a sus creadores, o a las compañías que las utilizan. En general, y realizando un recorrido por las distintas metodologías, podemos decir que la mayoría de las mismas, se basan en ciclos de vida guiados por iteraciones, sean estas llamadas iteraciones propiamente dichas, ciclos de adaptación, o *sprints*. ¿Pero por que se ha optado por cambiar las clásicas metodologías que sugerían el desarrollo en cascada por metodologías con ciclos iterativos? Por que en el contexto en el cual vivimos, es prácticamente imposible, evitar que los clientes no “encuentren” sus negocios, o se den cuenta de las cosas que obviaron, en la etapa de análisis, al final del proceso de desarrollo. Aún cuando se haya trabajado con prototipos o cualquier otro medio para fijar los requerimientos. Es muy común que el cliente, habiendo visto el producto desarrollado, y probándolo, se dé cuenta, que existen algunas cosas que se le podrían agregar o que “sería una buena idea” tener para cumplir con las necesidades del negocio.

Existe otra diferencia con respecto a la visión clásica del desarrollo de software, y es la del *cliente* y el *negocio* como reyes en el proceso. Se ha cambiado la visión del analista de sistemas como asesor del hombre de negocio, como aquel que sabe de tecnología frente a alguien que no, por la del analista que *debe* conocer el negocio, es un aprendiz, un hombre de negocios que conoce de tecnología. Es muy común escuchar a los líderes de proyecto decir *que si no se sabe como el cliente hace plata con su negocio, no se puede comenzar a desarrollar un producto que le permita incrementar sus ganancias*. Es ésta visión o punto de vista, la de cambiar la óptica sobre los encargados de desarrollar software, de un “gran maestro” a un “gran aprendiz” lo que también se ha visto reflejado en la evolución de las metodologías de desarrollo. Al poner al cliente como centro, al permitir priorizar los requerimientos, acomodar las *produc backlog list*¹⁹ según sus necesidades, al generar *entregables* al final de cada etapa, se esta buscando aprender cada vez mas de la visión del cliente tiene sobre su negocio.

Además de lo mencionado anteriormente, es destacable comentar que existen aquellos que

19 Lista de requerimientos utilizada por metodologías ágiles, creada por los informáticos para ser priorizada por los clientes.



pregonan la no utilización de metodologías, con frases como “*gente piola realizan proyectos piolas*”. Marcando su clara orientación a las personas en contraposición a la orientación a los procesos. Una visión de héroes, mas que de comunidad, de trabajo en “equipos de cirujanos”²⁰ más que de equipos autodirigidos. Quizás influenciada por malas experiencias en trabajos con metodologías. Y es aquí donde entran en juego, la buena elección de la misma. Debemos determinar cuales son las características básicas del proyecto para después seleccionar la herramienta que más se adapte a al mismo. De otra forma, sí la elegimos, por su fama o prestigio, por que esta de moda, o por que es la única que conocemos, es muy probable que alternemos entre éxitos y fracasos, por que bajo este punto de vista serán los proyectos los que se adapten a las metodologías en lugar de ser al revés.

A continuación expondré las metodologías de desarrollo de software, que a mi entender son las que mejor se adaptan al proyecto a desarrollar.

9.1 RUP (*Rational Unified Process*)

Según el libro “El proceso Unificado de Desarrollo de Software”²¹. RUP es un proceso de desarrollo de software.

Un proceso de desarrollo de software es un conjunto de actividades, necesarias para transformar los requisitos de un usuario en un sistema de software.

Además de lo mencionado anteriormente el libro continua diciendo que RUP es mas que un simple proceso; *es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.*

Este framework de trabajo esta basado en *componentes*, lo cual quiere decir que el sistema de software que se va a construir, está formado por *componentes de software* interconectados por *interfaces* bien definidas.

Por otro lado wikipedia establece otra definición que ayuda a aclarar el concepto:

“RUP es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es en realidad un refinamiento realizado por Rational Software del más genérico Proceso Unificado.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto

20 Equipos ejecutivos, donde uno realiza las operaciones y los demás asisten a aquel que trabaja directamente.

21 Autores: Ivar Jacobson, Grady Booch, James Rumbaugh – Editorial Addison Wesley



de metodologías adaptables al contexto y necesidades de cada organización.”

Debemos tener bien presente, que RUP es solo una herramienta, solo una guía para el desarrollo de las actividades durante el proceso de construcción de un software. Para nada son reglas estrictas, para, son 100% aplicables en todos los proyecto.

9.1.1 Características Principales

Los autores de RUP (Grady Booch, Ivar Jacobson y James Rumbaugh) establecen tres características esenciales:

1. **Proceso dirigido por Casos de Uso**
2. **Proceso centrado en la arquitectura**
3. **Proceso iterativo e incremental**

Se define un *Caso de Uso* como: *un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido.*

Los casos de uso son una técnica de captura de requisitos que fuerzan a pensar en aquellas cosas que le interesa al usuario, en vez de pensar en aquello que los desarrolladores consideran interesante incluir. Los Casos de Uso representan los requisitos funcionales del sistema. Pero los UC (Use Case, Caso de Uso) no solo sirven como una herramienta para trabajar con los requerimientos del sistema. Si no que intervienen a lo largo de todo el desarrollo a manera orientadora, para todas las demás fases del proyecto, e integrando las distintas partes del trabajo. Como toda herramienta, existen distintos formatos, de la misma. Existen casos de usos que realizan una traza exhaustiva de la funcionalidad, casi hasta el punto de transformarse en pseudos código para los programadores, y existen otros que sirven como una guía de pensamiento.

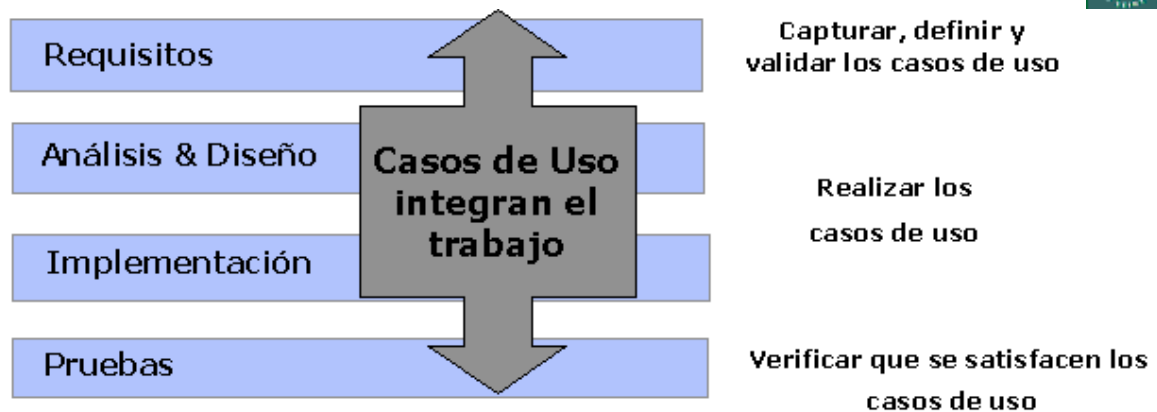


Figura 1: Los Casos de Uso integran el trabajo

La Arquitectura de un sistema no es un concepto fácil de explicar, es más, podemos observar la dificultad de la definición del concepto en una de las FAQ del estándar 1471²² de la IEEE, cuando pregunta: “entonces, ¿que es la arquitectura?” y continua respondiendo “No estamos seguros, pero reconocemos una cuando la vemos”.²³

A modo orientador podemos decir que una arquitectura de software es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo.

“Arquitectura es la estructura de los componentes más significativos de un sistema interactuando a través de interfaces con otros componentes conformados por componentes sucesivamente pequeños e interfaces.”

La arquitectura tiene que ver con los aspectos estáticos y dinámicos más significativos del sistema. Tiene que ver con las decisiones de cómo será construido el sistema y en que orden. La misma, se ve influenciada por la plataforma de software, sistema operativo, motores de bases de datos, protocolos y sistemas informáticos con los cuales se puede relacionar. Existe una fuerte relación entre los UC y la arquitectura, los primeros deben encajar en los segundos, y la arquitectura debe permitir que los UC se lleven a cabo.

El proceso Iterativo e incremental, esta dado en el orden de que un trabajo se divide en partes más pequeñas, o mini proyectos. Cada una de estas partes se puede considerar como una iteración, de la cual se obtiene un incremento en el trabajo final. Las mismas hacen referencia al flujo de trabajo y los incrementos al crecimiento del producto.

Una Iteración, puede considerarse como una cascada que va desde los requisitos para el

²² Estándar de arquitectura de software del *Institute of Electrical and Electronics Engineers (IEEE)*

²³ <http://www.iso-architecture.org/ieee-1471/ieee-1471-faq.html>

ciclo, hasta la integración con los resultados de las demás iteraciones.

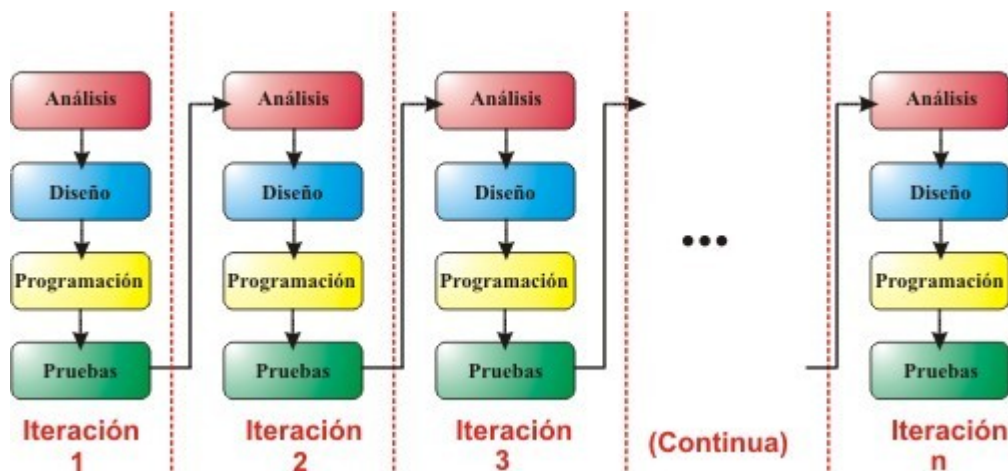


Figura 2: Iteraciones

Figura 2: Iteraciones

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones dependiendo su numero del cada proyecto en particular, y en las que se hace un mayor o menor hincapié en los distintas actividades. Como observación podemos destacar que tanto la gestión de configuración y cambios, la gestión del proyecto y el entorno se desarrollan a lo largo de todo el proyecto.

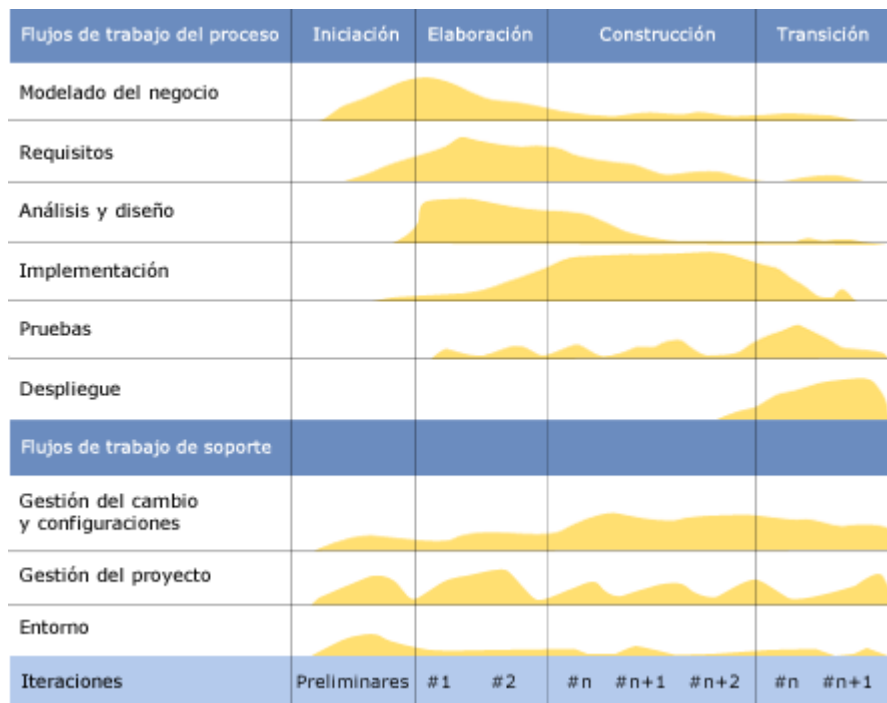


Figura 3: RUP

La idea básica de cada una de las iteraciones, y su paso por las distintas fases, es que la subsiguiente debe sumar valor. Debido a que se ha aprendido de la anterior.

Alguno los beneficios de un proceso iterativo controlado son:



1. Reduce el coste de los riesgos a un solo incremento. Es decir que si los desarrolladores tienen que repetir la iteración la organización solo pierde el esfuerzo mal empleado de la iteración y no de todo el proyecto.
2. Reduce el riesgo de no sacar al mercado el producto en los tiempos establecido por el calendario. Mediante la identificación de los riesgos en las fases tempranas del desarrollo.
3. Acelera el ritmo de los esfuerzos de desarrollo en su totalidad, debido a que los desarrolladores trabajan de manera más eficiente para obtener resultados claros a corto plazo. En lugar de tener un calendario largo.
4. Reconoce que las necesidades del usuario y sus correspondientes requisitos no pueden definirse completamente al principio.

9.2 SCRUM

Scrum es un framework de trabajo que forma parte de las llamadas “*metodologías ágiles*” de desarrollo.

“La definición moderna de desarrollo ágil de software evolucionó a mediados de los años 1990 como parte de una reacción contra los métodos de "peso pesado", muy estructurados y estrictos, extraídos del modelo de desarrollo en cascada...”²⁴

Como parte de las metodologías ágiles, Scrum hace referencia al trabajo coordinado y minimalista entre los distintos miembros del equipo. Además de ello, cuenta con una gran cantidad de “*artefactos*” para el trabajo diario en equipo. Es parte del espíritu de esta metodología, en contraposición con otras, festejar los cambios a realizar. Aunque suene muy difícil de entender, esta busca instalar en la mente del equipo que cada cambio es un acercamiento a lograr el producto final. ¿Pero que es SCRUM? Adhiero firmemente a la definición que expone Proyectoagiles.com:

Scrum es un proceso en el que se aplican de manera regular un conjunto de mejores prácticas para trabajar en equipo y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

²⁴ Fuente: Wikipedia: http://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_software

Otra definición a la cual adhiero es la “Scrum Alliance”²⁵ organismo que brinda la certificación a nivel internacional. Esta definición nos va dando la idea de los artefactos que Scrum utiliza y de los rituales necesarios para trabajar con este framework.

(Interpretación del Inglés)

Scrum es un framework de desarrollo ágil de software. El trabajo es estructurado en ciclos de trabajo llamados “Sprints” e interacciones de trabajo que típicamente tienen de dos a cuatro semanas de duración. Durante cada Sprint los equipos van trabajando con los requerimientos impuestos por el cliente volcados en una lista de prioridades llamada “backlog list”, dichos requerimientos o Historias de Usuarios, (User Stories) se van desarrollando de acuerdo al valor que el cliente le asigna. Al final de cada Sprint, el producto ha adquirido una mejora comprobable.

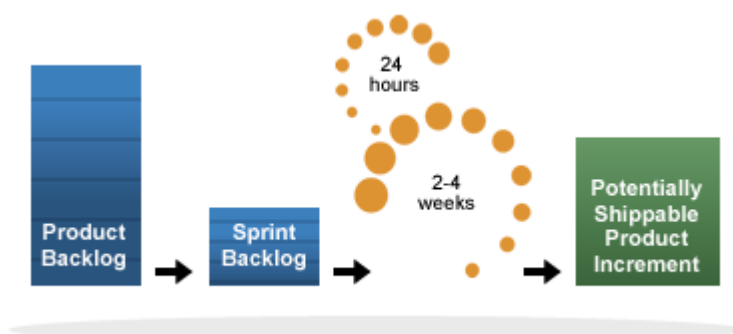


Figura 4: Ciclo de Vida de Scrum

9.2.1 SCRUM es un framework simple

Aunque existen varios refinamientos de la metodología, y en los últimos años se han agregado artefactos que dan soporte a los principales, Scrum es un framework simple. Ya que su espíritu descansa sobre tres roles, tres ceremonias y tres artefactos.

Los Roles

Los mismos se dividen a su vez en dos categorías: roles cerdo y roles gallinas. Los nombres de los mismos se inspiran en el chiste que a continuación se expone:

Un cerdo y una gallina se encuentran en la calle. La gallina mira al cerdo y dice: "Hey, ¿por qué no abrimos un restaurante?" El cerdo mira a la gallina y le dice: "Buena idea, ¿cómo se llamaría el restaurante?" La gallina piensa un poco y contesta: "¿Por qué no lo llamamos "Huevos con jamón?" "Lo siento pero no", dice el cerdo, "Yo estaría

²⁵ Fuente: Scrum Alliance (interpretación del Inglés) http://www.scrumalliance.org/pages/what_is_scrum



comprometido pero tú solamente estarías involucrada".

Pero debemos destacar que son los **roles cerdos** los que hacen al espíritu de SCRUM. Ellos son;

- El *Product Owner* o *Dueño del Producto* quien es el responsable del valor de negocio que tiene el proyecto. Es aquel que se encarga de priorizar los requerimientos que le aportaran valor al negocio. Necesariamente tiene que representar al cliente.
- El *scrumMaster* (*nótese la sintaxis particular de camel case*): es aquel que se encarga de que el equipo sea funcional y productivo. Tiene por principal tarea, eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint. No es un líder de equipo, ya que los equipos son auto-organizados. Si no que actúa como una protección ante cualquier influencia que los distraiga. Es el que hace que las reglas se cumplan.
- El *Equipo auto-organizado*: El equipo tiene la responsabilidad de entregar el producto. Un pequeño equipo de 5 a 9 personas con las habilidades transversales necesarias para realizar el trabajo (diseñador, desarrollador, etc).

Los **roles gallina** hacen referencia a aquellas personas interesadas en el proyecto y expertos en el negocio. Los mismos son: los *usuarios*, *Stakeholders* (clientes, proveedores, etc) y los *managers* o personas que establecen el ambiente para el desarrollo.

Las Ceremonias:

Son tres las principales reuniones que tiene Scrum. No por ello se deben descartar las reuniones ad-hoc que surgen del trabajo diario, ni de los cambios de requerimientos que el cliente plantee llamados comúnmente *requirements churn*. Ya que la metodología adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido, y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes. Ellas son:

- *La reunión de planificación del Sprint*: Es la reunión para organizar el próximo Sprint. En ella el *Product Owner* (PO) revisa la visión, el mapa de ruta, el plan de desarrollo y la *product backlog list* (PBL). Mientras tanto el equipo revisa las estimaciones para cada una de las características a implementar en la PBL y confirman o no que dichas características son desarrollables y alcanzables en el tiempo establecido. El equipo será el que determine cuanto trabajo será necesario para alcanzar dicha funcionalidad en base al tamaño del equipo, las horas disponibles, en nivel de productividad del equipo. Es



importante que el equipo extraiga de de las PBL las tareas que son alcanzables en el Sprint siguiente. Una vez que esto sucede, el scrumMaster liderará al equipo en “despedazar” esas funcionalidades en pequeñas tareas a realizar. Esta reunión tiene una duración máxima de cuatro horas, y tiene por resultado el sprint backlog.

- **Reunión Diaria:** en esta reunión la metodología es bastante prescriptiva. Sostiene que no tiene que ser más de 15 minutos, y que todos deben estar parados²⁶ para que la misma no se prolongue. En ella cada miembro del equipo debe responder 3 preguntas básicas: ¿Qué hice ayer?, ¿Qué voy a hacer hoy?, ¿Cuáles son los impedimentos que tengo en mi camino? Mientras que cualquiera puede asistir a estas reuniones, solo aquellos miembros que están comprometidos con la entrega del producto pueden hablar (Roles cerdo). Los objetivos, son tener una visión global del proyecto, descubrir alguna nueva dependencia, dirigir cualquier necesidad de compromiso personal, y ajustar cualquier plan de trabajo en tiempo real.
- **Reunión de Revisión:** Es una reunión con un límite temporal de 4 horas. La primera mitad del tiempo se basa en hacer una demostración de aquellas funcionalidades que fueron implementadas durante el sprint pasado. El dueño del producto es el encargado de liderar esta reunión y debe invitar a todos los interesados a participar. Se revisan el negocio, el mercado, y la tecnología. Se definen los objetivos para el siguiente sprint. La segunda mitad de la reunión le corresponde al equipo, en ella se realiza una retrospectiva del trabajo en equipo y se identifican las mejores prácticas, sin echar culpas a nadie, solo las mejores prácticas.

Los Artefactos:

Se utilizan para el seguimiento y priorización de las tareas a realizar.

- **Product Backlog:** cuando comienza el proyecto el dueño del producto prepara una lista con los requerimientos del sistema priorizados por el valor que los mismos le aportan al negocio. El equipo contribuye asistiendo al dueño del producto con estimaciones de cuanto costara desarrollar cada funcionalidad. Esta lista debe contener tanto los requerimientos visibles para el cliente, como los requerimientos técnicos necesarios para alcanzarlos.
- **Sprint Backlog:** mientras que la PB contiene un listado de las características a

26 Nota del Autor: Generalmente a esta reunión se la llama *standup (parado)daily*. Es más casi podríamos decir que la metodología obtiene su nombre de estas reuniones, al encontrarse todos parados frente a una pizarra los equipos conforman un semicírculo similar a los scrums de rugby.

implementar, la SB contiene un listado de las *tareas* necesarias para alcanzar esas características en el siguiente Sprint. A su vez estas tareas son despedazadas en mini tareas que requieran menos de dos días de desarrollo (16 hs. de desarrollo). Cuando SB es completada el total de horas debe coincidir con las horas de estimación de las funcionalidades descritas en la PB. Si la diferencia es considerable, se deberá negociar con el dueño del producto para incrementar las probabilidades de éxito.

- Burndown Chart: muestra el trabajo planificado en contraposición con el real.

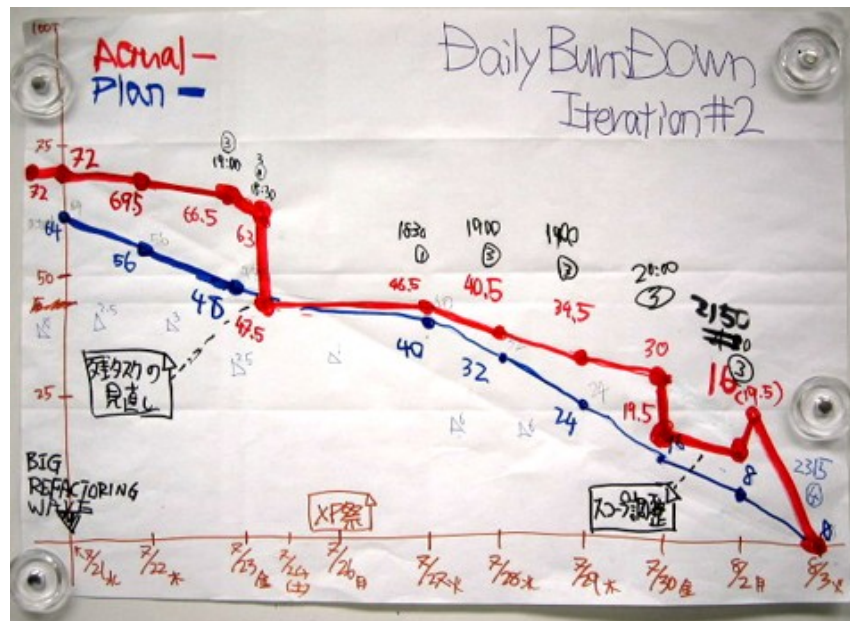


Figura 5: Ejemplo de Burndown Chart

En el eje de las abscisas se medirá los días del Sprint y en el eje de las ordenadas se mide el trabajo restante, el mismo puede ser en términos de tiempo o cualquier otra unidad de medida de tamaño con la cual trabaje la empresa. La idea de este grafico es determinar “cuanto nos falta para terminar”. También es común encontrar incluidos en estos gráficos métricas del promedio de trabajo del equipo.

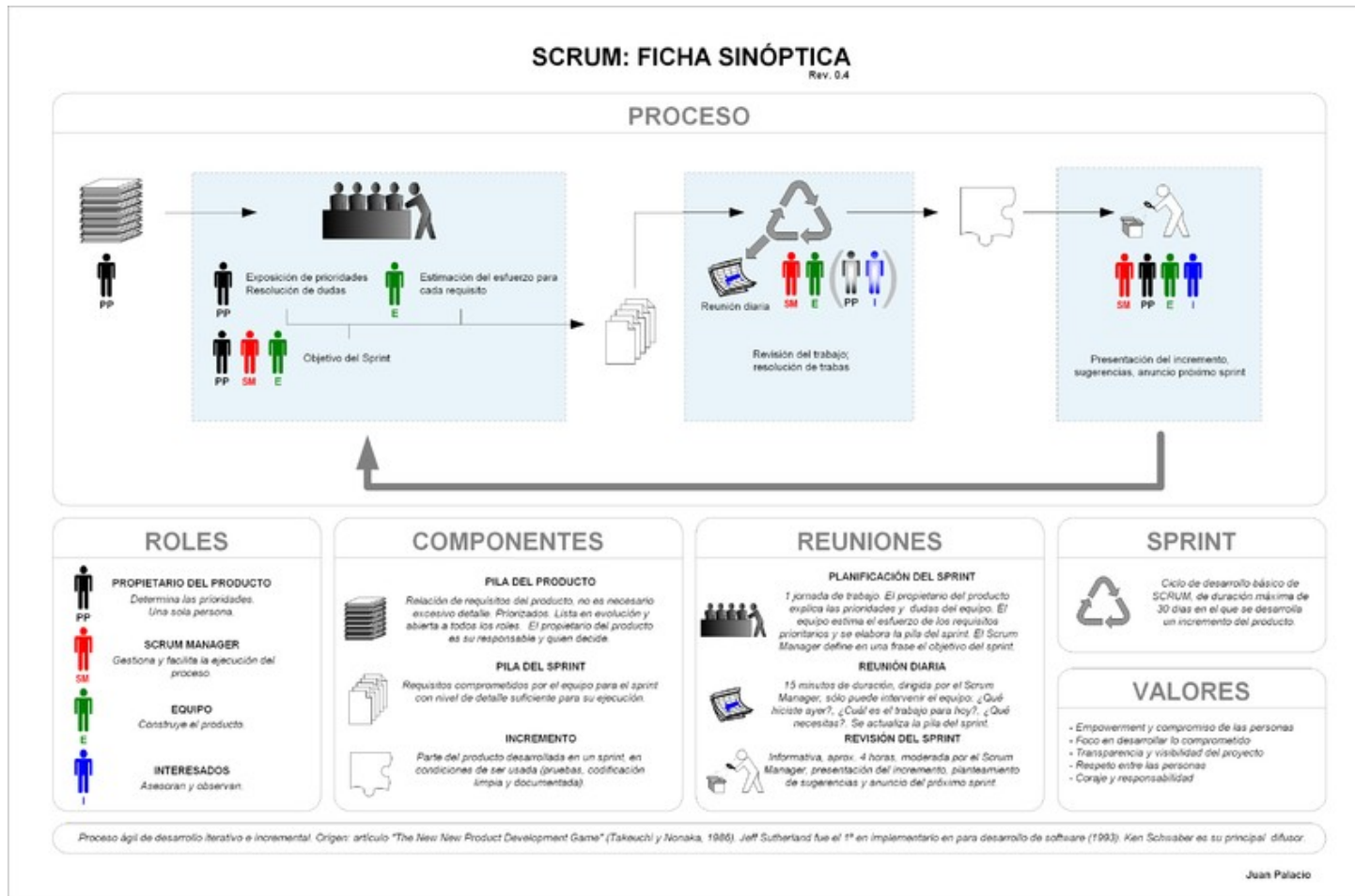


Figura 6: Ficha sumaria de SRUM



10 Lenguaje de Modelado Unificado

10.1 Consideraciones

UML por sus siglas en inglés de *Unified Modeling Language*, es uno de los lenguajes de modelado de software más utilizados en el mundo. Quizás por el prestigio de sus creadores (Grady Booch, Ivar Jacobson y Jim Rumbaugh), quizás por que incorpora las mejores características de cada una de las creaciones individuales de estos.

Wikipedia²⁷ adhiere a la definición:

“Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software”

Debemos destacar que se trata de un lenguaje y no de un método, se utiliza para definir un sistema de software, para detallar los entregables, documentar y construir.

UML es el complemento ideal para seguir a RUP por las notaciones estandarizadas y su colaboración al “proceso guiado por casos de uso”.

Este lenguaje propone modelar un sistema a partir de diferentes vistas del mismo. Cada modelo que compone este lenguaje representa al software desde una perspectiva específica.

Los Modelos de UML con los que se trabaja comúnmente son²⁸:

1. **Diagramas de Estructura Estática**
2. **Diagramas de Casos de Uso**
3. **Diagramas de Secuencia**
4. **Diagramas de Colaboración**
5. **Diagramas de Estado**
6. **Diagramas de Paquetes**

Al no ser este un proceso definido, cada sistema en particular, o cada arquitecto de software en particular, emplean cada uno de los modelos en la forma en las que considere conveniente. Es decir que existirán sistemas que contengan todos los diagramas y existirán otros que contengan solo algunos.

²⁷ Fuente: http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado

²⁸ Bibliografía: *El Lenguaje Unificado de Modelado* - G. Booch, J. Rumbaugh, I. Jacobson. - Addison Wesley Iberoamericana, 1999.

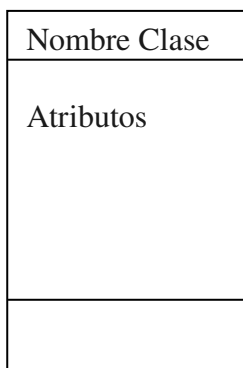


10.2 Diagramas de Estructura Estática

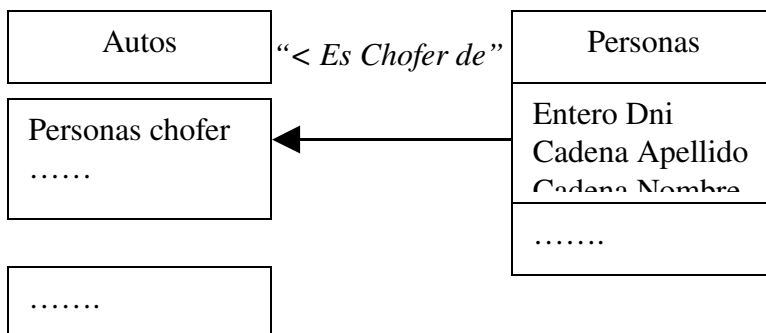
El diagrama de estructura estática, esta compuesto por el Modelo Conceptual de la Fase de Diseño de Alto Nivel y el Diagrama de Clases de Diseño. El primero de ellos modela elementos del dominio mientras que el segundo presenta elementos de la solución de software.

¿Pero que es una clase? Es un conjunto de objetos... ¿no muy claro verdad? En el mundo real existen entidades físicas y lógicas. Por ejemplo un auto es una entidad física. Este auto tiene un color, una marca, un fabricante, etc. Todos estos son atributos de un objeto llamado auto. A su vez este auto realiza acciones, como por eje., marchar, frenar, doblar una esquina, etc, estos son los comportamientos del objeto auto. Bueno la clase “Autos”, es el conjunto de autos.

Las clases se simbolizan con:



A su vez las clases se relacionan entre si en un diagrama de clases. Por ejemplo la clase Personas se relaciona con la clase Autos con a través de la asociación “es chofer de”



Estas relaciones pueden ser de varios tipos: relaciones de uso, relaciones de especialización y asociaciones. En la siguiente figura veremos como las distintas relaciones interactúan entre



si, conjuntamente con el nombre teórico de las mismas. Y la posterior explicación de la semántica de dichas relaciones.

Ejemplo 1 - Ariel Diaz Molina

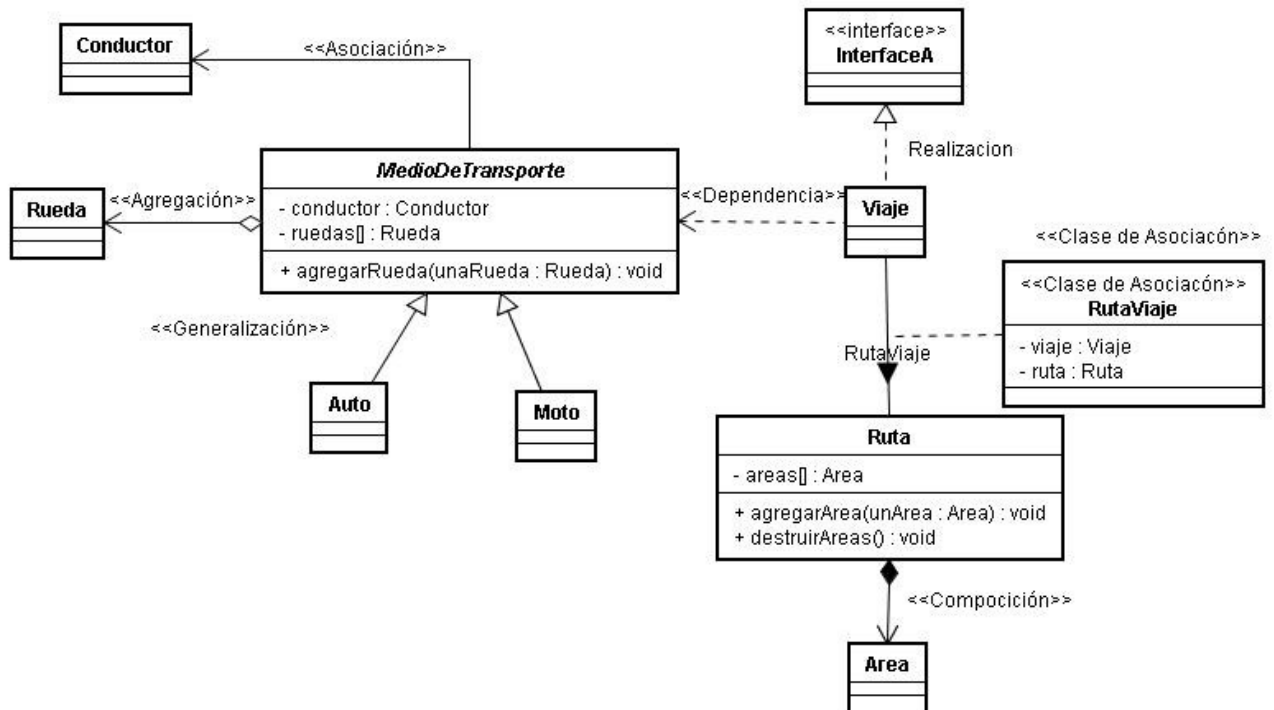


Figura 7: Ejemplo de Diagrama de Clases

10.2.1 Asociación:

Es una relación estructural entre clases, es decir que una clase contiene un atributo de la otra, en nuestro ejemplo podemos leer “*un medio de transporte tiene un conductor*”. La navegabilidad, (la flecha ->) nos muestra donde esta ubicado el atributo. Dicha navegabilidad es la forma en la que podemos leer y trasladarnos desde una clase a otra. Además de la navegabilidad, podemos encontrar la *multiplicidad* de las relaciones, es decir cuantos objetos de una clase contiene otra.

10.2.2 Agregación

Es una relación que se derivó de la asociación, por ser igualmente estructural, es decir que contiene un atributo, que en todos los casos, será una colección, es decir un array, vector, etc, y además de ello la clase que contiene la colección debe tener un método que agregue los elementos a la colección. En nuestro ejemplo podemos leer como “*un medio de*



transporte contiene una colección de rudas” y además de ello contiene un comportamiento para agregarlas.

Nos está diciendo que los objetos rueda forman parte del objeto medio de transporte. Pero, su ciclo de vida no está atado al del objeto medio de transporte. Es decir si el automóvil se destruye las ruedas pueden seguir existiendo independientemente. Realizando una analogía con la vida real, *si el automóvil tiene un accidente y se destruye (pierde su forma) las ruedas puede seguir existiendo para ser utilizadas en otros.*

10.2.3 Composición

Al igual que en la agregación, es una relación estructural pero a la misma se le suma, que tiene un método de destrucción de los objetos. Y a diferencia de la asociación, el ciclo de vida del objeto *Área* está relacionado con el del objeto *ruta*. Es decir que *si la ruta de viaje se levanta, las áreas que surgían a partir de ella desaparecen*. También se puede leer como que una ruta tiene varias áreas de cobertura.

Mucho se ha discutido a cerca de las agregaciones y las composiciones, el debate es casi tan intenso como el de los *include* y *extends* de los casos de uso. Un punto de vista interesante sobre éste debate, proviene de algunas corrientes de pensamiento que sostienen que los lenguajes orientados a objetos que contienen un *garbage collector*²⁹, no necesitan de un método de destrucción de objetos en los códigos escritos (relacionados a los ciclos de vida de una composición). Convirtiendo a la composición una agregación más. Pero con el avance las tecnologías de arquitectura, surgieron una nueva responsabilidad para la composición, de carácter meramente técnico, es la responsabilidad de los “repositorios” a la hora de crear y destruir los objetos en arquitecturas orientadas a servicios. Estudio que excede al carácter introducción y explicación de este apartado.

10.2.4 Clase de Asociación

Es una Clase que surge de una multiplicidad de muchos a muchos, y fue incorporada en UML para dar soporte a este caso. Se sacan los atributos de las clases involucradas y se los incorpora a una clase a parte. Al igual que las anteriores hace referencia a una relación estructural. En el ejemplo son los objetos *viaje* y *ruta*

²⁹ Mecanismo de los lenguajes de programación que se ocupan de la liberación de memoria y destrucción de objetos que ya no son utilizados.

10.2.5 Diagramas de Caso de Uso

Este es quizás el más importante de los diagramas de UML. Siempre que veamos un modelado de este tipo vamos a encontrar diagramas de casos de uso. Por que el diagrama de casos de uso representa lo que el sistema hace, es decir sus funcionalidades.

En el podemos ver como se desarrolla una función del sistema y la persona encargada de llevarla a cabo.

En este proyecto adhiero, totalmente a la interpretación que realiza Alistair Cockbur³⁰ de cómo debe modelarse e interpretarse un Caso de Uso (Use Case, UC). Las interpretaciones han variado de autor en autor, y quizás el punto más álgido de las discusiones sea la semántica de las relaciones existentes en los UC

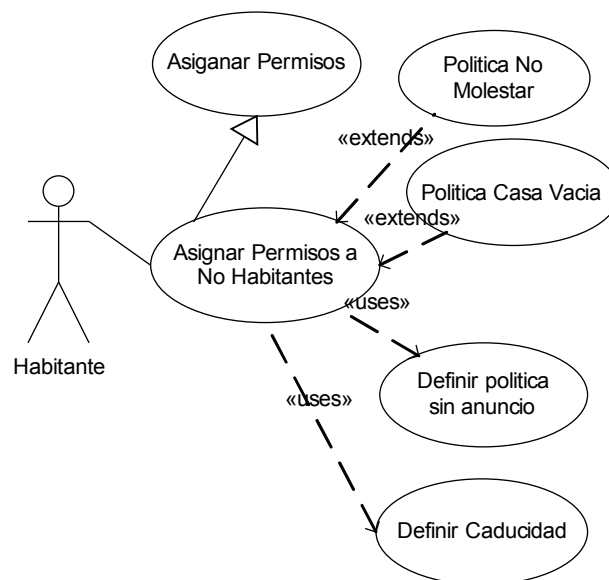


Figura 8: Caso de Uso

En el ejemplo anterior se ve como el usuario habitante se relaciona con el sistema para asignar permisos a una persona que no es habitante. Además vemos que existen tres tipos de relaciones entre los UC:

Include: es simplemente un paso que la persona debe realizar para llevar a cabo la tarea³¹. El caso de uso incluido representa un objetivo de menor nivel que el caso de Uso base, de donde parte la flecha.

30 Bibliografía: “Writing Effective Use Cases” - Alistair Cockbur

31 “Writing Effective Use Cases”- UML’s Include Relation: pagina 227



Extends: una relación de este tipo define que el caso de uso que extiende al UC base (el UC del cual parte la flecha) interrumpe al UC base bajo alguna circunstancia. El caso de uso base se ejecuta, es decir que especifica una secuencia de acciones, hasta que se da una determinada situación, a partir de la cual se empieza a ejecutar el caso de uso extendido, para luego continuar con la ejecución del base. Se puede definir dos situaciones claramente marcadas de cuando utilizar una extensión:

1. Naturalmente cuando existe un escenario alternativo que puede interrumpir el caso de uso base y por su importancia merezca ser sacado a fuera del caso de uso base.
2. Cuando estamos modelando incrementos de funcionalidades a un documento de requerimientos ya cerrado. Es decir que le estamos agregando funcionalidad a un caso de uso ya definido. Algunos dicen que la extensión es un parche del UC base.

Pero además de las mencionadas relaciones, se puede agregar relaciones de herencia, es decir relaciones que se piensan como... “es un tipo de...”. En mi ejemplo, “Asignar permisos a no habitantes” es un tipo de “Asignar Permisos”.

10.2.6 Diagrama de Secuencia

Modela las Interacciones que existen entre los objetos en un sistema. Mientras que el Diagrama de UC nos muestra detalles del modelado de “negocios” de un sistema el diagrama de secuencia nos muestra detalles de cómo implementar ese escenario que el UC plantea, incluyendo los objetos y clases que intervienen junto con los mensajes pasados entre ellos.

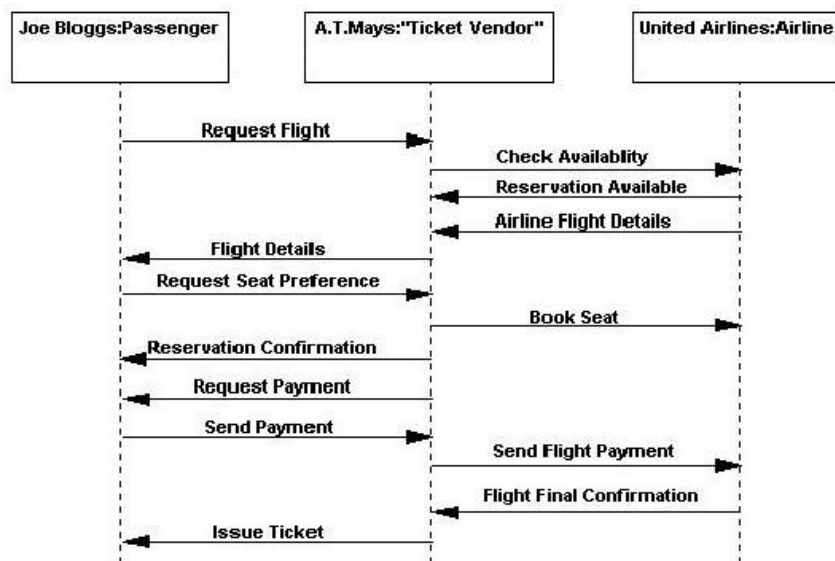


Figura 9: Diagrama de Secuencia

En el grafico anterior se muestran las intervenciones que existen entre un pasajero que va a comprar un pasaje, el vendedor y la aerolínea seleccionada.

Debemos destacar las interacciones entre los objetos están ordenadas en base a una secuencia cronológica.

10.2.7 Diagrama de Colaboración

Representa la alternativa al diagrama de secuencia para modelar interacciones entre objetos. Mientras que el diagrama de secuencias hace inca pie en la secuencia cronológica que estamos modelando, el diagrama de colaboración se concentra en modelar los efectos de un objeto determinado en un escenario planteado.

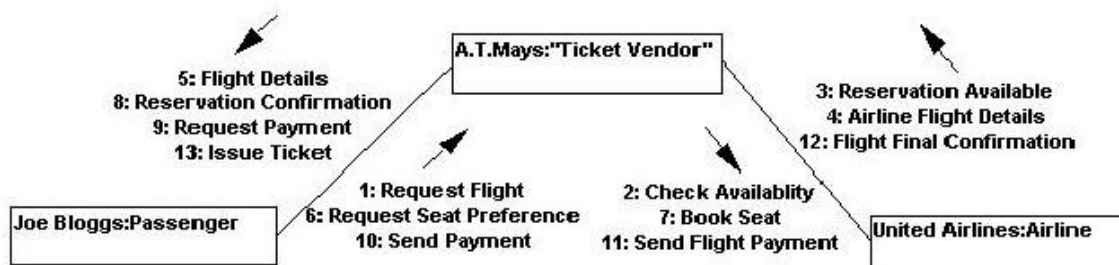


Figura 10: Diagrama de Colaboración

Para continuar con el ejemplo anterior del pasajero, el vendedor y la aerolínea.

10.2.8 Diagrama de Estado

Sirve para modelar el comportamiento en tiempo real de cada clase que tiene un comportamiento dinámico y significativo. Se puede usar también un diagrama de actividad, como una extensión del diagrama de estado, para mostrar detalles de las acciones realizadas por los objetos en respuesta a determinados eventos.

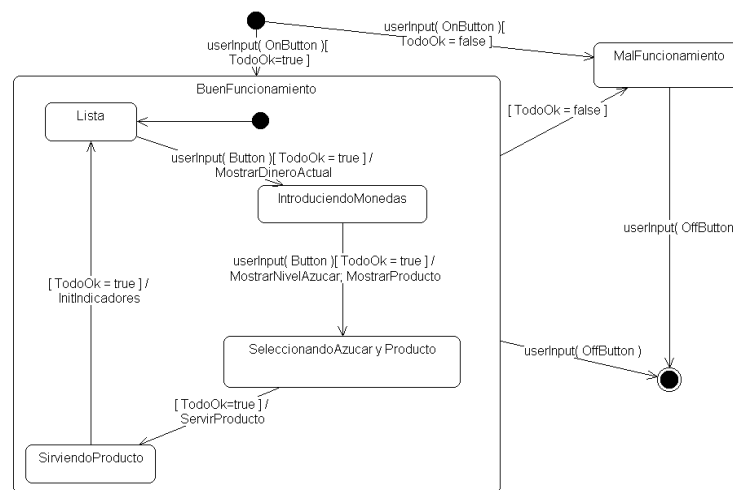


Figura 11: Diagrama de Estado

El anterior es un ejemplo descriptivo del diagrama de estado de una maquina de café. En el ve se todas las tareas necesarias para pasar de un estado inicial a un final.

10.2.9 Diagrama de Paquetes

Los paquetes son las unidades que conforman los sistemas. Cada sistema de tamaño considerable debe subdividirse en paquetes. Esta subdivisión responde a criterios tales como funcionalidad común, implementación relacionada, punto de vista común, etc.

UML no impone una regla para componer los paquetes, eso queda librado a cada diseño en particular.

Un diagrama de paquetes muestra como un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema.³²

32 Fuente: Wikipedia



10.2.10 Conclusión

En la etapa de iniciación de un proyecto se establece que los diagramas a utilizar son los de UC y los de Clases. Ambos en etapas primarias, con el objetivo de ir determinando lo que el sistema deberá hacer, y como podremos diseñar la solución

UML además brinda especificaciones para poder realizar comentarios que permitan entender con mayor facilidad la semántica de lo que graficamos. Ya que a pesar de ser un lenguaje profundamente difundido, el uso que se realiza de el, ya se por falta de formación, o por la diversidad que los diferente autores le asignan a su semántica, no siempre coincide.

Este lenguaje es una herramienta poderosa para comunicar el modelado de un sistema, pero no se debe perder la óptica de que es un lenguaje para comunicarse con personas, no con sistemas, ni herramientas CASE, por lo tanto, el adecuado y preciso uso que se haga de el dependerá de la persona que modela el sistema



INICIALIZACIÓN

- Project Charter
- Declaración de Alcance (Scope Statement)
- Historia Objetivo
- Requerimientos del Sistema
- Decisiones Estratégicas
- Estimaciones



11 Acta inicialización del proyecto (Project Charter)

El “Project Charter” es un documento que tiene por finalidad dar un lanzamiento formal al proyecto. En el se resumen tanto las características más importantes del trabajo a realizar, como así también las personas involucradas en este proyecto.

Acta de Inicialización – Project Charter

Este documento tiene por finalidad tanto notificar formalmente la inicialización del proyecto, como brindar la información global a los miembros que intervienen.

Información del proyecto

Información básica del proyecto

Autorización	Calixto Maldonado
Project Manager	Diaz Molina, Ariel
Fecha de Inicio	Febrero del 2009
Duración	5 Meses
Objetivo	Desarrollar un servicio de red social que de soporte a las principales actividades de una universidad. La aplicación resultante debe suministrar las herramientas básicas para responder tres de las principales preguntas que los estudiantes se realizar: ¿Quién me puede ayudar? ¿De donde estudiaste? ¿Qué te tomaron?

Interesados (Stake Holders)

Son aquellas personas que tienen o pueden tener un interés en que el proyecto se lleve a cabo.

Cliente Potencial	Universidades. Publicidad dirigida
Espónsor	Diaz Molina, Ariel
Team	- Director del Proyecto - Profesores-Asesores
Otros	- Miembros de la Universidad en General - Posibles inversores

Restricciones

Son todas aquellas restricciones tanto presupuestarias como de recursos humanos con las cuales cuenta el proyecto de antemano. Se refiere a gastos y recursos negados.

	El siguiente proyecto no cuenta con restricción presupuestaria alguna. Los gasto de movilización, comunicación, coordinación, y desarrollo corren por cuenta del responsable del proyecto.
--	--

Prioridades

Es la priorización de las restricciones con las que cuenta todos los proyectos. Alcance, tiempo, y recursos. Es decir aquello que se priorizará en detrimento de otro a los fines del alcanzar los objetivos



	Se debe tener en cuenta que el ALCANCE del proyecto debe ser considerado prioritario. Luego se tendrá en cuenta el tiempo y finalmente el costo.
--	--

Presupuesto estimado

Presupuesto aprobado para el proyecto.

	El siguiente proyecto está libre de presupuesto, por tratarse de un trabajo final de graduación. Los gastos en los cuales se incurra estarán a cargo del líder de proyecto.
--	---



12 Declaración de Alcance (Scope Statement)

El propósito de este documento es proveer un vehículo para documentar los esfuerzos de planificación iniciales. Se utiliza para lograr un nivel de aceptación mutua, entre los distintos interesados en el proyecto, respecto a los objetivos y el alcance del proyecto, antes de que se incurra en esfuerzos significativos

Nombre del Proyecto:	Red social Universitaria
Desarrollado Por:	Diaz Molina Jorge Ariel
Fecha inicio (DD/MM/AAAA):	Febrero del 2009

12.11. Sumario Ejecutivo

Breve descripción del Proyecto

El proyecto tiene como propósito principal, prestar soporte informático a un servicio de red social especializado. Se busca generar un habiente de cooperación entre los miembros de una universidad.

Esta información puede estar disponible en el Project Charter. También se provee un hipervínculo al Project Charter como referencia.

[Project Charter](#)

Tutor que Aprueba: Calixto Maldonado

Project Manager: Diaz Molina Ariel

Fecha Inicio: Febrero del 2009

Cliente: Universidades, empresas con necesidades de publicidad focalizada

Prioridad: Alcance del Proyecto

Nota: *en cualquier instancia donde existiese una discrepancia entre el Project Charter y el Project Scope Statement, la fechada a posterior, será considerada como valida.*

12.2 Objetivos

12.2.1 Objetivo General (Solución):

El siguiente trabajo busca establecer algunas bases teóricas, prácticas y metodológicas para analizar, diseñar, desarrollar e implementar una herramienta que permita dar soporte a las tareas y actividades, que desarrollan los distintos actores de una universidad al interactuar entre sí. Como ejemplo de desarrollo se seguirá las actividades de los principales miembros de la universidad siglo 21 durante el primer semestre del año 2009.

Al concluir el proyecto se espera contar con una herramienta informática o de software, para ser implementada y probada, capaz de dar soporte a las actividades de mensajería interna entre los distintos miembros, compartir material de estudio.

Además de lo mencionado anteriormente, se espera contar con la documentación de procedimientos que permita la puesta en marcha, su utilización, el posterior mantenimiento y escalabilidad de la herramienta.

12.2.2 Objetivos Particulares

- Permitir a un alumno conocer a otros alumnos que asisten a su universidad.
- Permitir a un alumno compartir material de estudio.
- Permitir a un alumno conocer novedades sobre una materia en particular.



- Permitir al alumno conocer novedades de las actividades de sus compañeros.
- Permitir al alumno declarar su perfil y ver el perfil de los demás compañeros.

12.2.3 Historia Objetivo

Es sábado a la noche. Carlos es alumno universitario y está preocupado por que tiene que rendir el primero de sus exámenes finales. Pero no ha podido recolectar los apuntes necesarios para estudiar. Y el compañero con el cual ha hablado por teléfono se encuentra en la misma situación que el. Y para peor, no encuentra la punta del ovillo para comenzar a estudiar la materia, hay temas que no los entiende, y durante el cursado no ha logrado interpretar lo que el profesor espera de el.

Casi frustrado entra a Internet para tratar de despejar un poco la mente de su preocupación, y chequeado sus correos encuentra la invitación que le ha enviado su compañero para formar parte de la red social de su universidad. Realiza un clic en el link de su mail de invitación y mientras la barra de progreso se va cargando, piensa que se va encontrar con una página de suscripción atestada de campos para rellenar, y con la preocupación que tiene en la cabeza ¡no puede perder tanto tiempo!

Pero para su sorpresa, encuentra una página de suscripción bastante simple, solo le solicita un email, la contraseña, donde y que estudia. El sistema lo lleva directamente a una página de inicio donde puede encontrar un menú que hace referencia a los apuntes de un compañero de clase y al lado de su foto hay un link que dice "apuntes de clases". Con un solo clic ¡ha tenido acceso a los apuntes de clases! Pero la preocupación no ha mermado, el tiempo apremia y todavía no sabe por donde comenzar. El sistema también le muestra un listado de los alumnos que ya han cursado la materia que tanto preocupa a Carlos, con un link que dice "Realizar Pregunta".

12.3 Descripción del Proyecto

Para ítem abajo se provee suficiente detalle para definir este proyecto adecuadamente:

12.3.1 Alcance del Proyecto

Incluye del entregables:

- Código Fuente
- Manual de Procedimientos
- Capacitación
- Documentación del Proyecto
- Presentación Final

No incluye:

- Management, capacitación o codificación que sobrepasen la finalidad de esta trabajo final

12.3.2 Dependencias Externas:

- Actualizaciones tanto de los ambientes de desarrollo, como de las herramientas necesarias para el mismo.

12.3.3 Se asume que:

- Las reuniones de control del proyecto serán realizadas en tiempo y forma
- El alcance del proyecto no tendrá variaciones mayores al 40% del aprobado en la primera etapa (cursado)
- Existirá claridad en las responsabilidades de solicitudes de cambio, siendo en la primera etapa el tutor de la materia. Y a posterioridad se le sumarán los miembros de la CAE



12.3.4 Limitaciones:

- No existen limitaciones, mas allá de las explícitamente declaradas en el reglamento de la materia Seminario Final

12.4 Hitos del Proyecto

Calendario Estimativo, Lista de Hitos del Proyecto

Hitos	Fecha Objetivo (DD/MM/AAAA)
• Inicio del Proyecto	
• Requerimientos del Proyecto	
• Diseño de soluciones del proyecto	
• Documentación y software del proyecto	
• Prueba	
• Proyecto Completo	

2 Reuniones del Proyecto

2.1 Reuniones de Reportes

Reporte	Propósito	Frecuencia
Semanal (Weekly meeting)	Brindar los avances en el proyecto durante el periodo de cursado de la materia. Además de ello esta reunión tiene por finalidad tanto capturar todos los cambios solicitados por el tutor, como así también asimilar la guía cognitiva que el brindará.	Semanal (convenir) – Por defecto Martes 09:00 a 10:00 AM
Ad hoc	Esta reunión tiene por finalidad, prestar asistencia inmediata a cambios solicitados. Se crea a razón de una finalidad que surja a partir de la actividad diaria y que no pueda esperar para ser considerada en la reunión semanal	Disparada por solicitud del profesor. O su aprobación para llevarse a cabo
Coloquios	Reuniones con los miembros de la CAE para sugerir y controlar cambios en el trabajo final de graduación.	A determinar

A continuación se describe brevemente como serán manejados cada uno de los ítems descriptos. También se proveen links a los documentos relevantes



2 Reuniones del Proyecto	
2.2 Administración de Publicaciones	<ul style="list-style-type: none"> • La publicación por defecto será el proyecto impreso que formará parte de la tesis de Grado y cualquier otro documento adjunto que forme parte de la totalidad del Proyecto. • Existirá complementariamente una versión digital en un documento de extensión .doc o .pdf y cuyo nombre estará formado por la conjunción "TrabajoFinalv"+el numero de versión. Siguiendo este numero de versión, el formato ([entrega cae] .[entrega tutor] .[cambios realizados]) • Las presentaciones al tutor de la carrera se realizaran en formato de borrador (draft). Dejando cuestiones de estética y cosmética de presentación, para la entrega final o definitiva. • El trabajo será entregado en forma impresa cuando así lo requiera específicamente (sea oral o escrito), el tutor o, persona responsable, en todos los demás casos se enviara vía mail al destinatario con copia adjunta al administrador de proyectos. • La entrega de más de una copia escrita debe ser solicitada por el tutor en el período que dure el cursado de la materia. • Tanto el tutor como los miembros de la CAE se encuentran autorizados a distribuir las copias digitales del trabajo final que consideren pertinentes para el mejor desempeño de sus tareas.
2.3 Administración de Cambios	<ul style="list-style-type: none"> • El procedimiento de control de cambios consistirá en la inscripción en la plantilla de los cambios sugeridos de, ideas, solicitudes, u observaciones, consideradas por las personas autorizadas a sugerir cambios. • La plantilla de cambios sugeridos deberá contener la fecha en la que se sugieren los cambios, la versión sobre la cual se deben realizar, el detalle de lo sugerido, el rol de la persona que sugiere los cambios, y la firma. • La presentación de la misma, deberá ser en todos los casos impresa. Exceptuando aquellos, en los que por cualquier motivo, no se pudiera concretar la reunión a sus efectos. • El estudio del efecto de los cambios sugeridos en el marco del proyecto en su conjunto deberá ser notificado a la persona que lo solicito en un plazo no superior a una semana, vía mail. • Cualquier cambio sugerido deberá implicar el estudio global del proyecto a los fines de mantener la consistencia del mismo
2.4 Administración de Comunicaciones	<p>La siguiente estrategia fue establecida a los fines de promover la efectiva comunicación durante y sobre el proyecto.</p> <ul style="list-style-type: none"> • El Administrador de proyectos presentara un informe semanal de avance de proyecto, el cual constara de una plantilla, donde se haga referencia al código del cambios sugerido, la versión que se esta cambiando, las actividades que se realizaron y el esfuerzo requerido. • La presentación del Informe de cambios debe ser en todos los casos impresa. Con la excepción presentada por el caso en que, por cualquier motivo, no se pudiera concretar la reunión ad hoc. • Una vez realizada las release de producción el líder de proyecto permanecerá en la modalidad "on call" por dos días. Lo que significa que realizara una guardia de 48 hs para responder a cualquier inquietud de quien lo solicite.



2 Reuniones del Proyecto	
2.5 Administración de Logros	El procedimiento de administración de logros consistirá en la confección de la planilla de historial de aprobaciones, en la cual constara el ítem, sección, entregable, o cualquier aspecto que necesite de aprobación, la versión, la fecha el nombre y rol del responsable de aprobaciones, y la firma. La PHA en todos los casos debe ser impresa y presentada a los efectos.
2.6 Administración de Recursos	La administración de los recursos con los cuales se llevará a cabo el proyecto quedará bajo la absoluta responsabilidad del autor y administrador del proyecto.

6. Autorizaciones	
12.5 El Scope Statement, WBS, Calendario del Proyecto, Plan de Riesgos son aprobados por el/los:	
<ul style="list-style-type: none"> • Tutor de la Cátedra : Ing. Calixto Maldonado • Miembros de la CAE 	
12.6 Los cambios en las líneas bases sobre el rendimiento del proyecto son aprobados por el/los:	
<ul style="list-style-type: none"> • Tutor de la Cátedra: Ing. Calixto Maldonado • Miembros de la CAE 	
12.7 Los entregables del Proyecto son aprobados/ aceptados por el/los:	
<ul style="list-style-type: none"> • Tutor de la Cátedra: Ing. Calixto Maldonado • Miembros de la CAE 	

7. Project Scope Statement Aprobaciones / Firmas			
Nombre del Proyecto:	Proyecto de realización de Red social Universitaria		
Administrador de Proyecto:	Diaz Molina Jorge Ariel Mat: Inf 451		
Revisé la información contenida en este documento y adhiero a ella:			
Nombre	Rol	Firma	Fecha (DD/MM/AAAA)

La firma Arriba indica el entendimiento de los propósitos y contenidos del presente documento por los firmantes.





13 Requerimientos del sistema

Para continuar con la metodología seleccionada, se presentaran los requerimientos del sistema que serán el punto de partida de los casos de uso, que dirigirán el desarrollo del sistema³³. En una primera instancia se presenta los requerimientos funcionales y a continuación los no funcionales.

Requerimientos Funcionales	
Código	Descripción
RF1	▪Unirse a la red social
RF2	▪Permitir el ingreso al sistema por dirección de correo electrónico y contraseña
RF3	▪Permitir el reseteo de la contraseña de usuario
RF4	▪El sistema deberá permitir a un usuario actualizar los datos del perfil a mostrar
RF5	▪El sistema deberá permitir invitar a amigos tipeando las direcciones de correo electrónico, el mismo deberá poder ingresar varias direcciones de mail al mismo tiempo.
RF6	▪Sistema deberá permitir saber cuales son las materias que un miembro de la universidad cursa
RF7	▪Permitir generar relaciones de amistad entre los distintos miembros de la comunidad y además se deberá poder modificar el estado de esas relaciones de amista.
RF8	▪Permitir al usuario poder enviar y recibir mensajes entre los miembros de una comunidad. Estos mensajes deberán respetar estados de pendientes de leer y leídos.
RF09	▪Permitir a los usuarios subir y bajar archivos
RF10	▪El sistema deberá permitir compartir URL (folcsonomías)
RF11	▪El sistema deberá mostrar un resumen de las estadísticas básicas o de la actividad reciente en la página de inicio.
RF12	▪El sistema deberá ser capaz de enviar preguntas a las personas que pueden ayudar al usuario automáticamente.
RF13	▪El sistema debe priorizar tanto las preguntas como los apuntes que han ayudado a los estudiantes a aprobar una materia.
RF14	▪Las noticias de los usuarios deben poder ser comentadas por los amigos del estudiante que publica la noticia
RF15	▪Los estudiantes deberán poder denunciar aquello que los usuarios suban, sean estas preguntas, comentarios, o apuntes.

33 Use Case Driven (proceso dirigido por casos de uso) – RUP



Requerimientos No Funcionales	
Código	Descripción
RNF1	▪Encriptación de los datos de cuenta
RNF2	▪Crear filtros de sanitización de datos contra sql injection
RNF3	▪Implementar estrategias de sanitización de datos contra ataques XSS
RNF4	▪Implementar estrategias de sanitización de datos contra ataques CSRF



14 Decisiones Estratégicas

Este apartado tiene por finalidad dejar sentado cuales son las elecciones estratégicas, tanto tecnológicas como metodológicas, asumidas para llevar a cabo el proyecto.

14.1 Metodología de Desarrollo

Se pretende utilizar un mix entre [RUP](#) y [SCRUM](#), ya que ninguna de las dos representan ambientes cerrados y prescriptivos, si no que las mismas centran sus practicas en la adaptación al proyecto a realizar. ¿Por que las dos? En el caso de RUP cuenta con excelentes artefactos y procedimientos, a la hora de gestionar los proyectos. Y en este proyecto en particular, no se puede contar con uno de los roles principales que plantea SCRUM, el dueño del producto. Es por ello que se hace necesaria la mayor cantidad posible de documentación en la gestión y administración del proyecto. Y es en esta etapa donde se puede aprovechar al máximo las virtudes de RUP. Sumado a los ciclos de desarrollo son similares a los que plantea SCRUM.

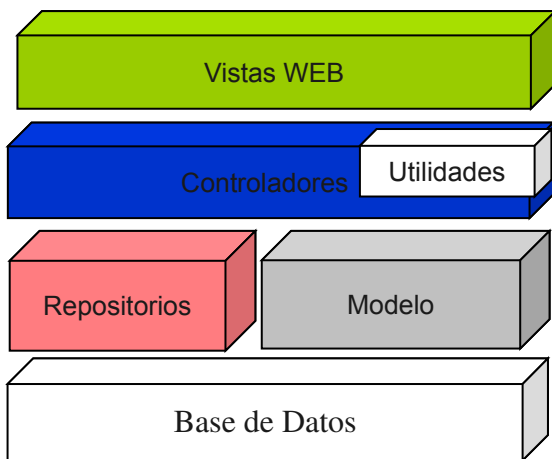
A la hora de trabajar en el desarrollo de la aplicación propiamente dicho, SCRUM plantea una mayor adaptación al cambio de requerimientos, y al conocimiento progresivo que se tiene sobre el sistema a desarrollar. Al trabajar a lo largo del proyecto con distintos profesores, se puede llegar a obtener distintos puntos de vista, y la adaptación a los cambios de requerimientos resultara fundamental para el éxito del proyecto.

14.2 Arquitectura Inicial

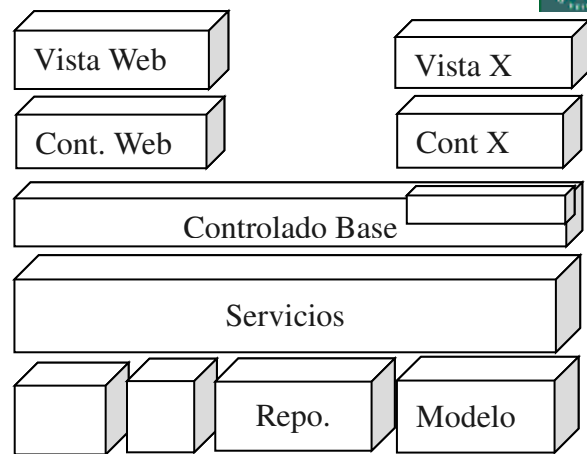
Ref. Marco Teórico: [Arquitectura del Sistema](#)

14.2.1 Arquitecturas posibles

Por las características del trabajo a desarrollar, se consideraron dos arquitecturas posibles. La primera de ellas (A) Es una arquitectura basada en el Patrón de Diseño de Sistemas MVC (Modelo-Vista-Controlador). La segunda, una arquitectura más sofisticada, está basada en los servicios.



A) Arquitectura Basada en MVC (**BMVC**)



B) Basada en Servicios (**BS**)

14.2.2 Basada en MVC

Nos permite una rápida y económica codificación de las principales funcionalidades, además de brindarnos la ventaja de que gran parte de las herramientas de programación existentes en la actualidad están orientadas al patrón MVC, permitiendo esto auto-generar código útil aproximadamente en un 30% de la totalidad. Si el sistema creciese en cuanto a funcionalidades se compartirán las capas *repositorio* y *modelo* en un 100% mientras que el porcentaje de reutilización de código de la capa *controlador* dependerá de la tecnología que emplearemos. Con ésta arquitectura equilibramos escalabilidad con rapidez en los resultados y posibilidades de realización en el tiempo fijado.

Como punto en contra de ésta arquitectura tendremos controladores, pesados, es decir con mayor cantidad de código lógico. Además de la dificultad de expandirse a otras tecnologías que no estén basadas en este patrón, y lenguajes de programación seleccionados.

14.2.3 Basada en Servicios

Ésta arquitectura, mucho más compleja y costosa que la anterior, en cuanto a tiempo y esfuerzo de programación nos permite una reutilización del código en un 70% cualquiera sea el crecimiento del sistema.

Todas aquellas tecnologías que trabajen con el sistema compartirán además de la capa de “*Servicios*” en un 100%, la capa de “*Controlador Base*” en grandes porcentajes.

La capa de servicios será la encargada de llevar a cabo la codificación de los UC del sistema por lo que cualquier tecnología podría tener acceso a la implementación de los casos de uso del sistema tan solo con enviar y recibir mensajes de esta capa. Por ejemplo una vez terminada la codificación



de la capa de servicios, y de aquellas sobre las cuales se asienta, se podría implementar una vista Web y un *Web Services* simultáneamente con un mínimo de esfuerzo.

Como puntos en contra, podemos destacar, que no existen herramientas en mercados que trabajen con este tipo de arquitecturas, por lo que su codificación es más artesanal, además de ello, requiere de un exhaustivo análisis de cada caso en particular y una fuerte carga de codificación al principio sin arrojar resultados visibles.

Es una arquitectura ideal cuando existen los recursos de programación y tiempos de desarrollo prolongados.

14.2.4 Selección de la arquitectura

En mi rol de líder de proyecto, me corresponde seleccionar cual será la arquitectura que llevaré a cabo a los fines de poder cumplir con los objetivos del proyecto en los tiempos establecidos.

Mi selección es la **arquitectura basada en el patrón MVC**, ya que con ella se equilibrará la escalabilidad del sistema con el tiempo de desarrollo. Además de lo mencionado anteriormente, no se cuenta con los recursos y el tiempo demandado por arquitectura basada en servicios, transformándose esto en un gran factor de riesgo que puede atentar contra los objetivos del proyecto.

La arquitectura seleccionada cumple con los requisitos de una buena arquitectura, además de ajustarse a las demandas de este proyecto en particular.

14.2.5 Especificaciones de la arquitectura

Responsabilidades de las distintas capas

Vista WEB:

- Tomar las peticiones de usuario
- Mostrar mensajes de error
- Implementar tecnologías de presentación
- Mostrar los resultados de los mensajes que provienen del controlador

Controladores:

- Implementar validaciones de forma
- Tomar y poner variables en sesión

- Despachar vistas
- Capturar excepciones de negocios
- Enviar mensajes a las capas de Repositorios y Modelo y Utilidades.

Utilidades:

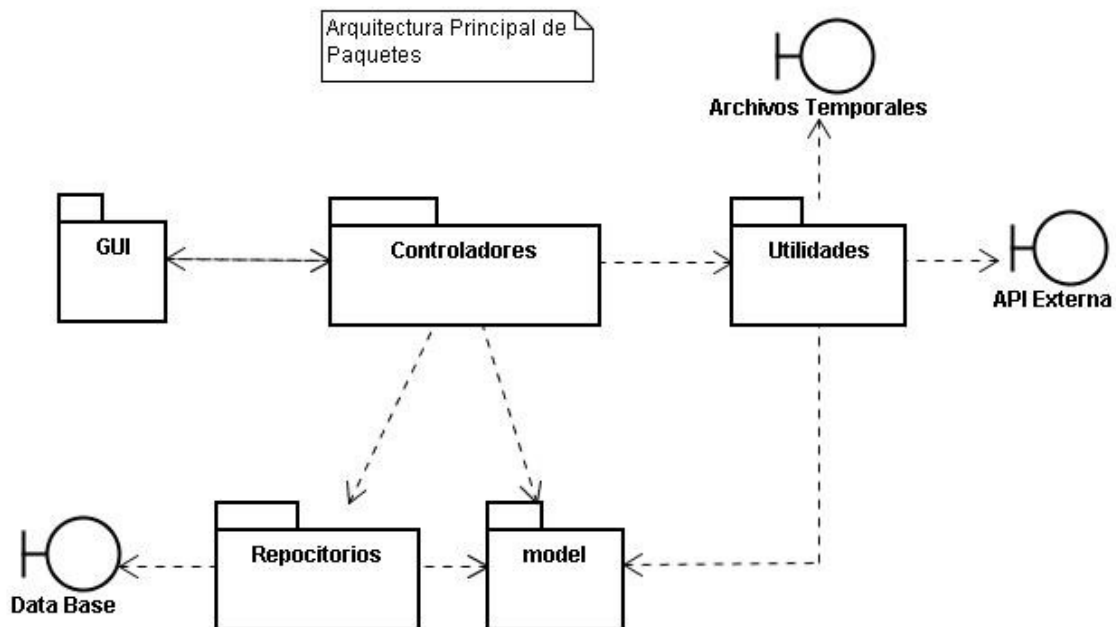
- traducir a objetos del modelo, aquellos datos que provengan de fuentes externas.
- Generar componentes que el sistema pueda reutilizar

Repositorios

- Consultas a la base de datos
- Creación de objetos

Modelo

- Mantener el estado de los objetos
- Guardar y borrar seguro
- Implementación de los casos de uso.



El gráfico expuesto muestra la organización de los paquetes y sus conexiones tanto a las base de datos, como a APIs y archivos. Existe una bidireccionalidad entre la interfaz gráfica y los Controladores por ser dependientes unos de otros.

Los archivos temporales, son archivos de importación de contactos, por eje. CSV.



No se considera otro tipo de archivos, es decir que persistan en la base de datos, como por ejemplos archivos de configuración, por no encontrarse este tipo de funcionalidades dentro del alcance del sistema. De ser necesarios se agregará un modulo que contenga estos archivos en futuras implementaciones.

14.3 Lenguaje de Programación

La selección del lenguaje parecería obvia.

Los dos lenguajes de programación que cuentan con mayores comunidades alrededor del mundo son java, y php. Al tratarse de un aplicación que pretende dar por resultado una red social, es decir, que por principio de las mismas, debe ser abierta a las comunidades, se elegirá aquel lenguaje que cuente con una mayor comunidad de programadores, que puedan en un futuro complementar la aplicación, con aplicaciones de terceros. En la actualidad ese lenguaje es PHP. Y con el advenimiento de PHP5 y su programación orientada a objetos, las virtudes de uno y otro parecerían equipararse. En contraposición, podemos destacar que PHP es un lenguaje interpretado, por lo que una incorrecta programación y no contar con los últimos parches de seguridad podría dar problemas de seguridad. A lo que responderé con políticas y estrategias de seguridad acordes a un trabajo final de graduación. Debo de resaltar que en esta decisión, ha pesado el caso de éxito llevado a cabo por la red social mas grande en la actualizad, facebook, y la porción de dicho éxito que se lleva la comunidad que desarrolla aplicaciones tales como tests, cuestionarios y juegos. Dicho servicio de red social está programado en PHP.



15 Estimaciones

15.1 Introducción

Cabe destacar que todos los valores rendidos en este apartado, hacen referencia a estimaciones iniciales. Ya que los verdaderos valores, solo podrán conocerse al final del proyecto.

Todo proyecto tiene una triple restricción; alcance, tiempo, costo. Se procederá a realizar estimaciones sobre las dos primeras, ya que las mismas pueden producir una alteración en los resultados esperados del proyecto. La última de las variables, el costo, será considerada “*Cæteris paribus* ³⁴” a los fines de este trabajo. Ya que todas las erogaciones derivadas de la realización de este proyecto serán afrontados por el líder del proyecto.

15.2 Estimación de alcance

La estimación del alcance del proyecto, se realizara a través de un artefacto del project managment, llamado “Work Breakdown Structure” (WBS). He seleccionado este artefacto para mostrar los entregables del proyecto como una lista de entregables de una *Project backlog list*

La WBS consiste en un árbol de entregables y las tareas necesarias para cumplir con esos entregables. La definición, tanto de las tareas como de los entregables, corresponderá a la necesidad de visualización del esfuerzo a realizar. Es decir que pueden variar tanto en tiempo, como en complejidad.

Del grafico “WBS de alcance del software” del anexo diagramas, se desprende que para desarrollar el software propuesto se deberá cumplir con **59** entregable finales y las tareas correspondientes para cumplir con dichos entregables.

Cabe destacar que la eliminación de algún entregable en la última fase responde al conocimiento y experiencia que se va acumulando a lo largo del desarrollo del proyecto, es decir a su curva de aprendizaje.

15.3 Estimación del tiempo

Para estimar los tiempos en los cuales incurrirá el proyecto, se acudió a la misma herramienta, (WBS) pero en este caso, a cada tarea se le asigno una cantidad de horas. La acumulación de las

34 Las variable se consideran constante a los fines del estudio encomendado



mismas nos revela el tiempo total del proyecto.

Del diagrama “TFG-WBS-vXXX” adjuntado en el anexo Diagramas, se desprende que el proyecto tendrá una totalidad de **855hs** siendo **640** las horas de dedicación directa a la construcción del software.

Algunas consideraciones con respecto a esta estimación:

- La estimación se realiza en cantidad de horas y no en días o meses
- Tanto las jornadas de trabajo como el calendario al cual se ajustará serán definidos en la planificación
- Esta metodología tiene la ventaja de ser la acumulación de pequeñas estimaciones, por lo que el error individual en la estimación de experto de una tarea específica, se ve disimulado en la totalidad.
- Esta metodología no tiene en cuenta la proyección de tareas en el tiempo, no se especifica los caminos críticos.
- Los tiempos estimados tienen en consideración las distintas iteraciones.
- La disminución en las horas estimadas para realizar cada tarea en las fases finales del desarrollo responden a la acumulación de experiencia y a la curva de aprendizaje.

15.3.1 Relaciones Entre los Entregables

En el caso de las tareas que conforman las *Fases* debemos denotar que **no** son independientes unas de otras. La alteración en las horas asignadas a un entregable está relacionada a las horas que se le deben asignar a otro entregable. De no ser así se afectara directamente a la **calidad de producto**.

- En el caso presentado, el entregable “**Estabilización**” es **dependiente de los entregables “Diagramas de Secuencia”, “Diagrama de Clase”, “Prototipos” y “Use Cases”**. Existe una relación inversamente proporcional entre las horas de unos y otros. Es decir, por ejemplo, que una disminución en una hora destinada a realizar los UC, afectaran en una proporción superior a uno, al incremento en las horas de estabilización.
- La asignación de horas a las tareas relacionadas al entregable “Testing” afectará en una relación directamente proporcional a la calidad del producto.



- El entregable “Código Fuente” es considerado un entregable de transición a “Estabilización”. Es decir que podremos decir que un código es limpio y responde a las necesidades de la aplicación cuando se convierta en el resultado del entregable “Estabilización”



PLANIFICACIÓN

- Plan de Fases
- Cronograma
- Factores Susceptibles de Afectar la Fecha de Entrega
- Estimaciones de Fecha de Entrega
- Roles y Responsabilidades



16 Plan de Fases

Cumpliendo con la metodología de desarrollo seleccionada (RUP), el siguiente proyecto se dividirá en 4 Fases las cuales tendrán una totalidad de 11 iteraciones. Cada una de las cuales representa un ciclo, donde las fases del de desarrollo de iniciación, elaboración, construcción y transición se llevaran a cabo. Al finalizar cada iteración se espera tener una versión del sistema que cumple con una determinada cantidad de requerimientos o funcionalidades.

Iteraciones - Proyecto Red Social Universitaria	
<i>Describe los circuitos a realizar en cada una de las iteraciones. Una iteración es un mini proyecto que al final da por resultado un entregable que aporta valor al software.</i>	
<p>Fase 1 (3 Iterac.) 156 Hs.</p>	<p><i>* Casos de Uso Posibles</i></p> <ul style="list-style-type: none"> • Suscribirse a la red social • Login de usuario • Actualizar Perfil • Actualizar Mas Información • Actualizar Información de Contacto • Actualizar Información Básica • Enviar mail • Invitar Contactos a formar parte de la red <p><i>* Actor</i></p> <ul style="list-style-type: none"> • Alumno
<p>Fase 2 (3 Itera.) 176 Hs.</p>	<p><i>*Casos de Uso Posibles</i></p> <ul style="list-style-type: none"> • Enviar invitaciones de amistad • Ver Invitaciones recibidas • Cambiar Estado de Invitaciones Recibidas de Amistad • Listar Amigos • Cambiar estado de los amigos • Redactar Mensaje • Ver bandeja de Entrada • Leer Mensaje • Contestar Mensaje • Cambiar Estado Mensaje <p><i>* Actor</i></p> <ul style="list-style-type: none"> • Alumno
<p>Fase 3 (2 Itera.) 145Hs</p>	<p><i>* Casos de Uso Posibles</i></p> <ul style="list-style-type: none"> • Agregar Evaluador • Registrar Examen • Agregar Tema a una Materia • Listar Tema • Listar Pregunta • Agregar Pregunta de Examen • Cambiar Estado Materia • Escribir Mi Pregunta • Listar Mis Preguntas • Ver Respuestas • Listar Preguntas a Responder • Responder Preguntas • Publicar Noticia



Iteraciones - Proyecto Red Social Universitaria	
	<ul style="list-style-type: none"> • Ver mis Noticias <p>*Actor</p> <ul style="list-style-type: none"> • Alumno
<p><u>Fase 4</u> (2 Itera.) 111hs</p>	<p>* Casos de Usos posibles</p> <ul style="list-style-type: none"> • Subir Archivo • Listar mis Archivos • Listar Archivos de Mis Amigos • Bajar Archivos • Ver Comentarios de Una noticia • Escribir Comentario • Cambiar Contraseña • Agregar Taxonomías • Ver mis Taxonomías • Ver Taxonomías de Mis Amigos • Listar Personas Que Me Pueden Ayudar <p>* Actor</p> <ul style="list-style-type: none"> • Alumno

Tanto las actividades de gestión de configuración, gestión de cambios, y administración de proyecto están incluidas en cada una de las iteraciones. Al finalizar todas las iteraciones se realizarán actividades de estabilización del proyecto en general. La disminución de la carga horaria de cada una de las iteraciones responde a cuestiones de alcance de cada una de las iteraciones y a la curva de aprendizaje y el supuesto del incremento en la velocidad de codificación de cada una de las iteraciones.



17 Cronograma

Una vez estimadas las horas sobre el entregable; “*Circuitos Básicos del Sistema*” (630) que el proyecto requerirá a través de la WBS, se procederá a estimar la jornada laboral para cada uno de los meses correspondientes. Concluida esta etapa se continuará con la estimación en meses y días, sometiendo las horas y la jornada laboral al ajuste del calendario.

La semana de trabajo se fija en cinco días por defecto, excluyendo los feriados y fines de semana.

Cronograma de Trabajo			
<i>Es la adaptación de las estimaciones de horas al calendario real del proyecto</i>			
Mes	Jornada Laboral (hs)	Nº Días Laborables	Total Horas/Mes
Mayo	4	19	76
Junio	4	21	84
Julio	6	22	132
Agosto	8	20	160
Septiembre	6	22	132
Octubre	4	15	60
Total			644

La diferencia en las horas laborables se debe al calendario personal, mientras que el número de días laborales corresponde a tanto a los días libres de fines de semana y feriados, como a los días que se requiere en el último mes para completar el proyecto.



18 Factores susceptibles de afectar la fecha de entrega (Riesgos / Oportunidades)

En este apartado se hará referencia a aquellos factores que pueden afectar la fecha de entrega. Se identificarán tanto los riesgos, como aquellos factores que pueden incidir positivamente. Al tratarse de estimaciones de experto, es decir realizadas a partir de las consultas a personas que trabajaron en proyectos de características similares³⁵, se definirá una ponderación sobre una escala de 1 a 10, siendo 1 un factor cuya influencia es casi imperceptible, y 10 un factor con influencia crítica o que podría atentar directamente a la conclusión exitosa del proyecto.

Factores que Influyen en la Fecha de Entrega del Software		
<i>La fecha de entrega real del proyecto se puede ver afectada por alguno de los factores listados a continuación</i>		
Factores de influencia	Ptos de Ponderación (PP)	Signo
Cambio de requerimientos	9	-
Cambios en la tecnología de desarrollo	7	-
Inexperiencia de los recursos afectados	5	-
Errores no reportados sobre la tecnología de desarrollo (bugs de los entornos de desarrollo y toda herramienta utilizada)	3	-
Evaluaciones o feedback inconsistentes	8	-
Trabajo fuera de las horas del cronograma (extra)	6	+
Reutilización de componentes de terceros	5	+
Optimización de la velocidad de desarrollo	7	+
Reutilización de código de terceros	6	+
Arquitectura inadecuada	8	-
Cambio en las fechas de entrega de milestones	8	-
Cambio en evaluadores o responsables del feedback	5	-
Escape de los procedimientos de cambio	8	-
Colaboración de especialistas tecnológicos	6	+
Responsabilidades diluidas	6	-

35 Buzzd Social network service development team



19 Estimaciones de Fecha de Entrega

Se fijará una fecha de entrega, como punto de partida, proveniente de las tareas de estimación realizada en el cronograma de trabajo. Dicha fecha de entrega se verá afectada por los puntos de ponderación listados en el apartado anterior. Es decir, que la fecha definitiva de entrega se verá sujeta a la ocurrencia, o no de los factores listados con anterioridad.

Estimaciones de Fecha de Entrega			
<i>La estimación de la fecha de entrega se encuentra directamente relacionada con los factores de influencia. La columna porcentaje de confianza, representa la confianza que se tiene sobre la entrega del trabajo en la fecha establecida.</i>			
Factores de Influencia	Fecha		% de Confianza
	Desde	Hasta	
Entre 15 y 30 puntos positivos	01 / 09 /2009	15 / 09 /2009	40 %
Entre 1 y 15 puntos positivos	16 / 09 / 2009	30 / 09 / 2009	50 %
Sin puntos Positivos ni negativos	01 / 10 /2009		70%
Entre 1 y 25 puntos negativos	02/ 10 /2009	01 / 11 / 2009	80 %
Entre 26 y 50 puntos negativos	02 / 11 /2009	15 / 12 /2009	90%
Entre 51 y 74 puntos negativos	16 / 12 / 2009	01 / 02 /2010	98%



20 Roles y Responsabilidades

En el siguiente apartado se describen los roles necesarios para llevar adelante el proyecto. Debo destacar que no se hace referencia a personas, si no a roles. Ya que a los fines de cumplir con el proyecto todos los roles y sus responsabilidades serán llevados a cabo por mi persona.

Además de lo mencionado, los roles serán establecidos en orden de prioridad y a la carga horaria que se realice en cada uno.

Roles y Responsabilidades del Proyecto	
Roles	Responsabilidades
Líder de Proyecto	El jefe de proyecto asigna los recursos, gestiona las prioridades, coordina as interacciones con los clientes y usuarios, y mantiene al equipo del proyecto enfocado en los objetivos. El jefe de proyecto también establece un conjunto de prácticas que aseguran la integridad y calidad de los artefactos del proyecto. Además, el jefe de proyecto se encargará de supervisar el establecimiento de la arquitectura del sistema. Gestión de riesgos. Planificación y control del proyecto.
Ingeniero de Software	Gestión de requisitos, gestión de configuración y cambios, elaboración del modelo de datos, preparación de las pruebas funcionales, elaboración de la documentación. Elaborar modelos de implementación y despliegue.
Analista de Sistemas	Captura, especificación y validación de requisitos, interactuando con el cliente y los usuarios mediante entrevistas. Elaboración del Modelo de Análisis y Diseño. Colaboración en la elaboración de las pruebas funcionales y el modelo de datos.
Programador Web	Elabora el código fuente, construye los prototipos, colabora en la elaboración de las pruebas funcionales, modelo de datos y las validaciones con el usuario
Diseñador Web	Es el encargado de diseñar la apariencia del sistema. Además de ello. Coordina sus actividades con los programadores

A continuación se presenta un grafico para determinar cual será la carga de acción durante el proyecto para cada uno de los roles establecidos. Este grafico muestra la carga principal de horas por cada una de las fases. Las relaciones de carga están establecidas en los números debajo de cada rol (ejemplo los las horas de programación son el triple de las horas de liderazgo)

Responsabilidades principales por fases				
Roles	Inicio	Elaboración	Construcción	Transición
<i>Líder de</i>	Coordinación general en el día a día. Establecimiento de procedimientos. Management			

Planificación / Roles y Responsabilidades



<i>Proyecto (1)</i>	político con el cliente. Elaboración y control de la arquitectura. Focalización en los objetivos			
<i>Ingeniero de Software (1)</i>	Requisitos. Entrevistas	Modelo de datos. Selección de herramientas	Configuración de cambios	Pruebas funcionales. Seguimiento de Inconvenientes
<i>Analista/ Programador (1)</i>	Capturar y validar requisitos. Entrevistas	Modelo de análisis y diseño.	Codificación conceptual	Pruebas funcionales (colabora).
<i>Programadores (3)</i>	Adaptación a las tecnologías. Entornos	Colaborar en el diseño	Código fuente	Estabilización
<i>Diseñador Web (1)</i>	Elaboración de prototipos		Codificación de la presentación	Cambios de apariencia



ELABORACIÓN / PRIMERA FASE

- Listado de Casos de Uso
- Casos de Uso del Sistema



21 Primera Fase

21.1 Listado de Casos de Uso

Ref. Marco teórico: [Caso de Uso \(Use Cases\)](#)

A continuación, se presentará el listado de los casos de uso con los cuales el sistema contará inicialmente. Estos UC son iniciales ya que tanto en la fase de construcción como en las distintas iteraciones del sistema puede ocurrir que surjan más UC devenidos de la maduración y entendimiento del negocio. Estos casos de uso hacen referencia a los requerimientos del sistema. La presentación de los mismos responde al orden planteado en el [plan de fases del sistema](#)

Casos de Uso Funcionales		
<i>Hacen referencia a las funcionalidades que el sistema debe realizar</i>		
<i>Notación código: Fase/N°UC</i>		
Reque.	Código	Nombre UC
Primera Fase		
RF1	F1/01	Subscribirse a la red social
RF2	F1/02	LogIn de usuario
RF3	F1/03	Actualizar perfil
RF4	F1/04	Actualizar Mas Información
RF5	F1/05	Actualizar Información de Contacto
RF6	F1/06	Actualizar Información Básica
RF7	F1/07	Enviar Mail de Invitación
RF8	F1/08	Invitar Contactos a formar parte de la red

A continuación tomaremos como referencia cada uno de los casos de uso, a los fines de cumplir con la guía que nos plantea la metodología RUP, y su principio que sostiene que es una metodología guiada por casos de uso. (*Use Case Driven*).



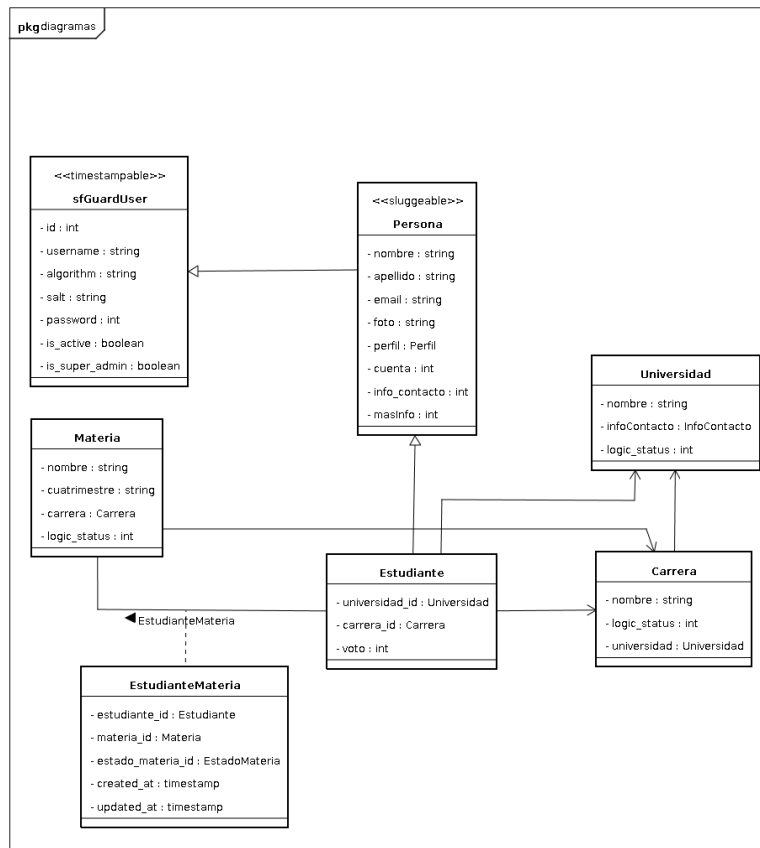
21.2 Subscribirse a la red social

21.2.1 Ficha de Caso de Uso

Subscribirse a la Red Social	
Código	F1 / 01
Actor	Invitado
Código WBS	142
Flujo Normal / User Stories	
<ol style="list-style-type: none"> 1. El estudiante llega a la página de suscripción de la red social. Ya sea por una invitación de un amigo o en forma independiente. 2. El estudiante ingresa su nombre de usuario, debe ser un e-mail valido, su contraseña, su nombre, apellido, la universidad a la que asiste y la carrera que cursa. 3. Presiona el botón [<i>Unirme ya!</i>] 4. El sistema lo redirecciona a una pagina de confirmación que la cuenta ha sido creada 5. El usuario hace clic en el vinculo: <u><i>ir a login</i></u> 	
Condiciones	
Pre Condiciones	Ninguna
Pos Condiciones	<ol style="list-style-type: none"> 1. Se crea un Estudiante 2. Se crea una Cuenta y se establece el logia estatus a pendiente de confirmación 3. Se Crean Tantas EstudiantesMateria como materias contenga el plan de carreras de la universidad y se establece su estado a "Libre"
Camino Alternativo / Extiende	
<ol style="list-style-type: none"> 1. El estudiante no ingresa un e-mail valido u pasa por alto alguno de los datos mandatarios. 2. El nombre de usuario ya existe en el sistema. 	

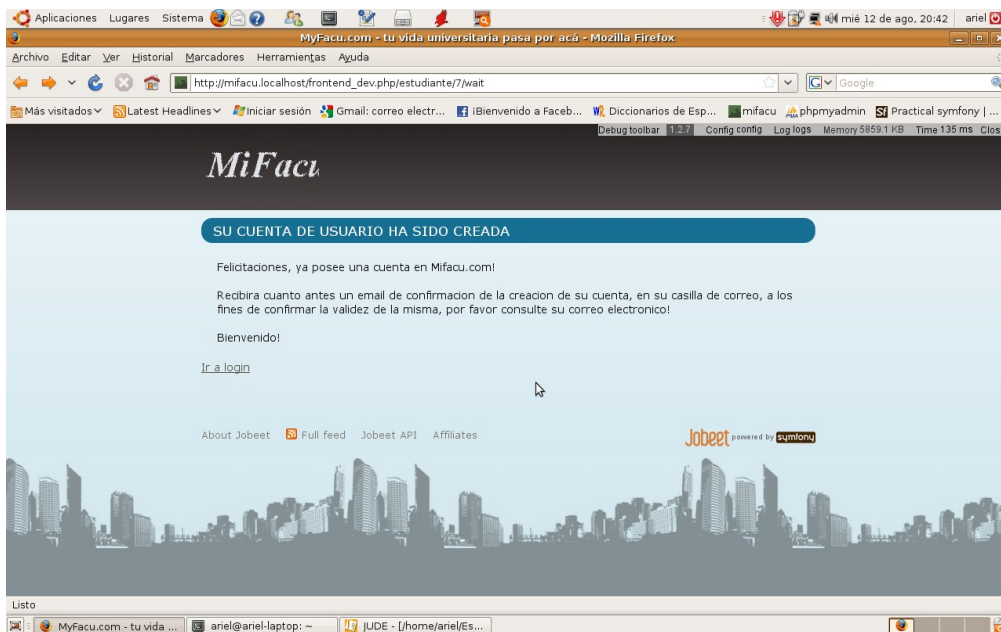


21.2.2 Diagrama de Clases

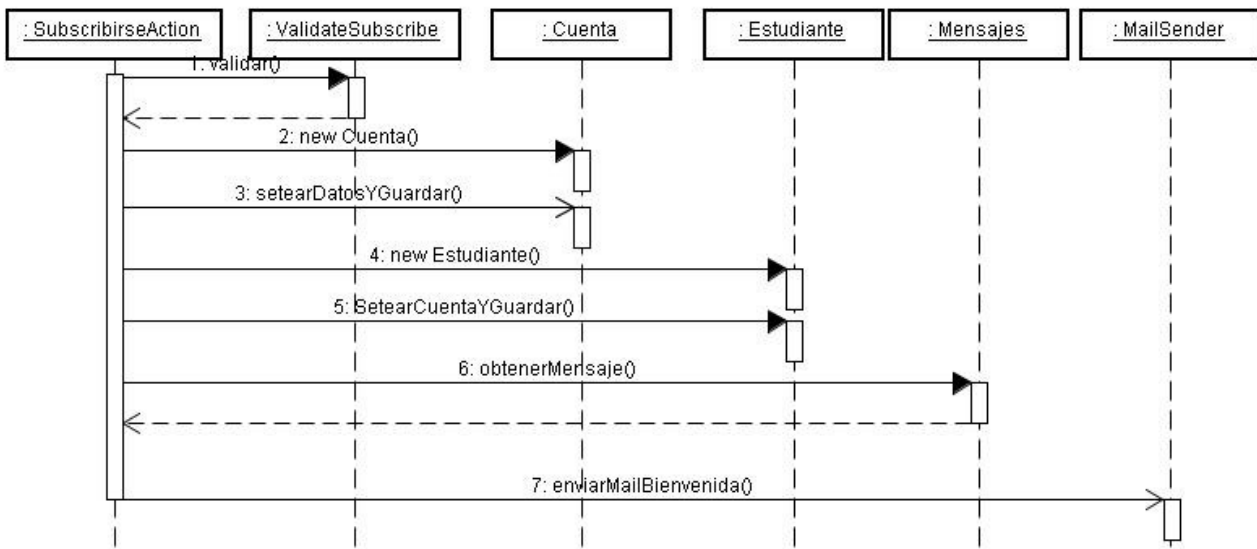


21.2.3 Pantallas





21.2.4 Diagrama de Secuencia



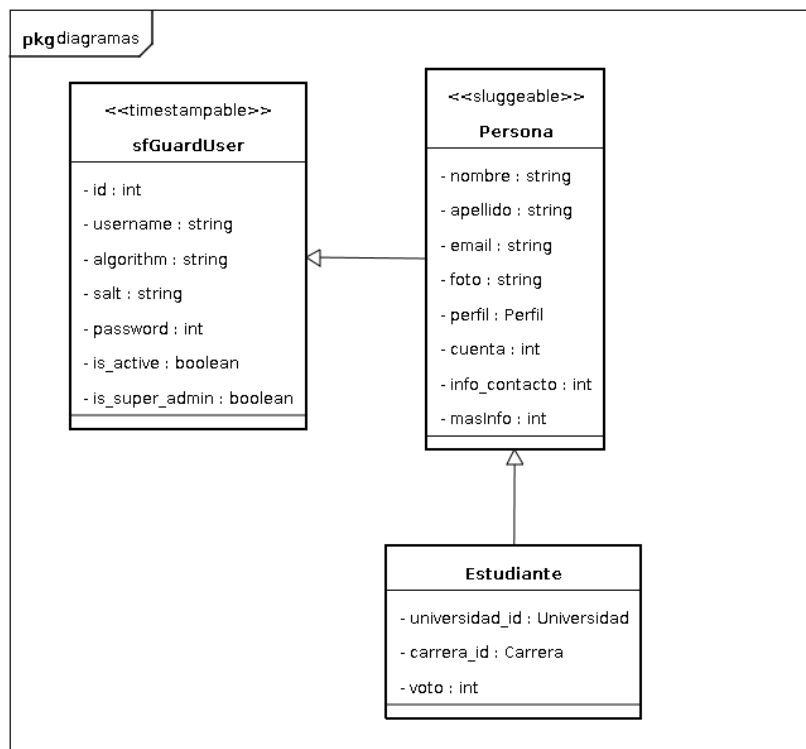


21.3 Login de Usuario

21.3.1 Ficha de Caso de Uso

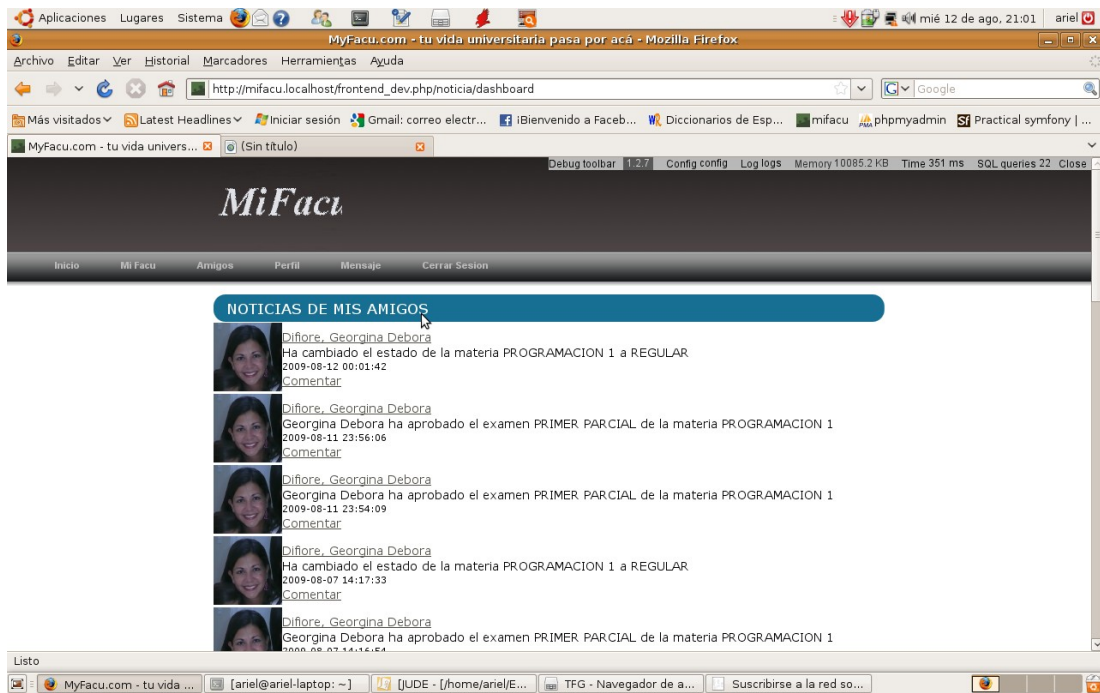
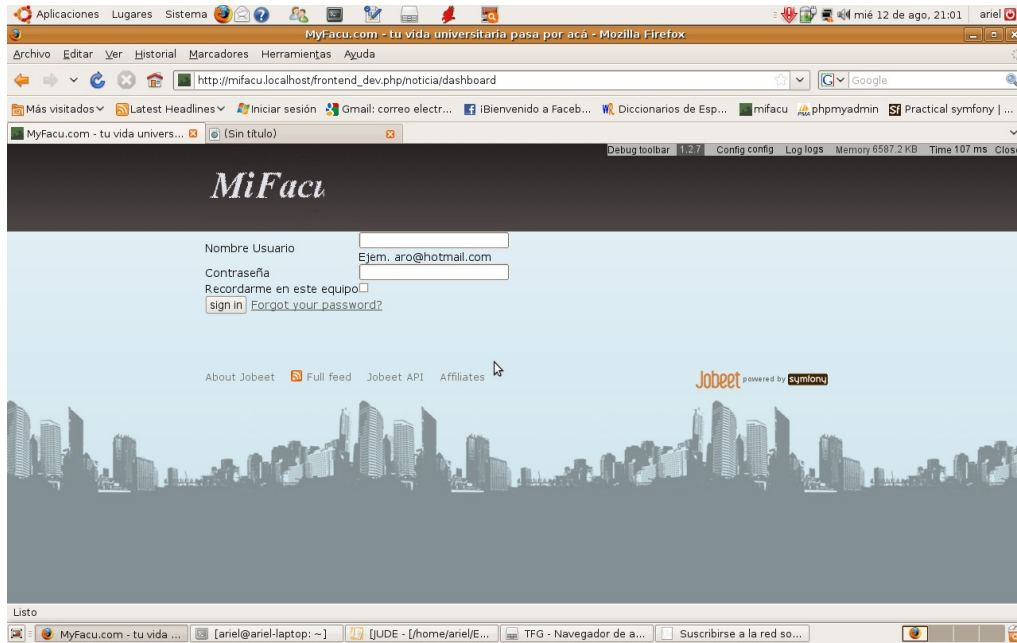
Login de Usuario	
Código	F1 / 02
Actor	Estudiante
Código WBS	143
Flujo Normal/ User Stories	
<ol style="list-style-type: none"> 1. El estudiante llega a la pantalla de login del sistema 2. Ingresa su nombre de usuario y contraseña 3. Hace clic en el Botón [Sign in] 4. El sistema lo redirecciona a la pagina de inicio de su cuenta. 	
Condiciones	
Pre Condiciones	<ol style="list-style-type: none"> 1. La Cuenta debe existir 2. El Estudiante Debe Existir
Pos Condiciones	<ol style="list-style-type: none"> 1. El estudiante es subido a la sesión de usuario
Camino Alternativo / Extiende	
<ol style="list-style-type: none"> 1. No existe el Usuario en el sistema. 2. No existe el nombre de usuario o la contraseña son incorrectas. 3. El usuario ingresar una dirección de mail invalida como nombre de usuario. 	

21.3.2 Diagrama de Clases





21.3.3 Pantallas



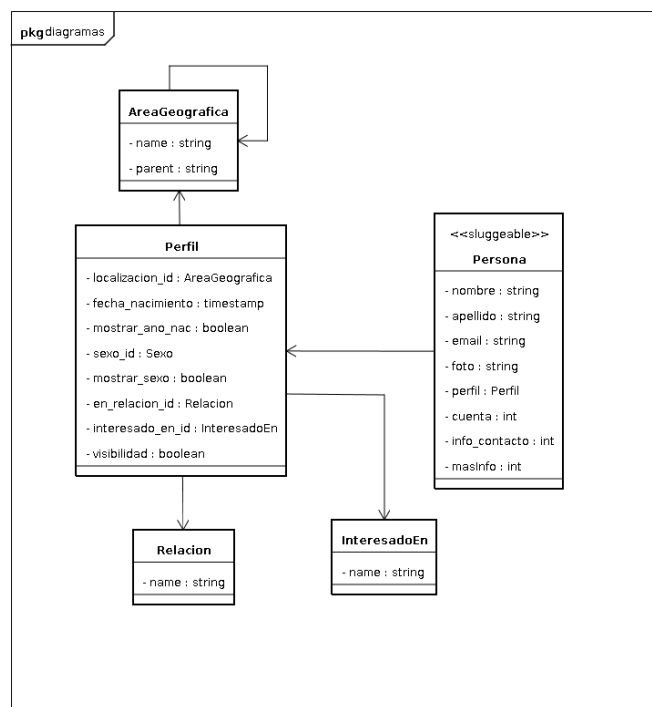


21.4 Actualizar Perfil

21.4.1 Ficha de Caso de Uso

Actualizar Perfil	
Código	F1 / 03
Actor	Estudiante
Código WBS	139
Flujo Normal / User Stories	
<ol style="list-style-type: none"> 1. El estudiante ingresa a la aplicación y en el menú de la misma hace clic en el botón Perfil / Información Básica 2. Selecciona en que lugar se encuentra 3. Selecciona su fecha de nacimiento y si desea que la misma sea mostrada en su perfil de usuario 4. Selecciona de que sexo es y si desea que el mismo se muestre en el perfil de usuario 5. Selecciona en que relación se encuentra y en que tipo de relación esta buscando. 6. Al finalizar hace clic en el botón [Guardar Cambios Realizados] 	
Condiciones	
Pre Condiciones	<ol style="list-style-type: none"> 1. El Estudiante debe existir 2. El AreaGeografica debe existir 3. La Relación debe existir 4. El InteresadoEn debe existir
Pos Condiciones	<ol style="list-style-type: none"> 1. Se Actualiza el Perfil del Estudiante
Camino Alternativo / Extiende	
Ninguno	

21.4.2 Diagrama de Clases





21.4.3 Pantallas



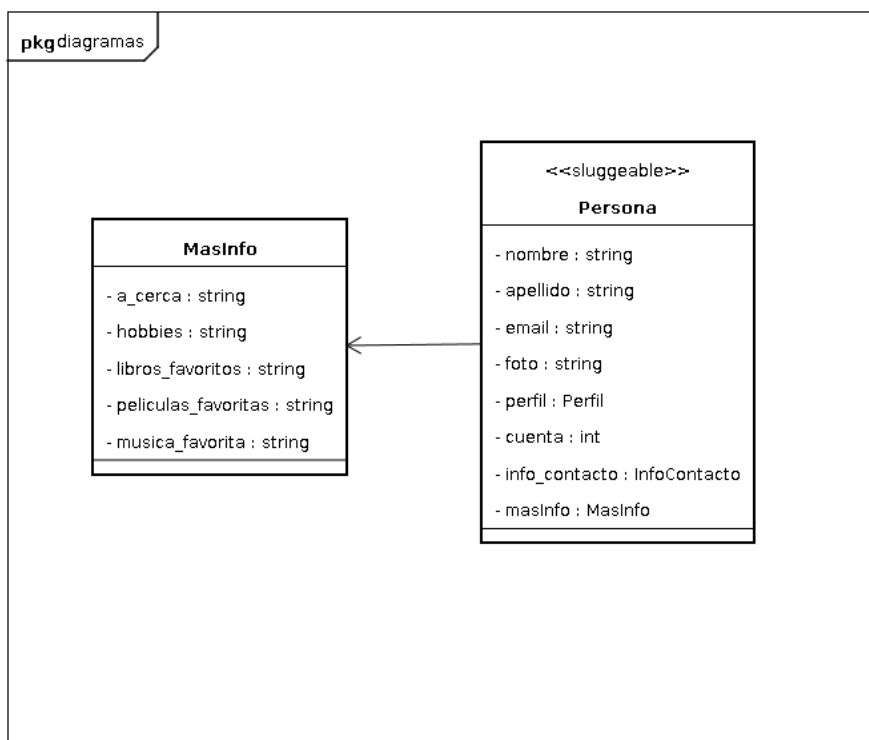


21.5 Actualizar Más Información

21.5.1 Ficha de Caso de Uso

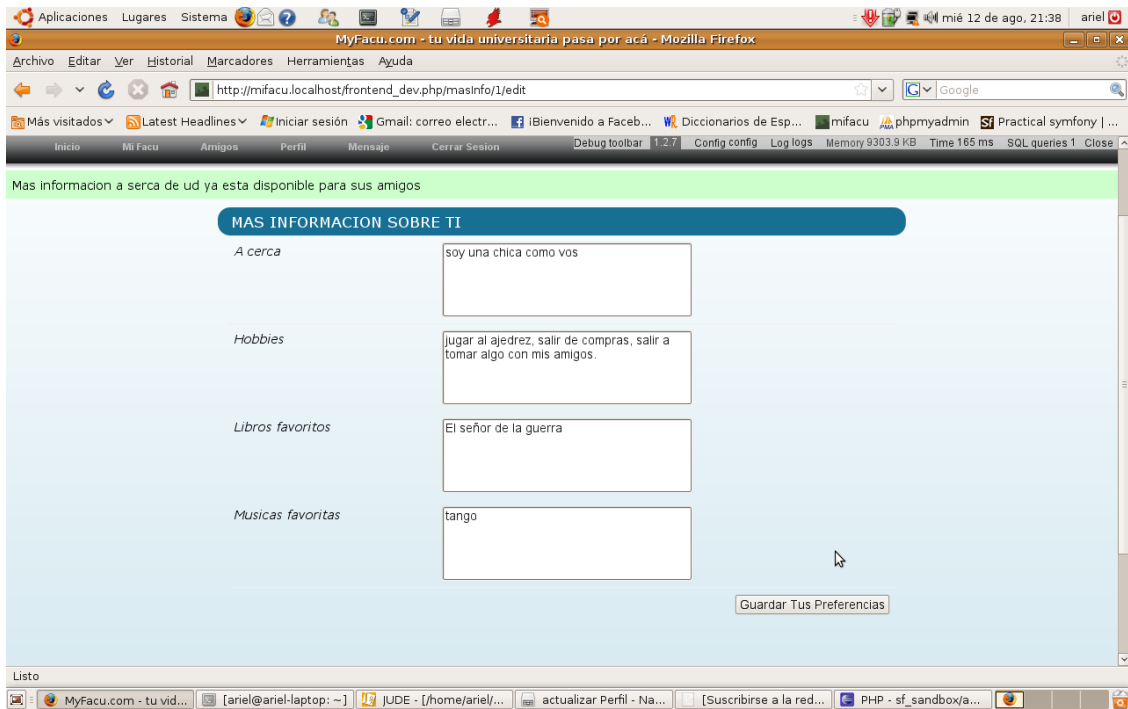
Actualizar Más Información	
Código	F1 / 04
Actor	Estudiante
Código WBS	140
Flujo Normal / User Stories	
1. El estudiante ingresa a la aplicación y hace clic en el menú Perfil en la opción Más Info 2. Ingresa un texto descriptivo a cerca de si mismo 3. Además de lo anterior tiene la opción de ingresar cuales son sus hobbies, libros y música favorita 4. Para finalizar hace clic en el botón [Guardar tus preferencias]	
Condiciones	
Pre Condiciones	La Persona debe existir
Pos Condiciones	Se actualiza la MasInformacion de la Persona
Camino Alternativo / Extiende	
Ninguno.	

21.5.2 Diagrama de Clases





21.5.3 Pantallas



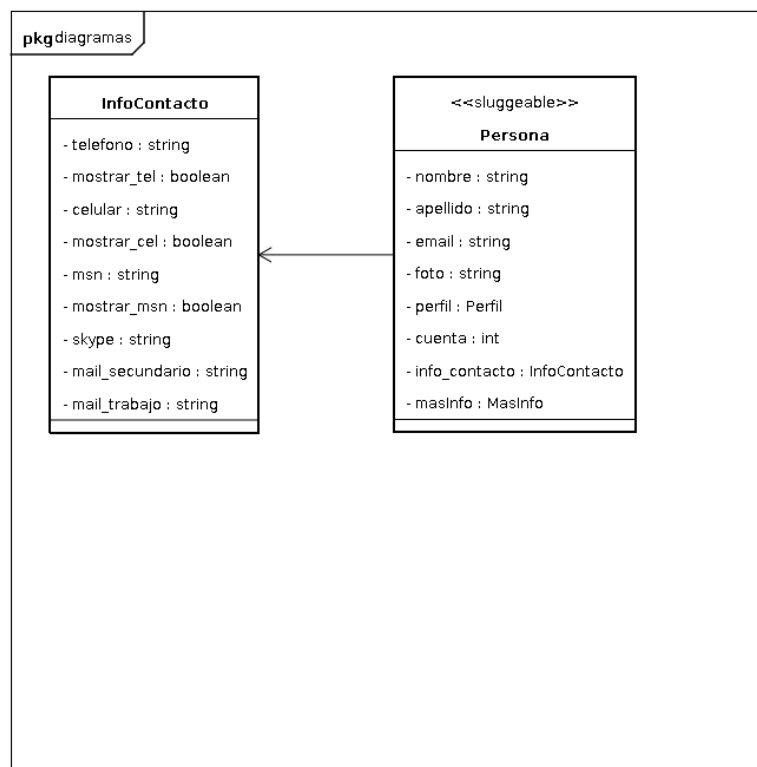


21.6 Actualizar Información de Contacto

21.6.1 Ficha de Caso de Uso

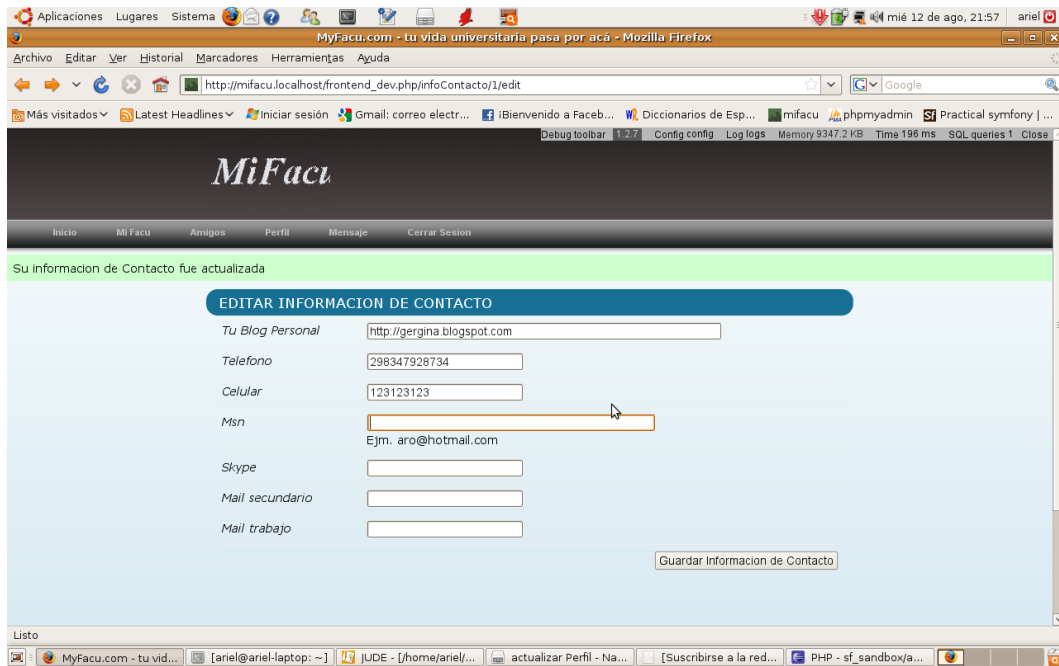
Actualizar Información de Contacto	
Código	F1 / 05
Actor	Estudiante
Código WBS	141
Flujo Normal / User Stories	
1. El estudiante ingresa al sistema y hace clic en el menú Perfil en la opción Información de Contacto 2. El estudiante tiene la opción de Ingresar cual es su página personal, teléfono, celular, msn, sakype, su mail secundario y su mail de trabajo. 3. Finalmente hace clic en el botón [Guardar Información de Contacto]	
Condiciones	
Pre Condiciones	La Persona debe Existir
Pos Condiciones	Se actualiza la Información de InfoContacto
Camino Alternativo / Extiende	
Ninguno	

21.6.2 Diagrama de Clases





21.6.3 Pantalla



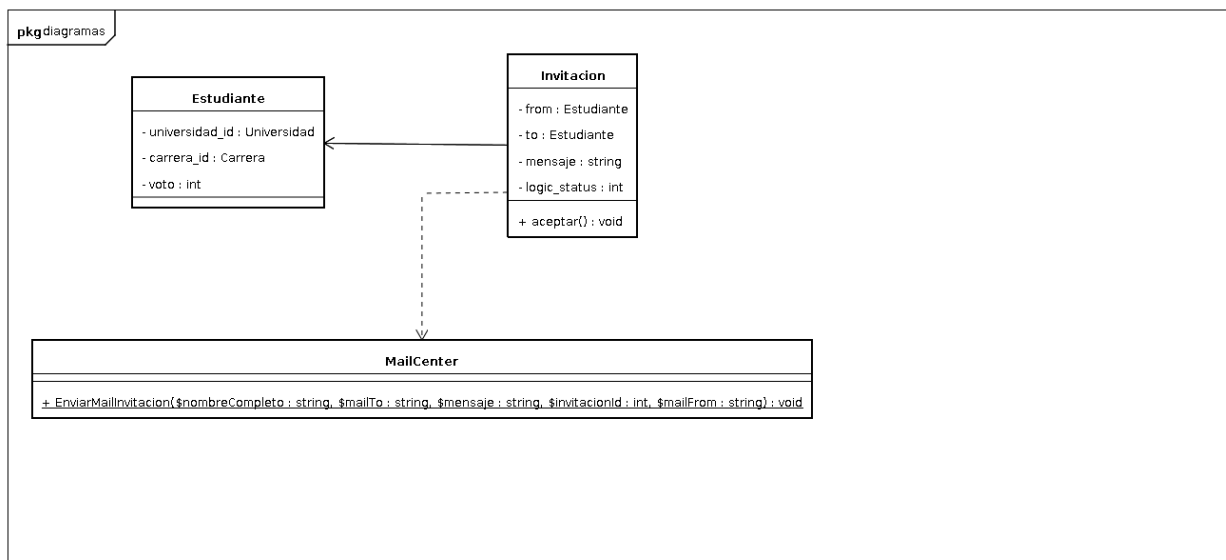


21.7 Enviar E-mail de Invitación

21.7.1 Ficha de Caso de Uso

Enviar E-mail de Invitación	
Código	F1 / 07
Actor	Estudiante
Código WBS	145
Flujo Normal / User Stories	
<ol style="list-style-type: none"> 1. El usuario entra al sistema y hace clic en invitar amigos 2. Ingresar la dirección del amigo o los amigos a los cuales le quiere enviar una invitación y un mensaje personalizado 3. Hace clic en enviar invitaciones 4. El sistema debe enviar una invitación de amistad vía mail a los amigos que el Estudiante ha registrado. 	
Condiciones	
Pre Condiciones	El Estudiante debe existir
Pos Condiciones	
Camino Alternativo / Extiende	
<ul style="list-style-type: none"> - No se puede enviar el email. - La dirección de e-mail es inválida. - Extiende el Caso de Uso Invitar Contactos a Formar parte de la Red. 	

21.7.2 Diagrama de Clases

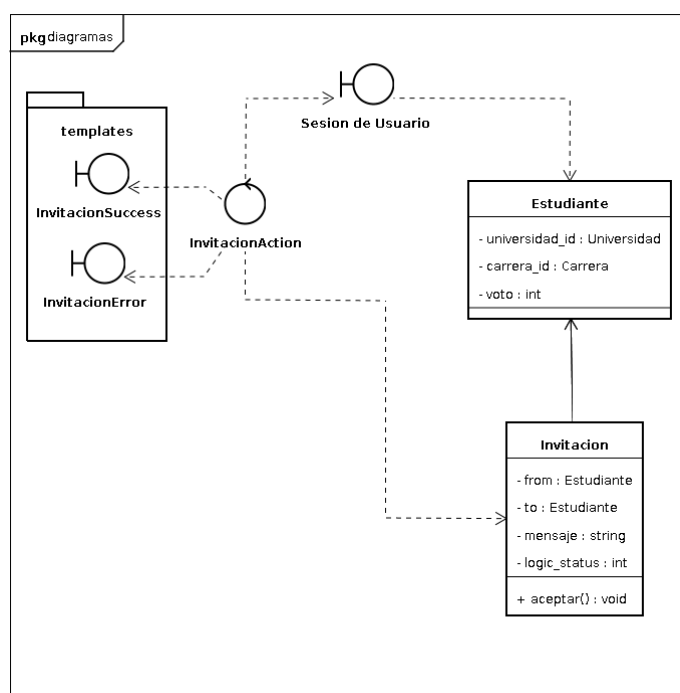


21.8 Invitar Contactos a Formar Parte de la Red

21.8.1 Ficha de Caso de Uso

Invitar a Contacto a Formar Parte de la Red	
Código	F1 / 08
Actor	Estudiante
Código WBS	146
Flujo Normal / User Stories	
1. El estudiante entra al sistema y va al menú amigos para hacer clic en la opción Invitar amigos 2. Ingresa los E-Mail de las personas a las que quiere invitar separadas por coma y el mensaje personalizado si es que lo desea. 3. Hace clic en el botón enviar invitaciones 4. El mensaje se envía a los mail de las personas que no tiene cuenta en el sistema.	
Condiciones	
Pre Condiciones	1. El Estudiante debe Existir 2. El destinatario no debe Existir
Pos Condiciones	1. Se crea una invitación 2. Se establece el logia_status de la invitación a pendiente 3. Incluye el caso de Uso: F1 / 07
Camino Alternativo / Extiende	
- El usuario ingresa direcciones de E-mail invalidas - Extiende el caso de Uso: F2 / 01 Enviar Invitaciones de Amistad.	

21.8.2 Diagrama de Clases





ELABORACIÓN / SEGUNDA FASE

- Listado de Casos de Uso
- Casos de Uso del Sistema



22 Segunda Fase

22.1 Listado de Casos de Uso

Debo de destacar que cada caso de uso adjuntara los diagramas que crea conveniente para el mejor entendimiento del mismo. En cuanto al flujo principal del caso de uso, se sostiene en historias de usuarios (User Stories³⁶) que no tienen una estructura formal de antemano.

Casos de Uso Funcionales		
<i>Hacen referencia a las funcionalidades que el sistema debe realizar</i>		
<i>Notación código: Fase/N°UC</i>		
Reque.	Código	Nombre UC
Segunda Fase		
RF9	F2/01	Enviar Invitaciones de Amistad
RF10	F2/02	Ver Invitaciones Recibidas
RF11	F2/03	Cambiar Estado de Invitaciones recibidas de Amistad
RF12	F2/04	Listar Amigos
RF13	F2/05	Cambiar Estado de los Amigos
RF14	F2/06	Redactar Mensaje
RF15	F2/07	Ver Bandeja de Entrada
RF16	F2/08	Leer Mensaje
RF17	F2/09	Contestar Mensaje
RF18	F2/10	Cambiar estado de un mensaje

³⁶ User Stories: Declaraciones verbales de un usuario sobre los requerimientos del sistema. Las mismas pueden o no tener una estructura formal.

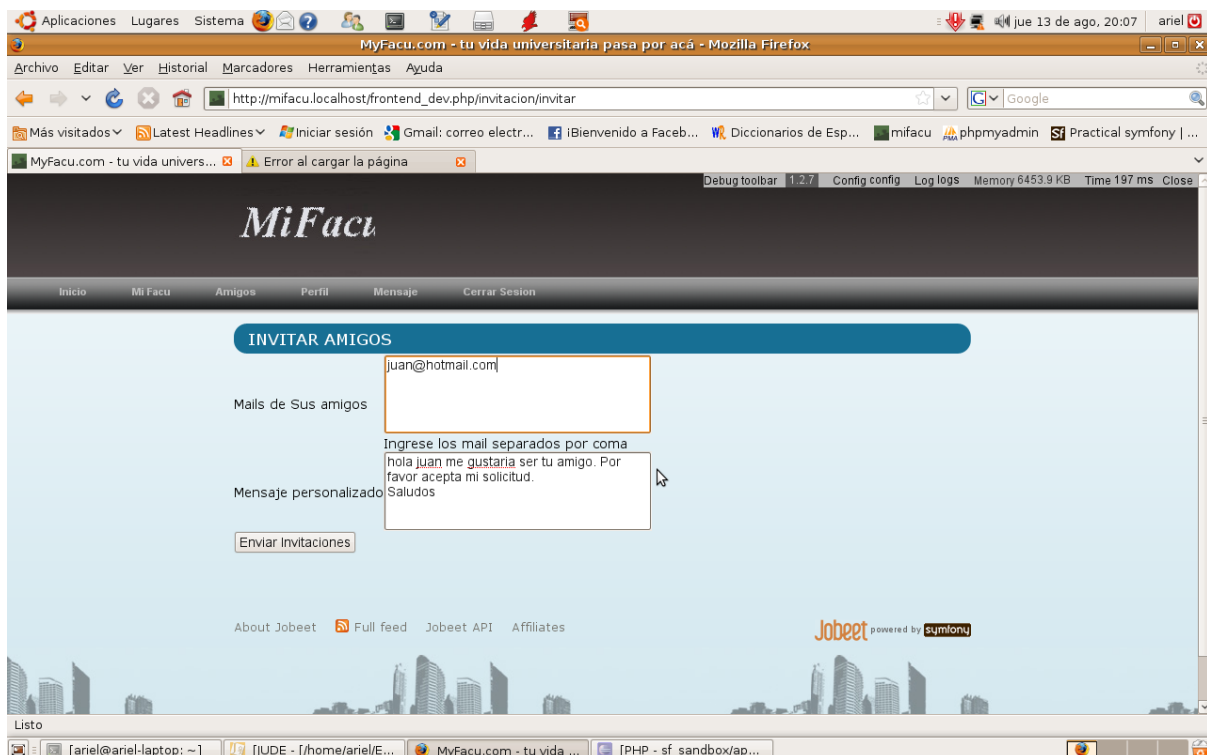


22.2 Enviar Invitación de Amistad

22.2.1 Ficha de Caso de Uso

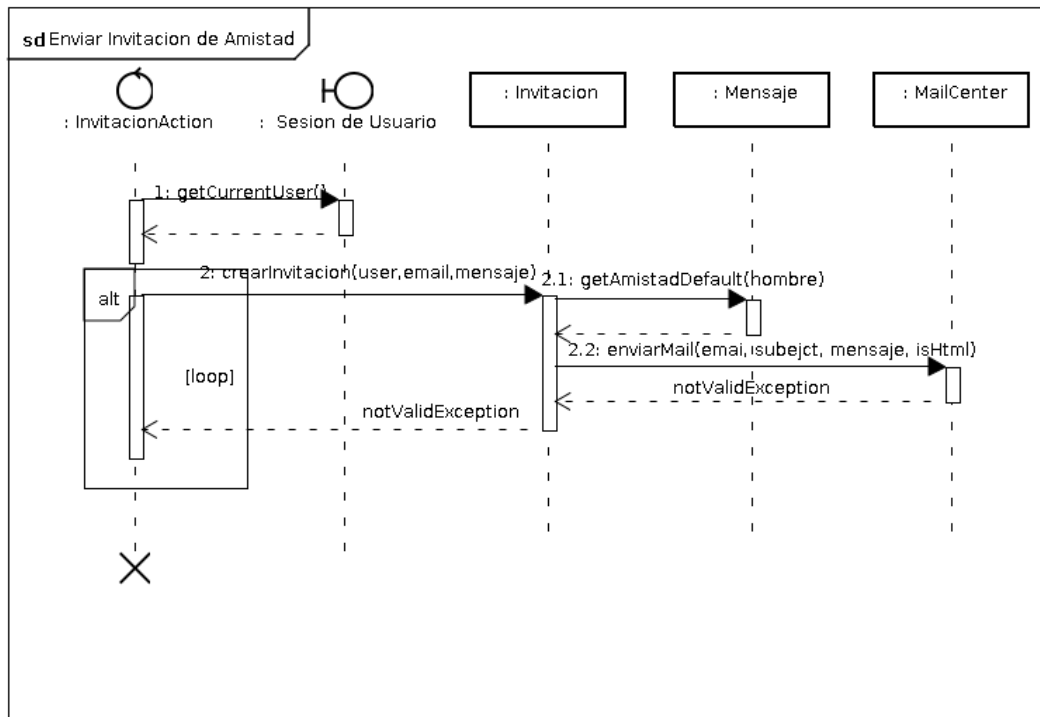
Enviar Invitación de Amistar	
Código	F2 / 01
Actor	Estudiante
Código WBS	204
Flujo Normal	
El estudiante ingresa al sistema y hace clic en el menú Amigos / Invitar Amigos Ingresar las direcciones de correo electrónico separadas por coma a las cuales quiere enviar las invitaciones para que sean sus amigos El estudiante hace clic en el botón [Enviar Invitaciones]	
Condiciones / User Stories	
Pre Condiciones	1. El usuario debe existir 2. No deben existir invitaciones pendientes de decisión entre el usuario y el contacto (logic_status)
Pos Condiciones	1. Se crea una Invitación 2. Se setea el logic_status de la invitación a pendiente o aceptada 3. Incluye el caso de uso: F1 / 08
Camino Alternativo / Extiende	
- El usuario ingresa direcciones de mail inválidas. - La invitación ya existe y tiene logic_status en pendiente o aceptada	

22.2.2 Pantallas





22.2.3 Diagrama de Secuencia



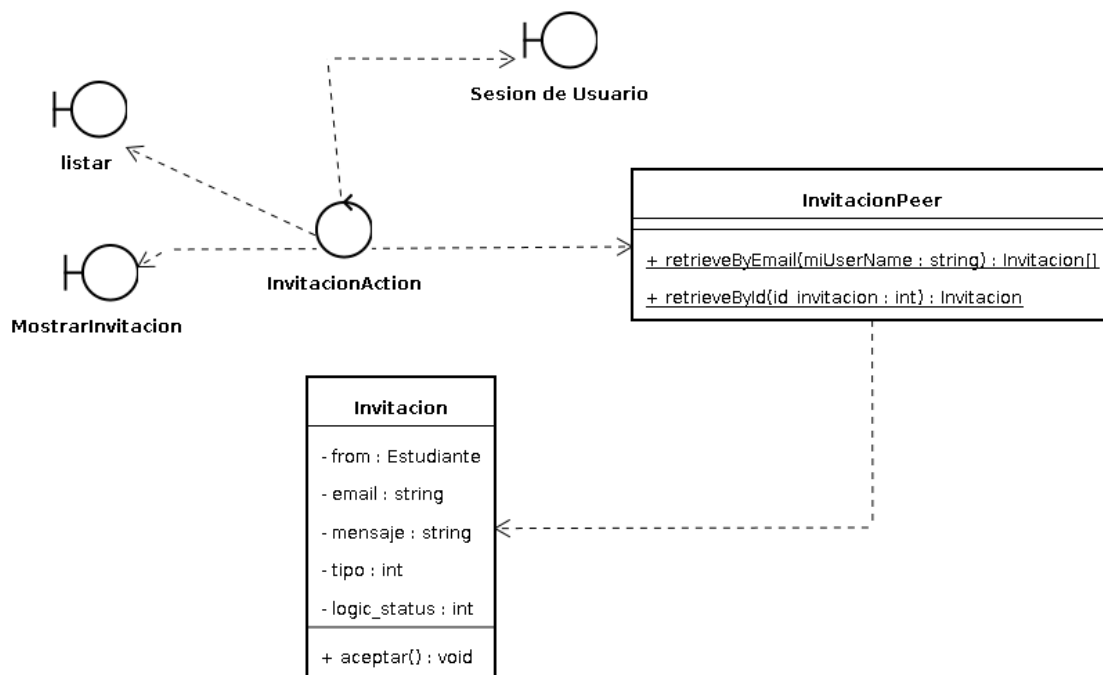


22.3 Ver Invitaciones Recibidas

22.3.1 Ficha de Caso de Uso

Ver Invitaciones Recibidas	
Código	F2 / 02
Actor	Estudiante
Código WBS	205
Flujo Normal / User Stories	
1. El Estudiante ingresa al sistema y hace clic en el menú Amigos, en la opción Invitaciones Recibidas 2. El Estudiante debe ser capaz de ver las invitaciones de amistad que le han enviado. 3. Además de lo anterior el Estudiante tendrá un link al lado de cada una de las invitaciones para poder ver mas detalles sobre las mismas 4. El Estudiante realizará un clic sobre el link ver Invitación y se mostrara mas detalle sobre la invitación recibida	
Condiciones	
Pre Condiciones	1. El estudiante que recibe la invitación debe existir 2. El estudiante que envía la invitación debe existir 3. La invitación debe existir y tener logic_status a pendiente
Pos Condiciones	Ninguna
Camino Alternativo / Extiende a	
Ninguno	

22.3.2 Diagrama de Clases





22.3.3 Pantallas





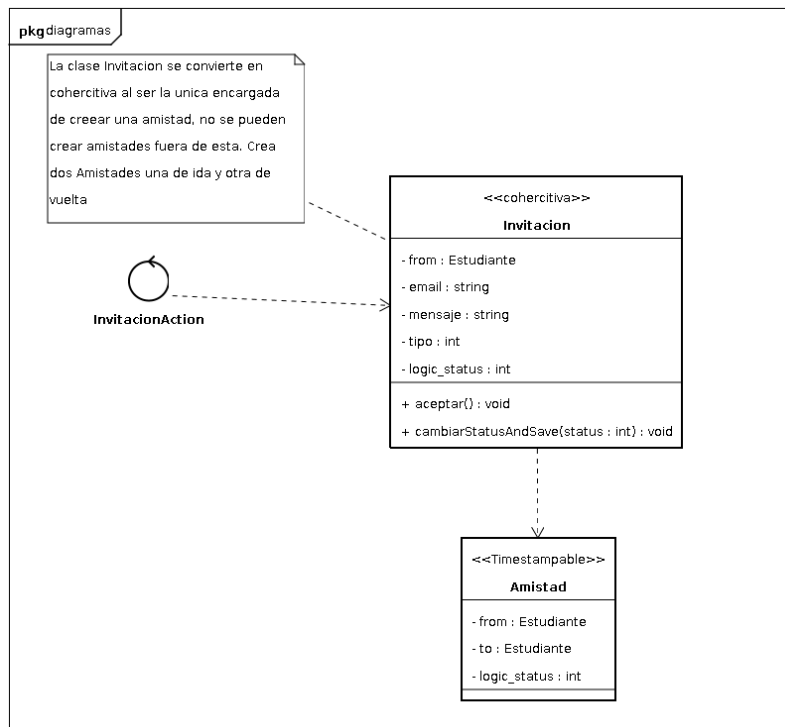
22.4 Cambiar Estado de las Invitaciones de Amistad

22.4.1 Ficha de Caso de Uso

Cambiar Estado de las Invitaciones de Amistad	
Código	F2 / 03
Actor	Estudiante
Código WBS	206
Flujo Normal / User Stories	
<ol style="list-style-type: none"> 1. El estudiante ingresa a la aplicación, luego hace clic en el menú amigos y en la opción Invitaciones Recibidas. 2. Luego hace clic en la invitación recibida a su elección en el link <u>Ver Invitación</u> 3. Finalmente hace clic en los links <u>aceptar</u> o <u>rechazar</u> la invitación. 4. El sistema lo redirecciona al listado de invitaciones pendientes y la invitación ya no debe existir. 	
Condiciones	
Pre Condiciones	Incluye el UC F2 / 02 La invitación debe estar en estado pendiente
Pos Condiciones	Se establece el estado de la invitación a Aceptada. Se crean una Amistad desde el estudiante al invitado y se establece el estado de la amistad a Vigente Se crea otra Amistad desde el invitado al Estudiante y se establece el estado de la amistad a Vigente
Camino Alternativo / Extiende	
- El estudiante rechaza la invitación y se establece el estado de la invitación a rechazada.	



22.4.2 Diagrama de Clases

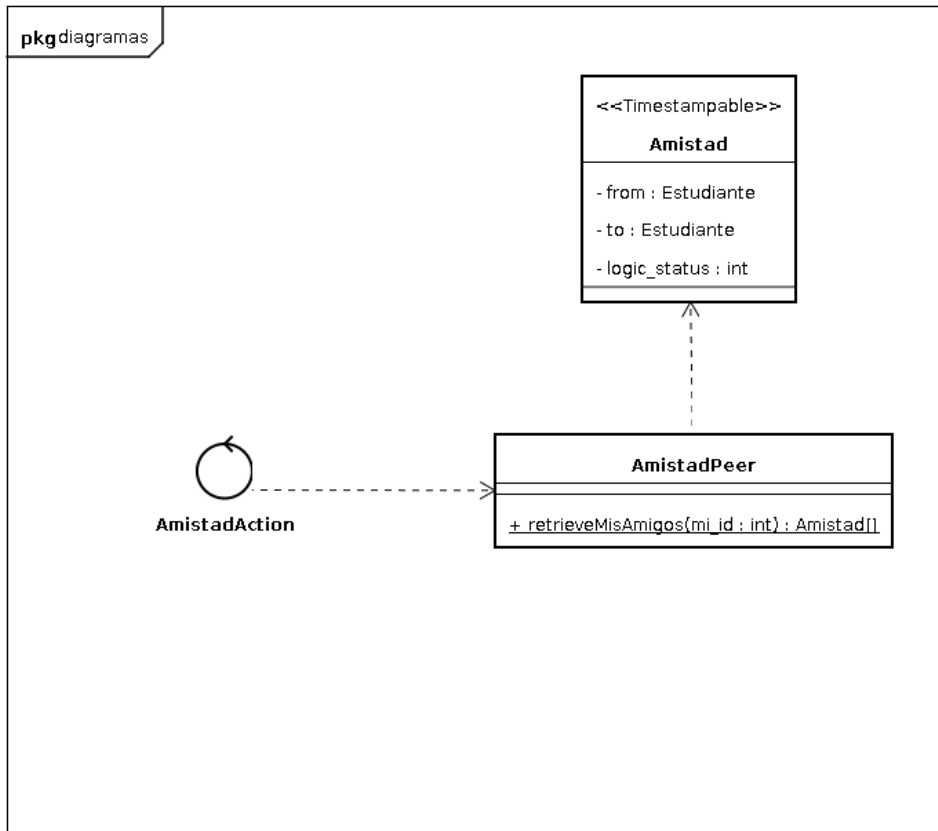


22.5 Listar Amigos

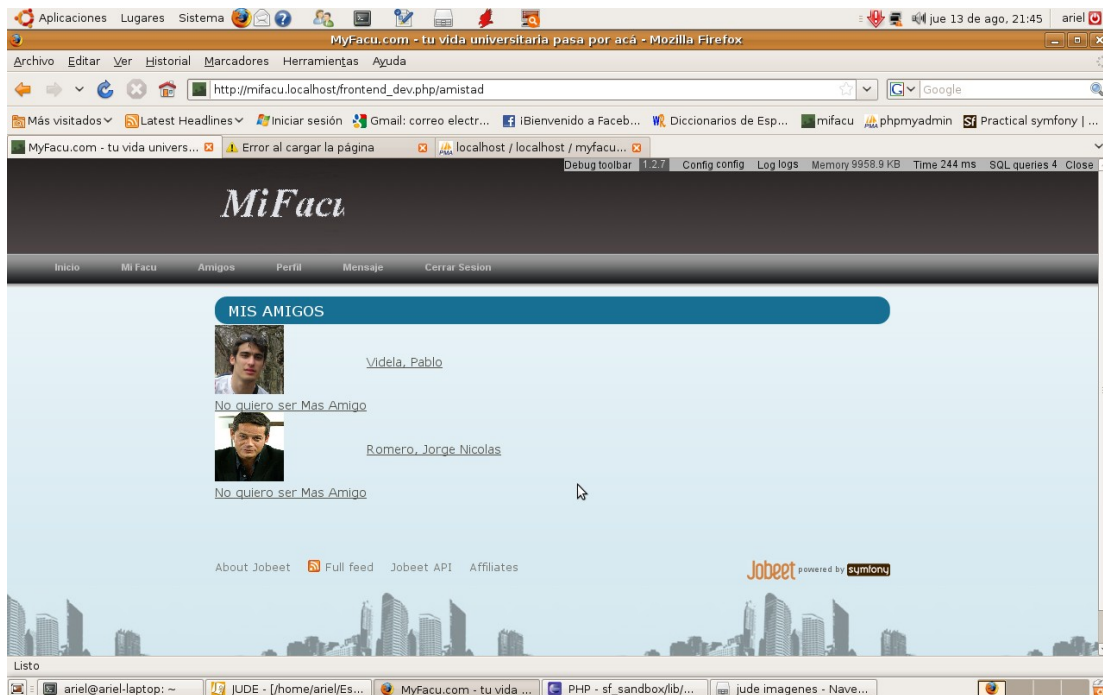
22.5.1 Ficha de Caso de Uso

Listar Amigos	
Código	F2 / 04
Actor	Estudiante
Código WBS	207
Flujo Normal / User Stories	
1. El estudiante ingresa al sistema y hace clic en el menú <u>Amigos</u> 2. El Sistema le mostrara todas las amistades que tiene junto con el link <u>No quiero ser mas amigo</u> 3. Las amistades que se mostraran serán únicamente las activas	
Condiciones	
Pre Condiciones	El esturidante debe existir Las Amistades deben existir y tener el estado de activas
Pos Condiciones	Ninguna
Camino Alternativo / Extiende	
Ninguna	

22.5.2 Diagrama de Clases



22.5.3 Pantallas



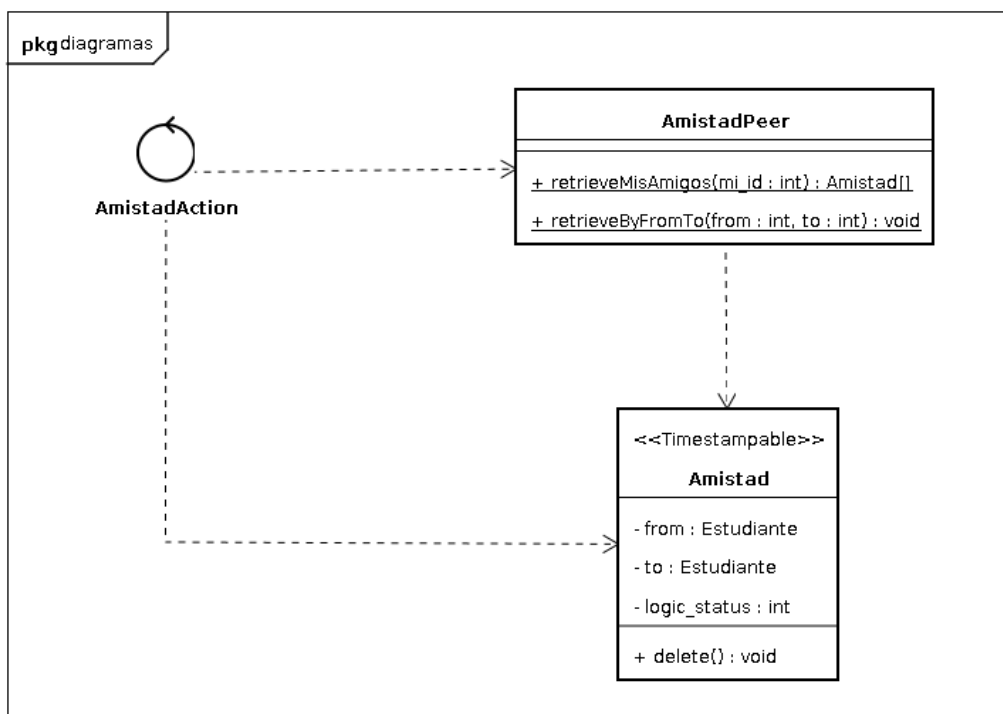


22.6 Cambiar Estado de los Amigos

22.6.1 Ficha de Caso de Uso

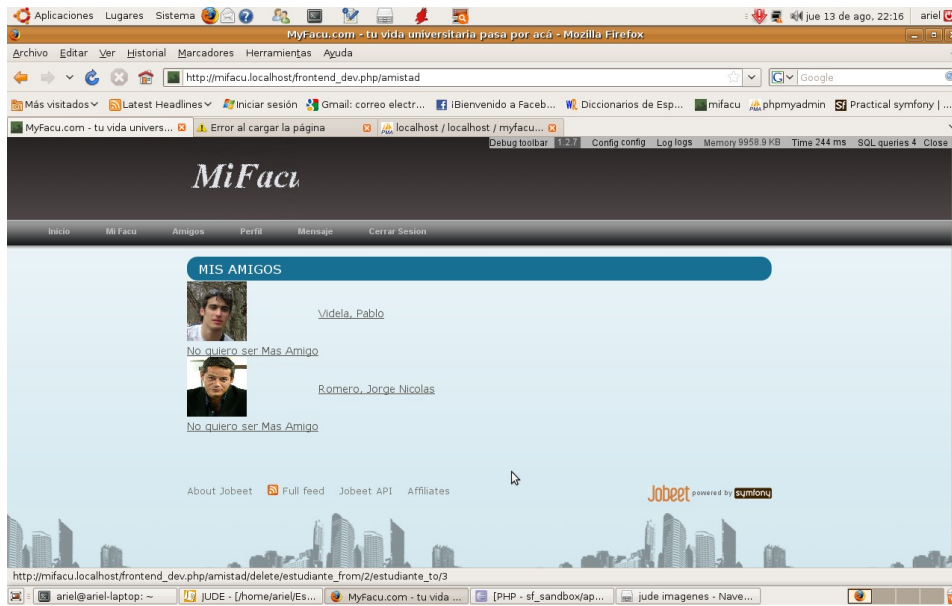
Cambiar Estado de los Amigos	
Código	F2 / 05
Actor	Estudiante
Código WBS	208
Flujo Normal / User Stories	
1. Incluye el Caso de Uso F2 / 04 2. El Estudiante Hace clic en el link <u>No quiero ser mas amigo</u> 3. El sistema solicita la confirmación de la acción 4. El estudiante confirma la acción	
Condiciones	
Pre Condiciones	El Estudiante debe existir El Estudiante Amigo debe existir La Amistad del Estudiante y el Amigo debe existir La Amistad del Amigo y el Estudiante debe existir
Pos Condiciones	Es borrada la Amistad Estudiante - Amigo Se borra la Amistad Amigo - Estudiante
Camino Alternativo / Extiende	
Ninguno	

22.6.2 Diagrama de Clases





22.6.3 Pantallas





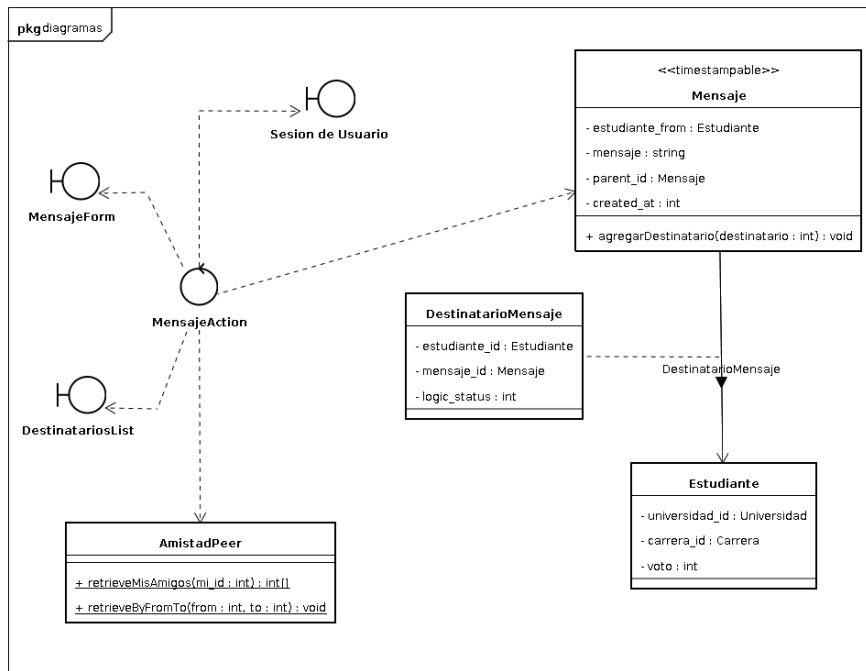
22.7 Redactar Mensaje

22.7.1 Ficha de Caso de Uso

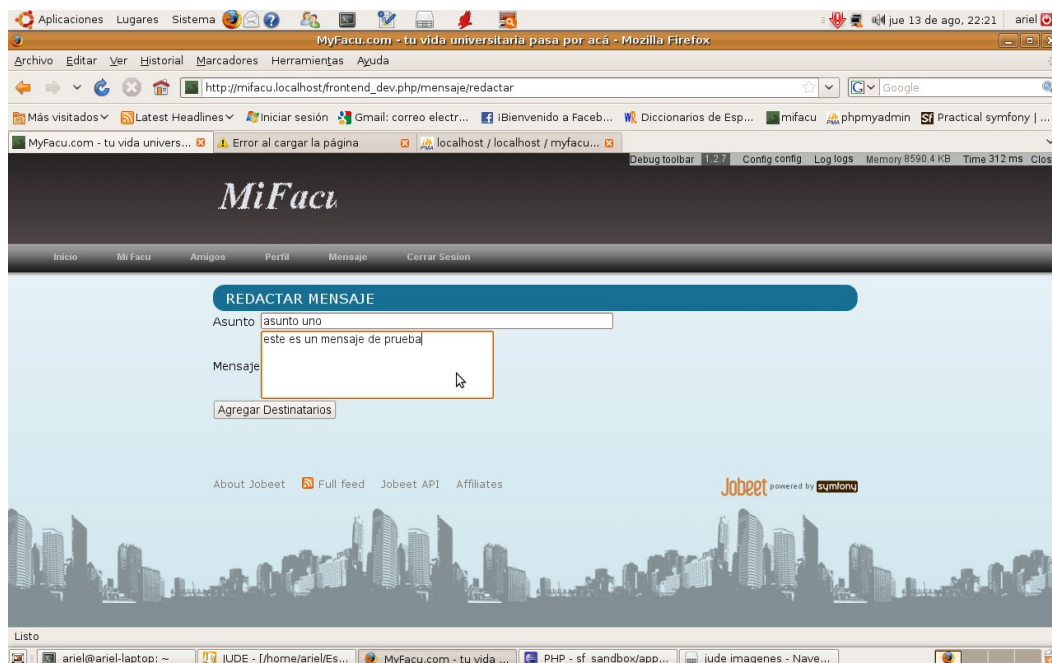
Redactar Mensaje	
Código	F2 / 06
Actor	Estudiante
Código WBS	209
Flujo Normal / User Stories	
<ol style="list-style-type: none"> 1. El Estudiante ingresa a la aplicación y hace clic en el menú Mensaje y luego en la opción Mensaje. 2. Ingresa cual es el asunto y el mensaje que quiere escribir 3. Hace clic en el botón [Agregar Destinatarios] 4. El sistema le permite agregar uno por uno los destinatarios a los cuales le quiere enviar el mensaje. 5. El sistema también debe permitir agregar todos los destinatarios con un solo clic 6. Finalmente hace clic en el botón [Enviar Mensaje] 	
Condiciones	
Pre Condiciones	<ol style="list-style-type: none"> 1. El estudiante debe existir 2. Los destinatarios del mensaje deben existir
Pos Condiciones	<ol style="list-style-type: none"> 1. Se crea un Mensaje. 2. Se crean tantos DestinatarioMensaje como destinatarios tenga el mensaje 3. Se establecen los estados de DestinatarioMensaje a no leído
Camino Alternativo / Extiende	
<ul style="list-style-type: none"> - El Estudiante vuelve a la pantalla de Redactar Mensaje - El estudiante no selecciona ningún destinatario - El destinatario hace clic en el link Agregar Amigos. Es Extendido por el UC F2 / 01 	

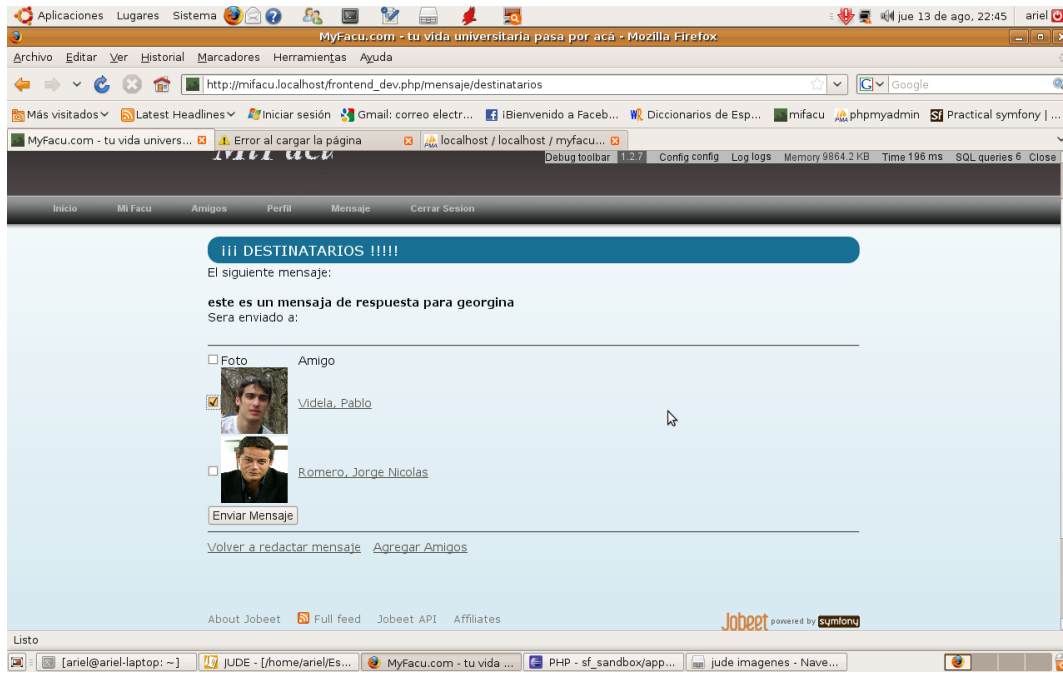


22.7.2 Diagrama de Clase

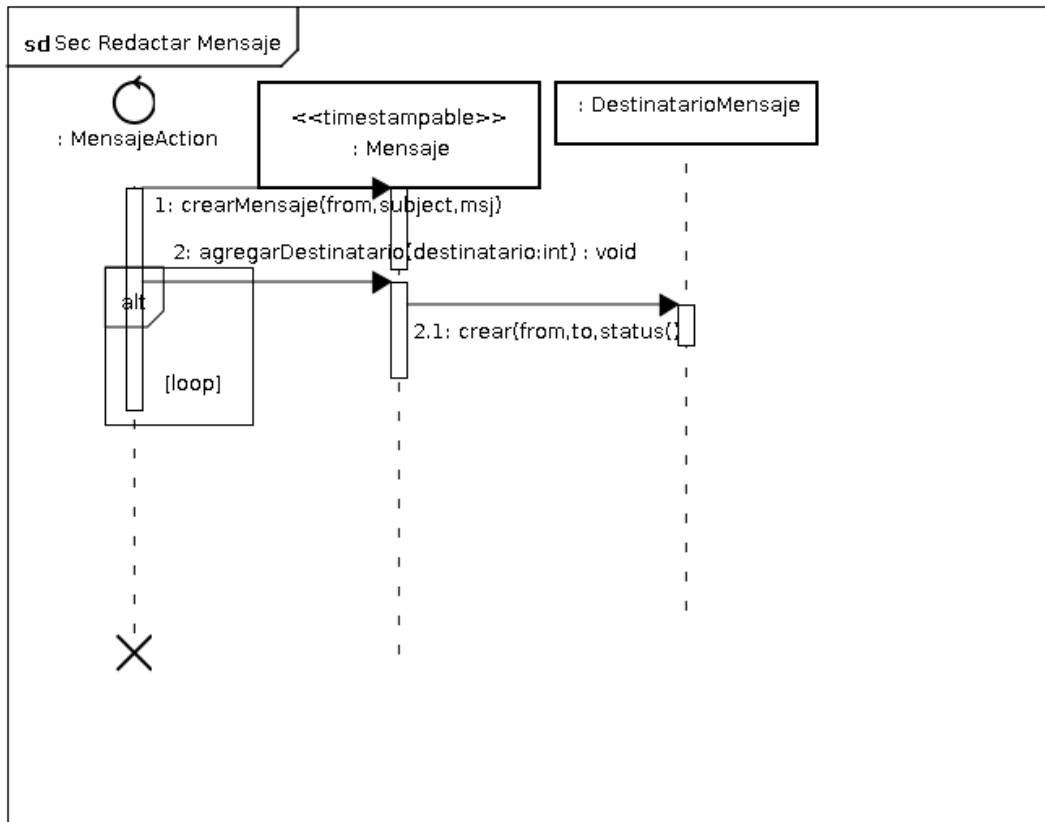


22.7.3 Pantallas





22.7.4 Diagrama de Secuencia



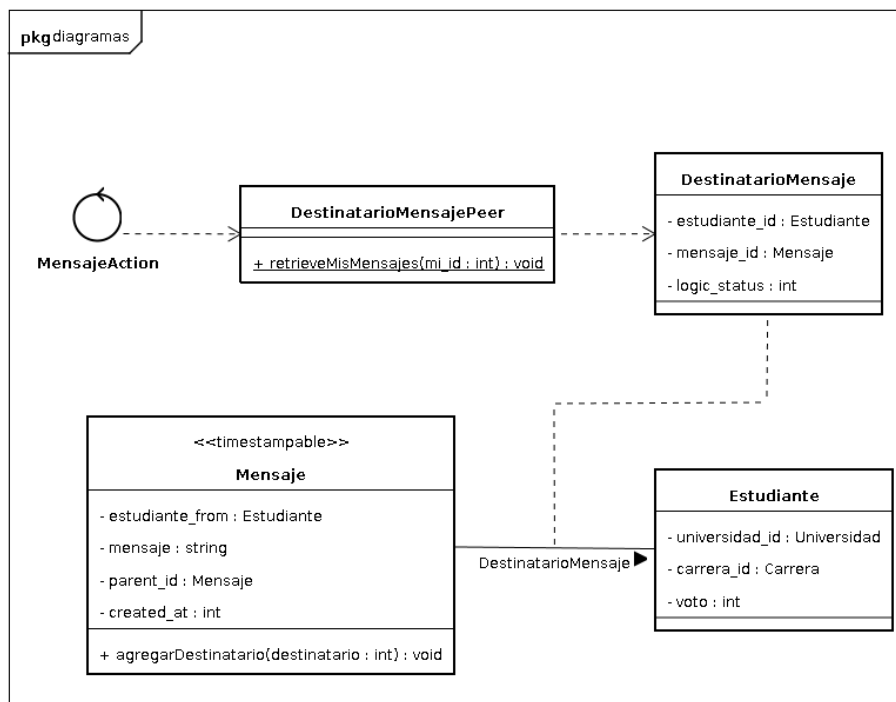


22.8 Ver Bandeja de Entrada

22.8.1 Ficha de Caso de Uso

Ver Bandeja de Entrada	
Código	F2 / 07
Actor	Estudiante
Código WBS	210
Flujo Normal / User Stories	
El estudiante ingresa al sistema y hace clic en el menú Mensajes y luego en la opción Bandeja de Entrada En la bandeja de entrada encontrara todos los mensajes que le enviaron, junto con el asunto y el mensaje como vinculo al mensaje Los mensajes se deberán ordenar de acuerdo a la fecha de emisión y si fueron leídos o no.	
Condiciones	
Pre Condiciones	El Estudiante debe existir Los Mensajes deben existir y tener el logic_status a leídos o pendientes de leer
Pos Condiciones	Ninguna
Camino Alternativo / Extiende	
ninguna	

22.8.2 Diagrama de Clases





22.8.3 Pantallas



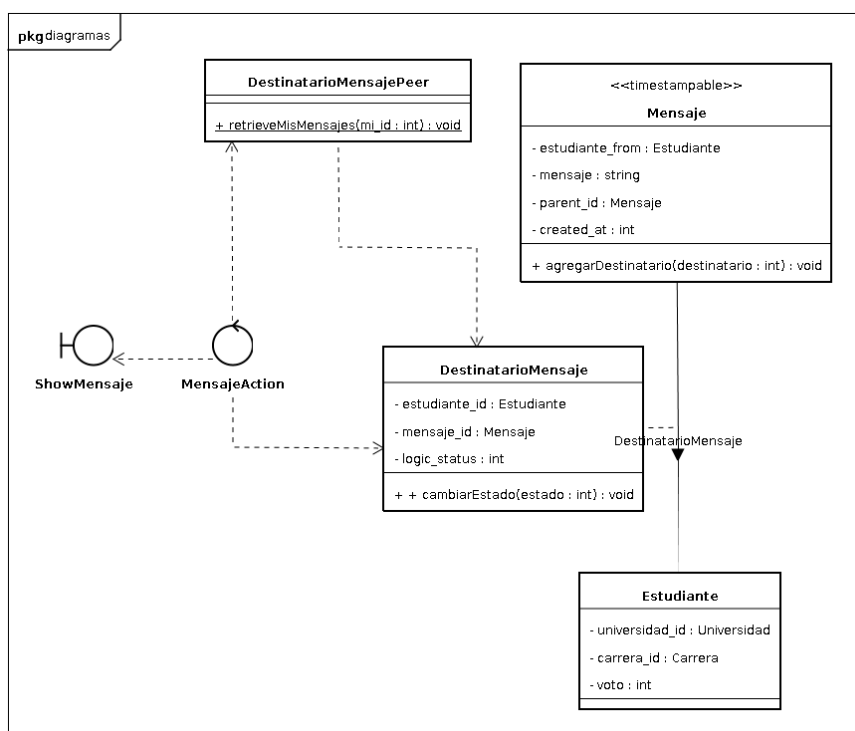


22.9 Leer Mensaje

22.9.1 Ficha de Caso de Uso

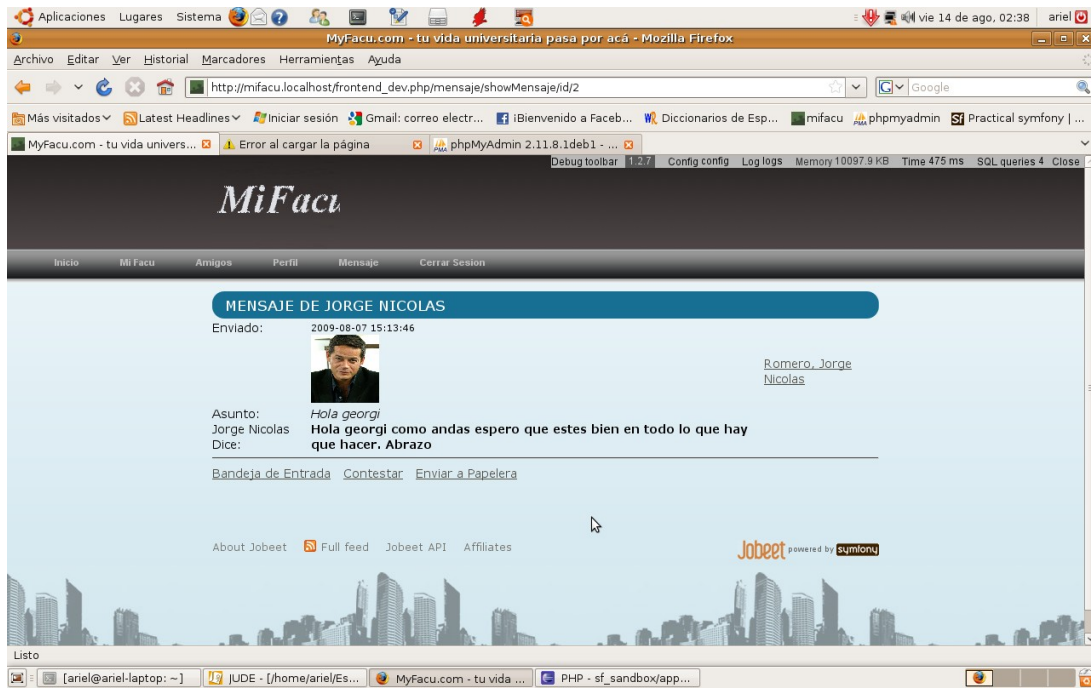
Leer Mensaje	
Código	F2 /08
Actor	Estudiante
Código WBS	204
Flujo Normal / User Stories	
El estudiante ingresa al sistema hace clic en el menú Mensaje y en la opción Bandeja de Entrada. Después selecciona uno de los mensaje y hace clic en el asunto o el mensaje del mismo El sistema mostrara el mensaje, el autor del mensaje, la fecha en que fue enviado y las opciones para enviar a la papelera o contestar el mensaje.	
Condiciones	
Pre Condiciones	El Estudiante debe existir El mensaje debe existir
Pos Condiciones	El logic_status del mensaje se establece en leído
Camino Alternativo / Extiende	
El Estudiante hace clic en volver a bandeja de entrada Extiende el caso de uso: F2 / 07	

22.9.2 Diagrama de Clase





22.9.3 Pantalla





22.10 Contestar Mensaje

22.10.1 Ficha de Caso de Uso

Contestar Mensaje	
Código	F2 / 09
Actor	Estudiante
Código WBS	212
Flujo Normal / User Stories	
<ol style="list-style-type: none"> 1. Incluye el Caso de Uso: F2 / 08 2. El estudiante ingresa al sistema y hace clic en el menú Mensaje y luego en la opción Bandeja de Entrada, selecciona un mensaje y hace clic en el vínculo con el <u>asunto</u> o el mensaje. 3. Luego de leer el mensaje el Estudiante hace clic en el vinculo <u>contestar</u> 4. En la ventana del mensaje de respuesta debe aparecer el mensaje original. 5. El asunto del mensaje de respuesta inmediatamente se llena con el asunto de mensaje a contestar y el prefijo 'RE:' 6. Luego de escribir el mensaje, finalmente el estudiante hace clic en el botón [Enviar Mensaje] 7. Aparece un mensaje flash de usuario que dice "Su mensaje ha sido enviado" 	
Condiciones	
Pre Condiciones	El Estudiante debe Existir El mensaje debe existir
Pos Condiciones	Se crea una nuevo Mensaje El mensaje al que responde se persiste como mensaje padre del mensaje enviado. Se establece el logic_status del DestinatarioMensaje Original como Contestado.
Camino Alternativo / Extiende	
Ninguno	



22.11 Cambiar Estado de Mensaje

22.11.1 Ficha de Caso de Uso

Cambiar Estado de Mensaje	
Código	F2 / 10
Actor	Estudiante
Código WBS	213
Flujo Normal / User Stories	
<p>Incluye el Caso de Uso: F2 / 07</p> <p>El estudiante ingresa al sistema y hace clic en el menú Mensaje y luego en la opción Bandeja de Entrada.</p> <p>Luego selecciona el mensaje al cual le quiere cambiar el estado haciendo clic en el checkbox al costado del mensaje</p> <p>Selecciona una acción para ejecutar (mensaje) de la lista desplegable</p> <p>Finalmente hace clic en botón que [Ejecutar Acción]</p>	
Condiciones	
Pre Condiciones	El Estudiante debe Existir El / Los DestinatarioMensaje deben existir
Pos Condiciones	Se establece el logic_status del DestinatarioMensaje al estado que el estudiante selecciono.
Camino Alternativo / Extiende	
Ninguno.	



ELABORACIÓN / TERCERA FASE

- Listado de Casos de Uso
- Casos de Uso del Sistema



23 Tercera Fase

23.1 Listado de Casos de Uso

La tercera fase cuenta con una mayor cantidad de casos de uso. Ya que se presupone el aprendizaje y maduración de algunos conceptos del negocio devenidos de la elaboración de la documentación, el desarrollo e implementación de las dos fases anteriores. Por la misma razón se procederá a un desarrollo menos prescriptivo en cuanto a la documentación, sin dejar de lado la justa y suficiente explicación de cada caso de uso en particular.

Casos de Uso Funcionales		
<i>Hacen referencia a las funcionalidades que el sistema debe realizar</i>		
<i>Notación código: Fase/N°UC</i>		
Reque.	Código	Nombre UC
Tercera Fase		
RF19	F3/01	Agregar Evaluador
RF20	F3/02	Registrar Examen
RF21	F3/03	Agregar Tema a una Materia
RF22	F3/04	Listar Temas
RF23	F3/05	Seleccionar Preguntas de Examen
RF24	F3/06	Agregar Pregunta de Examen
RF25	F3/07	Cambiar Estado Materias
RF26	F3/08	Escribir Mi Pregunta
RF27	F3/09	Listar Mis Preguntas
RF28	F3/10	Ver respuestas
RF29	F3/11	Listar Preguntas a Responder
RF30	F3/12	Responder Preguntas
RF31	F3/13	Publicar Noticia
RF32	F3/14	Ver Mis Noticias



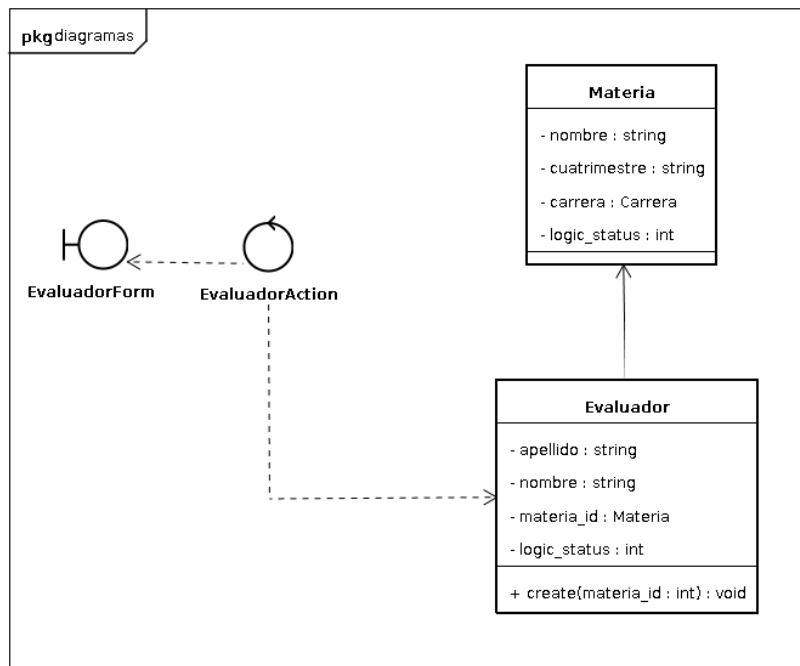
23.2 Agregar Evaluador

23.2.1 Ficha de Caso de Uso

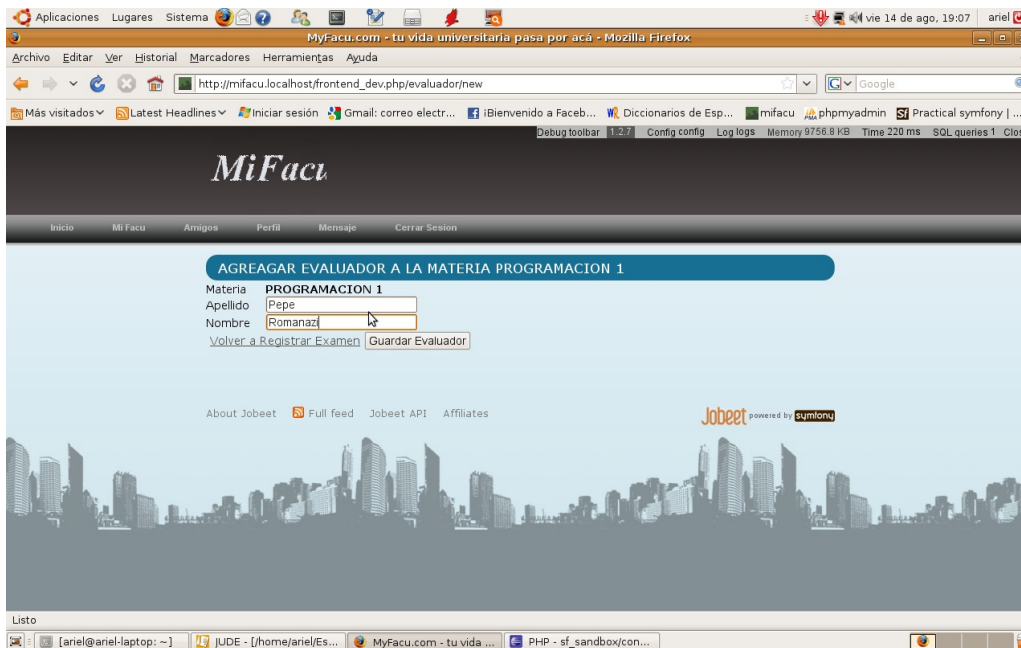
Agregar Evaluador	
Código	F3 / 01
Actor	Estudiante
Código WBS	288
Flujo Normal / User Stories	
<ol style="list-style-type: none"> 1. El usuario entra en el sistema y hace clic en el menú Mi Facu y la opción Mis Materias. 2. Después ante el listado de las materias, el estudiante hace clic en el link <u>registrar examen!!!</u> 3. Al ver que la persona que lo evaluó, no figura en el combo el estudiante hace clic en el link agregar evaluador. 4. El estudiante ingresa el apellido de la persona que lo evaluó y después como opcional el nombre. 5. Finalmente, hace clic en el botón [Guardar Evaluado] y el sistema lo redirecciona a la pantalla de registrar examen. 	
Condiciones	
Pre Condiciones	<ol style="list-style-type: none"> 1. El estudiante debe existir 2. La Materia a la cual se asocia el evaluado debe existir
Pos Condiciones	<ol style="list-style-type: none"> 1. Se crea un Evaluador y se establece el logic_status del evaluador a pendiente de revisión.
Camino Alternativo / Extiende	
Extiende al caso de Uso Registrar Examen (F3 / 02)	



23.2.2 Diagrama de Clases



23.2.3 Pantallas



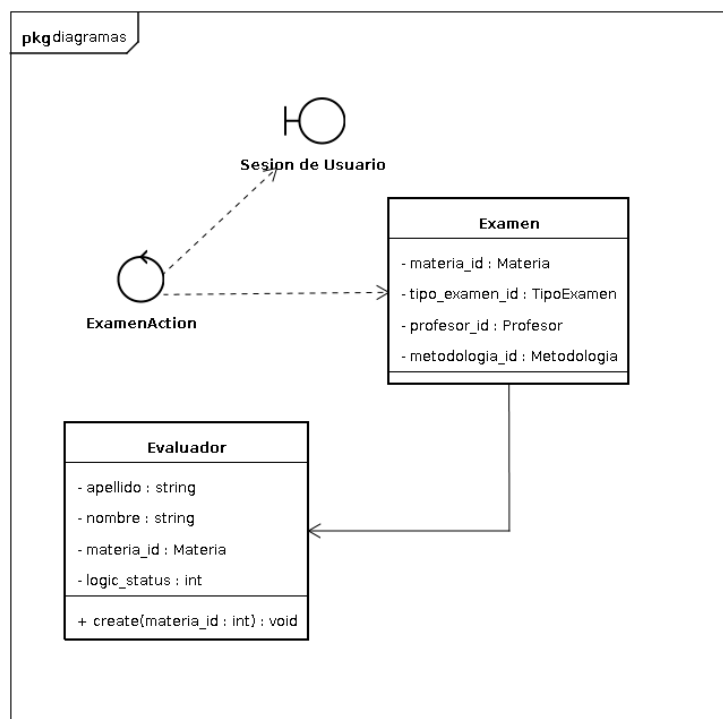


23.3 Registrar Examen

23.3.1 Ficha de Caso de Uso

Registrar Examen	
Código	F3 / 02
Actor	Estudiante
Código WBS	289
Flujo Normal / User Stories	
<ol style="list-style-type: none"> 1. El usuario entra en el sistema y hace clic en el menú Mi Facu y la opción Mis Materias. 2. Después ante el listado de las materias, el estudiante hace clic en el link <u>registrar examen!!!</u> 3. Que se encuentra al lado de cada materia en particular. 4. Selecciona quien fue la persona que lo evaluó, cual fue el tipo de examen rendido, y que metodología se utilizo para evaluarlo. 5. Finalmente selecciona guardar. 	
Condiciones	
Pre Condiciones	<ol style="list-style-type: none"> 1. El estudiante debe existir 2. La materia debe existir
Pos Condiciones	Se crea un Examen y se lo relaciona a la Materia rendida
Camino Alternativo / Extiende	
El estudiante no encuentra la persona que le tomo para seleccionar	

23.3.2 Diagrama de Clases



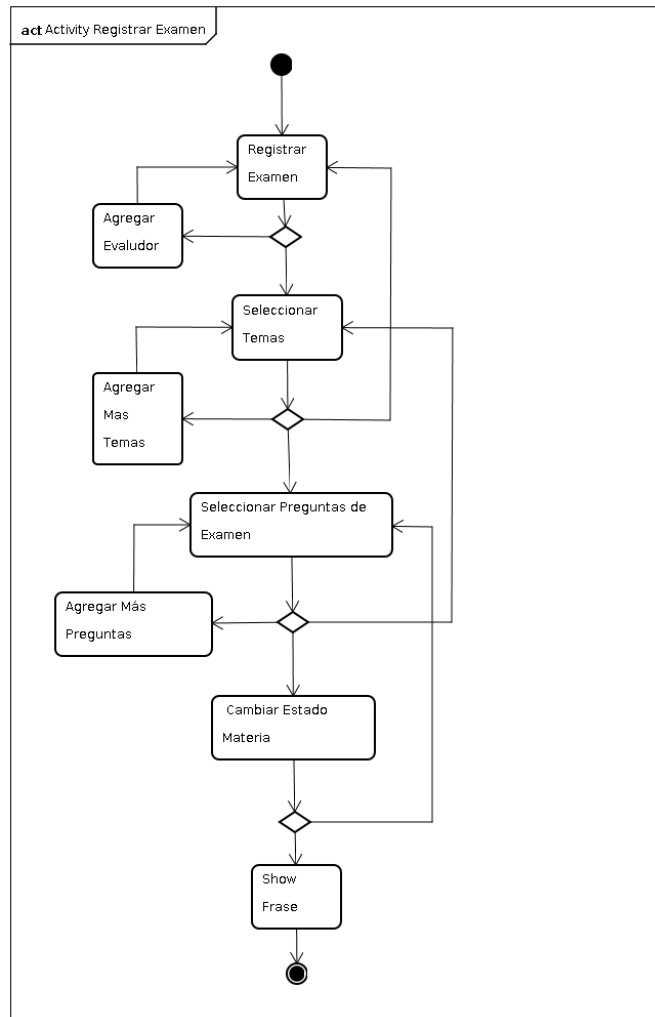


23.3.3 Pantalla





23.3.4 Diagrama de Actividad



23.4 Agregar Tema a una Materia

23.4.1 Ficha de Caso de Uso

Agregar Tema a una Materia	
Código	F3 / 03
Actor	Estudiante
Código WBS	290
Flujo Normal / User Stories	
1. El estudiante hace clic en el link <u>agregar mas temas a la materia xxxx</u> 2. Ingresar el nombre del tema. No hace falta que el mismo sea un nombre oficial del programa de la materia. 3. Ingresar el grado de dificultad o cuan difícil le pareció a él ese tema en una escala de 1 a 10 y una breve descripción del tema.	
Condiciones	
Pre Condiciones	1. El estudiante debe existir y haber sido subido a sesión



	2. La Materia debe Existir
Pos Condiciones	Se crea un Tema y se establece el logic_status a pendiente
Camino Alternativo / Extiende	
Extiende al caso de uso: Listar Temas (F3 / 04)	



23.5 Listar Temas

23.5.1 Ficha de Caso de Uso

Listar Temas	
Código	F3 / 04
Actor	Estudiante
Código WBS	291
Flujo Normal / User Stories	
1. Es incluido el UC: Registrar Examen (F3 / 02) 2. El usuario podrá ver cuales son los temas de la materia disponibles. 3. Estos temas serán públicos. Es decir que corresponderán a todos los usuarios del sistema. 4. Se priorizarán aquellos que mayor grado de ponderación tengan, en cuanto a votos.	
Condiciones	
Pre Condiciones	1. El Estudiante debe existir 2. La Materia debe existir
Pos Condiciones	Ninguna.
Camino Alternativo / Extiende	
Ninguna	

23.5.2 Pantallas





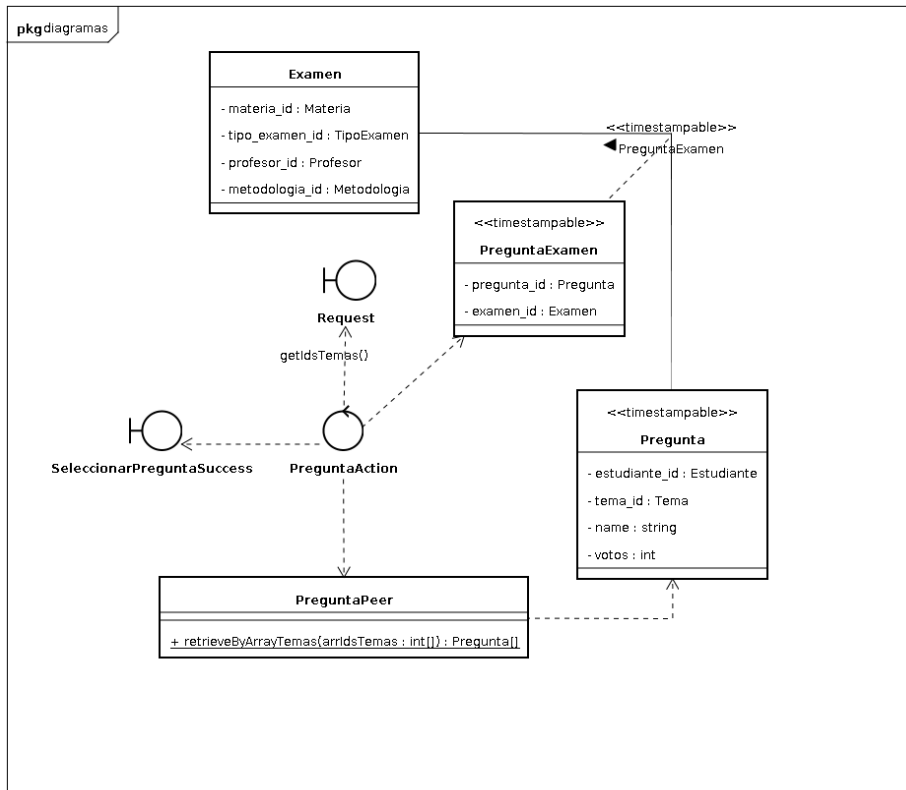
23.6 Seleccionar Preguntas de Examen

23.6.1 Ficha de Caso de Uso

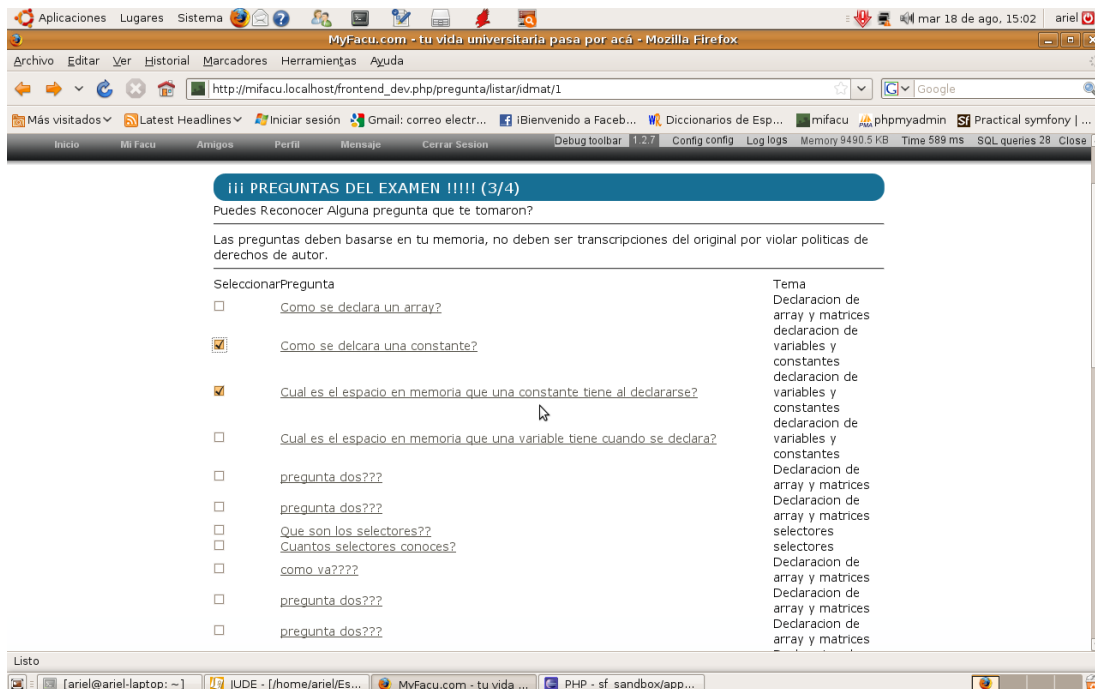
Seleccionar Preguntas de Examen	
Código	F3 / 05
Actor	Estudiante
Código WBS	292
Flujo Normal / User Stories	
<ol style="list-style-type: none"> 1. El estudiante ingresa al sistema, hace clic en el menú mi Facu y a continuación en la opción mis materias. Luego hace clic en el vínculo <u>Registrar Examen!!!</u> 2. Registra el examen, luego selecciona los temas que le tomaron y finalmente tiene que seleccionar las preguntas que le realizaron. 3. Dichas preguntas se listaran con un casillero a su izquierda para poder seleccionar aquellas que ya se han realizado. 4. Además de ello se deberán mostrar ponderadas por sus votos. 	
Condiciones	
Pre Condiciones	<ol style="list-style-type: none"> 1. El estudiante debe existir 2. La materia debe existir. 3. Los Temas deben Existir
Pos Condiciones	<ol style="list-style-type: none"> 1. Se crea nuevas PreguntaExamen 2. Se le asignará un voto_examen a la Pregunta.
Camino Alternativo / Extiende	
<ol style="list-style-type: none"> 1. El estudiante cancela el asistente. 2. Si no existiesen preguntas sobre el tema. El caso de Uso será extendido directamente por el Caso de Uso Agregar Preguntas de Examen (F3 / 06) 	



23.6.2 Diagrama de Clase



23.6.3 Pantallas



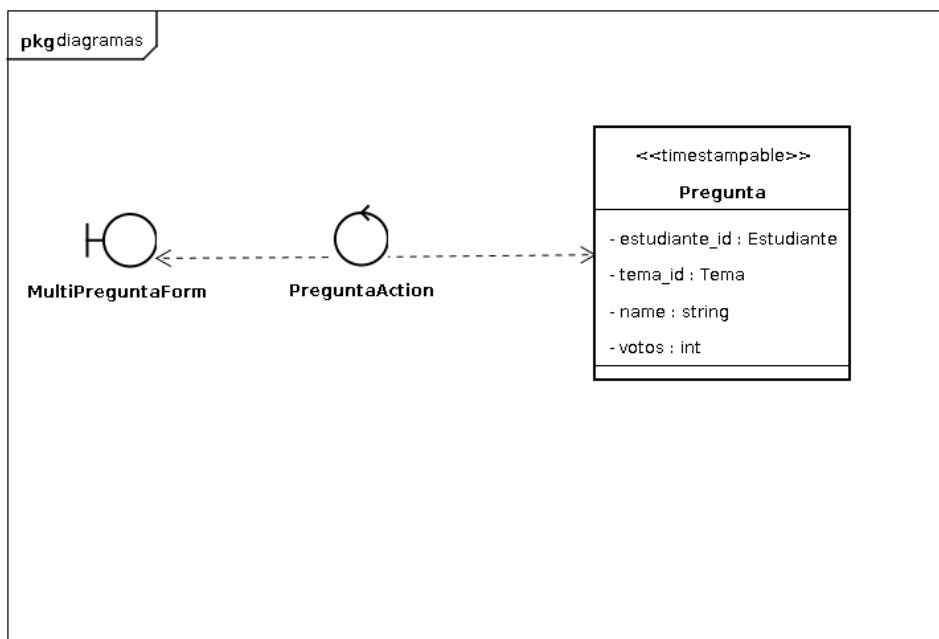


23.7 Agregar Pregunta de Examen

23.7.1 Ficha de Caso de Uso

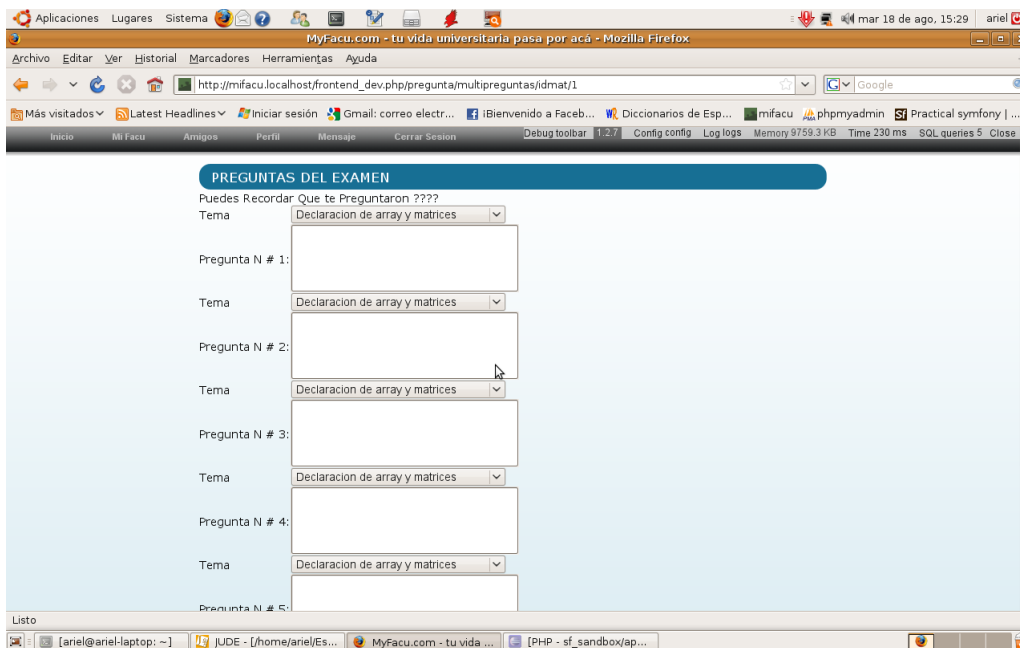
Agregar Pregunta de Examen	
Código	F3 / 06
Actor	Estudiante
Código WBS	293
Flujo Normal / User Stories	
<ol style="list-style-type: none"> Una vez que el Estudiante ha registrado el examen, ha seleccionado los temas que el tomaron y no encuentra las preguntas que el tomaron en el listado que el sistema le ofrece, el estudiante podrá ingresar las preguntas haciendo clic en el vinculo <u>agregar más preguntas que me realizaron</u>. El sistema redireccionará al Estudiante a un “multiformulario”. Donde en un solo envío podrá registrar más de una pregunta. Las preguntas ingresadas deberán aparecer en el listado para ser seleccionarlas. 	
Condiciones	
Pre Condiciones	<ol style="list-style-type: none"> El Estudiante debe existir Los Temas deben haber sido seleccionados. La Pregunta no debe existir o tener una ponderación muy baja.
Pos Condiciones	Se crean nuevas Pregunta
Camino Alternativo / Extiende	
Extiende al Caso de Uso: Seleccionar Preguntas de Examen (F3 / 05)	

23.7.2 Diagrama de Clase





23.7.3 Pantallas





23.8 Cambiar Estado Materias

23.8.1 Ficha de Caso de Uso

Cambiar Estado Materia	
Código	F3 / 07
Actor	Estudiante
Código WBS	294
Flujo Normal / User Stories	
<ol style="list-style-type: none"> Una vez que el Estudiante avanza en el asistente para registrar los exámenes, los temas y las preguntas que le tomaron, el estudiante podrá registrar si dicho examen ha producido un cambio en el estado de la materia. Para ello deberá seleccionar el cambio de estado en el listado de estados de la materia. Y hacer clic en el botón [Guardar Cambios] 	
Condiciones	
Pre Condiciones	<ol style="list-style-type: none"> El Estudiante debe existir La Materia debe Existir Los Temas deben Existir y haberle asignado un voto examen Las preguntas deben Existir y asignarle un voto Examen
Pos Condiciones	Se cambia el estado de la MateriaEstudiante
Camino Alternativo / Extiende	
El estudiante termina el asistente, si registrar cambio alguno.	



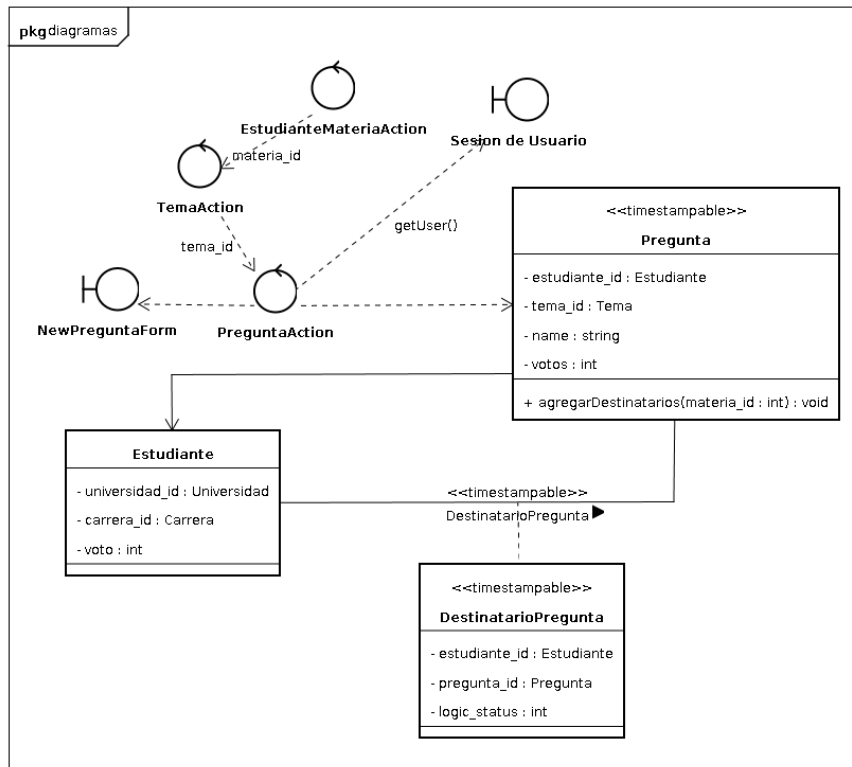
23.9 Escribir Mi Pregunta

23.9.1 Ficha de Caso de Uso

Escribir Mi Pregunta	
Código	F3 / 08
Actor	Estudiante
Código WBS	295
Flujo Normal / User Stories	
<ol style="list-style-type: none"> 1. El estudiante ingresa al sistema y hace clic en el menú Mi Facu, luego selecciona la opción Mis Materias. 2. Para escribir una pregunta el usuario deberá hacer clic en el vínculo que dice <u>Realizar Pregunta</u> que figura al lado de cada materia. 3. El sistema le solicitará que seleccione el tema sobre el cual quiere hacer una pregunta. Si el tema no estuviera listado el estudiante podrá agregar temas. 4. El Estudiante escribe la pregunta a realizar y hace clic en el botón [Enviar Pregunta] 5. El sistema deberá enviar las preguntas a todos aquellos amigos que lo pueden ayudar. 	
Condiciones	
Pre Condiciones	<ol style="list-style-type: none"> 1. El Estudiante debe Existir 2. El estudiante debe tener amigos
Pos Condiciones	<ol style="list-style-type: none"> 1. Se crea una Pregunta 2. Se crean varios DestinatarioPregunta 3. Se establece el estado de DestinatarioPregunta a Enviada.
Camino Alternativo / Extiende	
<ol style="list-style-type: none"> 1. El Estudiante Cancela la pregunta 2. Es Extendido por el caso de uso: Agregar Tema a una Materia (F3 / 03) 	

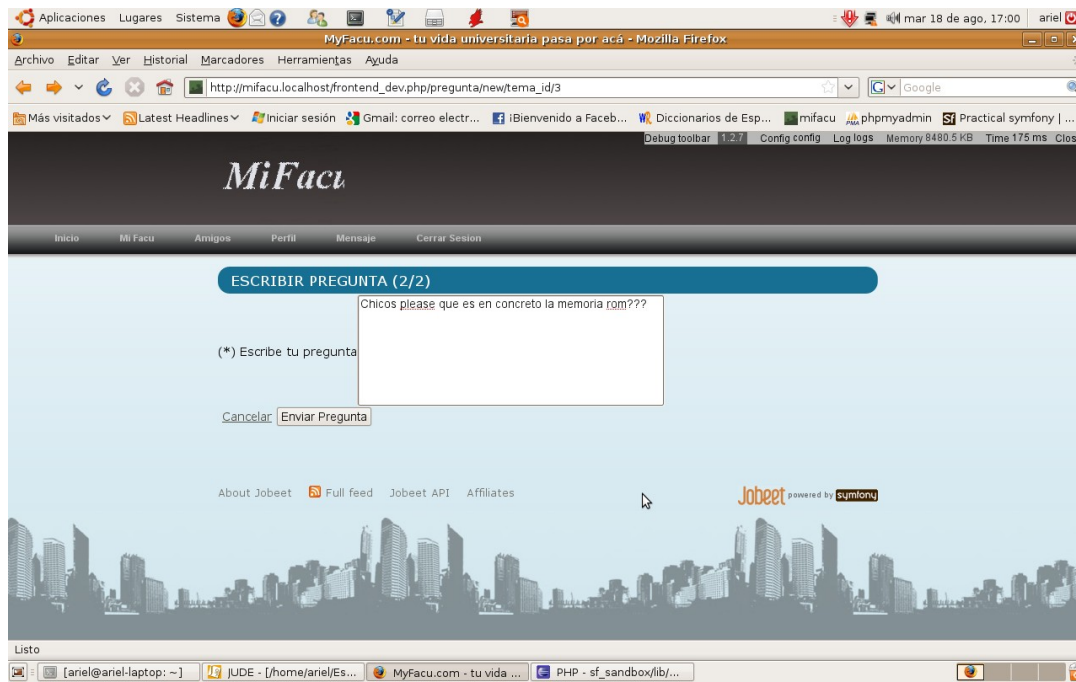


23.9.2 Diagrama de Clase

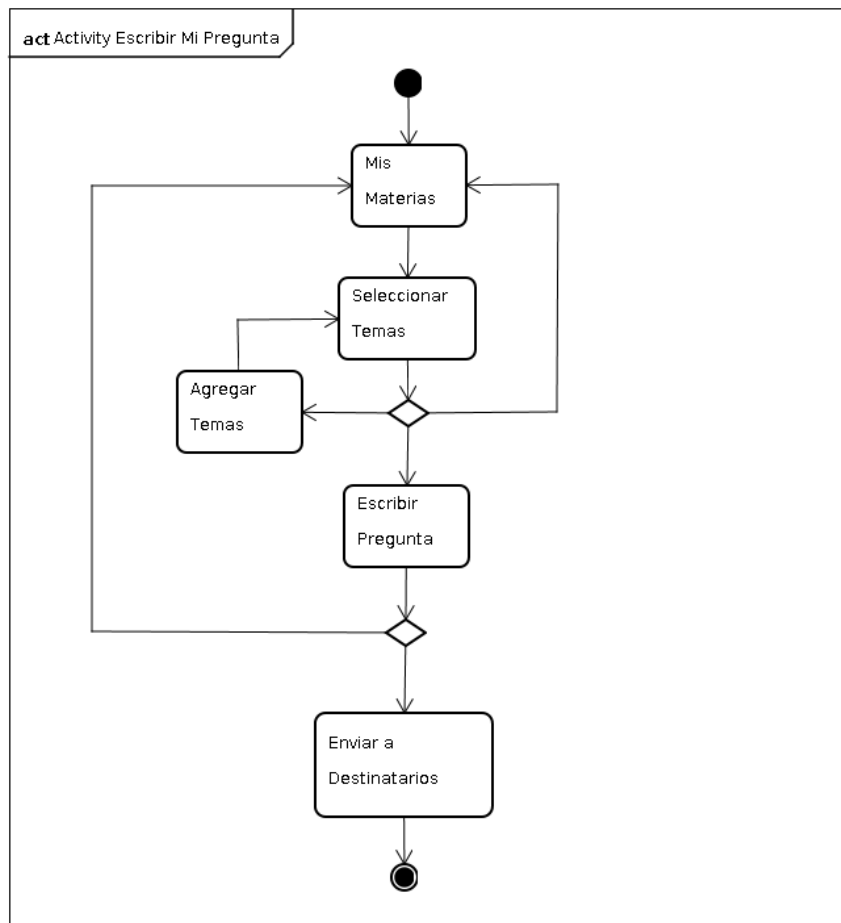


23.9.3 Pantallas





23.9.4 Diagrama de Actividad



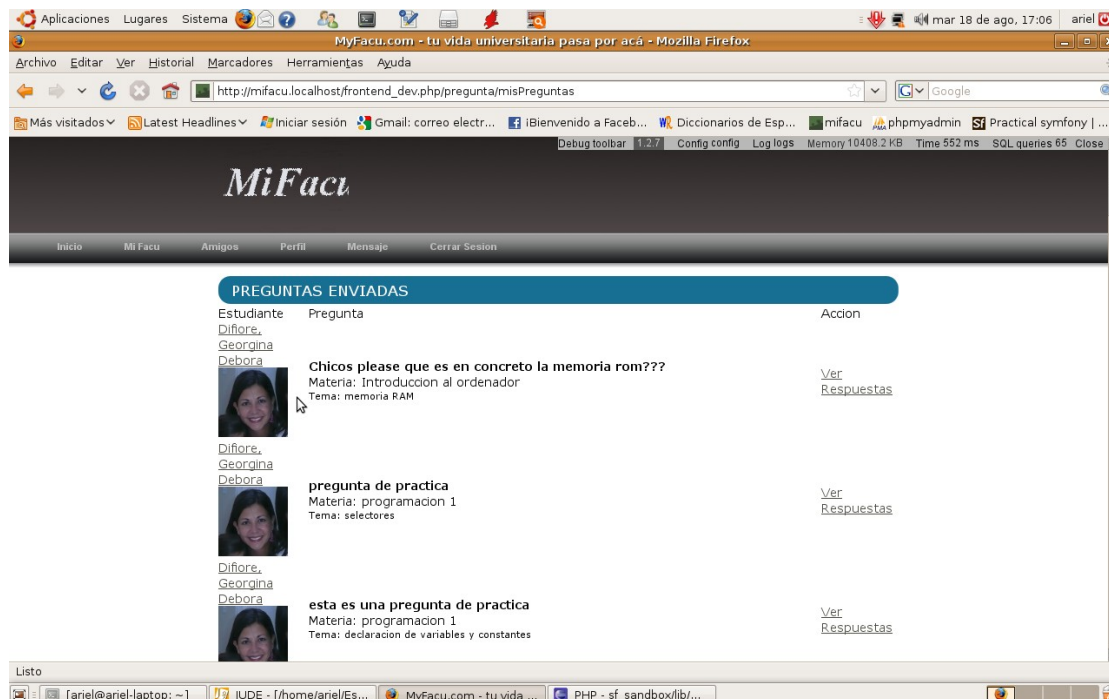


23.10 Listar Mis Preguntas

23.10.1 Ficha de Caso de Uso

Listar Mis Preguntas	
Código	F3 / 09
Actor	Estudiante
Código WBS	296
Flujo Normal / User Stories	
1. El estudiante ingresa al sistema y hace clic en el menú Mi Facu, luego selecciona la opción Ver Respuestas. 2. Las Preguntas Enviadas aparecerán listadas de acuerdo a la fecha en que fueron realizadas.	
Condiciones	
Pre Condiciones	1. El estudiante debe existir 2. El estudiante debe haber realizado preguntas a sus amigos.
Pos Condiciones	Ninguna.
Camino Alternativo / Extiende	
Ninguna.	

23.10.2 Pantallas





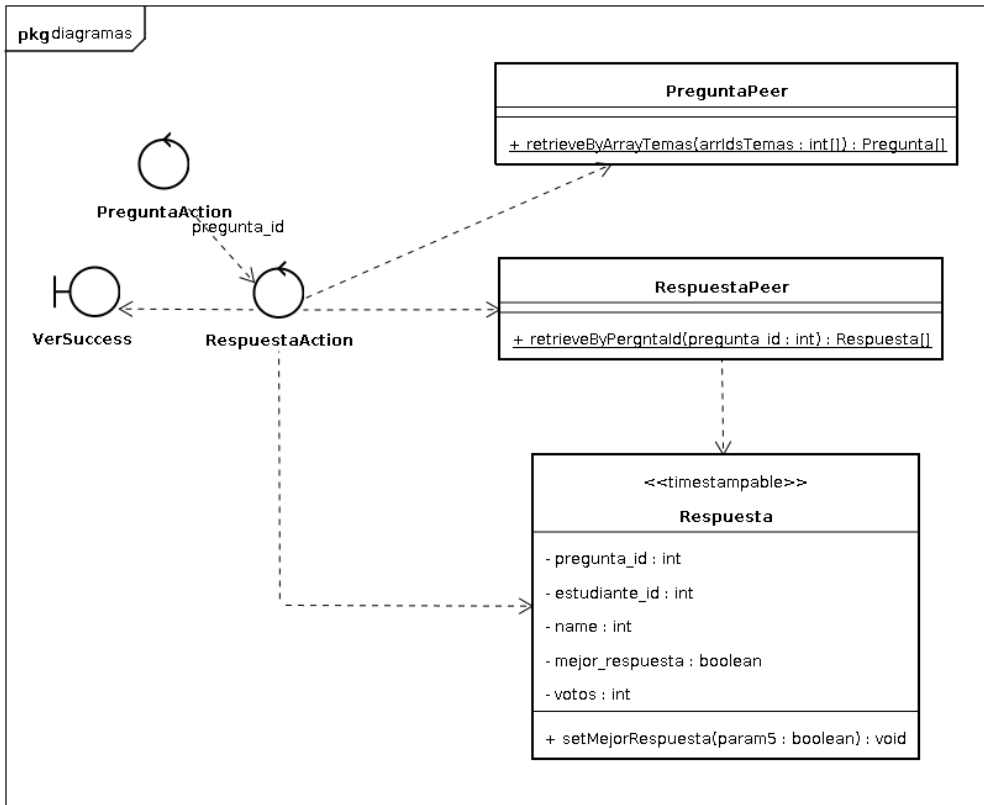
23.11 Ver Respuestas

23.11.1 Ficha de Caso de Uso

Ver Respuestas	
Código	F3 / 10
Actor	Estudiante
Código WBS	297
Flujo Normal / User Stories	
<ol style="list-style-type: none"> 1. El estudiante ingresa al sistema y hace clic en el menú Mi Facu, luego selecciona la opción Ver Respuestas. 2. Selecciona una de las preguntas realizadas y hace clic en el vinculo que dice <u>Ver Respuestas</u> 3. El Sistema deberá mostrarle todas las respuestas enviadas por sus amigos, y permitirle establecer cual de ellas es la mejor respuesta en su opinión. 	
Condiciones	
Pre Condiciones	<ol style="list-style-type: none"> 1. El Estudiante debe Existir 2. La Pregunta debe Existir 3. La/s Respuesta/s Deben Existir
Pos Condiciones	La Respuesta Seleccionada como mejor Respuesta establece su mejor_respuesta a verdadero.
Camino Alternativo / Extiende	
Ninguno.	



23.11.2 Diagrama de Clase



23.11.3 Pantallas





23.12.3Pantallas



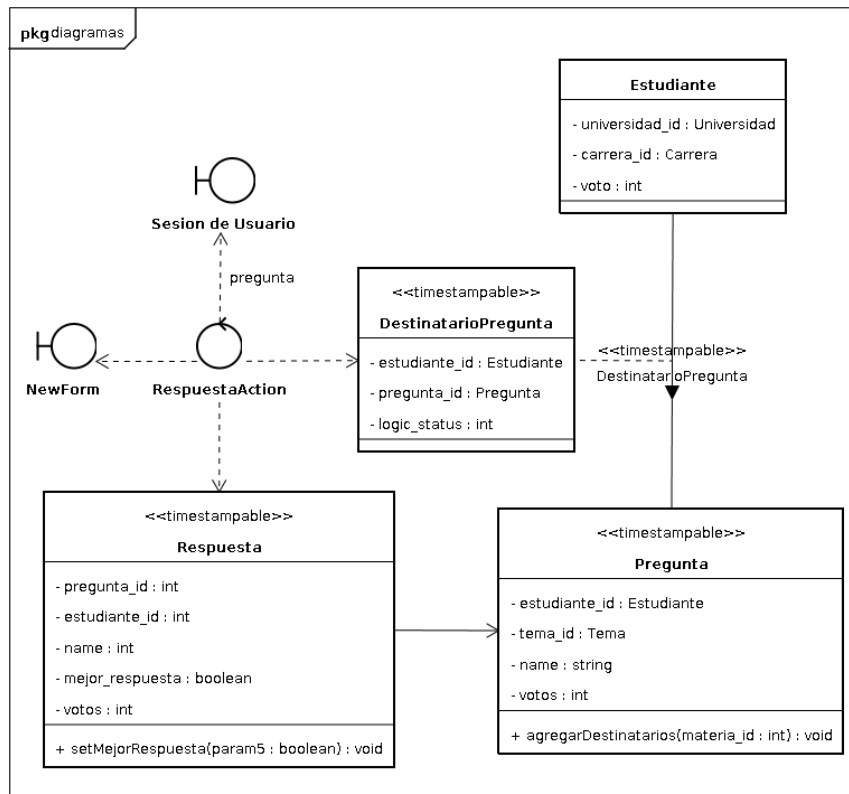


23.13 Responder Preguntas

23.13.1 Ficha de Caso de Uso

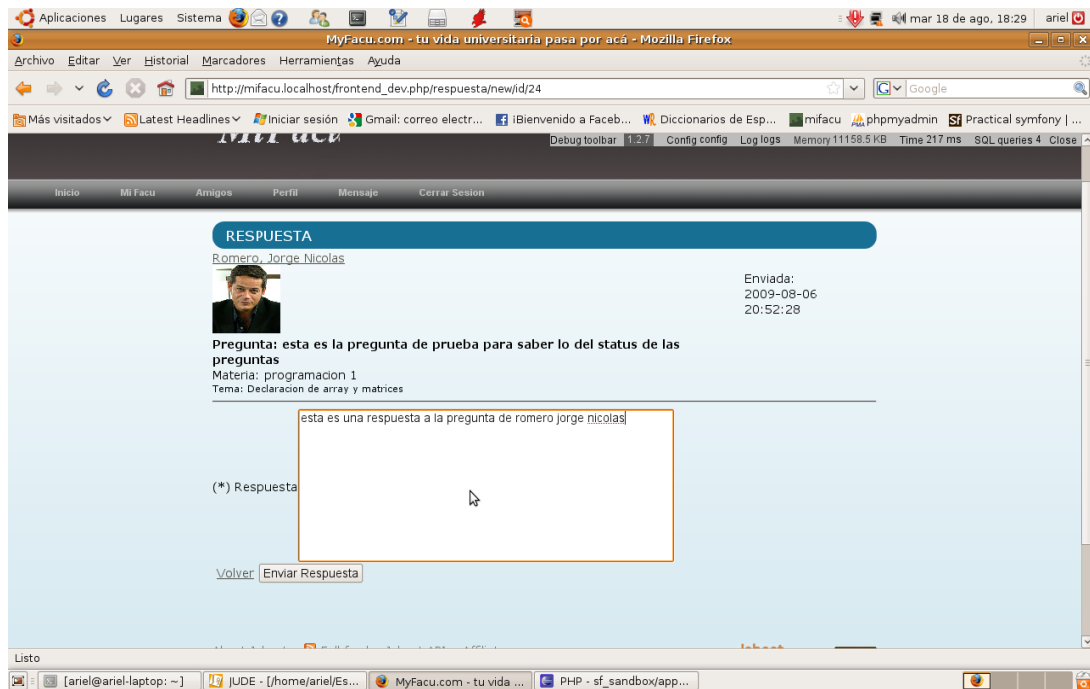
Responder Preguntas	
Código	F3 / 12
Actor	Estudiante
Código WBS	299
Flujo Normal / User Stories	
1. Incluye al caso de Uso: listar Preguntas a Responder (F3 / 11) 2. El Estudiante hace clic en el vinculo <u>Responder</u> 3. En la pantalla puede ver la pregunta que el amigo realizo e ingresar la respuesta que considera. 4. Finalmente hace clic en el botón [Enviar Respuesta]	
Condiciones	
Pre Condiciones	1. El estudiante debe existir y haber iniciado sesión. 2. La pregunta debe existir y tener estado activo
Pos Condiciones	1. Se crea una respuesta 2. Se cambia el estado de la DestinatarioPregunta a CONTESTADA
Camino Alternativo / Extiende	
El Estudiante cancela la respuesta	

23.13.2 Diagrama de Clase





23.13.3 Pantallas



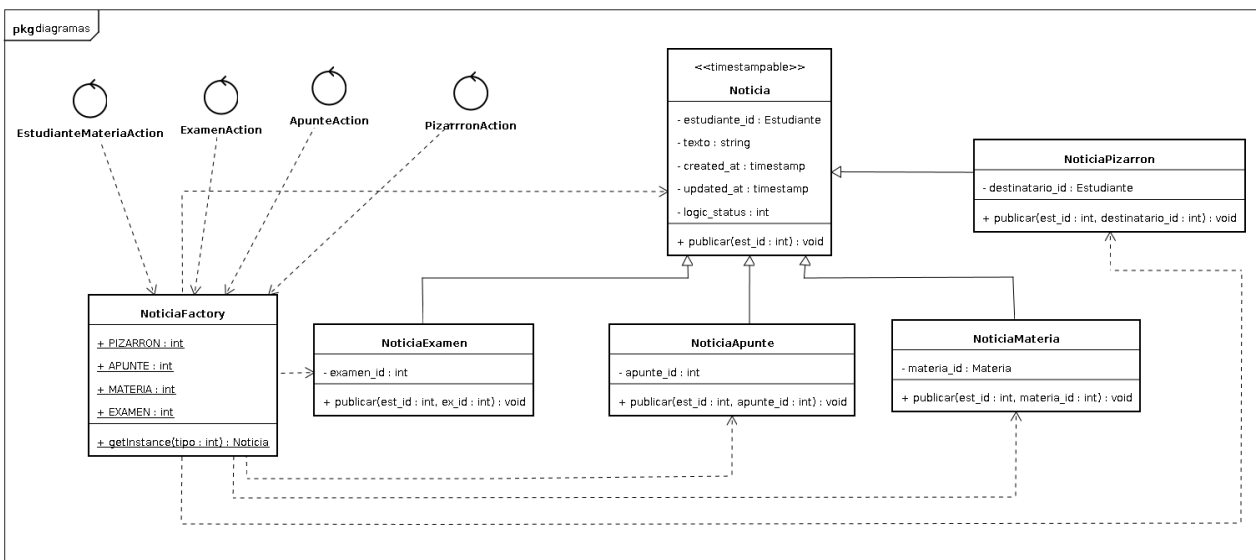


23.14 Publicar Noticia

23.14.1 Ficha de Caso de Uso

Publicar Noticia	
Código	F3 / 13
Actor	Estudiante
Código WBS	300
Flujo Normal / User Stories	
<ol style="list-style-type: none"> 1. Cuando el Estudiante Realice una acción que se considere que deba publicarse se creara una noticia que aparecerá en la pagina de inicio de sus amigos y de el mismo. 2. Se crearan noticias cuando un estudiante cambie el estado de una materia, cuando registre un examen, cuando suba un apunte o cuando escriba en el pizarrón de otro estudiante. 3. Las noticias mostradas serán solo de los amigos. 	
Condiciones	
Pre Condiciones	<ol style="list-style-type: none"> 1. El Estudiante debe existir 2. Tiene que haber existido algún hecho que dispare la creación de la noticia.
Pos Condiciones	<ol style="list-style-type: none"> 1. Se crea una Noticia 2. La noticia se publica.
Camino Alternativo / Extiende	
Ninguno.	

23.14.2 Diagrama de Clase



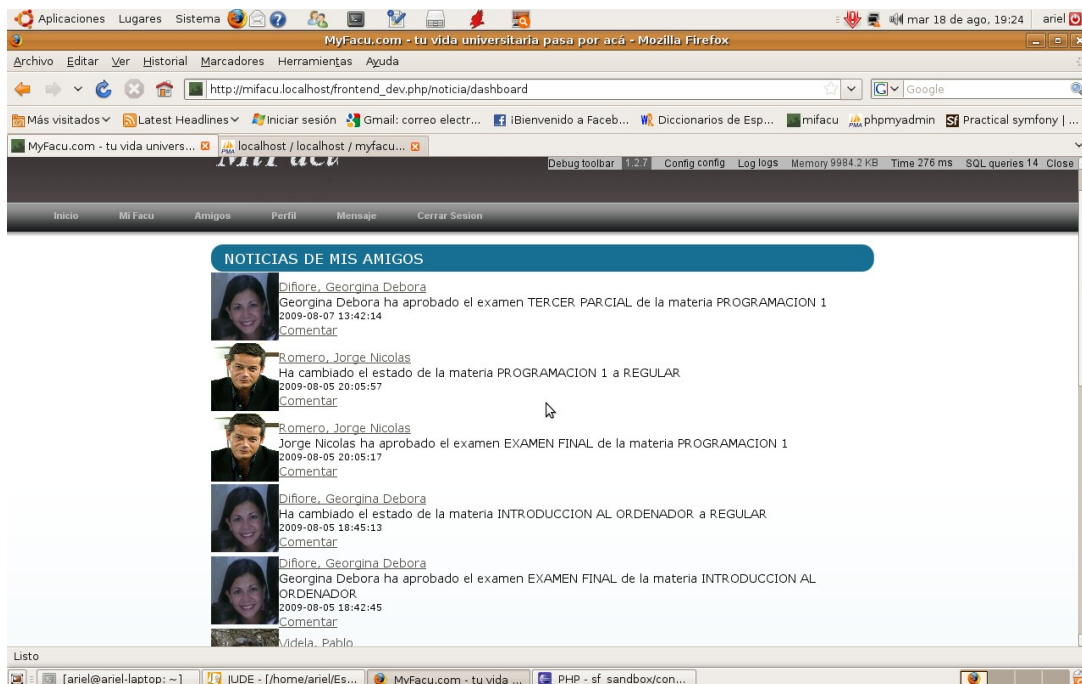


23.15 Ver Mis Noticias

23.15.1 Ficha de Caso de Uso

Ver Mis Noticias	
Código	F3 / 14
Actor	Estudiante
Código WBS	301
Flujo Normal / User Stories	
<ol style="list-style-type: none"> 1. Cuando el estudiante Ingresa al sistema, inmediatamente el sistema se redirecciona a la pagina de noticias. 2. El “dashboard” muestra las noticias publicadas por los amigos del estudiante ordenadas por la fecha y hora de su publicación. 3. Además de lo anterior cada noticia deberá contener un link <u>comentar</u>. 	
Condiciones	
Pre Condiciones	<ol style="list-style-type: none"> 1. El estudiante debe existir 2. El estudiante debe tener amigos
Pos Condiciones	Ninguna
Camino Alternativo / Extiende	
Ninguna	

23.15.2 Pantallas





ELABORACIÓN / CUARTA FASE

- Listado de Casos de Uso
- Casos de Uso del Sistema



24 Cuarta Fase

En esta última fase, se trabajará sobre los casos de uso restantes que se estimaron para cumplir con los requisitos de la materia “Trabajo Final de Graduación”. Además, y al finalizar esta etapa en un anexo, se incluirá la documentación referente a algunas otras funcionalidades adicionales que se le agregaron al trabajo y que surgieron del trabajo sobre la backlog, en la interacción con los miembros de la CAE.

24.1 Listado de Casos de Uso

La cuarta fase implementará casos de uso avanzados o que requieran de una mayor experiencia en el desarrollo con las herramientas y la arquitectura seleccionada.

Se llega a esta instancia con un conocimiento relativamente alto del negocio al que la aplicación intenta dar soporte, además de una optimización de los recursos de desarrollo.

En contraste con la fase 3, la fase 4, contiene un menor número de casos de uso, ya que se tiene en cuenta los cambios que la aplicación va a sufrir cuando sea sometida a los ambientes de *Testing*, y *Stage* integrados por el cuerpo de evaluadores de la CAE.

A su vez, gran parte del *núcleo del negocio (core business)* descansará sobre los casos de uso a desarrollar en esta última fase.

Casos de Uso Funcionales		
<i>Hacen referencia a las funcionalidades que el sistema debe realizar</i>		
<i>Notación código: Fase/N°UC</i>		
Reque.	Código	Nombre UC
Cuarta Fase		
RF33	F4/01	Subir Archivo
RF34	F4/02	Listar mis Archivos
RF35	F4/03	Listar Archivos de mis Amigos
RF36	F4/04	Bajar Archivos
RF37	F4/05	Ver Comentarios de una Noticia
RF38	F4/06	Escribir Comentario
RF39	F4/07	Cambiar Contraseña (Candidato)
RF40	F4/08	Agregar Taxonomía (Candidato)
RF41	F4/09	Ver mis Taxonomías (Candidato)
RF42	F4/10	Ver Taxonomías de mis Amigos
RF43	F4/11	Listar Personas Que me Pueden Ayudar



24.2 Subir Archivo

24.2.1 Ficha de Caso de Uso

Subir Archivo	
Código	F4/ 01
Actor	Estudiante
Código WBS	359
Flujo Normal / User Stories	
<ol style="list-style-type: none"> 1. El estudiante ingresa al sistema, luego hace clic en mi Facu. 2. A continuación hace clic en compartir apunte de la materia con la cual tiene relevancia el apunte. 3. A continuación el estudiante se encontrara con un listado de todos los apuntes sobre esa materia subidos por sus compañeros. Y además un formulario para que el pueda subir sus propios apuntes. 4. En el formulario el estudiante ingresara el nombre con el cual quiere que los estudiantes conozcan su apunte, una breve descripción de lo que el apunte contiene, seleccionara el apunte de su computadora y finalmente decidirá si quiere que lo vean todos o solamente sus amigos. 5. Una vez que el estudiante ha subido el apunte, el mismo debe aparecer primero en la lista de apuntes subidos, y crearse una noticia para que los amigos puedan ver la noticia. Incluye el caso de Uso: Publicar Noticia. 	
Condiciones	
Pre Condiciones	<ol style="list-style-type: none"> 1. El Estudiante debe existir 2. La Materia debe existir 3. El EstudianteMateria debe existir.
Pos Condiciones	<ol style="list-style-type: none"> 1. Se crea un apunte 2. Se relaciona un Estudiante y una materia al apunte. 3. Se crea una NoticiaApunte
Camino Alternativo / Extiende	
Ninguno.	



MyFacu.com - tu vida universitaria pasa por acá - Mozilla Firefox

http://mifacu.localhost/frontend_dev.php/apunte?materiald=1

Inicio MI Facu Amigos Perfil Mensaje Cerrar Sesión

¿QUERES COMPARTIR ALGÚN APUNTE?

(*) Campos Obligatorios

Nombre *


Breve Descripción

Archivo * Examinar...


Disponible para

Compartir Apunte

APUNTES DE LA MATERIA

Estudiante	Apunte	Votos	Acción
 Difiore, Georgina	Empl de casos de uso Este es un ejemplo de casos de uso programacion 1 2009-09-12 00:33:40 2009-09-12 00:34:07	4	download Borrar Apunte


MI CUENTA



Difiore, Georgina
Mensajes Nuevos: 3

PUBLICIDAD

Hacé las compras



Listo

ariel@ariel-laptop: ~ | JUDE - [/home/ariel/Es... | [TFGv1.0.1 - Navegad... | [TFGv1.0.1 - Navegad... | MyFacu.com - tu vida ...

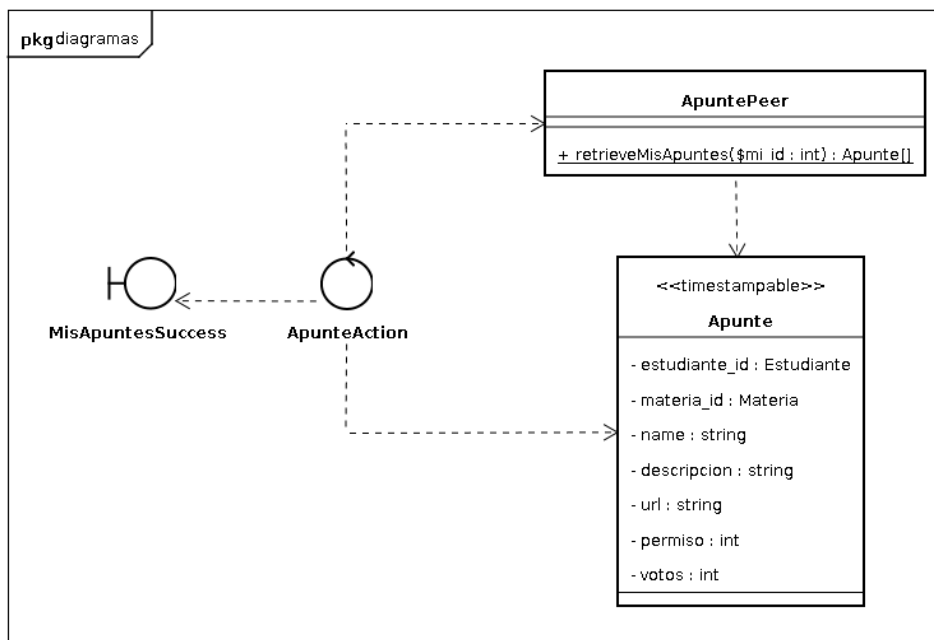


24.3 Listar Archivos

24.3.1 Ficha de Casos de Uso

Listar Archivos	
Código	F4/ 02
Actor	Estudiante
Código WBS	360
Flujo Normal / User Stories	
1. El estudiante ingresa al sistema, luego hace clic en mi Facu. 2. A continuación hace clic en compartir apunte de la materia con la cual tiene relevancia los apuntes que quiere ver. 3. El estudiante debería ver cuales son los archivos que ha subido.	
Condiciones	
Pre Condiciones	1. El Estudiante debe existir 2. Los apuntes deben existir.
Pos Condiciones	
Camino Alternativo / Extiende	
Ninguno.	

24.3.2 Diagrama de Clases



24.3.3 Pantallas



MyFacu.com - tu vida universitaria pasa por acá - Mozilla Firefox

http://mifacu.localhost/frontend_dev.php/apunte?materiald=1

¿QUERES COMPARTIR ALGÚN APUNTE?

(*) Campos Obligatorios

Nombre *

Breve Descripción

Archivo * Examinar...

Disponible para

Compartir Apunte

MI CUENTA

Difiore, Georgina
Mensajes Nuevos: 3

PUBLICIDAD

Hacé las compras

APUNTES DE LA MATERIA

Estudiante	Apunte	Votos	Acción
Difiore, Georgina	Empleo de casos de uso Este es un ejemplo de casos de uso programacion 1 2009-09-12 00:33:40 2009-09-12 00:34:07	4	download Borrar Apunte
Diaz Molina, Ariel	Empleo de casos de uso Este es un ejemplo para ver si no se mezclan	1	download Denunciar

MyFacu.com - tu vida universitaria pasa por acá - Mozilla Firefox

http://mifacu.localhost/frontend_dev.php/apunte/showApunteMateria

Inicio Mi Facu Amigos Perfil Mensaje Cerrar Sesión

Noticias Apuntes Preguntas Preguntas de Examen Temas

Buscador Vivo

APUNTES MAS BAJADOS - PROGRAMACION 1

Estudiante	Apunte	Votos	Acción
Difiore, Georgina	plantilla de casos de uso Esta es la plantilla de casos de uso programacion 1 2009-09-11 22:54:42 2009-09-12 00:24:22	6	download Borrar Apunte
Difiore, Georgina	Empleo de casos de uso Este es un ejemplo de casos de uso programacion 1 2009-09-12 00:33:40 2009-09-12 00:34:07	4	download Borrar Apunte
Diaz Molina, Ariel	plantilla de casos de uso es una plantilla de casos de uso programacion 1 2009-09-12 00:28:43 2009-09-12 00:28:48	2	download Denunciar

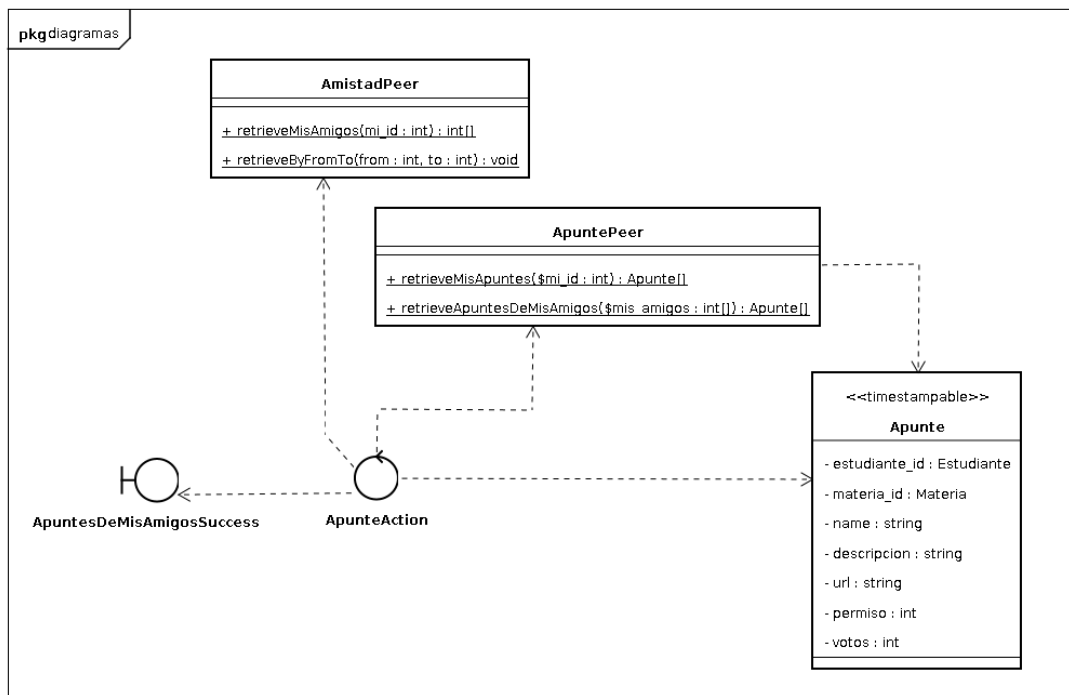


24.4 Listar Archivos de Mis Amigos

24.4.1 Ficha de Casos de Uso

Listar Archivos de Mis Amigos	
Código	F4/ 03
Actor	Estudiante
Código WBS	361
Flujo Normal / User Stories	
1. El estudiante ingresa al sistema, luego hace clic en mi Facu. 2. A continuación hace clic en compartir apunte de la materia con la cual tiene relevancia los apuntes que quiere ver. 3. El estudiante debería ver cuales son los archivos que han subido sus amigos.	
Condiciones	
Pre Condiciones	1. El Estudiante debe existir 2. Los apuntes deben existir.
Pos Condiciones	
Camino Alternativo / Extiende	
Extiende el caso de uso: Listar Archivos	

24.4.2 Diagrama de Clases





24.4.3 Pantallas

The screenshot shows a web browser window with the URL `http://mfacu.localhost/frontend_dev.php/apunte/showApunteMateria`. The page title is "MyFacu.com - tu vida universitaria pasa por acá - Mozilla Firefox". The browser's address bar shows the URL. The page content includes a navigation menu with "Noticias", "Apuntes", "Preguntas", "Preguntas de Examen", and "Temas". Below the menu is a search bar with the text "Buscador Vivo" and a prompt "Escribe una palabra relacionada". The main content area is titled "APUNTES MAS BAJADOS - PROGRAMACION 1" and displays a table of notes.

Estudiante	Apunte	Votos	Accion
Difiore, Georgina	plantilla de casos de uso Esta es la plantilla de casos de uso programacion 1 2009-09-11 22:54:42 2009-09-12 00:24:22	6	download Borrar Apunte
Diaz Molina, Ariel	plantilla de casos de uso es una plantilla de casos de uso programacion 1 2009-09-12 00:28:43 2009-09-12 00:28:48	2	download Denunciar
Diaz Molina, Ariel	Emplo de casos de uso Este es un ejemplo para ver si no se mezclan programacion 1 2009-09-12 00:32:39 2009-09-12 00:32:39	1	download Denunciar

The browser's taskbar at the bottom shows several open windows, including "ariel@ariel-laptop: ~", "JUDE - [home/ariel/Es...", "TFGv1.0.1 - Navegad...", and "MyFacu.com - tu vida ...".

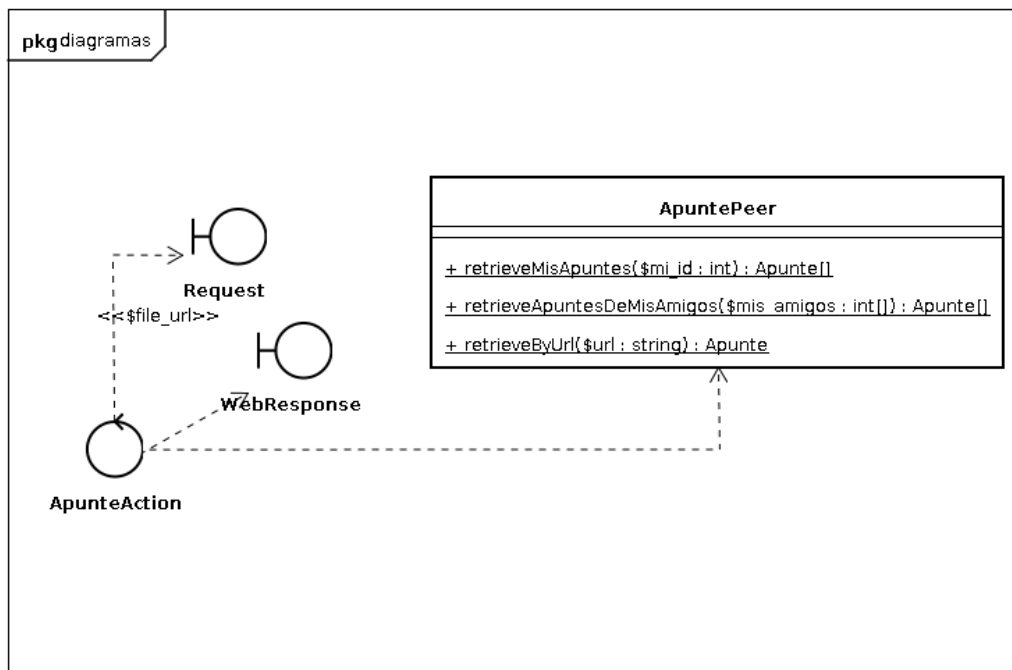


24.5 Bajar Archivos

24.5.1 Ficha de Casos de Uso

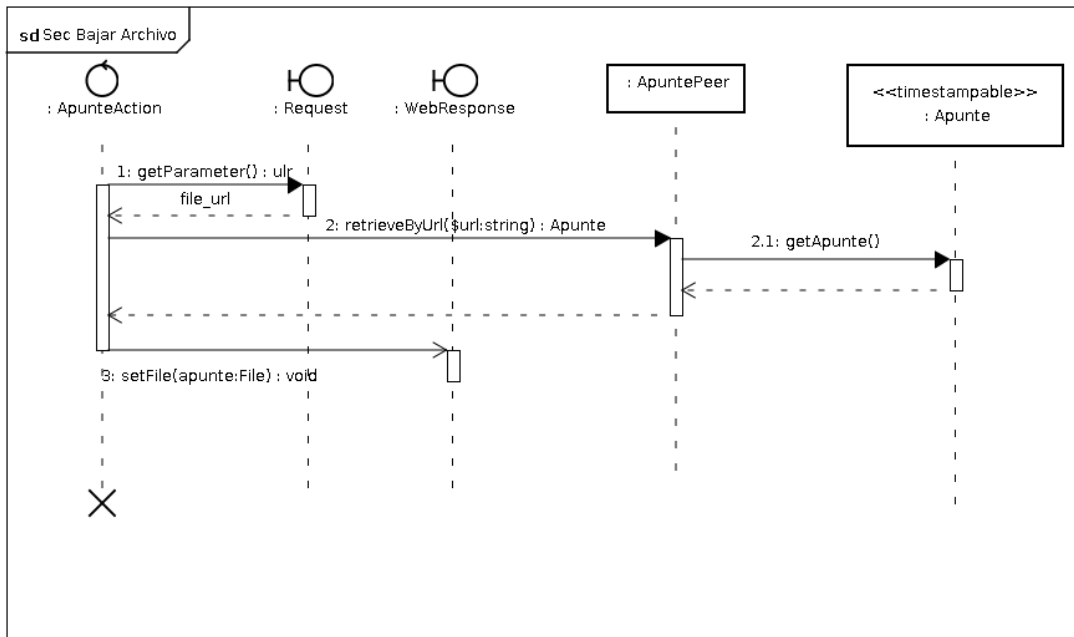
Bajar Archivos	
Código	F4/ 04
Actor	Estudiante
Código WBS	362
Flujo Normal / User Stories	
1. El estudiante podrá bajar un apunte en cualquier lado donde el mismo aparezca. Es condición de que sea de un amigo, o el mismo sea público.	
Condiciones	
Pre Condiciones	<ol style="list-style-type: none"> 1. El Estudiante debe existir 2. El apunte deben existir. 3. El estudiante debe ser amigo de la estudiante que subió el apunte o el mismo debe haber sido posteado con estatus público.
Pos Condiciones	
Camino Alternativo / Extiende	
Ninguno.	

24.5.2 Diagrama de Clases

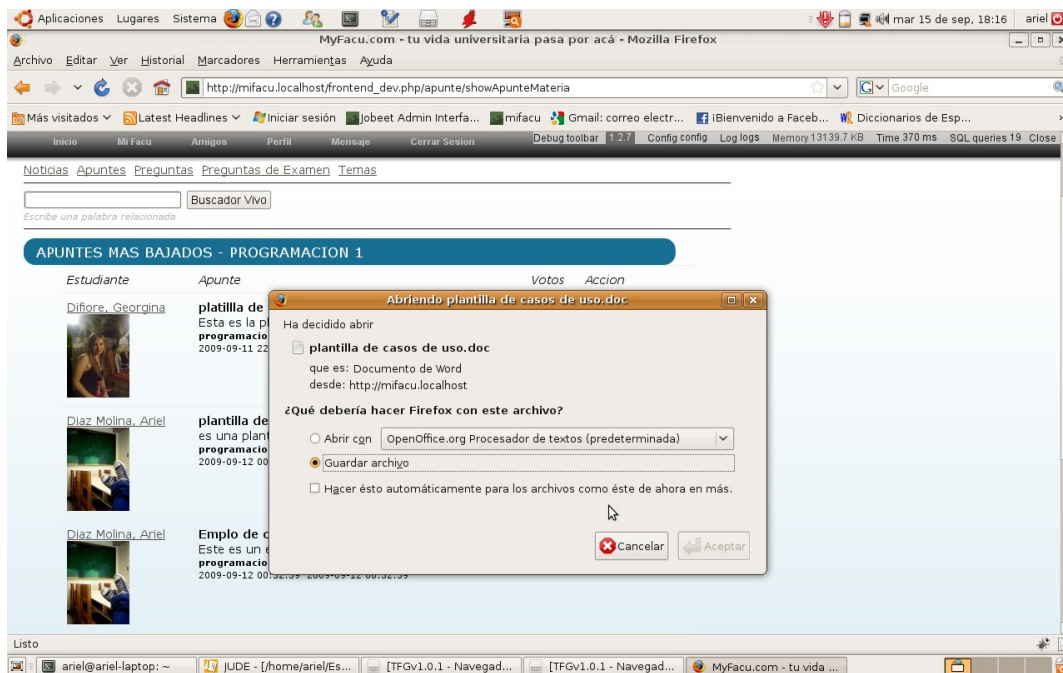




24.5.3 Diagrama de Secuencia



24.5.4 Pantallas





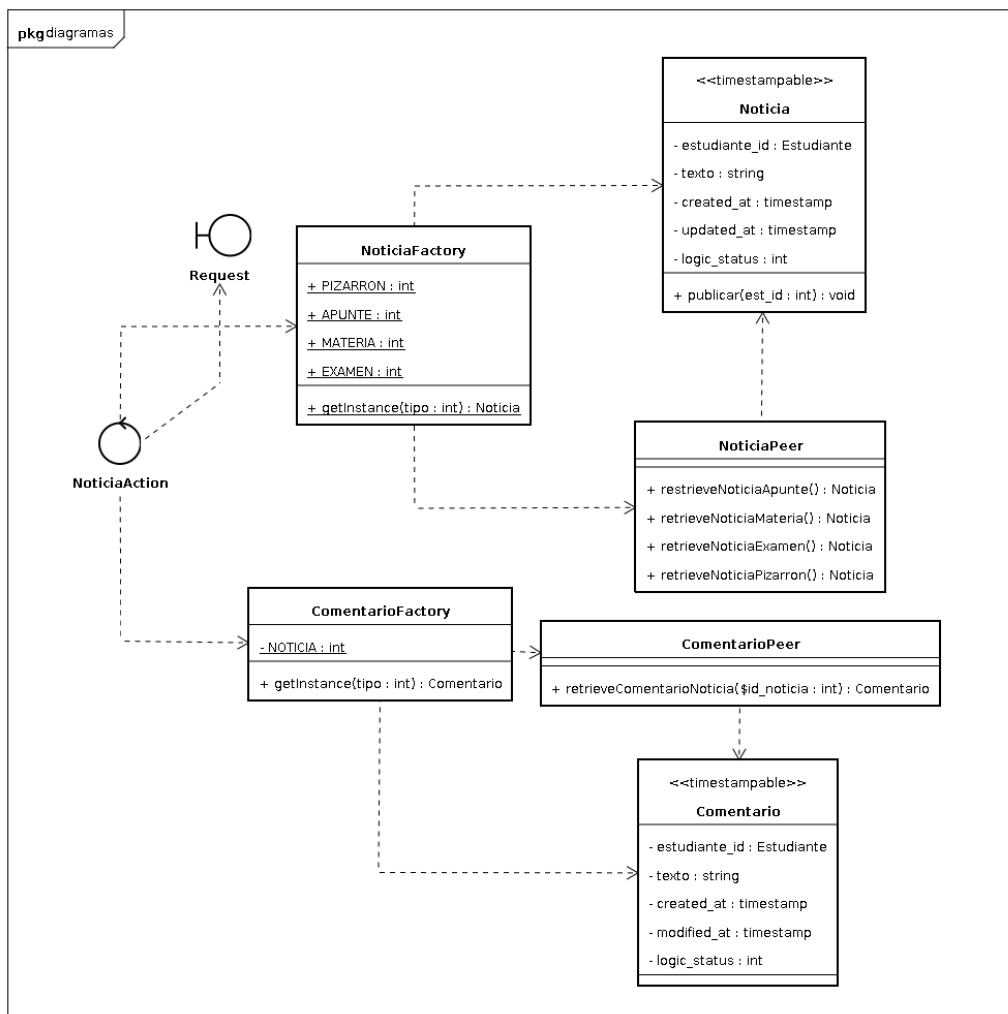
24.6 Ver Comentarios de Una Noticia

24.6.1 Ficha de Caso de Uso

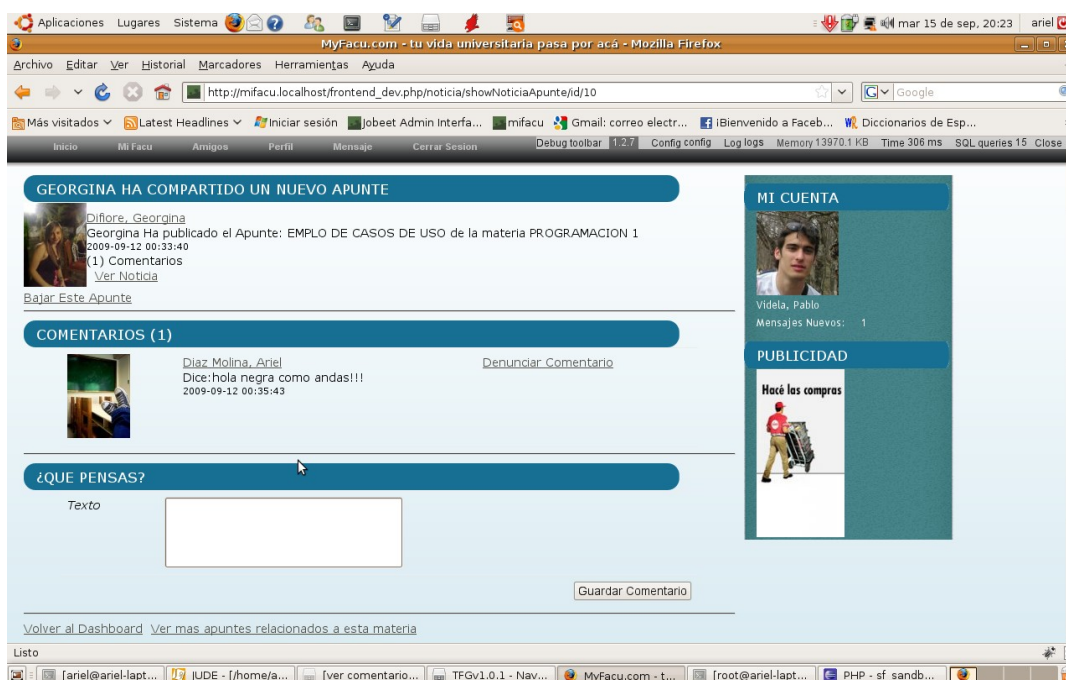
Ver comentario de una Noticia	
Código	F4/ 05
Actor	Estudiante
Código WBS	363
Flujo Normal / User Stories	
<ol style="list-style-type: none"> 1. El estudiante podrá ver los comentarios que de una noticia, haciendo clic en el vínculo que dice ver noticia. 2. Ya sea en el “dashboard” de la aplicación o en la sección destinada a cada una de las materias. 	
Condiciones	
Pre Condiciones	<ol style="list-style-type: none"> 1. El Estudiante debe existir 2. La Noticia debe existir 3. Los comentarios deben existir 4. Las noticias deben estar relacionadas a los amigos del estudiante.
Pos Condiciones	
Camino Alternativo / Extiende	
Ninguno.	



24.6.2 Diagrama de Clases



24.6.3 Pantallas



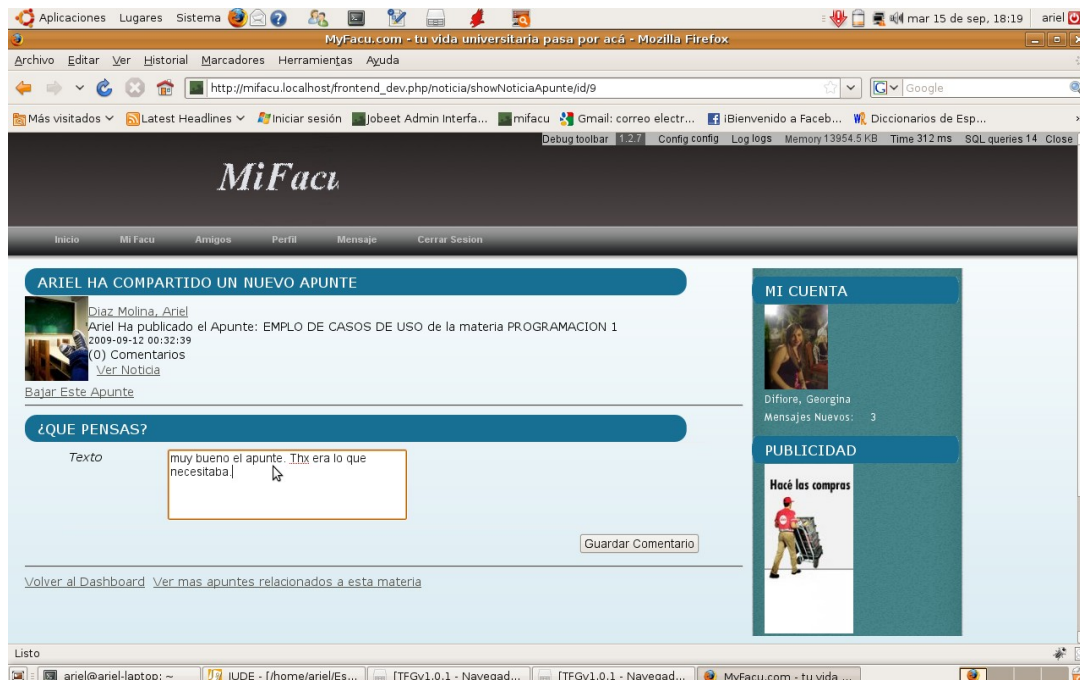


24.7 Escribir Comentario

24.7.1 Ficha de Caso de Uso

Escribir Comentario	
Código	F4/ 06
Actor	Estudiante
Código WBS	364
Flujo Normal / User Stories	
<ol style="list-style-type: none"> 1. El estudiante podrá escribir un comentario sobre una noticia, en cualquier lado donde la misma sea publicada. 2. Deben existir comentarios para noticias de apuntes, noticias sobre una materia, y noticias sobre un examen. 3. Incluye los casos de Uso: Escribir comentario sobre una noticia de un apunte, Escribir comentario sobre una noticia de un examen, Escribir comentario sobre una noticia de una materia. 	
Condiciones	
Pre Condiciones	<ol style="list-style-type: none"> 1. El Estudiante debe existir 2. La Noticia debe existir.
Pos Condiciones	<ol style="list-style-type: none"> 1. Se crea una un ComentarioNoticia.
Camino Alternativo / Extiende	
Ninguno.	

24.7.2 Pantallas



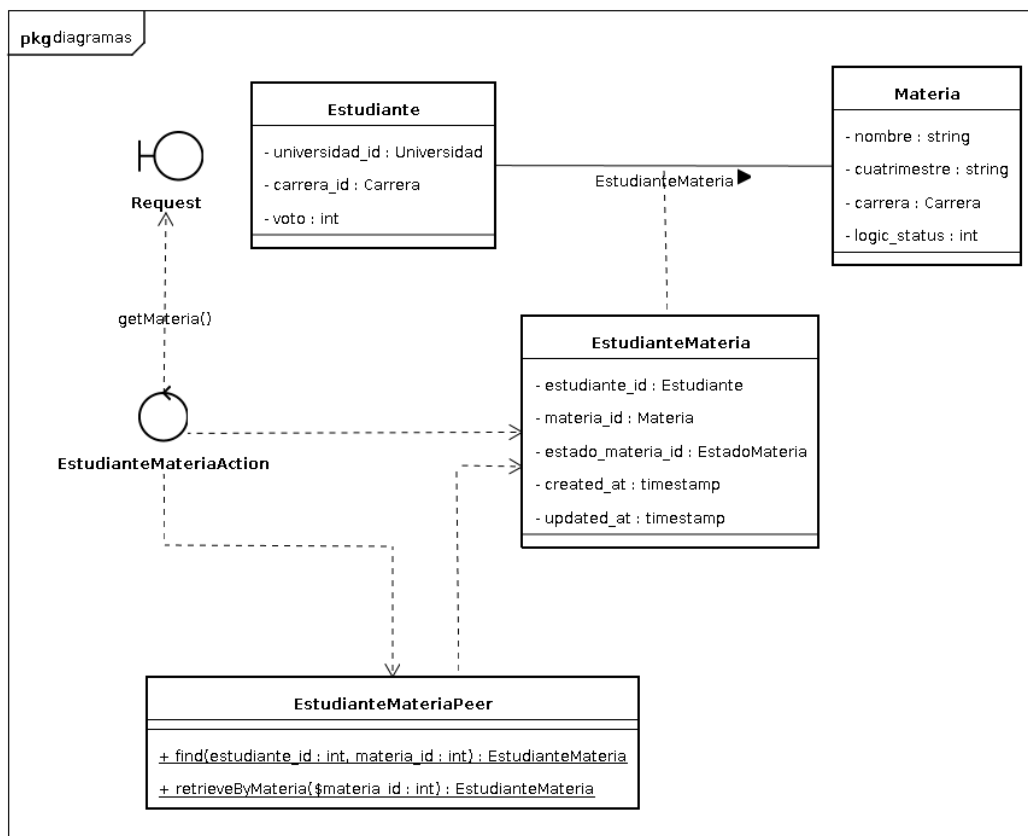


24.8 Listar Personas que me Pueden Ayudar

24.8.1 Ficha de Caso de Uso

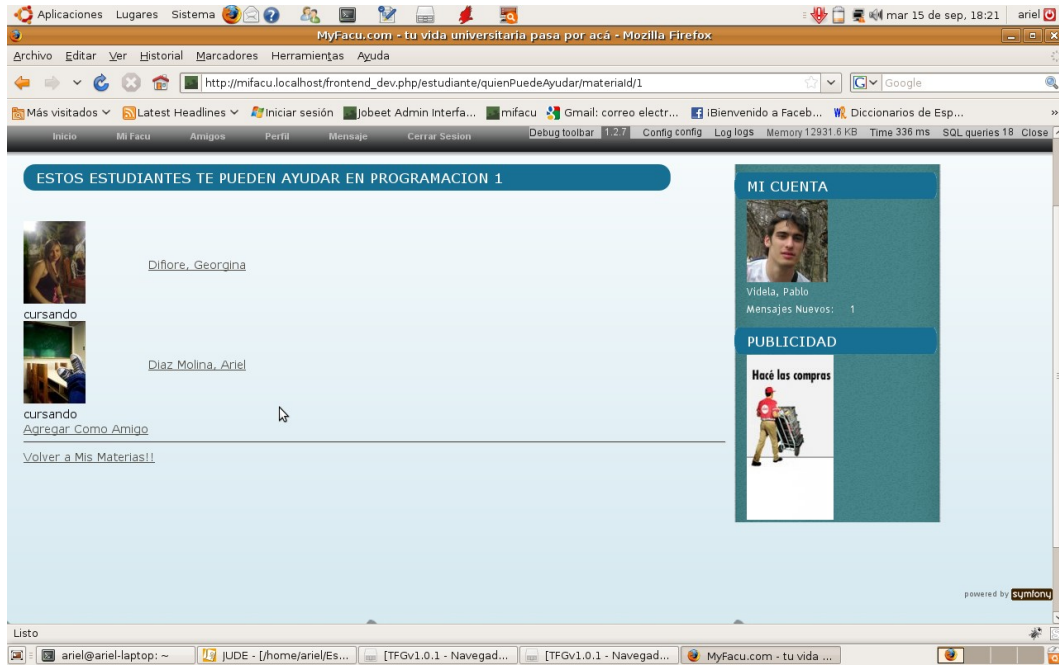
Listar Personas que me Pueden Ayudar	
Código	F4/ 11
Actor	Estudiante
Código WBS	358
Flujo Normal / User Stories	
4. El estudiante podrá escribir un comentario sobre una noticia, en cualquier lado donde la misma sea publicada. 5. Deben existir comentarios para noticias de apuntes, noticias sobre una materia, y noticias sobre un examen. 6. Incluye los casos de Uso: Escribir comentario sobre una noticia de un apunte, Escribir comentario sobre una noticia de un examen, Escribir comentario sobre una noticia de una materia.	
Condiciones	
Pre Condiciones	3. El Estudiante debe existir 4. La Noticia debe existir.
Pos Condiciones	2. Se crea una un ComentarioNoticia.
Camino Alternativo / Extiende	
Ninguno.	

24.8.2 Diagrama de Clases





24.8.3 Pantallas





CONCLUSIONES

- Conclusiones Sobre el Proyecto
- Conclusiones Personales



25 Conclusiones Sobre el Proyecto

Durante el desarrollo del presente proyecto he llegado a distintas conclusiones tanto en relación a las herramientas que contribuyen directamente a la construcción de un software, como así también en relación a aquellas que tienen que ver con las actividades, ceremonias, artefactos y roles que dan soporte a las metodologías de desarrollo, es decir toda la parte “blanda” del proceso de desarrollo de software.

25.1 Herramientas que contribuyen directamente al desarrollo de Software.

25.1.1 Arquitectura de una Aplicación

Es indudable que a la hora de empezar a codificar debemos contar con una arquitectura que nos sirva de marco de contención de lo que vamos a desarrollar en un futuro. Por supuesto que no conocemos que vamos a hacer, que el negocio esta difuso, que los requerimientos parecerían simples y que en un primer momento siempre tenemos esa falsa sensación de que ya hemos hecho algo parecido y que solo tenemos que “cortar” y “pegar” desde otros proyectos pasados.

Muchas veces he sentido que la arquitectura no era necesaria, que podía prescindir de ella, que no debía sobrecargar mis modelos agregando capas y capas de código e interfases. Pero a la hora del trabajo diario me fui dando cuenta que la arquitectura me servía de marco de referencia para una fácil ubicación en el código y además de ello contribuía a que mi aplicación cumpla con el principio DRY (“*Don't Repeat Yourself*”³⁷) de toda buena aplicación. En mi forma de ver (Y en esto le doy la derecha a RUP, en su “*centrado en la arquitectura*”) es la evolución de un paradigma, como es el de la programación orientada a objetos. La arquitectura nos permitirá escalar los sistemas con el mínimo esfuerzo, saber a donde vamos a ubicar esa clase que andamos buscando, saber donde vamos a poner ese método que nos cuesta hasta determinar el nombre. Y por ultimo, Poder dar ese, “incremento comprobable en las funcionalidades” que pregonan SCRUM o FDD³⁸. En el rol de asesor de negocio que el informático posee (A mi entender y lo explicare en el próximo apartado) en la actualidad, la escalabilidad lo es ¡todo! Y la arquitectura gran parte de ese todo.

37 “*No te repitas a ti mismo*” (Literal) principio que sostiene que debemos buscar no realizar el mismo trabajo en distintos lados.

38 *Feature Driven Development*: Desarrollo guiado por funcionalidades o características.



25.1.2 Buenas Prácticas en la Programación y Patrones utilizados.

En este proyecto, las buenas prácticas en cuestiones de programación se tornaron necesarias. La verdad, en cuanto a la asignación de nombres he optado por las buenas prácticas de “Sun Microsystems” sobre la codificación en java. A pesar de que en PHP existen, y son muy variadas, solo he utilizado estas, en el nombre de las variables, finalizando ya casi el proyecto. Esto se debió a la mejor legibilidad, de la notación de *guiones bajos*, frente a la *camel case* de java. No he utilizado prefijos³⁹ en la denominación de las variables, por no considerarlos necesarios, ya que la suma de las clases y la orientación a objeto me ha dado suficiente legibilidad. Se suele utilizar prefijos con las primeras letras del tipo de dato ya que en php por defecto no es necesario definir el tipo de dato en su declaración, pero esto era común antes de la versión orientada a objetos de php. En cuanto a los patrones de diseño existentes, yo solo he utilizado algunas “*fabricas*”⁴⁰, pero solo a los fines de mantener una interfaz para posibles escalabilidades. No me han sido de demasiada ayuda en el presente proyecto. Además, el framework de desarrollo que he utilizado, Symfony, implementaba patrones de diseño tales como *routing* (o enrutador) y *wrapper* (o decorador).

25.1.3 Framework de Desarrollo

En este proyecto he utilizado Symfony. Un framework de desarrollo en php basados en buenas prácticas y en la conjunción de otros frameworks que se encargan de las distintas capas de una aplicación. Es así que Symfony incorpora herramientas para el trabajo con la capa de persistencia (ORM⁴¹), manejo del las url, creación de formularios, creación de unidades de prueba, etc.

El trabajo con frameworks de desarrollo, ya no es una opción, es una necesidad, ya que los mismos optimizan los tiempos de desarrollo, nos dan la seguridad de trabajar sobre código probado, y nos permiten organizar mejor una aplicación.

Pero no todo es *color de rosas*, aprender a utilizar estos frameworks requiere horas de capacitación, la correcta elección de la herramienta en los inicios del proyecto, y la permanente actualización y “*parchado*” de la misma. A mi forma de ver, estas herramientas te solucionan el 80% de los problemas que una aplicación plantea, pero el 20% restante, suele ser muy difícil de solucionar, por no decir imposible. Además de lo anterior, si la herramienta es demasiado joven y no cuenta con

39 Ejemplo: `obj_usuario` para una variable del tipo `Usuario`, `int_usuario`; para definir que se esta hablando de un entero que podría ser el id del usuario.

40 Patrón *Concrete Factory*: Es un patrón de diseño de creación de objetos que nos brinda un método estándar para crear objetos de una familia de objetos.

41 Object Relational Mapping: sirve para crear un vínculo entre la base de datos y los objetos del modelo, abstrayéndonos de la forma en la que persiste los datos en la base de datos.



una gran comunidad de desarrolladores, la solución errores de programación y la maduración del producto suele ser muy lenta y muy difícil de rastrear por nosotros mismos.

De pronto me encontré parchando algunos errores de programación de las librerías base del framework, o abriendo tickets para la solución de bugs y sentándome a esperar que los mismos sean resueltos. Por suerte, symfony cuenta con una gran comunidad de usuarios y gran cantidad de documentación, y digo por suerte, por que desconocía esto a los inicios del proyecto.

26 Herramientas que contribuyen Indirectamente al desarrollo de software

26.1.1 Metodología de Desarrollo

En el proyecto desarrollado, he combinado dos metodologías. Ellas son.; [RUP](#) y [SCRUM](#).

Debo de destacar que cuando hago referencia a las metodologías no hago referencia a un paquete cerrado de ceremonias, roles, y artefactos. Si no más bien que he elegido principalmente los artefactos, después los roles y finalmente las ceremonias, por ser un trabajo unipersonal.

En relación a la planificación, planteamiento y toda etapa temprana del proyecto, los artefactos planteados por una de las “*capas*” de RUP que hace referencia al *Project Management*, a mí entender son los adecuados. Ellos son la Declaración de Alcance, el Acta de Inicialización del Proyecto, La Declaración de Trabajo y el Acta de Cambios y Modificaciones. En contraposición SCRUM no posee demasiadas consideraciones en estas etapas. Además de lo anterior, también ha influido el tipo de proyecto, donde las personas que evalúan los avances del proyecto, no pueden tener una dedicación absoluta al mismo.

En relación al desarrollo de la aplicación, su día a día, y la codificación, SCRUM es el adecuado. Ya que el mismo, plantea ciclos de vida mas cortos, los “*Sprints*”, que van desde quince a días hasta un mes. Es decir RUP, con su ciclo de vida “*iterativo e incremental*” plantea un desarrollo similar, pero su rigidez, falta de artefactos útiles en el desarrollo, la falsa sensación de sobre-documentación hace que trabajar con scrum sea más agradable y cómodo.

Algunos artefactos interesantes son la *Product Backlog list* y la *Sprint Backlog List*. La primera de ellas es muy útil a la hora de definir cuales serán las funcionalidades a implementar, esto es importante, ya que nos mantiene concentrados en el valor que ese “algo” le aporta al negocio y no en cuestiones de forma o meramente “informáticas”. Por ejemplo agregar un link, puede llegar a ser más importante que agregar una nueva funcionalidad, por más aburrido que esto nos resulte.



La segunda de ellas, es una lista de las funcionalidades a desarrollar en el sprint, que me ha servido para organizar mi trabajo y saber en cada momento cual es el siguiente paso a tomar.

26.1.2 Lenguaje de Modelado

En esto no hay duda, ni discusión, UML es el indicado. Precisamente los diagramas de clases y algunos otros diagramas para casos especiales. Pero la construcción de un adecuado diagrama de clases con suficiente información, transmite el 90% de lo que he querido transmitir.

He optado por no recargar de diagramas cada uno de los casos de uso. Hay que entender que no por que un diagrama aplique para una circunstancia determinada hay que hacerlo si o si. Prefiero el punto de vista de construirlo cuando la idea no ha sido transmitida completamente.

26.1.3 Herramientas de Soporte

Con esto hago referencia a herramientas tales como *issues trackers*⁴² y *repositorios*⁴³.

He utilizado repositorios a los fines de mantener copias de respaldo del código. Ya que al ser un trabajo unipersonal, no me ha sido necesario el control del versionado ni el almacenamiento, salvo en los casos en los que le he solicitado revisiones de código⁴⁴ a algún amigo.

En el caso de los “*issues trackers*” he utilizado una herramienta pública puesta a disposición pro google que es el “[google code](#)⁴⁵”. Esta herramienta permite crear un proyecto con wikis, repositorios, e *issues trackers*. Esta ultima, me ha permitido materializar una *product backlog*. Permiéndome cambiar los estados, agregar los problemas que iba encontrando en el camino y priorizando las actividades.

Es mi conclusión, que este tipo de herramientas son necesarias si o sí en un proyecto en el cual se trabaje en equipos, ya que permiten la coordinación de los esfuerzos, refuerzan la consistencia del código escrito y mantienen diferenciadas las distintas versiones de una aplicación. En otras palabras, nos permiten comunicarnos, visualizar, y tener puntos en común entre los distintos miembros de un equipo.

42 Herramienta que tiene por finalidad registrar y asignar a los miembros de un equipo; los problemas, errores de programación y tareas a realizar a lo largo del desarrollo.

43 Herramienta de almacenamiento y versionado del código fuente realizado por un equipo de desarrollo.

44 *Code Review*: Procedimiento en el que solicita la revisión del código escrito a un tercero. Tiene por finalidad la exhaustiva inspección del mismo y su optimización.

45 Google Code: <http://code.google.com/intl/es-AR/> - Septiembre del 2009



27 Conclusiones Personales

Las conclusiones personales a las que he arribado finalizando este proyecto, pueden dividirse en conclusiones sobre la industria de este tipo de aplicaciones, y conclusiones personales sobre el rol del informático de cara al futuro.

27.1 Conclusiones sobre la Industria

La industria de las aplicaciones WEB 2.0 parecería estar en auge, todo parecería ser redes sociales en Internet en la actualidad. Si queremos aprender un idioma o simplemente mantenernos actualizados de las cosas que hacen nuestros amigos, tenemos que recurrir a aplicaciones de este tipo. Pero ya se está hablando de la WEB 3.0, este concepto no difiere tanto de los conceptos de comunicación, cooperación y comunidad, lo que cambia son las tecnologías que les darán soporte. En lugar de tecnologías asíncronas como ajax, se habla de *widgets*⁴⁶ y tecnologías adaptadas para dispositivos móviles. Además, el avance de los monitores y pantallas táctiles facilitan el trabajo con este tipo de aplicaciones. De todas formas, las 3 C llegaron para quedarse en la industria de Internet. En lo personal el desarrollo de este proyecto me ha permitido cambiar la óptica de las aplicaciones a desarrollar en mi futuro. He dejado de creer en las aplicaciones que llamo EyC (Entregar y Correr), es decir aplicaciones que se programan de acuerdo, y sin desviarse ni un centímetro, de un listado de funcionalidades establecidas al inicio de un proyecto a cara de perro, se entregan y se procede a discutir todo cambio solicitado por el cliente, bajo la frase; “*es lo que vos me pediste y firmaste acá, todo lo demás te va a costar extra*”. La industria requiere de nuevos vendedores de software que no vendan aplicaciones a las cuales el cliente se tenga que adaptar, si no más bien que vendan la cooperación, comunidad y comunicación entre el informático y el negocio.

27.2 Conclusiones sobre el Rol de Informático

Creo que el informático debe de cambiar su rol; de ser un *gran maestro* tecnológico que asesora a una persona de negocios que no sabe de tecnología, por el rol de un *gran aprendiz* del negocio, que sabe de tecnología. Una interfaz entre el mundo real y el mundo tecnológico. Una persona que pueda asistir a los gerentes o directores hablándoles al oído, abriéndole los ojos sobre las ventajas competitivas que una aplicación le puede brindar a su negocio. Establecer lazos de unión inseparables. No solo profesionales que brindan la “*solución*” a algún problema, si no también

46 Pequeña aplicación o programa distribuidos en archivos que son ejecutados por una maquina virtual de widgets. O widget Engine



verdaderos creativos de nuevos negocios. Gerentes tecnológicos.

Es aquí donde este proyecto logró cambiar mi paradigma mental. Donde descubrí que una aplicación informática no tiene fin, ni alcance a determinar, más que para un periodo determinado, o para hablar en términos de desarrollo, una fase. Pero jamás hay que perder la visión de continuidad que todo proyecto informático trae intrínsecamente.

Las aplicaciones informáticas, deben ser ventajas competitivas que logren posicionar a los negocios por encima de sus competidores, no solo herramientas de usos diarios o meramente transaccionales.

Deben nacer con este espíritu y desarrollarse dentro de este marco.



ANEXO: GLOSARIO



28 Anexo Glosario

- **Arquitectura de Software:** Son formas y guías generales en el desarrollo de software a través de las cuales se pueden resolver problemas. Es un nivel de diseño que hace foco en aspectos "más allá de los algoritmos y estructuras de datos de la computación; el diseño y especificación de la estructura global del sistema es un nuevo tipo de problema". – 21/09/2009 - http://es.wikipedia.org/wiki/Arquitectura_de_software
- **Artefacto:** es un producto tangible resultante del proceso de desarrollo de software. Algunos artefactos como los casos de uso, diagrama de clases u otros modelos UML ayudan a la descripción de la función, la arquitectura o el diseño del software. Otros se enfocan en el proceso de desarrollo en sí mismo, como planes de proyecto, casos de negocios o enfoque de riesgos. El código fuente compilado para el testeo se suele considerar un artefacto, ya que el ejecutable es necesario para el plan de testeo. – 21/09/2009 - [http://es.wikipedia.org/wiki/Artefacto_\(diseño_de_software\)](http://es.wikipedia.org/wiki/Artefacto_(diseño_de_software))
- **Blog:** en español también una bitácora, es un sitio web periódicamente actualizado que recopila cronológicamente textos o artículos de uno o varios autores, apareciendo primero el más reciente, donde el autor conserva siempre la libertad de dejar publicado lo que crea pertinente. -21/09/2009 - <http://es.wikipedia.org/wiki/Blog>
- **Cæteris páribus:** frecuentemente escrita como caeteris, ceteris o ceteris paribus, es una locución latina que significa permaneciendo el resto constante. Ceteris significa lo demás o el resto, como en et cetera (y el resto) del que deriva la palabra etcétera. Par significa igual, como en la expresión castellana a la par. – 21/09/2009 - http://es.wikipedia.org/wiki/Ceteris_paribus
- **CASE: (Computer-Aided Software Engineering)** es la aplicación científica de un conjunto de herramientas y métodos a un sistema de software, dicha aplicación tiene por finalidad generar productos de software con de gran calidad, libres de defectos y fáciles de mantener. – 21/09/2009 - http://en.wikipedia.org/wiki/Computer-aided_software_engineering - Interpretación del Ingles: Ariel Diaz Molina.
- **Caso de Uso:** es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico – 21/09/2009 - http://es.wikipedia.org/wiki/Caso_de_uso
- **CSRF: (Cross-site request forgery)** también conocido como “ataque de un solo clic” o “montar una sesión” es un tipo de condigo malicioso que aprovecha las vulnerabilidades de un sitio Web a los fines de transmitir un comando no autorizado por intermedio de un usuario en el cual el sitio Web confía. – 21/09/2009 - http://en.wikipedia.org/wiki/Cross-site_request_forgery - Interpretación del Ingles: Ariel Diaz Molina.
- **Facebook:** es un sitio web gratuito de redes sociales creado por Mark Zuckerberg. Originalmente era un sitio para estudiantes de la Universidad de Harvard, pero actualmente está abierto a cualquier persona que tenga una cuenta de correo electrónico. Los usuarios pueden participar en una o más redes sociales, en relación con su situación académica, su lugar de trabajo o región geográfica. – 21/09/2009 – <http://es.wikipedia.org/wiki/Facebook>
- **Feed:** es un medio de redifusión de contenido web. Se utiliza para suministrar información actualizada frecuentemente a sus suscriptores. – 21/09/2009 – <http://es.wikipedia.org/wiki/Feed>
- **Folcsonomía:** es una indexación social, es decir, la clasificación colaborativa por medio de etiquetas simples en un espacio de nombres llano, sin jerarquías ni relaciones de parentesco



predeterminadas. Se trata de una práctica que se produce en entornos de software social cuyos mejores exponentes son los sitios compartidos como del.icio.us (enlaces favoritos), Flickr (fotos), Tagzania (lugares), floc (lugares) o 43 Things (deseos). – 21/09/2009 - <http://es.wikipedia.org/wiki/Folcsonomía>

- **Framework:** en el desarrollo de software, es una estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. – 21/09/2009 - <http://es.wikipedia.org/wiki/Framework>
- **LinkedIn:** es un sitio de red orientada a los negocios, fundado en diciembre de 2002 y lanzada en mayo de 2003 (comparable a un servicio de red social), principalmente para red profesional. – 21/09/2009 - <http://es.wikipedia.org/wiki/LinkedIn> - Sitio Oficial: <http://www.linkedin.com/>
- **Livemocha:** red social para aprendizaje de idiomas con mas de 2M de usuarios. – 21/09/2009 – Sitio Oficial: <http://www.livemocha.com/>
- **Look and Feel:** es un término utilizado en el ámbito de productos y campos tales como; el diseño de productos, mercadeo, gestión de marcas y gestión de registros de marcas, para describir las principales características de la apariencia de los mismos – 21/09/2009 - http://en.wikipedia.org/wiki/Look_and_feel - Interpretación del Ingles: Ariel Diaz Molina.
- **MySpace:** es un sitio web de interacción social formado por perfiles personales de usuarios que incluye redes de amigos, grupos, blogs, fotos, vídeos y música, además de una red interna de mensajería que permite comunicarse a unos usuarios con otros y un buscador interno – 21/09/2009 – <http://es.wikipedia.org/wiki/MySpace>
- **Open Source:** es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones morales y/o filosóficas las cuales destacan en el llamado software libre. – 21/09/2009 – http://es.wikipedia.org/wiki/Open_Source
- **Project Charter:** (También: Definición del Proyecto o Términos de una Referencia) es una breve declaración del alcance, objetivos y participantes de un proyecto.- 21/09/2009 - http://en.wikipedia.org/wiki/Project_charter - Interpretación del Ingles: Ariel Diaz Molina.
- **Proyecto:** es una planificación que consiste en un conjunto de actividades que se encuentran interrelacionadas y coordinadas; la razón de un proyecto es alcanzar objetivos específicos dentro de los límites que imponen un presupuesto y un lapso de tiempo previamente definidos. La gestión de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para satisfacer los requisitos del proyecto. – 21/09/2009 - <http://es.wikipedia.org/wiki/Proyecto>
- **Red Social:** es una estructura social que se puede representar en forma de uno o varios grafos en el cual los nodos representan individuos (a veces denominados actores) y las aristas relaciones entre ellos – 21/09/2009 – http://es.wikipedia.org/wiki/Redes_sociales
- **RUP:** es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. – 21/09/2009 - <http://es.wikipedia.org/wiki/RUP>
- **Scope Statement:** (Declaración del Proyecto) Detalla los entregables y principales objetivos de un proyecto.- 21/09/2009 - http://en.wikipedia.org/wiki/Scope_Statement - Interpretación del Ingles: Ariel Diaz Molina.



- **Scrum:** es un proceso de desarrollo de software iterativo y creciente utilizado comúnmente en entornos basados en el desarrollo ágil de software. – 21/09/2009 - <http://es.wikipedia.org/wiki/Scrum>
- **Serialización:** (o marshalling en inglés) consiste en un proceso de codificación de un Objeto (programación orientada a objetos) en un medio de almacenamiento (como puede ser un archivo, o un buffer de memoria) con el fin de transmitirlo a través de una conexión en red como una serie de bytes o en un formato humanamente más legible como XML o JSON, entre otros. – 21/09/2009 – <http://es.wikipedia.org/wiki/Serializacion>
- **Servicio de Red Social:** se concentra en construir comunidades en línea de personas que comparten intereses y/o actividades, o de aquellos que están interesados en explorar los intereses y actividades de otros. La mayoría de los servicios de redes sociales están soportados por aplicaciones Web y proveen gran variedad de medios para que los usuarios interactúen. – 21/09/2009 - http://en.wikipedia.org/wiki/Social_network_services -Interpretación del Inglés: Ariel Diaz Molina.
- **Sprint:** Periodo de desarrollo en el cual se divide una aplicación cuando se trabaja con Scrum. Usualmente tiene una duración de entre 15 y 30 días. – 21/09/2009 – Fuente: <http://es.wikipedia.org/wiki/Scrum> - Definición del Autor.
- **Sql Injection:** es una vulnerabilidad informática en el nivel de la validación de las entradas a la base de datos de una aplicación. El origen es el filtrado incorrecto de las variables utilizadas en las partes del programa con código SQL. Es, de hecho, un error de una clase más general de vulnerabilidades que puede ocurrir en cualquier lenguaje de programación o de script que esté incrustado dentro de otro. – 21/09/2009 – http://es.wikipedia.org/wiki/SQL_injection
- **UML: (Lenguaje Unificado de Modelado)** es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. – 21/09/2009 - <http://es.wikipedia.org/wiki/UML>
- **Web 2.0:** acuñado por Tim O'Reilly en 2004 para referirse a una segunda generación en la historia del desarrollo de tecnología Web basada en comunidades de usuarios y una gama especial de servicios, como las redes sociales, los blogs, los wikis o las folcsonomías, que fomentan la colaboración y el intercambio ágil de información entre los usuarios de una comunidad o red social – 21/09/2009 – http://es.wikipedia.org/wiki/Web_2.0
- **Widget:** es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de *widgets* o Widget Engine. - 21/09/2009 - <http://es.wikipedia.org/wiki/Widget>
- **Wiki:** es un sitio web cuyas páginas web pueden ser editadas por múltiples voluntarios a través del navegador web. Los usuarios pueden crear, modificar o borrar un mismo texto que comparten. Los textos o «páginas wiki» tienen títulos únicos. Si se escribe el título de una «página wiki» en algún lugar del wiki, esta palabra se convierte en un «enlace web» a la página web. – 21/09/2009 - <http://es.wikipedia.org/wiki/Wiki>
- **XSS: (Cross-site scripting)** es un tipo de inseguridad informática o agujero de seguridad basado en la explotación de vulnerabilidades del sistema de validación de HTML incrustado. – 21/09/2009 – <http://es.wikipedia.org/wiki/XSS>





ANEXO: DIAGRAMAS

- WBS de Tiempos del proyecto
- WBS de Progreso del Proyecto



ANEXO: PROJECT MANAGMENT

- Hoja de Cambios
- Historial de Aprobaciones



29 Anexo Administración de Proyectos

Hojas de Cambios			
<p><i>Este documento tiene por finalidad mantener un historial de cambios sugeridos por las autoridades correspondientes. Todos los cambios sugeridos, serán sometidos a estudio a los fines de mantener la consistencia y la integridad del proyecto. Aquellas versiones objetivos señalizadas con "X" representan versiones futuras sin especificar.</i></p>			
Componente / modulo	Descripción	Versión revisada	Versión Objetivo
Draft/ Objetivos	Reemplazar palabras como "sentar las bases teóricas". Por dar presunción de absoluto.	0.1.1	0.1.2
Draft/limites/ fuera	Contemplar la posibilidad de incluir un estudio de usabilidad de las pantallas	0.1.1	0.1.2
Draft/limites/ dentro	Contemplar agregar al los limites del sistema un captcha para validar la no alta automatizada por parte de robots	0.1.1	0.1.2
Draft/limites/ dentro	Contemplar agregar a los limites del sistema el mantenimiento de los mensajes en base de datos	0.1.1	0.1.2
Draft/Alcance	Agregar la asociación de los puntos del alcance a los puntos numerados de los limites	0.1.1	0.1.2
Draft/Alcance	Contemplar la inclusión al alcance del sistema el proceso de recuperación de contraseña	0.1.1	0.1.2
Draft/Alcance	Considerar la utilización del protocolo Ldap para directorios.	0.1.1	0.x.x
Draft/Alcance	Considerar incluir en el alcance del software alguna herramienta de folksonomia	0.1.1	0.1.2
Draft/Alcance	Estudiar la posibilidad de compartir archivos a través de terceros –(ejem Rapidshare)	0.1.1	0.1.2
Draft/ Objetivos	Aclara que la herramienta puede ser utilizada por cualquier universidad, no solo para la siglo 21	0.1.1	0.1.2
Tickets levantados:			10
Ticket Resueltos:			9
Draft/limites	Invertir el orden de los limites	0.2.3	0.3.1
Draft/limites	Generalizar los lo que queda fuera del los limites del sistema	0.2.3	0.3.1
Draft/UC	Modelo de UC	0.2.3	0.x.x
Draft/Modelo de Datos	Modelo de Datos	0.2.3	0.x.x
Draft/Mock ups	Prototipos	0.2.3	0.x.x
Tickets levantados:			2



Hojas de Cambios

Este documento tiene por finalidad mantener un historial de cambios sugeridos por las autoridades correspondientes. Todos los cambios sugeridos, serán sometidos a estudio a los fines de mantener la consistencia y la integridad del proyecto. Aquellas versiones objetivos señalizadas con "X" representan versiones futuras sin especificar.

Componente / modulo	Descripción	Versión revisada	Versión Objetivo
Ticket Resueltos:			2
Introducción	Agregar la versión minimalista de la visión.	0.3.3	0.4.1
Objetivos	Ajustes generales	0.3.3	0.4.1
Objetivos/particulares	Ajustes generales	0.3.3	0.4.1
Limites y Alcances	Ajustes Generales	0.3.3	0.4.1
Marco Teórico/Scrum	Construcción	0.3.3	0.4.1
Product Backlog List	Modificación de requerimientos: se eliminaran requerimientos de taxonomitas por requerimientos de control del ingreso realizado por los alumnos. Que no atenten contra la misión de la aplicación.	0.4.1	1.0.0
Glosario	Construcción de Un glosario con contenga la descripción de los términos utilizados conjuntamente con las fechas en los que fueron investigados	0.4.1	1.0.0
Tickets Levantados			7
Tickets Resueltos			7



Historial de Aprobaciones

Este documento tiene por finalidad mantener un historial de aprobaciones sobre los distintos componentes del trabajo. Cada apartado será considerado como un componente independiente.

Fecha	Sección / Apartado	Firma Docente
24/04/2008	Introducción	Calixto Maldonado
14/05/2008	Aprobación del Draft con observaciones	Calixto Maldonado
12/08	Aprobación con observaciones de versión 0.4.1	Calixto Maldonado
09/09	Aprobación con observaciones versión 0.4.5	CAE



BackLog List (4ta Fase)

La backlog se crea a partir de la cuarta fase debido a la posibilidad de agregar funcionalidades extras a la aplicación. Se establece las prioridades de las tareas a desarrollar a partir del issue tracker publicado en <http://code.google.com/p/mifacu/issues/list>

ID	Tipo	Estado	Prioridad	Resumen
3	Tarea	cerrado	Alta	numero de respuestas
5	Tarea	cerrado	Alta	Hacer temas Mas tomados
6	Tarea	cerrado	Alta	Hacer Preguntas mas tomadas
7	Tarea	cerrado	Alta	Hacer listado de apuntes mas bajados
9	Defecto	cerrado	Alta	Error en estudiantes que pueden ayudar
12	Tarea	cerrado	Alta	ver estudiantes que tienen apuntes sobre esta materia
34	Mejora	cerrado	Alta	Prevenir ataques xss
35	Defecto	cerrado	Alta	Mensaje persiste en sesión
36	Tarea	cerrado	Alta	agregar link para en la sidebar para editar el perfil
14	Tarea	Nuevo	Alta	crear modulo folcsonomías
16	Tarea	Nuevo	Alta	Enviar mail de preguntas a destinatarios
17	Tarea	Nuevo	Alta	Crear cambiar contraseña
20	Tarea	Nuevo	Alta	Denunciar Tema.
28	Defecto	Nuevo	Alta	Error en realizar pregunta
37	Tarea	Nuevo	Alta	El borrado de las preguntas por el admin. debe ser lógico no físico
38	Tarea	Nuevo	Alta	el borrado de los comentarios por el admini debe ser lógico no físico
11	Mejora	Nuevo	Baja	Agregar una descripción de lo que se pretende en el invitar amigos.
4	Tarea	cerrado	Crítica	Crear Componente de Publicidad Dirigida
15	Tarea	cerrado	Crítica	Crear la baja de los archivos
18	Tarea	cerrado	Crítica	Denunciar Comentario Noticia.
19	Tarea	cerrado	Crítica	Dar de baja comentario
21	Tarea	cerrado	Crítica	Denunciar Pregunta



BackLog List (4ta Fase)

La backlog se crea a partir de la cuarta fase debido a la posibilidad de agregar funcionalidades extras a la aplicación. Se establece las prioridades de las tareas a desarrollar a partir del issue tracker publicado en <http://code.google.com/p/mifacu/issues/list>

ID	Tipo	Estado	Prioridad	Resumen
24	Tarea	cerrado	Crítica	Denunciar Mensaje
25	Tarea	cerrado	Crítica	Denunciar Respuestas
26	Tarea	cerrado	Crítica	Denunciar Apunte
27	Tarea	cerrado	Crítica	Agregar buscador vivo de preguntas
39	Defecto	cerrado	Crítica	buscador vivo de materias
40	Defecto	cerrado	Crítica	buscador vivo de Apuntes
42	Defecto	cerrado	Crítica	Agregar Actualizar Publicidad en admin
43	Defecto	cerrado	Crítica	Crear modulo para resolver denuncias de comentarios
44	Tarea	cerrado	Crítica	Crear modulo para resolver denuncias de preguntas
45	Tarea	cerrado	Crítica	Crear modulo para resolver denuncias de Apuntes
22	Tarea	Nuevo	Crítica	Baja Tema
23	Tarea	Nuevo	Crítica	Baja Pregunta.
41	Tarea	Nuevo	Crítica	buscador vivo de preguntas de examen
8	Defecto	cerrado	Media	Crear target en el Perfil
10	Mejora	cerrado	Media	Agregar estudiantes que subieron apuntes
13	Defecto	cerrado	Media	mis preguntas a responder - no muestra invitar
1	Defecto	Invalido	Media	doctrine:build-all-reload doesn't work properly with sfDoctrine-GuardPlugin
2	Defecto	Nuevo	Media	Mostrar año del perfil no muestra la edad
29	Tarea	Nuevo	Media	Apuntes del perfil
30	Tarea	Nuevo	Media	Noticia del Perfil
31	Tarea	Nuevo	Media	Pregunta de Perfil
32	Tarea	Nuevo	Media	Estado Materia del Perfil
33	Tarea	Nuevo	Media	Examen Perfil
Ticket Abiertos				45
Ticket Resueltos				27