



Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento de las Ciencias de la Computación

**Herramienta docente para estimar tiempos de desarrollo de
proyectos de software**

Memoria para optar al título de Ingeniero Civil en Computación

Francisco Fabian Poblete Chavez

Profesor guía:
José Alberto Pino Urtubia

Miembros de la comisión:
María Cecilia Bastarrica Piñeyro
Agustín Antonio Villena Molina

Santiago de Chile
Agosto 2007

Resumen

El presente trabajo tiene por objetivo la construcción de una herramienta que apoye a los alumnos del Departamento de Ciencias de la Computación de la Universidad de Chile, que cursen ramos donde se deba desarrollar un proyecto de software de manera grupal, en la labor de dividir dicho proyecto en las tareas que lo componen, de forma que puedan realizar estimaciones acerca de la duración de estas tareas en equipo, y puedan mejorar la calidad de sus estimaciones a través de la práctica y el estudio de los resultados obtenidos anteriormente.

Para construir la aplicación se estudiaron diferentes recursos y herramientas disponibles, considerando factores como la facilidad de uso o comprensión para los usuarios, su interrelación con otros componentes del sistema y si poseían una política de código abierto, de forma de poder adaptar las soluciones encontradas a la aplicación que se deseaba implementar. Finalmente se decidió utilizar los recursos de bases de datos, Chat, foro y cartas Gantt provistos por las herramientas MySQL, X7 Chat, Simple Machines Forum y JpGraph.

Para modelar la división de los proyectos en sus tareas, se consideraron los conocimientos previos de los alumnos en el área de desarrollo de software impartidos en los ramos de la facultad, de forma de utilizar conceptos e ideas manejadas por los alumnos.

Para el diseño de las sesiones de estimación se tomó como base la metodología Wideband Delphi, debido a las ventajas que ofrece para realizar estimaciones de manera grupal. Tras analizar distintos modelos de discusión argumentativa, de forma de estudiar las ventajas que podrían agregar a las estimaciones, se realizaron cambios a la metodología Wideband Delphi original, modificando las fases existentes y agregando acciones comunicativas esperables en cada fase de forma de poder trasladar este nuevo modelo a la aplicación.

Se realizó una sesión de trabajo con los alumnos de un curso del Departamento de Ciencias de la Computación, donde debieron dividir el proyecto asignado a su grupo en las tareas que lo componen, y para cada tarea estimar de manera grupal la duración que consideraban tendría cada una. Si bien fue una versión simplificada de un caso de uso real de la herramienta, la mayoría de los alumnos consideró que el utilizarla podría ayudar a mejorar la calidad de sus estimaciones.

Finalmente, tras comparar las ventajas que agrega el modelo presentado frente a una reunión de estimación que utilice el modelo Wideband Delphi original, y tras ver el uso y las opiniones que dieron los alumnos sobre la aplicación, se puede concluir que su primer acercamiento con la herramienta fue positivo, y que su uso puede permitir que los alumnos logren estimar de mejor manera, aprendiendo de sus errores así como de las estimaciones y argumentos de los demás.

Agradecimientos

No se como partir los agradecimientos. Han pasado tantas cosas en todo este tiempo en la universidad que todo se agolpa en mi cabeza, pensando en que escribir primero.

Quiero agradecer a mi madre por todo el cariño que me ha entregado desde pequeño, por su infinita paciencia, por desvelarse en ocasiones conmigo cuando tenia que quedarme estudiando, por animarme cuando sentía que ya no podría seguir y por hacer de mi una persona de la que puedo sentirme orgulloso y feliz. Agradezco a mi padre por ser un ejemplo de esfuerzo y superación, por inculcarme a ser ordenado y disciplinado, y por mostrarme que en mis trabajos podía dar siempre más de lo que yo creía.

También quiero agradecer a Osvaldo por su incondicional apoyo, tanto en lo académico como en lo personal, por ayudarme a crecer como persona y a preocuparme por los demás, por enseñarme a vivir y disfrutar el presente, por sus dibujos, por su humor, por todo ese universo de animales parlanchines y por muchísimas cosas más que se resumen en un “gracias por habernos conocido”. Ashul.

Agradezco también a Pochi y a Gaby por haberse acercado en un momento muy difícil para mí, por todo su cariño, por todas las fiestas que hemos tenido la oportunidad de pasar juntos y por cuidarme en aquellos años en los que pensaba que nada podía salir bien. Espero poder estar ahí para ustedes también, en las buenas y en las malas. Sinceramente, sin su apoyo el término de mi carrera no habría sido posible.

A Mauricio, por todas las veces que nos hemos reído como menso por cosas como el verbo H y la “perfección” y por nuestros duelos eternos que traspasan generaciones de consolas. También agradezco a él y a su familia por hacerme sentir que a pesar de los años aún tengo un hogar fuera de mi casa.

A Paola y Rodrigo, por cuidarme en mis momentos depresivos y por ayudarme con mis estudios.

A Andrés, Carolita, Claudita, Maquita y Anita por ser un grupo tan bueno de amigos y por haberme aceptado como uno más de ustedes. Me frustra no haber podido conocerlos antes, pero en este caso es cierto que más vale tarde que nunca.

Finalmente, agradezco a toda la gente que me ayudó a llegar hasta el final de mi carrera de una forma u otra, pero que perdimos el contacto por diferentes motivos. Gracias por su apoyo en mis estudios y en mi desarrollo como persona.

Índice de capítulos

Resumen	2
Agradecimientos.....	3
Índice de capítulos	4
Índice de ilustraciones.....	6
Índice de tablas	7
1- Introducción.....	9
2- Justificación.....	11
3- Objetivo general	13
4- Objetivos específicos	13
5- Metodología.....	14
6- Revisión bibliográfica	17
7- Contexto académico	22
7.1 Estándar E.S.A.....	22
7.1.1 Fases y entregables	22
7.1.2 Roles.....	23
7.1.3 Valor pedagógico del estándar.....	25
7.2 Metodología “extreme programming”	25
7.2.1 Ideas principales.....	25
7.2.2 Valor pedagógico de la metodología	26
7.3 Estructura de descomposición del trabajo (EDT)	27
7.3.1 Ideas principales.....	27
7.3.2 Valor pedagógico de la EDT.....	27
8- Diseño de la aplicación	28
8.1 Integración con los conocimientos del alumno	28
8.1.1 Integración con estándar de la E.S.A.....	28
8.1.2 Integración con metodología extreme programming.....	28
8.1.3 Integración con el EDT	28
8.2 Consideraciones adicionales al contexto académico	28
8.3 Definición de servicios de la aplicación	29
8.4 Recursos	30
8.4.1 Recursos de comunicación	30
8.4.2 Recursos de implementación de código	31
8.4.3 Recursos de almacenamiento	32
8.4.4 Recursos de representación de progreso	32
8.5 Elección de recursos.....	33
8.5.1 Recursos de comunicación	33
8.5.2 Recursos de implementación de código	33
8.5.3 Recursos de almacenamiento de datos	33
8.5.4 Recursos de representación de progreso	33
8.6 Arquitectura.....	34
8.6.1 Arquitectura P2P.....	34
8.6.2 Arquitectura cliente servidor	34
8.6.3 Arquitectura orientada al servicio	34
8.6.4 Arquitectura elegida.....	34
8.7 Vista de capas.....	35
9- Modelo de estimación	37
9.1 Modelo argumental “IBIS”	37
9.2 Modelo “Dialogue Mapping”.....	38
9.3 Modelo de argumentación de Toulmin	40
9.4 Modelo “Beliefs, reasons and moves”	41
9.5 Modelo de Quignard y Baker	42

9.6 Búsqueda de un modelo aplicable a Wideband Delphi	43
9.6.1 Fase de presentación de las especificaciones.....	43
9.6.2 Fase de discusión del contexto de estimación.....	44
9.6.3 Fase de respuesta a formularios	45
9.6.4 Fase de resumen de estimaciones realizadas	45
9.6.5 Fase de discusión de estimaciones.....	45
9.6.6 Fase de análisis de éxito	46
9.7 Análisis de las acciones comunicativas encontradas.....	46
9.8 Acciones comunicativas planteadas.....	47
9.9 Fases planteadas.....	48
9.9.1 Fase de discusión.....	48
9.9.2 Fase de planteamiento del contexto.....	48
9.9.3 Fase de estimación.....	49
9.9.4 Fase de votación de estimaciones	49
9.9.5 Fase de acuerdo en estimación	49
9.10 Acciones comunicativas en cada fase.....	50
9.11 Diagrama resumen del modelo presentado.....	52
9.12 Beneficios y problemas del modelo	53
9.12.1 Ganancias en una reunión grupal	53
9.12.2 Pérdidas en una reunión grupal	53
9.12.3 Ganancias en una reunión grupal utilizando Wideband Delphi y la metodología propuesta..	54
9.12.4 Pérdidas en una reunión grupal utilizando Wideband Delphi y la metodología propuesta....	55
10- Funcionamiento del sistema	58
10.1 Actores del sistema.....	58
10.2 Diagramas de casos de uso	58
10.2.1 Diagrama de casos de uso para administración	58
10.2.2 Diagrama de casos de uso para proyectos.....	59
10.2.3 Diagrama de casos de uso para tareas	60
10.2.4 Diagrama de casos de uso para herramientas	61
10.2.5 Diagrama de casos de uso para estimaciones	61
10.3 Estructura de la aplicación.....	62
10.4 Diagramas de secuencia	63
10.4.1 Diagrama de secuencia para creación de proyectos	63
10.4.2 Diagrama de secuencia para creación de tareas.....	64
10.4.3 Diagrama de secuencia para eliminar proyectos o tareas	65
10.4.4 Diagrama de secuencia para ver carta Gantt.....	66
10.4.5 Diagrama de secuencia para modificar tarea.....	67
10.4.6 Diagrama de secuencia para ingreso a estimación	68
10.4.7 Diagrama de secuencia para fase de estimación	68
10.5 Diagramas de flujo de datos	69
10.5.1 Flujo de datos para creación de proyectos	69
10.5.2 Flujo de datos para creación de tareas	70
10.5.3 Flujo de datos para eliminar proyectos	71
10.5.4 Flujo de datos para eliminar tareas	72
10.5.5 Flujo de datos para ver carta Gantt.....	73
10.5.6 Flujo de datos para modificar tarea	74
11- Implementación de la aplicación	75
11.1 Análisis costo efectividad de herramientas.....	75
11.1.1 Características deseadas para recursos de comunicación.....	75
11.1.2 Características deseadas para recursos de almacenamiento	77
11.1.3 Características deseadas para recursos de representación	77
11.1.4 Análisis para los recursos de comunicación	78
11.1.5 Análisis para los recursos de almacenamiento	79
11.1.6 Análisis para los recursos de representación.....	80
11.1.7 Herramientas de apoyo a la aplicación	81

11.2 Instalación de herramientas iniciales	82
11.3 Implementación del sistema	85
11.3.1 Modelo de datos	85
11.3.2 Permisos	90
11.3.3 Modificaciones a código de herramienta de Chat	90
11.3.4 Modificaciones a código de herramienta de foro.....	91
11.3.5 Modificaciones a herramienta de representación.....	91
11.3.6 Modificaciones a herramienta de representación de estructura de árbol	92
12- Uso de la aplicación	93
12.1 Modo administración	93
12.1.1 Registro del equipo docente.....	93
12.1.2 Creación de categorías	94
12.1.3 Creación de foros	94
12.1.4 Creación de grupos de usuarios.....	95
12.1.5 Registro de alumnos.....	96
12.2 Modo alumno	97
12.2.1 Vista de mantenimiento	97
12.2.2 Vista de estimación	104
12.2.3 Vista de herramientas.....	108
13- Discusión.....	113
13.1 Discusión sobre el modelo de estimación presentado	113
13.2 Discusión sobre la aplicación.....	114
13.2.1 Experiencia de uso de la aplicación	114
13.2.2 Discusión sobre experiencia de uso.....	115
13.2.3 Discusión sobre modificaciones a la aplicación	116
14- Conclusiones.....	118
15- Referencias	119
Anexos.....	122
A- Respuestas de alumnos a cuestionario de evaluación	122

Índice de ilustraciones

Ilustración 1: flujo de proceso con Wideband Delphi	14
Ilustración 2: ciclo de vida de software según estándar de la E.S.A.	23
Ilustración 3: arquitectura de la aplicación.....	35
Ilustración 4: capas que componen la aplicación.....	35
Ilustración 5: estructura básica del modelo IBIS	38
Ilustración 6: mapa de diálogo de una reunión acerca de la planificación de presupuestos.....	39
Ilustración 7: mapa de diálogo usando oraciones para representar ideas	40
Ilustración 8: modelo de argumentación de Quignard y Baker	43
Ilustración 9: diagrama del modelo presentado	52
Ilustración 10: diagrama de casos de uso para administrador	58
Ilustración 11: diagrama de casos de uso para proyectos.....	59
Ilustración 12: diagrama de casos de uso para tareas	60
Ilustración 13: diagrama de casos de uso de herramientas	61
Ilustración 14: diagrama de casos de uso de estimaciones	61
Ilustración 15: estructura de la aplicación	62
Ilustración 16: diagrama de secuencia para creación de proyectos	63
Ilustración 17: diagrama de secuencia para creación de tareas.....	64
Ilustración 18: diagrama de secuencia para eliminar proyectos o tareas.....	65
Ilustración 19: diagrama de secuencia para ver carta Gantt	66
Ilustración 20: diagrama de secuencia para modificar tarea	67
Ilustración 21: diagrama de secuencia para ingreso a estimación	68
Ilustración 22: diagrama de secuencia para fase de estimación	68
Ilustración 23: flujo de datos para creación de proyectos.....	69

Ilustración 24: flujo de datos para creación de tareas	70
Ilustración 25: flujo de datos para eliminar proyectos	71
Ilustración 26: flujo de datos para eliminar tareas.....	72
Ilustración 27: flujo de datos para ver carta Gantt	73
Ilustración 28: flujo de datos para modificar tareas.....	74
Ilustración 29: Modelo inicial de datos	85
Ilustración 30: modelo de datos para tablas relacionadas con cuentas de usuario	87
Ilustración 31: modelo de datos para tablas relacionadas con modificaciones	87
Ilustración 32: modelo de datos para tablas relacionadas con modificaciones	88
Ilustración 33: modelo de datos para tablas relacionadas con estimaciones.....	88
Ilustración 34: modelo de datos para tablas relacionadas con el glosario	89
Ilustración 35: registro del equipo docente	93
Ilustración 36: creación de categorías	94
Ilustración 37: creación de foros	95
Ilustración 38: creación de grupos de usuarios.....	95
Ilustración 39: registro de alumnos desde foro	96
Ilustración 40: registro de alumnos desde aplicación	97
Ilustración 41: creación de proyectos.....	98
Ilustración 42: creación de tareas	99
Ilustración 43: vista de carta Gantt.....	99
Ilustración 44: sala de discusión de proyectos.....	100
Ilustración 45: vista de log de discusión.....	100
Ilustración 46: formulario de modificación de proyecto.....	101
Ilustración 47: formulario para eliminar un proyecto	101
Ilustración 48: vista de detalle para una tarea	102
Ilustración 49: formulario de modificación de tarea.....	102
Ilustración 50: formulario para eliminar una tarea.....	103
Ilustración 51: sala de discusión para una tarea.....	103
Ilustración 52: modificar opciones de estimación.....	104
Ilustración 53: formulario de ingreso a estimación.....	104
Ilustración 54: formulario de discusión	105
Ilustración 55: formulario de contexto de estimación	106
Ilustración 56: formulario de estimación.....	106
Ilustración 57: formulario de opinión de estimaciones	107
Ilustración 58: formulario de acuerdo en la estimación.....	107
Ilustración 59: vista de ajuste de estimaciones.....	108
Ilustración 60: cuestionario de fortalezas.....	109
Ilustración 61: glosario	109
Ilustración 62: formulario para estimador de complejidad	110
Ilustración 63: herramienta de foro.....	111
Ilustración 64: actualización de glosario	112

Índice de tablas

Tabla 1: acciones del modelo “Beliefs, reasons and moves”	42
Tabla 2: respuestas a acciones del modelo “Beliefs, reasons and moves”	42
Tabla 3: acciones comunicativas planteadas.....	47
Tabla 4: acciones comunicativas en cada fase.....	50
Tabla 5: ganancias de una reunión grupal.....	53
Tabla 6: pérdidas en una reunión grupal	53
Tabla 7: ganancias en una reunión grupal con Wideband Delphi y la metodología propuesta.....	55
Tabla 8: pérdidas en una reunión grupal con Wideband Delphi y la metodología propuesta	55
Tabla 9: rangos de puntaje para características de herramientas.....	75
Tabla 10: características para Chat	76
Tabla 11: características para foro.....	76

Tabla 12: características para base de datos	77
Tabla 13: características para herramienta de representación.....	78
Tabla 14: análisis de características para alternativas de Chat.....	78
Tabla 15: análisis de características para alternativas de foro	79
Tabla 16: análisis de características para alternativas de base de datos	80
Tabla 17: análisis de características para herramientas de representación.....	80
Tabla 18: herramientas de apoyo a la aplicación.....	81
Tabla 19: parámetros modificados de php.....	83
Tabla 20: llaves foráneas iniciales	85
Tabla 21: llaves foráneas del modelo	89
Tabla 22: modificaciones a herramienta de Chat.....	90
Tabla 23: modificaciones a herramienta de foro	91
Tabla 24: modificaciones a herramienta de representación	92

1- Introducción

La estimación de tiempos de desarrollo de software es un tema que ha sido estudiado constantemente por más de tres décadas. Es un proceso que debe realizarse de manera continua a lo largo de la creación de una aplicación, con el fin de incluir las diferentes tareas que van surgiendo a medida que dicha aplicación va tomando forma. Desde la toma de requisitos del usuario, en donde se establece qué es lo que se desea que haga el software, hasta el momento en que la aplicación es instalada en el lugar solicitado por el cliente, todo necesita una estimación de tiempo asociada a la tarea que se debe realizar. Sin embargo, obtener una estimación ajustada al tiempo real de desarrollo no es fácil.

Tomando como base el modelo de licitación para desarrollos de software, es posible que una estimación mal ajustada lleve a una empresa a caer en una situación desfavorable, según la estimación haya sido inferior o superior a la necesidad real. En el caso de haber asignado un tiempo inferior al real que tomará la implementación de una herramienta, este error forzará a la empresa a dedicar más tiempo del planificado inicialmente a seguir trabajando en el proyecto, incurriendo con esto en costos tanto de salario del personal que debe seguir trabajando hasta finalizar el proyecto, como de los posibles clientes con los que se podría haber iniciado un proyecto nuevo. Además, estos gastos que genera el tiempo de desarrollo no planificado deben ser costeados con un presupuesto diferente al que se había planificado inicialmente. Si, por el contrario, una empresa asigna un tiempo considerablemente mayor al que de verdad tomará un proyecto de software con el fin de, por ejemplo, tener un período de holgura con el cual manejar posibles contratiempos, es posible que otra empresa que compita por ganar el mismo proyecto termine adjudicándose por ofrecer un tiempo de desarrollo inferior, perdiendo así un potencial cliente. Así pues, ambas situaciones son un grave problema, sobre todo para las empresas que están comenzando a aparecer en el mercado. La solución a buscar entonces debe ser una que permita ofrecer una estimación lo más realista posible.

Para resolver este problema se han ideado diversas metodologías de estimación, intentando abarcar la mayor cantidad de los factores que afectan el desarrollo de un proyecto, con el fin de lograr resultados más fieles a la realidad. Dentro de las metodologías de estimación pueden distinguirse dos grupos: las metodologías paramétricas, que realizan una estimación basadas en una expresión matemática que considera diferentes factores como las líneas de código aproximadas del proyecto o las funciones que contiene; y las basadas en juicios e intuiciones, que utilizan material histórico y las experiencias de los participantes para producir una estimación.

Dentro de las metodologías basadas en juicios, el objetivo que se busca es reunir los puntos de vista de todos los integrantes de un grupo de desarrollo para llegar a un acuerdo común, cuya estimación será mejor que las estimaciones que cada uno podría haber hecho por separado. Una de estas metodologías es llamada Wideband Delphi, creada por Barry Boehm y John A. Farquhar cerca de 1970 [1], modificación de la metodología de Juicio Experto propuesta por estos mismos autores, la cual consiste en revisar las opiniones de un grupo de expertos referentes a un tema en particular, comentar estas estimaciones, y realizar nuevas rondas de estimación basándose en las respuestas recién discutidas.

Se han realizado estudios con el fin de determinar si un tipo de metodología de estimación logra entregar mejores resultados que el otro, sin embargo los resultados no parecen favorecer a ningún grupo completamente [2]. Esto se debe a que ningún desarrollo es completamente

similar a otro, por lo que ningún modelo podrá englobar completamente todas las características que tenga un proyecto. Si un proyecto basado en un lenguaje de programación Web es desarrollado por el mismo equipo que acaba de completar exitosamente un proyecto anterior en un lenguaje orientado a objetos, no implica que este nuevo proyecto vaya a ser igual de exitoso.

Dentro del ámbito universitario, la estimación de los tiempos de desarrollo de los proyectos suele ser más difícil aún para los estudiantes, considerando la poca experiencia que poseen. Además el intentar realizar comparaciones entre las estimaciones realizadas utilizando metodologías paramétricas por cada grupo que conforma el curso no es posible, debido a las diferencias tanto en los lenguajes que usarán para programar como en las herramientas que utilizarán, lo que afectará posibles factores de los modelos como las líneas de código o los puntos de función.

El objetivo de esta memoria es crear una herramienta orientada a la formación del estudiante utilizando una versión modificada del esquema propuesto en Wideband Delphi. Esta herramienta les permitirá, junto al grupo que les haya sido asignado en alguno de los ramos de la carrera, realizar divisiones de un proyecto en sus tareas correspondientes, realizar estimaciones de tiempo para estas tareas, asignarles responsables y analizar el desempeño del grupo en cada una de ellas. De esta forma, la hipótesis de esta memoria pretende demostrar que el alumno puede mejorar su capacidad para realizar estimaciones a través de la práctica junto a sus compañeros. Al utilizar constantemente la herramienta, las estimaciones deberían ser progresivamente más ajustadas.

2- Justificación

El problema de estimación de tiempos de desarrollo de software es un tema que ha acompañado al alumno desde que ingresa a la Universidad, al necesitar estimar cuanto tiempo considera que debe otorgar a cada una de las tareas que le han sido asignadas en sus ramos de computación. Más adelante se enfrenta a situaciones de trabajo reales en los ramos CC51A, Ingeniería de Software, y CC61A, Proyecto de Software, en los que deberá desarrollar aplicaciones en la duración de un semestre, período que dura cada uno de los cursos. Si bien es cierto que la duración de cada uno de los proyectos en estas asignaturas han sido adecuadas por el equipo docente para que su duración no exceda la del semestre académico, el alumno debe lograr distribuir de forma correcta su tiempo con el objetivo de tener prototipos operacionales de la solución de software solicitada para cada una de las iteraciones que se hayan establecido, sin llegar al final del semestre con un producto que no cumpla lo acordado con su cliente.

Sin embargo, debido a la falta de experiencia que poseen algunos de los alumnos, las estimaciones que ellos realizan no resultan completamente efectivas, y por lo general se debe realizar trabajo adicional en el último momento para poder finalizar los proyectos a tiempo.

Al terminar los estudios, el alumno egresado deberá formar parte de grupos de desarrollo donde cada miembro puede poseer una ideología y forma de estimación completamente diferente al resto, con lo que se dificulta el proceso mediante el cual se logra llegar a una estimación final adecuada.

Como se puede ver, es un problema que siempre está presente, pero no se refuerzan estas habilidades en el alumno durante sus estudios. La idea tras la creación de esta herramienta es que permita que un alumno que cursa alguno de los ramos de la carrera de Ingeniería Civil en Computación logre adquirir habilidades básicas de estimación en grupo mediante la utilización con sus pares de esta aplicación, refinando sus estimaciones a medida que interactúan con ella, ganando experiencia en la materia a lo largo del proceso.

La elección de Wideband Delphi como metodología base para realizar estimaciones se basó en que en los cursos que incluyen el desarrollo de una aplicación como su actividad principal de evaluación a lo largo del semestre, se debe trabajar en grupos donde no todos poseen los mismos conocimientos acerca del problema a resolver, con lo que la estimación de la complejidad y tiempo de desarrollo que se le asignará a cada tarea que de él se desprenda puede variar mucho de una persona a otra.

Utilizando una versión modificada de la metodología Wideband Delphi, donde los expertos son reemplazados por los alumnos integrantes del grupo de desarrollo, la hipótesis que se pretende probar es que los estudiantes pueden aumentar incrementalmente su capacidad de estimación a través de la interacción con la aplicación y la realización continua de estimaciones, compartiendo sus experiencias previas en la implementación de aplicaciones, para que así se incorporen factores que otros miembros del grupo no hayan tenido la oportunidad de experimentar aún, llegando con esto a una estimación de tiempo más realista que la que podría hacer cada miembro del grupo por separado.

Si se realizan estimaciones continuamente, con el fin de verificar qué tan acertadas fueron las estimaciones anteriores, es posible que el alumno acumule conocimiento sobre el como estimar mejor a través de la práctica [20].

3- Objetivo general

Crear una aplicación Web que sea utilizada con fines académicos para estimar el tiempo que tomará el desarrollo de un software.

4- Objetivos específicos

- Crear una herramienta que permita participar y practicar a los alumnos del Departamento de las Ciencias de la Computación con la actividad de estimar el tiempo que tomarán las diferentes tareas de las que se compone el desarrollo de un software, en alguno de los ramos que esté cursando.
- Dar a conocer a los alumnos las ventajas de los métodos de estimación, tanto en su vida estudiantil como profesional.
- Realizar comparaciones entre las estimaciones de los alumnos que utilizan la herramienta, con aquellas de los alumnos que no la utilizan, con el fin de verificar si la herramienta los lleva a estimaciones más realistas y precisas.
- Almacenar información acerca del uso que se le da a la aplicación, para poder comprobar qué partes de ésta son más útiles a los alumnos.
- Efectuar cerca del final del curso nuevas estimaciones con los alumnos que utilizaron la herramienta, para comprobar si mejoraron de alguna manera las formas en las cuales realizan sus estimaciones y validar con esto su aprendizaje en la materia.
- Medir por medio de cuestionarios la aceptación o rechazo de la herramienta por parte de los alumnos, solicitando sugerencias para realizar modificaciones posteriores a la aplicación.
- Estudiar la efectividad real de la herramienta en un ambiente de desarrollo de software basado en los resultados de los objetivos anteriores.

5- Metodología

El programa permitirá la participación de los alumnos al momento de estimar tiempos de desarrollo basado en la metodología llamada Wideband Delphi. Esta metodología ha sido definida como “una forma para estructurar el proceso de comunicación en grupo, de manera de permitirle actuar como un todo en la resolución de un problema complejo” [3].

La metodología puede resumirse en los siguientes pasos [4].

- 1- Se planifica una reunión a la cual se invitará un grupo de personas llamadas “expertos”, que posean conocimientos en el tema que se desea discutir.
- 2- Se celebra la reunión, en la cual se explica la situación en la que se encuentra actualmente el problema. De esta forma todos los participantes sabrán a qué se enfrentan y podrán medir sus conocimientos en el tema.
- 3- Luego de esta reunión inicial, cada participante debe reunir información adicional por su propia cuenta, a fin de mejorar sus conocimientos y poder realizar aportes de mayor valor en los siguientes pasos.
- 4- Posteriormente el grupo vuelve a reunirse, esta vez con el fin de opinar respecto al problema y realizar pronósticos o analizar diferentes formas de resolverlo, según sea la naturaleza del tema.
- 5- Cuando las opiniones ya han sido entregadas, se recopilan y agrupan según sus puntos de vista.
- 6- El último paso corresponde al análisis de las diferentes opiniones entregadas, intentando comprender los puntos de vista de los demás participantes con el fin de ampliar el rango de visión del grupo, mostrándoles detalles que posiblemente no habían considerado.

Luego del paso 6, una de dos situaciones puede darse:

- Las aproximaciones o soluciones entregadas concuerdan o son muy similares, diferenciándose en factores considerados menores por los miembros del grupo. En este caso, la aproximación o solución es aceptada como válida, por lo que puede aplicarse.
- Las aproximaciones o soluciones entregadas son muy distintas entre sí, y tras discutir las razones de porque ocurrió esto, muchos de los participantes se dieron cuenta que habían factores que no habían considerado en su análisis. En este caso, puede realizarse una nueva iteración del proceso, con el fin de obtener nuevas soluciones o aproximaciones al problema, repitiéndose cuantas veces sea necesario hasta que el grupo considere que una solución es satisfactoria.

El siguiente diagrama ilustra el ciclo recién descrito.



Ilustración 1: flujo de proceso con Wideband Delphi

En este proyecto, se utilizará una versión adaptada del modelo Wideband Delphi original, donde los expertos serán reemplazados por los alumnos que estén cursando alguno de los ramos de la carrera. Para aplicar esta metodología, se les ofrecerá a los alumnos en la aplicación una sección donde podrán exponer sus ideas y preguntas referidas al proyecto que deben desarrollar, y tendrán que realizar una estimación inicial anónima del tiempo que creen les tomará finalizar la implementación e implantación de todas las tareas que compongan el software. Luego, deberán opinar respecto a las estimaciones que habrán realizado los demás miembros del equipo, analizando las diferentes razones que ofrecerá cada uno. En base a este análisis, deberán realizar una nueva estimación, hasta encontrar una respuesta que represente a la mayoría del grupo. Este proceso debe realizarse continuamente con cada una de las tareas que se desprendan del análisis del proyecto en el que estén, con el objetivo de que adquieran práctica y puedan mejorar sus estimaciones.

La razón por la cual el anonimato en las respuestas es importante es debido a que de esta forma se evita la presión que puede ejercer una persona sobre las otras, que un grupo de personas elimine la posible diversidad de respuestas que podrían surgir por ser mayoría en el diálogo, o que se lleguen a discusiones por diferencias radicales en los puntos de vista de los participantes [5].

Es importante destacar que la herramienta deberá tener un uso cíclico, realizando estimaciones de tiempo cada vez que aparezcan nuevas tareas a causa del desarrollo, así como al finalizar el curso, para poder comprobar efectivamente el aprendizaje de los alumnos.

Los servicios que se pretende entregar con la aplicación son los siguientes:

1. **Recolección de información del proyecto:** para apoyar a la planificación y reunión inicial, se ofrecerá a los miembros del grupo la posibilidad de que ingresen los datos que caracterizarán su proyecto como son un nombre identificador, un margen inicial de fechas entre las que el proyecto debe ser realizado, y una descripción de en qué consistirá el proyecto.
2. **Formularios guías:** previo a las tareas de estimación, cada alumno perteneciente al grupo podrá tener la posibilidad de responder un formulario con preguntas básicas acerca de su experiencia como desarrollador, con el fin de que comience a considerar los diferentes factores que podrían influir en el tiempo de desarrollo de un software y en el rol que podría tomar en el grupo, de forma que pueda complementar sus habilidades con las de los demás. Algunas de estas preguntas podrían corresponder a:
 - ¿Se exige algún lenguaje especial para desarrollar el software? ¿Conozco ese lenguaje?
 - ¿Están claramente diferenciadas las tareas que componen este software? ¿Cuántas son? ¿Cuáles son?
 - ¿Conozco a mis compañeros de grupo? ¿He trabajado antes con ellos?
 - ¿Poseo habilidades de líder? ¿Trato más de escuchar que de hablar?
3. **Herramienta de discusión:** debido a que gran parte del proceso para llegar a una estimación final se desarrolla a través de la discusión y el análisis en conjunto de los resultados, es necesario ofrecer herramientas para que los miembros del grupo puedan opinar de forma organizada referente al tema, hacer sus estimaciones de forma anónima, y poder discutir luego acerca de ellas. Para esto se creará un servicio de comunicación que permita tener a la vista los temas a tratar, destacando el tema actual,

las estimaciones o respuestas dadas por los miembros del grupo referentes al tema actual, y un área de conversación para que los alumnos puedan exponer los puntos que ellos consideren importantes referentes a la estimaciones que se estén discutiendo. Esta parte de la herramienta cubrirá los pasos de Reunión de estimación, Tareas de recopilación y Revisión se resultados. Puede ser de gran utilidad para los alumnos que estas conversaciones y predicciones sean almacenadas, para revisar temas ya tratados con el fin de evitar repetir discusiones ya realizadas.

4. **División de tareas:** cuando se tenga una visión inicial de cuánto durará el proyecto, éste se puede dividir en las tareas que lo compondrán, asignando responsables a cada una de ellas, y realizando con esto nuevas estimaciones para estas tareas. De esta manera se podrá tener una idea más detallada del problema a resolver, cuyo análisis puede ser comparado con la estimación inicial realizada por los alumnos cuando aún no conocían bien el problema, obteniendo con esto una mejor visión del tiempo total de desarrollo [6]. Esta división de un proyecto en las tareas que lo componen y su respectiva asignación de tiempos de desarrollo servirá para controlar el trabajo realizado por cada uno de los miembros del grupo y ver el estado de avance total del proyecto en función de las tareas que ya han sido completadas. Es importante que los alumnos actualicen constantemente sus estimaciones de tiempo por cada tarea con el objetivo de afinar sus estimaciones a través de la práctica.
5. **Gráficos de planificación:** Para que los alumnos puedan llevar las estimaciones de tiempo realizadas a la práctica, se ofrecerá la posibilidad de crear gráficos utilizando las mediciones que hayan realizado dentro de la misma aplicación, basados en los tiempos que han asignado a las diferentes tareas. Se ofrecerá la posibilidad de crear cartas Gantt debido a que son ampliamente utilizadas en otros ramos de la Facultad. En estos mismos gráficos se podrá observar el porcentaje de trabajo realizado que lleva cada uno de los encargados de las tareas, incluyendo pequeñas anotaciones de forma que todos puedan saber en que se ha estado trabajando últimamente.
6. **Formulario de estimación de complejidad:** Una herramienta que serviría a los alumnos para obtener una medición de la complejidad del problema que intentan resolver es un formulario que les solicite aproximaciones según sus propios criterios a los factores de complejidad de un estimador algorítmico, como es el caso del modelo COCOMO [7]. El análisis de los valores que otorgaron a cada uno de los factores del algoritmo que calcula el estimador de complejidad puede discutirse utilizando esta misma herramienta, analizándolo según Wideband Delphi.
7. **Pizarra de mensajes:** Será posible que cada miembro del grupo pueda expresar su avance referente a una tarea, o su opinión respecto a un tema que se esté discutiendo dejando notificaciones en una pizarra de mensajes dentro del mismo programa. Esto es de gran utilidad, sobre todo para celebrar pre reuniones de forma asíncrona, ya que no todos los participantes podrán estar siempre presentes cuando se realicen discusiones. Así, cuando se celebre la reunión efectiva, cada persona poseerá la información necesaria acerca del estado de avance en que se encuentra el grupo referente al problema [8].
8. **Glosario:** Se proveerá de un glosario con la terminología habitual utilizada en las labores de desarrollo de software, que podrá servir de ayuda al alumno en la búsqueda de información referida a los diferentes temas que sean tratados en sus cursos.

6- Revisión bibliográfica

Se ha optado por la metodología Wideband Delphi para el desarrollo de este proyecto, dado que fomenta el trabajo en grupo y es ideal para analizar la duración de tiempos que tomará cada una de las partes que compongan el desarrollo de un software [6], [9].

En el paper “Software Cost Estimation” [10], se comenta que el juicio experto es utilizado ampliamente para estimar costos, basado tanto en las normas de la industria que se han mantenido vigentes en el tiempo, como en la experiencia adquirida en diferentes proyectos. Este factor, la experiencia, es algo que sólo se puede lograr a través de la práctica. Considerando la poca experiencia profesional que se tiene al estar cursando los ramos de la carrera, es común cometer errores de estimación al momento de tener que evaluar la duración de las tareas que componen alguno de los proyectos asignados en los ramos que se estén cursando. Más aún, si al momento de tomar los requisitos para este proyecto estos no fueron obtenidos de forma clara, o si se pasó por alto el considerar alguna tarea importante en la planificación del proyecto, obligará a realizar nuevas tomas de requerimientos y reuniones con el cliente, lo que llevará a tener que efectuar una nueva estimación que abarque efectivamente estos hitos.

Considerando que en estos ramos se recomienda realizar la asignación de las tareas que componen un proyecto de forma grupal, con el objetivo de poder favorecer la entrega de ciertos temas a personas que posean mayores habilidades en ellos, es natural pensar que el método de estimación de tiempos que se utilice deba permitir que todos los participantes puedan dar su opinión y sus puntos de vista, tanto en la asignación como en la estimación de tiempos de estas tareas. Wideband Delphi se beneficia de las opiniones que surgen del trabajo en grupo, sin embargo, en el paper recién mencionado [10] se destacan los siguientes puntos en contra de este método:

- 1- El enfoque no es repetible y la forma para obtener un estimado no son explícitos.
- 2- Es difícil encontrar estimadores altamente expertos para cada proyecto nuevo.
- 3- La relación entre el costo y el tamaño del sistema no es lineal. El costo tiende a aumentar exponencialmente con el tamaño. El juicio experto es apropiado sólo cuando el proyecto actual y los anteriores son similares.
- 4- Manipulaciones de presupuesto enfocadas a evitar sobrecostos hacen cuestionables los datos de las experiencias anteriores.

Referente al punto 1, dado que el objetivo de esta memoria es mantener un registro de las razones y factores considerados por cada miembro del grupo para poder llegar al acuerdo final, el método con el cual se llegó al acuerdo será siempre visible.

El argumento expuesto en el punto 2 es cierto, no es fácil encontrar estimadores que estén preparados para cada uno de los posibles proyectos que pueda intentar realizar una empresa. En respuesta a esto, y considerando el alcance académico del presente proyecto, utilizando la metodología modificada basada en Wideband Delphi se pretende formar de a poco mejores estimadores a través de la práctica constante y la discusión con el resto de sus compañeros de grupo.

El punto 3 apunta a que en el mundo laboral es difícil encontrar proyectos donde el conocimiento previo pueda ser utilizado de igual manera en un proyecto actual. Si bien esto es cierto, no implica que el conocimiento previo no pueda ser utilizado como argumento por alguno de los miembros del grupo como factor determinante de su estimación de tiempo, y que pueda servir a que otro participante lo considere como un factor que no había sido tomado en cuenta. Nuevamente dentro del ámbito académico, los proyectos realizados (tanto en el ramo Ingeniería de Software como en Proyecto de Software) son presentados con una duración límite inicial similar (el tiempo de duración del semestre), por lo que parte de la planificación realizada en un ramo puede ser útil para el otro en el caso de que se trate con tecnologías similares, e incluso cuando ambas planificaciones involucren trabajar por un cierto tiempo aprendiendo tecnologías desconocidas.

Finalmente, en el punto 4, la información y experiencia previa que manejan los alumnos para realizar sus estimaciones no depende de la información almacenada por una organización externa, sino de sus propios conocimientos acerca de sus potencialidades y limitaciones, por lo que sabrá definir personalmente cuán buena fue la experiencia adquirida anteriormente y cómo transferirla al nuevo proyecto.

Al momento de elegir qué metodología de estimación de tiempo de desarrollo elegir para el presente trabajo se consideraron también otros modelos entre los que se destacan algunos paramétricos como COCOMO y SEER-SEM. Este tipo de metodologías están basadas en un modelo algorítmico que sigue una ecuación similar a:

$$\text{esfuerzo} = C \times PM^S \times M$$

donde C es un factor de complejidad, PM es una métrica asociada con el producto que se desea desarrollar, ya sea en su tamaño o en el número de funcionalidades que se desea que tenga, M es un multiplicador que se crea combinando diferentes atributos del producto y proceso de desarrollo, y S es un exponente que varía según sea el tamaño del proyecto, aunque por lo general su valor es 1 [6].

Estas metodologías poseen como falencia la dificultad que representa estimar el valor de PM al inicio de un proyecto donde se tiene poca información del mismo, o que los valores que toman C y M son asignados en una forma más bien subjetiva.

Esta subjetividad puede ser reducida si se analiza el mismo proyecto con un grupo de personas con conocimientos en la materia, el cual es precisamente el objetivo del método Wideband Delphi. Debido a esto, se decidió mantener en la aplicación la posibilidad para que los alumnos interactúen con el modelo COCOMO, ofreciéndoles la posibilidad que produzcan ellos mismos sus factores de multiplicación y métricas de complejidad.

Inicialmente el modelo COCOMO fue diseñado por Barry Bohem en 1981 con el objetivo de entregar una estimación del número de meses-hombre que tomaría desarrollar un producto de software [11]. Sin embargo con el paso del tiempo la forma en la que se trabajaba cambió de trabajar con computadores centrales o “mainframes” a trabajar con computadores personales, donde fue posible reutilizar código y software ya preparado para facilitar el desarrollo, por lo que el modelo fue modificado para reflejar este hecho.

La ventaja del modelo COCOMO es que es posible adaptarlo a distintas condiciones de trabajo, ya que considera factores como la información con la que se cuenta acerca del proyecto, las

características que posee el equipo de trabajo, el hardware a utilizar en el desarrollo y el nivel de calidad que se espera del producto final.

Una de las desventajas que presenta tanto el modelo COCOMO como COCOMO II [12] para ser utilizado como metodología base en esta memoria es que uno de los parámetros que utiliza corresponde al tiempo estimado de desarrollo calculado en personas-mes de trabajo. Sin embargo, en el ámbito académico en el que se pretende utilizar la herramienta, esta tipo de estimación dista mucho de la que se haría basada en el mundo laboral, considerando los horarios que se deben cumplir con los demás ramos que estén cursando los alumnos. También en el ámbito académico, se debe considerar que el tiempo asignado por un alumno a un proyecto puede no ser constante debido a la posible sobrecarga académica que tenga con otros ramos, situación que no se da en el ámbito laboral real (exceptuando los casos en que algún profesional tenga diversos proyectos asignados) donde la persona tendrá un horario de trabajo fijo sujeto a las regulaciones de la empresa en la cual se encuentre. Esta última situación se asemeja más a la realidad para la cual fue pensada este modelo.

Otro punto a considerar es que cada factor de complejidad está previamente definido, frenando la libertad de los alumnos de definir por sí mismos cuáles factores podrían llegar a influir en el desarrollo de un proyecto, y en qué medida. Además la catalogación existente de los diferentes factores podría centrar al alumno en ellos, limitándolo a pensar sólo en ellos en lugar de considerar otros que podrían existir pero cuyos valores no aparezcan tabulados.

SEER-SEM, otro modelo paramétrico, considera diferentes elementos para proveer sus estimaciones de esfuerzo, duración, personal necesario y defectos. Entre estos se incluyen el tamaño del software a desarrollar, el progreso actual alcanzado, las herramientas con las que cuentan los desarrolladores, la factibilidad de desarrollo con las herramientas actuales y los cambios en el producto final al aplicar restricciones de personal y planificación inicial [13].

Al igual que COCOMO, se basa en estadísticas para realizar sus estimaciones, utilizando varios factores como complejidad del proyecto y puntos de función [13]. Con esto, se genera el problema de tener que pensar el proyecto según las características solicitadas por el modelo. Existe ya una implementación de este modelo, sin embargo, ésta posee una visión más comercial y la licencia del programa no es de código abierto [14].

Comparado con otros métodos de estimación basados en ponderadores que buscan obtener resultados exactos, Wideband Delphi se beneficia del diálogo que generan sus participantes para establecer como posible acuerdo un rango de valores aceptables o intervalo de confianza [15]. Además, es importante mencionar que los modelos de estimación paramétricos utilizan métricas que no son siempre equivalentes al momento de realizar estimaciones. Por ejemplo, un proyecto para realizar una aplicación Web no poseerá las mismas líneas de código o puntos de función que una aplicación realizada para ser instalada y ejecutada dentro de un equipo en particular. Aún más, dentro de este último tipo de aplicaciones existen diferencias en el valor de estas variables, dependiendo de si se trabaja en un lenguaje orientado a objetos o en un lenguaje funcional [2].

Junto con los métodos recién descritos se analizó también la metodología PERT, que en su primera etapa de planificación posee un comportamiento que se asemeja levemente a Wideband Delphi [16].

La metodología PERT fue desarrollada en 1958 por la compañía Booz Allen Hamilton, bajo

petición del Departamento de Defensa de los Estados Unidos. Su principal objetivo fue simplificar la planificación de proyectos grandes y complejos a través del análisis de las tareas que lo componen, e identificando el tiempo mínimo necesario para completarlos.

La principal ventaja que posee este modelo es que permite incluir la incerteza que exista en el desarrollo de cada una de las tareas que compongan un proyecto, siendo posible su planificación sin conocer completamente la duración o los detalles de cada una de sus actividades.

Esta metodología puede utilizar en un principio la interacción entre los miembros de un grupo estimador para determinar las actividades que compondrán el proyecto y sus posibles tiempos de duración. Sin embargo, al momento de analizar qué caminos del desarrollo son considerados críticos, pueden presentarse problemas si a medida que avanza el proyecto van surgiendo nuevos riesgos, generando con esto nuevas rutas críticas. El hecho de crear rutas críticas llama la atención de los desarrolladores para dedicar mayor preocupación a las tareas que las compongan, lo que podría llevar a distribuir de mala manera los recursos de desarrollo con los que se cuenta para resolverlas prontamente, descuidando el resto de las tareas [17]. De esta forma se puede decir que el buscar tareas críticas puede resultar ser una fuente de errores para las estimaciones que se intenten hacer.

Una forma de estimación que es posible utilizar en los ambientes en los que se utilice la metodología de desarrollo extreme programming corresponde al llamado “Juego de planificación” [18].

Para aplicar este método debe fijarse primero una reunión entre desarrolladores y clientes en la que se define, mediante tarjetas con casos de uso o “user story cards”, que requisitos se debe completar para el final de la iteración de trabajo actual, la funcionalidad que se debe agregar y las fechas en las que estos deben ser entregados. Para definir estas fechas los desarrolladores analizan los casos de uso presentados por los clientes en las tarjetas y estiman el tiempo que consideran necesario para completarlos. Luego los objetivos presentados en las tarjetas son ordenados por prioridad y se establecen posibles cambios en los casos de uso o funcionalidades solicitadas.

Con los elementos a implementar elegidos, los desarrolladores sostienen una reunión en la que analizan que tareas se desprenden de las tarjetas con los casos de uso entregadas por los clientes. Posteriormente es necesario estimar el tiempo que tomara realizar cada una de estas tareas. Finalmente, los desarrolladores eligen algunas de estas tareas de forma de distribuir equitativamente la carga entre todo el equipo.

Para realizar este tipo de estimaciones se establecen un conjunto de principios entre los que se destacan el tener que estimar como un solo equipo tanto clientes como desarrolladores, además de realizar estimaciones de manera frecuente y para entregas que incluyan un número no muy elevado de funcionalidades nuevas [19]. Sin embargo, en ambientes donde no se utilice la metodología extreme programming, el estimar con los clientes puede resultar complejo, ya que por lo general ellos no se consideran a sí mismos como parte del grupo de trabajo.

En el paper “A review of studies on expert estimation of software development effort” [2], se reunieron 15 investigaciones realizadas entre 1990 y 2002 referentes al desempeño que presentaba la metodología de juicio experto contra estimaciones realizadas basadas en otros modelos de estimación. De todos los estudios reunidos, se concluye que ninguno de los

modelos anteriormente presentados es completamente efectivo a la hora de realizar estimaciones. Sin embargo muchos de los modelos de estimación utilizan en cierta medida el juicio experto al asignar algunos de los valores a los diferentes factores que componen la fórmula del modelo, sobre todo cuando se refiere a líneas de código a escribir en un software para el cual aún se están reuniendo requisitos. Por esto, y por el valor pedagógico que posee, permitiendo acercar a los miembros de un grupo de trabajo mediante reuniones frecuentes de estimación, se ha elegido la metodología Wideband Delphi como base para este proyecto.

El objetivo que existe tras el hecho que se motive en los alumnos la práctica constante de realizar estimaciones está basado en un modelo constructivista de aprendizaje [20]. Este modelo se beneficia de los siguientes factores [21]:

- Los estudiantes pueden trabajar para clarificar y para ordenar sus ideas y también pueden contar sus conclusiones a otros estudiantes.
- Esto les da la oportunidad de elaborar lo que aprendieron.
- Lo que aprenden puede someterse a la crítica de los otros alumnos.
- Esta crítica les permite descubrir defectos e inconsistencias.

Opuesto al conductivismo, donde el alumno aprende a base de la teoría enseñada por un profesor, y luego la aplica obteniendo una respuesta del medio que deberá razonar, el constructivismo se basa en el aprendizaje logrado mediante la práctica y la respuesta inmediata por parte del medio frente a la acción realizada por el alumno. Es un aprendizaje donde es el estudiante quien modela la realidad en base a las experiencias a las que él se somete, reconstruyendo con esto modelos mentales que le han sido enseñados por otras personas y su medio. Tomando esto en cuenta, el software pretende guiar al alumno siguiendo un modelo constructivista histórico-social, donde el profesor le transmite un conocimiento al alumno basado en un medio ambiente que ya está estructurado, pero es el alumno quien decide cómo interactuar con él.

Otro de los factores a los que se desea dar importancia en esta aplicación es al trabajo en equipo, sugiriendo roles en base a las aptitudes que cada miembro del grupo cree tener, para así complementar las cualidades que los alumnos posean, formando un equipo de desarrollo consciente de sus habilidades y limitantes. Un grupo bien estructurado sabrá las fortalezas de cada miembro, resolviendo con esto problemas que un grupo poco cohesionado no podría resolver [22].

7- Contexto académico

El entorno de trabajo hacia el cuál está dirigida la aplicación corresponde al desarrollo de software en equipos en los cursos que así lo requieran. Dentro de estos cursos se le enseñan al alumno estándares y metodologías de desarrollo con las cuales puede enfrentar la implementación de la aplicación asignada. Una de las ideas tras la creación de esta aplicación es ayudar a los alumnos a planificar mejor sus recursos, en especial su tiempo, basados en la experiencia previa que pueda tener y en los conocimientos de estas herramientas que se le entregan en la facultad. De estas herramientas, las más utilizadas en el desarrollo grupal de software dentro de la facultad se describen a continuación.

7.1 Estándar E.S.A.

El estándar de la E.S.A. (European Space Agency) describe las características que debe poseer un software preparado para esta organización, tanto por su personal interno como por un equipo de desarrollo externo [23]. Corresponde a un conjunto de fases y elementos entregables que sirven como guía en el desarrollo de una aplicación.

7.1.1 Fases y entregables

El estándar de la E.S.A. define las siguientes fases en el desarrollo del software.

7.1.1.1 Definición de requisitos de usuario (UR)

Corresponde a la fase en la que se debe identificar el problema desde el punto de vista del usuario y/o cliente. Esto se logra por medio de entrevistas donde se deben identificar y aislar cada uno de los requisitos que solicite deba cumplir la aplicación. El entregable de esta fase corresponde al “documento de requisitos de usuario”, el cuál debe ser revisado (fase UR/R) antes de poder pasar a la siguiente etapa.

7.1.1.2 Definición de requisitos de software (SR)

En esta fase se analiza cómo el software a desarrollar puede cumplir con los requisitos solicitados por el usuario. Para verificar que cada requisito de software satisfaga los requisitos de usuario se realizan muestras de prototipos de la aplicación. El entregable de esta fase corresponde al “documento de requisitos de software”, que contiene todos los requisitos de software encontrados y debe ser revisado (fase SR/R) antes de pasar a la siguiente etapa.

7.1.1.3 Definición del diseño arquitectónico (AD)

Se debe crear un modelo de la implementación de la aplicación, indicando sus componentes principales, las funciones que poseen y la forma en que se relacionan entre ellas. Su entregable corresponde al “documento de diseño arquitectónico”, que debe ser revisado (fase AD/R).

7.1.1.4 Diseño detallado y producción del código (DD)

Basándose en los documentos obtenidos de las fases anteriores se debe detallar el diseño de

la aplicación, y comenzar con las labores de codificación, documentación y posterior prueba ante fallas. Los entregables de esta fase son el “documento de diseño detallado”, el código fuente de la aplicación y el manual del usuario, los cuales deberán ser revisados (DD/R).

7.1.1.5 Transferencia del software a operaciones (TR)

En esta fase la aplicación debe ser instalada en el lugar donde será utilizada por los clientes/usuarios, con el fin de verificar que se cumplan efectivamente los requisitos expuestos en el “documento de requisitos de usuario”. El entregable en esta etapa corresponde al “documento de transferencia de software”, que debe indicar todas las actividades, detalles y alcances del proceso de transporte e instalación de la aplicación en los equipos de operación.

7.1.1.6 Operación y mantenimiento (OM)

Esta fase involucra utilizar la aplicación creada con el objetivo de comprobar si supera las pruebas de aceptación definidas por el equipo de desarrollo y de usuarios. Al finalizar esta fase se debe preparar un “documento histórico del proyecto”, que contenga toda la información acumulada a lo largo del proyecto.

El ciclo de vida definido puede verse a continuación.

PHASES ITEMS	UR User Requeriments Definition	UR/R	SR Software Requeriments Definition	SR/R	AD Achtitectural Design	AD/R	DD Detailed Design and Production	DD/R	TR Transfer	OM Operations and Manteinance		
MAJOR ACTIVITIES	<ul style="list-style-type: none"> determination of operational environment identification of users requirements 		<ul style="list-style-type: none"> construcción of logical model identification of software requirements 		<ul style="list-style-type: none"> construcción of phisical model definition of major components 		<ul style="list-style-type: none"> module design coding unit test integration test system test 		<ul style="list-style-type: none"> installation provisional acceptance tests 	<ul style="list-style-type: none"> final acceptance test operations maintenance of code and documentation 		
DELIVERABLE ITEMS	User Requirements Document pág. 72	URD →	Software Requirements Document pág. 73	SRD →	Architectural Design Document pág. 74	ADD →	Detailed Design Document págs. 75/76 Code SUM Software User Manual	DDD → Code → SUM →	Software Transfer Document pág. 76	STD →	Project History Document pág. 77	PHD
REVIEWS (See Checklist Apendix D)		tech. review	tech. review	tech. review	tech. review	tech. review	tech. review	tech. review				
MAJOR MILESTONES		URD approved	SRD approved		ADD approved		Code/DDD/SUM approved		STD delivered		PHD delivered	
									Provisional Acceptance		Final Acceptance	

Ilustración 2: ciclo de vida de software según estándar de la E.S.A.

7.1.2 Roles

Junto con la enseñanza de este estándar, se asignan a los miembros de cada grupo diferentes roles con el fin de distribuir de forma equitativa el trabajo del desarrollo. Estos roles no forman

parte del estándar, pero son aplicables a cada una de las fases del ciclo de vida. Se describen a continuación.

7.1.2.1 Administrador o jefe de proyecto

Corresponde al principal encargado del proyecto. Su misión es el lograr coordinar el trabajo de todos los miembros del grupo, así como apoyarlos en cada una de las fases y ver que se estén cumpliendo los plazos y objetivos propuestos.

7.1.2.2 Analista

Encargado de realizar las entrevistas con los clientes o usuarios, donde se deberán obtener los requisitos que se espera satisfaga el software. El “documento de requisitos de usuario” y el “documento de requisitos de software” son creados por él, junto con las revisiones que el tester aportará.

7.1.2.3 Diseñador

Su misión es realizar el diseño de la aplicación, basándose en el “documento de requisitos de software”, así como generar y validar prototipos de la aplicación junto con los implementadores. Debe generar el “documento de diseño detallado” donde se explique la arquitectura de la aplicación, sus componentes y las relaciones entre ellas.

7.1.2.4 Implementador

Su principal labor corresponde a la codificación de la aplicación. También debe establecer un plan que refleje la forma en la que se llevará a cabo la implementación y generar prototipos para validar el cumplimiento de los requisitos.

7.1.2.5 Tester

Es el encargado de validar las entregas tanto en documentación como en código, ver que se cumplan los requisitos especificados y entregar documentos con los problemas o errores detectados.

7.1.2.6 Cliente

Corresponde a la contraparte del equipo de trabajo. Es la persona que detecto una necesidad no satisfecha en su entorno de trabajo y solicitó la creación de una aplicación que la resuelva. Parte de los requisitos que debe cumplir el sistema se deben extraer del cliente, así como la validación final.

7.1.2.7 Usuario

Son usuarios las personas que utilizarán la aplicación solicitada. En algunos casos el cliente corresponde también a uno de los usuarios del sistema. El grupo de desarrollo, en especial el analista, querrá obtener los requisitos de los usuarios juntos con los del cliente, para verificar que no se produzcan contradicciones entre estos.

7.1.3 Valor pedagógico del estándar

Si bien el estándar está enfocado a proyectos de gran tamaño, es posible efectuar pequeñas modificaciones con el fin de adaptarlo a proyectos de menor tamaño [24]. De esta forma se logró crear un modelo que permite realizar y monitorear proyectos en la duración de un semestre académico, donde algunas de las tareas que realizan los miembros del equipo se pueden realizar en forma paralela, realizando además entregas incrementales de cada una de las fases del estándar.

7.2 Metodología “extreme programming”

Extreme programming corresponde a una metodología ágil de desarrollo de software, la cual define un conjunto de valores y reglas que le permitan ser lo más adaptable posible frente a cambios en los requerimientos o condiciones de trabajo, con el objetivo de minimizar los problemas que puedan traer estos cambios y maximizar el valor generado en el desarrollo de software [25,26].

7.2.1 Ideas principales

7.2.1.1 Valores

Esta metodología propone cuatro valores principales que deben mantenerse en un ambiente de desarrollo que utilice extreme programming.

Comunicación: hace referencia a la comunicación que debe existir tanto entre los desarrolladores como entre éstos y los usuarios, de forma que se mantenga siempre una visión compartida del sistema a desarrollar.

Simplicidad: se debe mantener siempre un diseño sencillo, refactorizando lo que ya se tiene con el objetivo de acercarse incrementalmente al sistema completo. Al mantener un diseño simple se facilita la comunicación entre los miembros del equipo de desarrollo acerca de los componentes del sistema.

Feedback: debe existir feedback desde el sistema mismo, el cual es posible obtener mediante pruebas a sus diferentes componentes, que indiquen acerca del estado del sistema una vez se hayan efectuado cambios en él. También debe haber feedback desde los clientes, el cual se puede obtener mediante pruebas de aceptación, que permiten comprobar la satisfacción de los clientes con el software.

Valor: se refiere a no tener miedo de programar pensando en las necesidades de hoy, sin por eso negarse a preparar el código de forma de que sea adaptable a futuros componentes que sean agregados. Además implica no tener miedo a desechar parte del código que haya quedado obsoleto o impida que ciertas pruebas no se completen.

En una versión posterior de la presentación de esta metodología se agrego el valor respeto, que corresponde a aceptar que se trabaja en un grupo, y por ende se debe velar principalmente por el funcionamiento del sistema, respetando el trabajo de los demás sin poner prioridad sobre el código de autoría propia.

7.2.1.2 Etapas

Junto con estos valores, se reconocen cuatro etapas principales en el proceso de desarrollo de software, donde cada etapa posee su conjunto propio de reglas.

Planificación: se refiere a la forma en la que el proyecto será desarrollado a lo largo del tiempo. Las principales reglas de la planificación corresponden a dividir el proyecto en iteraciones, realizar entregas pequeñas y frecuentes del código desarrollado, y evitar que una persona pase mucho tiempo encargado de una tarea e ignorando las otras, debiendo realizar rotaciones entre los miembros del equipo.

Diseño: corresponde a la definición de las componentes del sistema. Sus reglas incluyen la simplicidad, no agregar demasiada funcionalidad inmediatamente al software y refactorizar el código de forma de eliminar componentes obsoletos o mejorar el desempeño de otros.

Escribir código: se refiere a la escritura de la aplicación. Algunas de sus reglas corresponden a escribir primero las pruebas de validación antes que los mismos componentes, a programar siempre en equipo e integrar frecuentemente el código escrito por estos equipos.

Pruebas: se refiere a validar el correcto funcionamiento del sistema. Para esto se establece que toda parte del código debe tener una prueba asociada, y que todo el código debe pasar las pruebas a cada una de sus unidades antes de poder ser liberado.

7.2.2 Valor pedagógico de la metodología

En los ramos en los que los proyectos a desarrollar son de una longitud considerable y no se desea establecer diferencias entre los miembros del grupo mediante la asignación de roles, se suele utilizar la metodología “extreme programming” [27].

Con esta metodología se fomenta la idea de que todos los miembros del equipo de trabajo deben tener conocimiento del problema completo, y no enfocarse a resolver partes aisladas del mismo. Esto con el fin de evitar las pérdidas de tiempo generadas cuando la única persona del equipo que sabe resolver un problema en particular no se presenta a trabajar. Para esto se recomienda trabajar de a pares al codificar. De esta forma, nunca el conocimiento de una parte del software recaerá completamente en una sola persona.

Dentro de los ramos donde se utiliza esta metodología se realizan entregas del software solicitado por medio de pequeños incrementos en el desarrollo de la aplicación, donde cada entrega del mismo constituye un producto por si solo. Así el cliente tiene mayor control sobre el producto desarrollado al ver los cambios que han transcurrido entre una entrega y otra; y el equipo programador tiene mayor libertad de acción en caso de que cambie un requisito a mitad de camino. Con esto, se otorga a los alumnos la experiencia para saber como actuar frente al cambio, y también el como administrarlo.

Con esta metodología se busca también fomentar una interacción más cercana con el cliente o usuario. En lugar de realizar una entrevista de toma de requisitos al comienzo del desarrollo y una reunión de verificación de la aplicación al final, se realizan reuniones de manera frecuente intentando integrar al cliente al equipo de desarrollo, presentándole prototipos y avances.

7.3 Estructura de descomposición del trabajo (EDT)

7.3.1 Ideas principales

La principal idea tras la estructura de descomposición del trabajo (Work breakdown structure, WBS) corresponde a la división de una tarea de gran tamaño en tareas hijas de menor tamaño, utilizando una estructura de árbol jerárquico. Con el objetivo de realizar una buena división, el total del trabajo realizado por las tareas hijas debe equivaler al trabajo que realizaría la tarea padre [28,6].

7.3.2 Valor pedagógico de la EDT

Esta estructura es utilizada en el estándar de la E.S.A. para dividir el proyecto completo asignado en las diferentes fases que componen su representación del ciclo de vida del software, donde cada fase contiene a su vez diferentes tareas que corresponden a la obtención de requisitos, creación y validación de documentación, entre otros. Además puede ser utilizada en la metodología extreme programming para definir un plan inicial de trabajo. Éste no es uno de sus objetivos principales debido a que al aplicar extreme programming se está suponiendo que habrán cambios a la división del proyecto planteada debido a la incertidumbre inicial de requerimientos y posibles cambios en lo que el cliente necesite.

8- Diseño de la aplicación

8.1 Integración con los conocimientos del alumno

Para poder diseñar la aplicación de forma que su terminología y modo de uso sean intuitivos para el alumno, se deben tener en cuenta los conocimientos que él posee, según fueron expuestos en los capítulos 7.1: “Estándar E.S.A.”, 7.2: “Metodología extreme programming” y 7.3: “Estructura de descomposición del trabajo”. A continuación se explica la forma en que estos serán utilizados dentro de la aplicación.

8.1.1 Integración con estándar de la E.S.A.

Se diseñará la herramienta de forma que utilice el estándar en la asignación de responsables a las actividades que se desprendan de un proyecto, aunque sin asignar roles explícitamente, donde estas actividades pueden corresponder a una entrega en código o en documentación. Esto con el fin de que cualquier miembro del grupo se muestre como disponible para realizar una actividad, sin importar si posee un rol específico dentro del grupo. Además la aplicación debe permitir especificar restricciones en el orden en que se deben completar las actividades que se propongan, como es por ejemplo el poseer una versión revisada del “documento de requisitos de software” antes de poder comenzar con la creación del “documento de diseño detallado”, y ofrecer alguna forma de visualizar esto en cartas Gantt.

8.1.2 Integración con metodología extreme programming

Si bien será posible asignar responsables a cada actividad de un proyecto, la herramienta no asignará roles a cada uno de los participantes, por lo que todos los miembros del grupo tendrán la misma responsabilidad en cuanto a asignar y modificar los tiempos de desarrollo estimados de su parte de la aplicación, con el fin de informar al resto del grupo respecto a sus avances. Además, considerando la variación en los requisitos que se asume al utilizar esta metodología, y para lograr una mejor adaptación al cambio utilizando como referencia experiencias anteriores, se permitirá que la aplicación pueda llevar un registro de las múltiples modificaciones a los requisitos que expone el cliente al grupo desarrollador. Así, el feedback entregado por el cliente a cada tarea no se perderá.

8.1.3 Integración con el EDT

Para poder realizar estimaciones más ajustadas a la realidad, se espera que el proyecto asignado sea dividido en cada una de las actividades que lo componen, donde el proyecto no estará completo hasta que lo esté cada una de sus partes. Dependerá del alumno el realizar una buena división de su proyecto para que no queden partes de éste sin considerar, pero se espera que sus técnicas de análisis y división mejoren con el tiempo y la práctica.

8.2 Consideraciones adicionales al contexto académico

Además de los conocimientos que poseen los alumnos, el estar en un ramo de la facultad en donde se debe trabajar de forma grupal involucra otros factores que deben ser tomados en cuenta al analizar el diseño de la aplicación.

Necesidad de herramientas de apoyo al desarrollo: al cursar los ramos en los que se debe desarrollar software de manera grupal, se hace necesario llevar un registro de la cantidad de trabajo realizado, así como cuánto se ha avanzado en el proyecto y los resultados de reuniones anteriores. Muchos alumnos se ven en la necesidad de buscar herramientas externas durante el desarrollo del curso para poder satisfacer estas necesidades. Se espera que esta herramienta ofrezca una alternativa para los alumnos, que les permita visualizar cuanto se ha avanzado en el proyecto, así como llevar un registro de las discusiones que estos tengan.

Horarios de alumnos: el horario de desarrollo del proyecto por parte de los alumnos tiende a ser muy variado, considerando la carga académica que poseen, por lo que las instancias de reunión en las que todos puedan estar presentes para dar su opinión son escasas. Se debe diseñar la herramienta de forma que se pueda utilizar aún cuando no todos los miembros del grupo puedan estar en el mismo lugar físico.

Interacción entre alumnos: dentro de un mismo grupo pueden darse alumnos con personalidades muy variadas, donde es posible que alguien pueda hacer las veces de líder e influencie a los demás con sus puntos de vista referentes a un problema. Es importante evitar esto, otorgando un ambiente donde las opiniones tengan un valor equivalente.

Interacción entre docentes y alumnos: a lo largo del desarrollo de la aplicación se les asigna a los alumnos un miembro del equipo docente cuya labor dependerá del ramo que se esté cursando. En ocasiones el personal docente cumplirá las funciones de cliente, donde deberá especificar los requisitos del software que desea obtener y validar si se está satisfecho con el producto presentado. En otras ocasiones ocupará el rol de miembro de apoyo al desarrollo, dando consejos al grupo basado en su experiencia laboral y verificando los avances obtenidos. Se espera que el equipo docente pueda revisar y opinar acerca del avance de los alumnos, y poder participar de algunas actividades si fuese necesario.

8.3 Definición de servicios de la aplicación

Se definen a continuación los servicios que ofrecerá la aplicación, tomando en cuenta las consideraciones efectuadas en el capítulo 8.1: “Integración con los conocimientos del alumno” y en el capítulo 8.2: “Consideraciones adicionales al contexto académico”.

Creación y mantención de proyectos: se entenderá por “proyecto” en el contexto de la aplicación al tema asignado por el equipo docente a cada grupo de alumnos. Se podrá crear una representación del proyecto en la aplicación, ingresando datos como su nombre, fecha de inicio, término y una descripción. También será posible modificar y eliminar la información ingresada respecto a ellos. Una vez que el proyecto esté creado, se podrá ver una representación gráfica de la duración de éste. Un proyecto debe ser analizado y dividido en actividades más pequeñas, de forma que el conjunto de todas ellas formen el proyecto completo.

Creación y mantención de tareas: se entenderá por “tarea” en el contexto de la aplicación a cada una de las actividades que compongan un proyecto. Se espera que los alumnos dividan el proyecto asignado en las tareas que lo componen, siguiendo la estructura de descomposición del trabajo. A cada tarea que se desee crear se le debe asignar un nombre, una fecha de inicio, término, responsables y restricciones con otras tareas del proyecto si es que las hay. Esta información puede ser modificada o eliminada con posterioridad. Por cada tarea se tendrá una

sala de Chat, donde se podrán reunir los miembros para tomar decisiones respecto al plan de desarrollo a seguir.

Estimación de duración de tareas: se ofrecerá una parte de la aplicación en la que se pueda discutir acerca de las fechas de inicio y de término de las tareas que han sido creadas por los alumnos, conversar de los factores que influyen en la elección de estas fechas y observar un historial de las estimaciones realizadas anteriormente, con el objetivo de verificar su aprendizaje.

Almacenamiento de información generada: la aplicación mantendrá un registro de las conversaciones generadas en las salas de discusión de las tareas, así como en los foros y salas de estimación, de forma que los estudiantes puedan basarse en las experiencias anteriores obtenidas en el desarrollo del software al tener que buscar respuestas a nuevos problemas.

Herramientas de apoyo al estudiante: se ofrecerá un cuestionario al alumno donde podrá obtener información respecto a sus fortalezas como miembro de un grupo, un formulario donde podrá calcular la complejidad de alguna actividad mediante la metodología COCOMO, y un glosario con los términos más utilizados en los ramos donde se utiliza el desarrollo grupal de aplicaciones.

8.4 Recursos

Para satisfacer estas necesidades y problemáticas se consideraron las siguientes alternativas, las cuales fueron analizadas según sus fortalezas y debilidades.

8.4.1 Recursos de comunicación

Foro

Fortalezas: permite que los alumnos se comuniquen de forma asíncrona, sin necesidad de que el grupo se encuentre reunido en el mismo lugar o en el mismo horario. Cada alumno puede opinar con un seudónimo si así lo desea, sin sentir la presión de otros miembros del grupo. El equipo docente puede controlar las discusiones de los alumnos, mediante roles especiales asignados en esta herramienta, así como participar en ellas. Dada su naturaleza, permite llevar un registro de las discusiones realizadas. Durante el transcurso de su carrera, el alumno se ve enfrentado a herramientas de discusión de este tipo, por lo que sabrá como utilizarlo.

Debilidades: el alumno debe revisar constantemente el foro para saber si existe algún mensaje nuevo. Si el alumno no se encuentra motivado por el proyecto de todas formas puede no escribir comentarios en las discusiones.

Listas de correo

Fortalezas: permite que los alumnos se comuniquen de forma asíncrona, sin necesidad de que el grupo se encuentre reunido en el mismo lugar o en el mismo horario. Los miembros del equipo docente pueden estar registrados en la lista de correo y recibir los mensajes que los alumnos envíen, así como enviar sus propios mensajes.

Debilidades: no todos los alumnos conocen este medio de comunicación, por lo que se

debería presentar el concepto y su forma de uso dentro de la herramienta. Si el alumno no se encuentra motivado por el proyecto de todas formas puede no escribir comentarios en las discusiones. Por ser una lista de correo, el correo electrónico utilizado por el alumno para enviar sus comentarios lo identificará a los demás usuarios.

Grupos de noticias

Fortalezas: permite que los alumnos se comuniquen de forma asíncrona, sin necesidad de que el grupo se encuentre reunido en el mismo lugar o en el mismo horario. Los miembros del equipo docente pueden estar y ver los mensajes que los alumnos envíen, así como enviar sus propios mensajes.

Debilidades: no todos los alumnos conocen este medio de comunicación, por lo que se debería presentar el concepto y su forma de uso dentro de la herramienta. Para utilizar todas sus funcionalidades, se debe tener una aplicación externa de lectura de grupos de noticias.

Chat

Fortalezas: permite que los alumnos se comuniquen de forma asíncrona, sin necesidad de que el grupo se encuentre reunido en el mismo lugar. Los miembros del equipo docente pueden ingresar y recibir los mensajes que los alumnos envíen, así como enviar sus propios mensajes. Es un medio de comunicación que suele ser conocido por los alumnos. Según sea su licencia, pueden modificarse sus funcionalidades originales para ofrecer un registro de los temas tratados anteriormente. Algunos sistemas de Chat permiten a sus usuarios utilizar seudónimos para realizar conversaciones anónimas.

Debilidades: si un miembro del grupo no puede estar presente en el momento que se lleve a cabo la discusión, no podrá participar en ella.

8.4.2 Recursos de implementación de código

php

Fortalezas: ofrece funciones que se integran fácilmente con muchos de los recursos presentados. Posee abundante documentación en su sitio Web y en sitios relacionados con el lenguaje. La mayoría de los tutoriales de programación Web disponibles se basan en este lenguaje. Posee diversas herramientas ya preparadas para ser incluidas en aplicaciones Web.

Debilidades: se pueden utilizar variables que no han sido declaradas previamente, y realizar operaciones sobre ellas. No se define el tipo de las variables explícitamente. Poca consistencia en el nombre de sus funciones.

ruby on rails

Fortalezas: se basa en ofrecer convención sobre configuración, facilitando la tarea del programador al generar automáticamente algunos de los archivos necesarios para el funcionamiento de la aplicación basado en el modelo vista controlador (MVC). Facilita la creación de consultas a la base de datos

Debilidades: el código generado queda repartido en diferentes carpetas, por lo que la modificación de un archivo puede llevar a dejar sin actualizar todos los archivos que se generaron automáticamente de él y que no han sido corregidos. Mucho del código

generado automáticamente puede ser válido para una versión específica de ruby on rails, y quedar obsoleto para versiones posteriores.

8.4.3 Recursos de almacenamiento

Base de datos basadas en SQL

Fortalezas: permite obtener información mediante consultas realizadas en un lenguaje estructurado. Es posible almacenar los datos en forma organizada a través de la creación de tablas y relaciones entre ellas. Su sistema de consulta está optimizado para obtener información de forma eficiente.

Debilidades: para sacar el mayor provecho se requiere utilizar un sistema administrador de bases de datos. Entre diferentes sistemas administradores de bases de datos existen diferencias en la sintaxis SQL utilizada.

Bases de datos orientadas a objetos

Fortalezas: permiten modelar las entidades participantes a través de objetos. Favorece la creación de aplicaciones multimedia.

Debilidades: ninguno de los recursos de implementación de código se integra con esta alternativa. Algunas de las herramientas de comunicación presentadas utilizan bases de datos basadas en SQL para sus funciones. No es necesario orientar la aplicación y sus participantes a objetos.

Archivos de texto plano

Fortalezas: no se necesitan aplicaciones adicionales para trabajar con ellos. Debido a esto, el espacio total utilizado al almacenar datos puede ser menor.

Debilidades: no existe una forma estandarizada de obtener información de un archivo de texto. Se debe leer gran parte del archivo al buscar una información en particular. No permite una representación adecuada de las diferentes entidades que intervienen en la aplicación.

8.4.4 Recursos de representación de progreso

Gráficos de barras

Fortalezas: utilizando el eje y para representar el porcentaje de avance y el eje x para las diferentes partes de un proyecto, se puede utilizar para ver el avance total.

Debilidades: salvo mostrar el avance en cada tarea, no es posible obtener más información de ellas.

Modelo PERT

Fortalezas: junto con obtener información acerca del orden en que se deben ejecutar las tareas, otorga datos importantes de ellas como son fecha de inicio y fecha de término, y posibles problemas entre tareas.

Debilidades: según sea la cantidad de datos que muestre, su lectura puede no resultar inmediata. Dado el formato de árbol que utiliza para expandir su contenido, un proyecto que posea varias actividades utilizará gran espacio.

Carta Gantt

Fortalezas: se obtiene información acerca de las fechas de inicio, término y relación con otras actividades a simple vista. Permite representar el porcentaje avanzado en cada actividad dentro de sus barras de duración.

Debilidades: en su diseño original no contempla la representación de posibles actividades críticas que puedan interferir con el desarrollo de otras.

8.5 Elección de recursos

Para elegir los recursos que finalmente se utilizarán en la aplicación, junto con los factores descritos en los capítulos 8.1 y 8.2, se consideraron los servicios que se desea entregar, descritos en el capítulo 8.3. A continuación se listan los recursos elegidos, junto al uso que se les dará en la aplicación Web.

8.5.1 Recursos de comunicación

Los recursos elegidos corresponden al foro y al Chat, debido a que se necesita una herramienta tanto para discutir asincrónicamente en la pizarra de mensajes, como sincrónicamente en la herramienta de discusión. Ambas alternativas son las más conocidas por los alumnos y no se necesitan herramientas externas, salvo un navegador Web, para acceder a ellas.

8.5.2 Recursos de implementación de código

El lenguaje elegido corresponde a php, debido a la gran cantidad de herramientas y soporte existente para él, además de darle mayor control al programador sobre la estructura que tendrá la aplicación y los archivos que la compongan. Este lenguaje será la base para implementar todos los servicios que se ofrecerán.

8.5.3 Recursos de almacenamiento de datos

Se utilizará una base de datos basada en SQL debido a que muchos de los recursos elegidos utilizan este tipo de bases de datos. Existe además una gran variedad de sitios que poseen soporte y tutoriales referente a su uso. Se almacenará en la base de datos todas las discusiones realizadas por los alumnos en la pizarra y en la herramienta de discusión, así como los datos de sus proyectos y tareas.

8.5.4 Recursos de representación de progreso

Se utilizarán las cartas Gantt para mostrar las tareas y el progreso de estas en el proyecto debido a la rapidez y facilidad con la que es posible extraer información de ellas.

8.6 Arquitectura

Para elegir la arquitectura que se utilizaría al implementar el sistema, se debe considerar que ésta debe permitir que los usuarios interactúen en tiempo real. Se tomaron en cuenta las siguientes alternativas.

8.6.1 Arquitectura P2P

Fortalezas: la ventaja que presenta el crear el sistema considerando esta arquitectura es que no se depende de un servidor central para que la aplicación funcione, sino que se puede utilizar los recursos que los mismos usuarios del sistema poseen. De esta forma, si uno de los clientes cae, la aplicación puede seguir funcionando al no tener un punto único de falla.

Debilidades: es necesario instalar la aplicación en todos los equipos utilizados por los usuarios, de forma que puedan funcionar como nodos de la red. Debido a esto, la aplicación debería diseñarse de forma que sea instalable en diferentes sistemas operativos.

8.6.2 Arquitectura cliente servidor

Fortalezas: toda la información de la aplicación está almacenada en un servidor central, permitiendo así tener un mejor control sobre la seguridad. Sólo las personas con acceso al servidor podrán realizar cambios a la aplicación. Permite actualizar los datos del sistema de forma más eficiente que en otras arquitecturas.

Debilidades: al encontrarse toda la aplicación en el servidor, se genera un punto único de falla. Si el servidor deja de funcionar, las peticiones de los clientes no podrán ser satisfechas. Si se utiliza un solo servidor, éste puede saturarse al ser accedido por muchos clientes al mismo tiempo.

8.6.3 Arquitectura orientada al servicio

Fortalezas: permite reutilizar componentes ya creados en proyectos futuros debido a que sus interfaces se definen siguiendo estándares. Los componentes se diseñan considerando que sólo su entrada y salida serán interesantes para quien la solicito, sin importar el lenguaje en que fue escrita o su funcionamiento interno.

Debilidades: Esta arquitectura define que en el diseño de los servicios no se debe mantener ni depender de ningún estado. No se encontraron implementaciones de los recursos a utilizar en la aplicación que hayan sido diseñados con esta arquitectura en mente.

8.6.4 Arquitectura elegida

La arquitectura elegida para la aplicación corresponde al modelo cliente/servidor de tres capas, montando las capas de lógica y datos en un solo servidor.

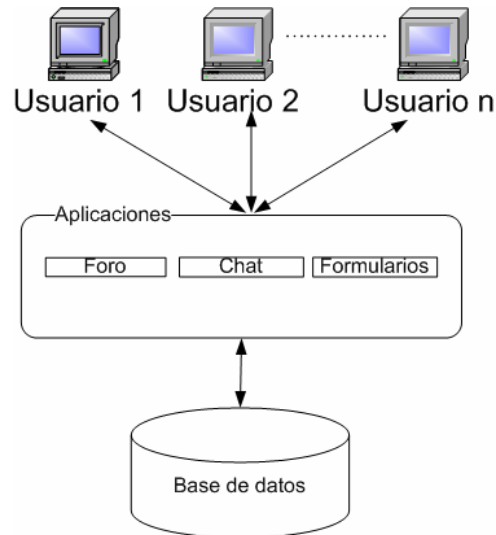


Ilustración 3: arquitectura de la aplicación

La elección de esta arquitectura fue basada en la necesidad de actualizar rápidamente la información enviada por los usuarios, debiendo ingresarla una vez en la base de datos y de ahí propagarla según sea solicitada por los usuarios, a cambio de otros enfoques donde la información se transmite de cliente en cliente de forma más lenta, debiendo realizar así chequeos de quién posee la última versión de la información.

8.7 Vista de capas

En la siguiente ilustración se aprecia la relación entre la capa lógica y la capa de datos. El acceso entre las distintas capas es desde el nivel superior al inferior.

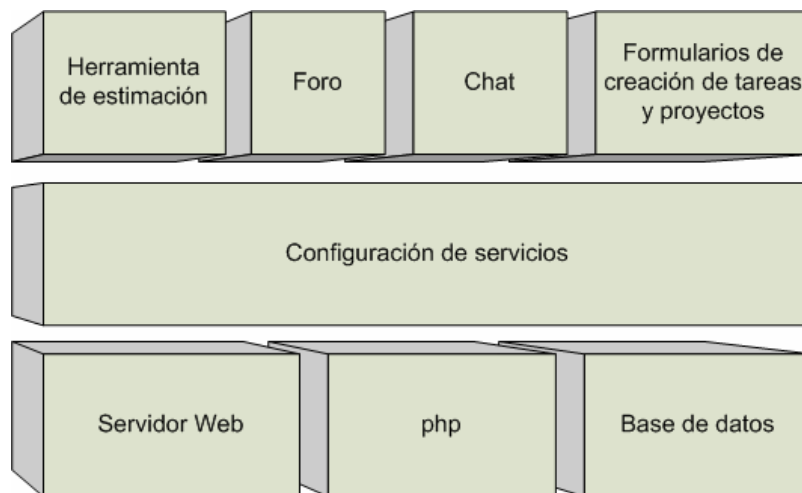


Ilustración 4: capas que componen la aplicación

Herramienta de estimación: son los archivos que contienen la información necesaria para presentar la sección de estimación a los usuarios, y utilizan las funciones definidas en las capas de configuración de servicios.

Foro: corresponde a los archivos que se encargan de presentar el contenido del foro al cliente, y utilizan las funciones definidas en las capas de configuración de servicios.

Chat: corresponde a los archivos que se encargan de presentar el contenido del Chat al cliente, y utilizan las funciones definidas en las capas de configuración de servicios.

Formularios de creación de tareas y proyectos: son los archivos que presentan los formularios al usuario donde podrá crear nuevas tareas o proyectos, asignarles responsables y fechas límite. Esto incluye también los formularios guía.

Configuración de servicios: corresponden a los archivos de configuración de los servicios definidos en la capa superior. En esta capa se definen las funciones de acceso a la base de datos y los parámetros que indican a la aplicación el entorno en el que se está ejecutando.

Servidor Web: es el servidor HTTP que se utilizará para poder mostrar las páginas que conforman la aplicación.

php: es el lenguaje de programación de páginas Web que será utilizado para programar los distintos archivos que compondrán la aplicación, y será utilizado por el servidor para interpretar las páginas que deban ser mostradas a los clientes.

Base de datos: corresponde a la aplicación que se encargará de almacenar la información ingresada por los usuarios referentes a sus proyectos y tareas, discusiones y también algunas de las variables que utilicen las herramientas de capas superiores.

9- Modelo de estimación

Para poder llevar a cabo una buena sesión de estimación utilizando el método Wideband Delphi, se debe establecer un modelo de comunicación entre los participantes del grupo de forma de crear una discusión estructurada, donde todos tengan la misma igualdad en sus opiniones y que estas estén respaldadas con argumentos. Para poder analizar los tipos de diálogo que se podrían presentar entre los participantes, se estudiaron los siguientes modelos que pueden en cierta medida representar o ser utilizados en una sesión de estimación.

9.1 Modelo argumental “IBIS”

Este modelo surgió como una respuesta a la resolución de los problemas apodados “wicked”, debido a su dificultad para ser resueltos siguiendo métodos tradicionales [29,30]. Estos problemas poseen las siguientes características.

- No son fáciles de definir de forma que todos los participantes en la resolución concuerden en el problema a resolver.
- Requieren juicios complejos acerca del nivel de abstracción en el cuál se definirá el problema.
- No tienen reglas de detención claras.
- Tienen soluciones mejores o peores, no buenas o malas.
- No tienen una medición objetiva de éxito.
- Requieren iteraciones, cada intento para crear una solución cambia el problema.
- A menudo están en una dimensión moral, política o profesional.

El término de la resolución de un problema de este tipo es generalmente forzada a través de acuerdos alcanzados entre los miembros de la resolución del problema. Según H. J. Rittel, uno de los creadores de esta definición, se debe establecer una forma de resolver este tipo de problemas basada en la argumentación, donde los argumentos entregados son enfrentados sistemáticamente a los diferentes puntos de vista de los miembros del equipo. De esta forma se creó la base para los sistemas de información basados en problemas, o de su sigla en inglés IBIS. Los elementos principales de este modelo corresponden a los problemas o temas a resolver, las posiciones que toman los miembros del equipo frente a estos problemas, y los argumentos que entregan los miembros frente a las posiciones que toman. La relación entre estas componentes puede verse en la siguiente figura.

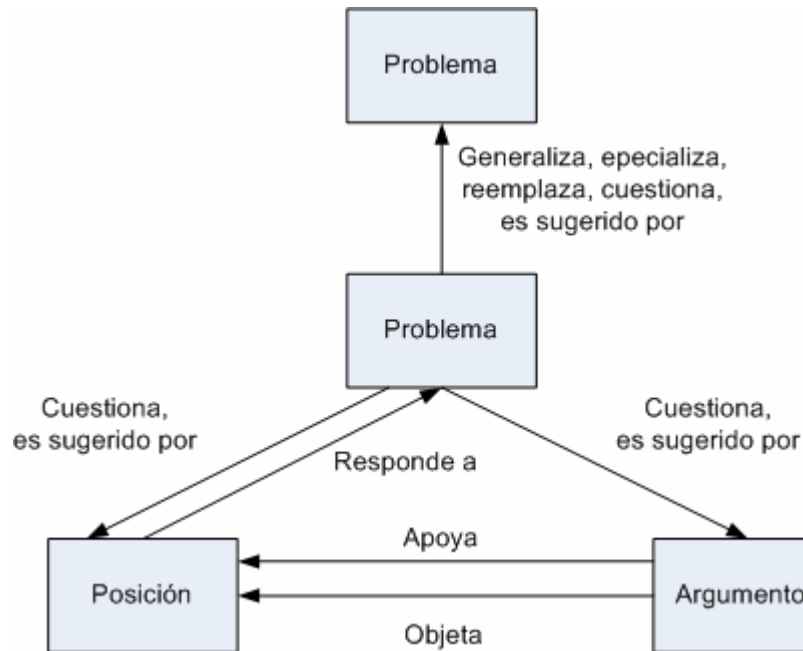


Ilustración 5: estructura básica del modelo IBIS

Este modelo refleja muy bien las características del problema de estimar la duración de un proyecto de software, y refleja las actividades y los tipos de relaciones que deben establecerse entre los participantes para poder llegar a un acuerdo que de solución al tema a resolver.

9.2 Modelo “Dialogue Mapping”

Dialogue Mapping corresponde a un modelo de discusión cuya gramática está basada en la presentada por el modelo IBIS. Su nombre proviene de la actividad de representar mediante un mapa las distintas opiniones entregadas por los miembros de un equipo que intentan resolver un problema [31,32]. En este modelo se distinguen tres componentes principales.

- El grupo de trabajo, que intenta encontrar una solución a un problema particular mediante el diálogo.
- Un proyector que refleja el estado de la discusión mediante un mapa donde se ilustran las ideas expuestas por los participantes, así como los puntos a favor y en contra de cada idea, preguntas adicionales y archivos anexos utilizados en la reunión.
- Un facilitador, quien es el encargado de mantener el mapa del diálogo a lo largo de la reunión, basado en las ideas expuestas por los participantes.

El diagrama creado permite a los participantes ver como sus comentarios se relacionan con los de otros miembros, ver las ideas expuestas para evitar repetir temas ya tratados, junto con entregar a todos la misma visión del problema y del punto en el que se encuentran en la resolución del mismo. El ejemplo de un mapa de diálogo creado con este método puede verse en la figura siguiente.

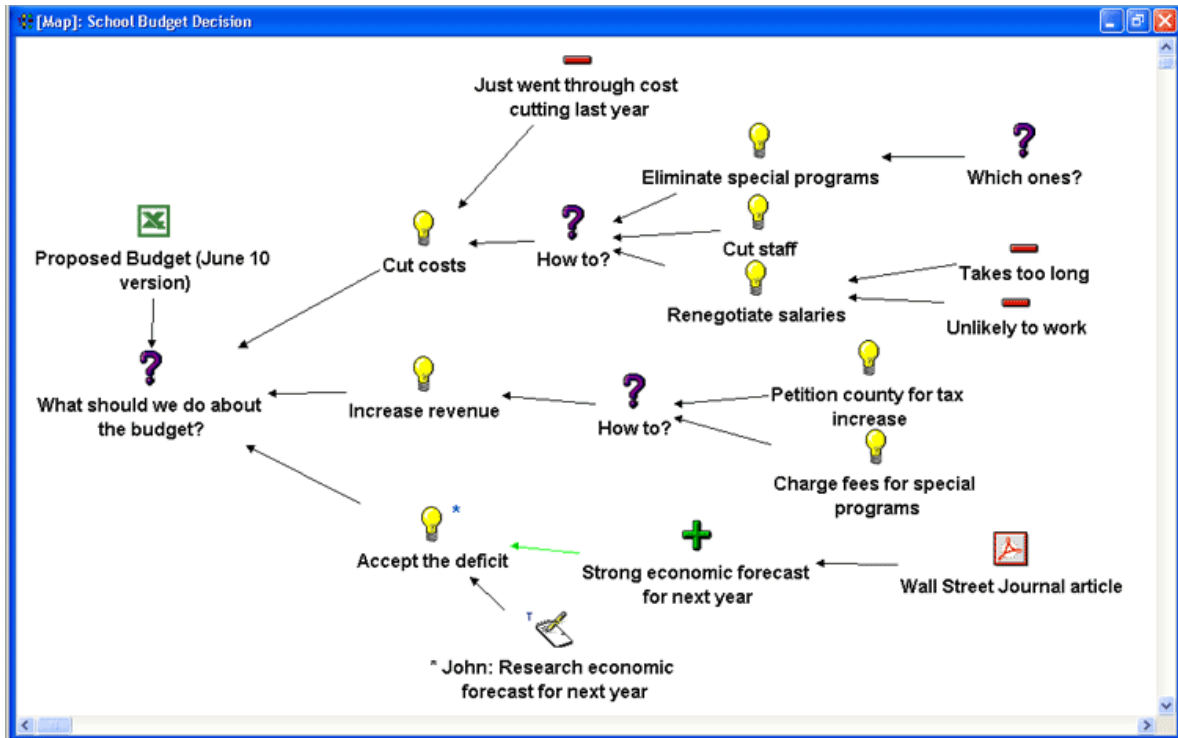


Ilustración 6: mapa de diálogo de una reunión acerca de la planificación de presupuestos

Este método permite obtener rápidamente una visión del problema general que se está tratando, gracias a su representación con íconos, ideas breves y las relaciones entre éstas.

Sin embargo su utilización supone el uso de un software especial que permita realizar este tipo de diagramas, y una sala con el equipo que permita tener este tipo de reuniones. En este caso, sólo los participantes del grupo que puedan asistir a la reunión podrán reflejar sus opiniones en el mapa.

Otro problema de este modelo es que no permite realizar adecuadamente estimaciones basadas en argumentos, dado que para que el mapa sea comprensible a simple vista, las ideas expresadas deben ser frases breves. Esto no permitiría expresar adecuadamente la estimación efectuada con su correspondiente argumento sin llenar el espacio de trabajo con nubes de palabras. La siguiente figura muestra un mapa en donde se utilizan grandes párrafos de texto para expresar ideas, lo que hace que el diagrama aumente su tamaño, perjudicando la visualización y dificultando su lectura.

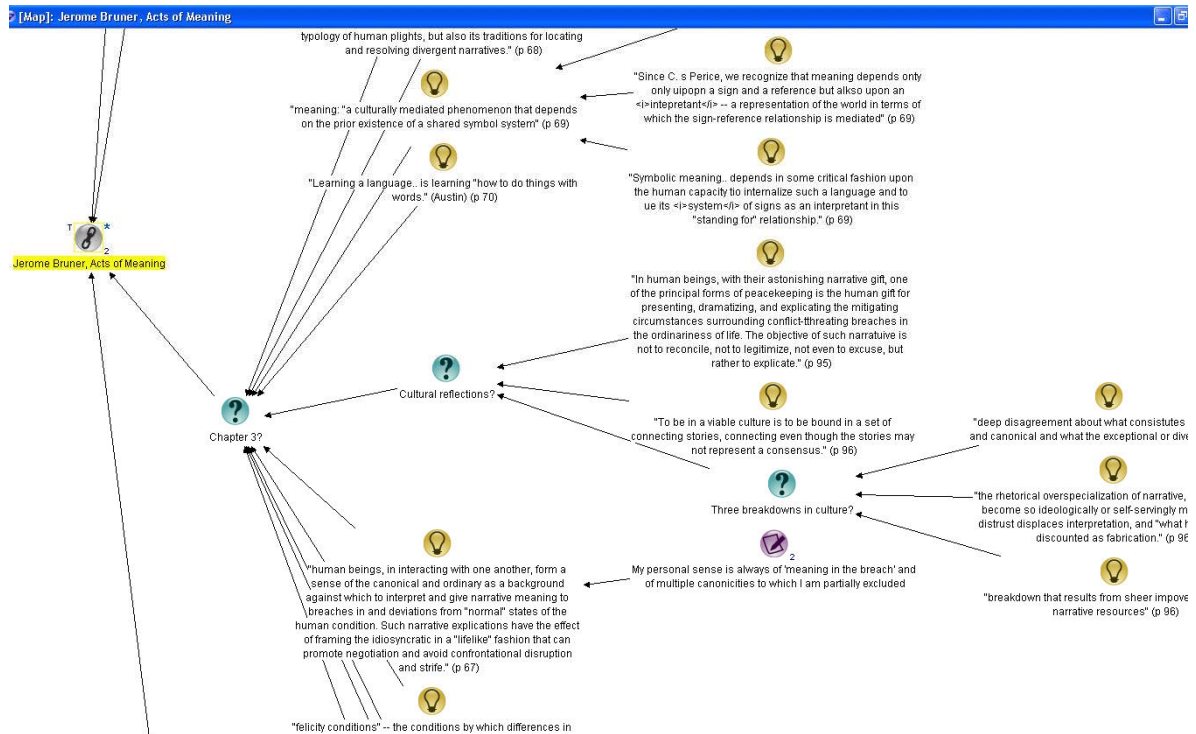


Ilustración 7: mapa de diálogo usando oraciones para representar ideas

Debido a esto, dialogue mapping no se observa como una buena alternativa para ser utilizada como modelo de discusión en una herramienta Web.

9.3 Modelo de argumentación de Toulmin

Stephen Toulmin, en su libro "The Uses of Argument" [33,34] propuso un esquema de discurso argumentativo que contiene seis componentes diferentes que se interrelacionan para formar una discusión. Las tres primeras componentes descritas son consideradas necesarias, mientras que las tres restantes pueden no estar presentes en una argumentación.

- Aserción: es la idea principal que se intenta exponer, defender o resolver. Se distinguen cinco clases de aserciones: factuales, valorativas, políticas, causales y definitorias.
 - **Factuales:** realizan inferencias acerca de hechos en el pasado, presente o futuro. Ejemplo: "La deserción escolar ha decrecido".
 - **Valorativas:** pretenden valorar o dar mérito a una idea o práctica según un conjunto de reglas presentadas por el argumentador. Ejemplo: "El estudio del latín facilita el aprendizaje de una segunda lengua".
 - **Políticas:** indican las acciones a tomar frente a un determinado problema para resolverlo. Ejemplo: "La Universidad debe invertir más en herramientas tecnológicas y en registro de publicaciones en línea que en materiales impresos".
 - **Causales:** expresan el por qué ha ocurrido u ocurrirá algún suceso, y las consecuencias que esto traerá. Ejemplo: "El uso del computador ha aumentado las habilidades escritas de los estudiantes".
 - **Definitorias:** pretenden describir un fenómeno o una idea. Ejemplo: "Todo texto es argumentativo".

- Evidencia: corresponden a las razones, ejemplos, opiniones expertas o datos que apoyan la aserción. Se propone evaluar la evidencia frente a los criterios de actualidad, imparcialidad y que exprese hechos y no opiniones.
- Garantía: son los supuestos que existen tras la argumentación. Pueden corresponder a las creencias o valores generalmente aceptados en la sociedad o cultura, siendo así el terreno común entre el autor de la conclusión y la audiencia. Ejemplo: “El niño tiene fiebre (evidencia). Es porque tiene una infección (aserción). La fiebre es un indicio de infección (garantía)”.
- Calificador modal: debido a que una conclusión o tesis es más una posibilidad que algo cierto, se utilizan calificadores que expresen el grado de certeza del autor sobre la tesis. Ejemplos son: “quizá”, “tal vez”, “probablemente”, “la mayoría”.
- Reserva: corresponde a las negaciones u objeciones que puede expresar la audiencia frente a la aserción o tesis propuesta.
- Respaldo: es información adicional que permite dar fortaleza y veracidad a las garantías entregadas. Ejemplo: “La lectura de textos literarios aumenta la capacidad argumentativa de los estudiantes (aserción). Los alumnos con buenos hábitos de lectura participan más en las discusiones (evidencia). La literatura enseña a pensar (garantía). Trabajos previos afirman que la lectura de textos literarios enseña a resolver problemas y obligan al lector a realizar inferencias profundas que luego transfieren a sus escritos (respaldo)”.

9.4 Modelo “Beliefs, reasons and moves”

Este modelo establece un conjunto de reglas con el objetivo de representar un tipo de diálogo en el que los participantes intentan llegar a un acuerdo o desacuerdo referente a un cierto tema [35]. Se distinguen tres componentes principales en este modelo.

- Creencias: corresponden a las creencias que establece cada participante de la discusión referente al tema que se está tratando, y referente a los otros miembros de la conversación.
- Argumentos: se pueden extraer de las creencias expuestas por los participantes, y apoyan o rechazan el tema en discusión.
- Acciones: están basadas en los argumentos entregados por los participantes, y su objetivo es explicar las diferencias entre sus puntos de vista, intentar entender estas diferencias, aceptar o rechazar una explicación dada y proveer de información a la discusión.

Las acciones aceptadas por el modelo son expresadas en la siguiente tabla.

Tabla 1: acciones del modelo “Beliefs, reasons and moves”

Acción	Descripción
¿Por qué?	El participante pide argumentos para una afirmación realizada por otro miembro.
¿Es/no es así?	El participante intenta confirmar si otros miembros tienen la misma creencia en el tema que él.
Si/no	El participante responde a una solicitud de confirmación
Informo	El participante da información variable o aseveraciones sin argumentos que la respalden.
Porque	El participante da un argumento.
Si pero	Se acepta una información, pero al mismo tiempo se dan razones de mayor peso para otra.
Desacuerdo/acuerdo	Tiene que ver con la aceptación de los puntos de vista de los demás participantes. Indica si se llegó a un acuerdo o a un desacuerdo.

Además, se establecen las siguientes reglas para responder a una acción.

Tabla 2: respuestas a acciones del modelo “Beliefs, reasons and moves”

Acción previa	Acción siguiente	Intención
¿Por qué?	Porque	Proveer de información
Si pero	¿Por qué?	Buscar información
	¿Es/no es así?	Buscar información
	Informo	Explicar diferencias
	Porque	Explicar diferencias
	Si pero	Aceptar/rechazar
	Si/no	No se aplica
	Desacuerdo	Expresar desacuerdo
	Acuerdo	Expresar acuerdo
¿Es/no es así?	Si/no o si pero	Proveer información

Esta representación ofrece un conjunto de acciones básicas que permiten modelar el flujo de una discusión de forma estructurada y basándose siempre en información argumentada. Además los factores que en el intervienen son fácilmente identificables en una discusión normal.

9.5 Modelo de Quignard y Baker

Este modelo fue presentado para estudiar la relación entre la argumentación y el cambio en la forma en que se adquiere conocimiento en interacciones colaborativas para la resolución de problemas [36].

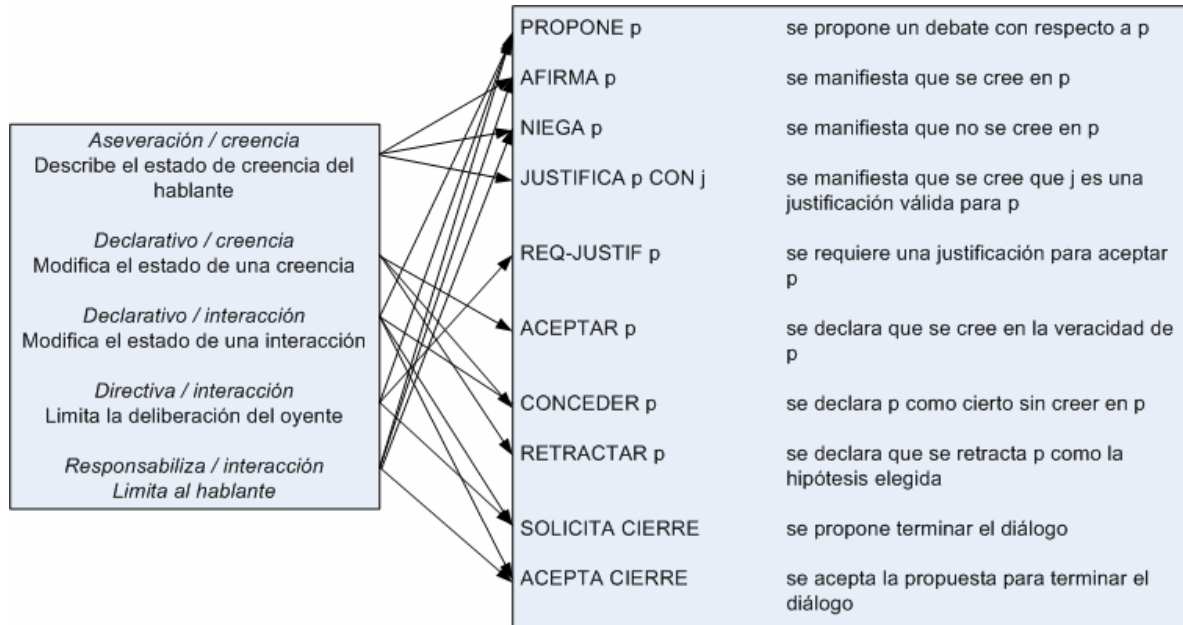


Ilustración 8: modelo de argumentación de Quignard y Baker

La principal diferencia de este modelo con los anteriores es la forma en la que se termina la discusión, donde uno de los miembros solicita a los demás participantes el término de ésta. Las demás acciones como realizar una afirmación y respaldarla con argumentos también están presentes en los demás modelos.

9.6 Búsqueda de un modelo aplicable a Wideband Delphi

Tomando en cuenta los modelos estudiados, se debe establecer como podrían ser utilizados al momento de realizar las estimaciones dentro de la herramienta, de forma que se adapten lo mejor posible a la metodología Wideband Delphi. El objetivo no es aplicar todos los modelos al mismo tiempo, sino establecer basados en ellos un modelo propio que permita realizar estimaciones anónimas donde se llegue a un acuerdo común. Para esto se analizarán las distintas fases que se observan en una sesión de Wideband Delphi, y se establecerá que opciones de comunicación existen entre los participantes.

9.6.1 Fase de presentación de las especificaciones

Esta fase es la que inicia una reunión de estimación basada en Wideband Delphi, y corresponde a aquella en la que se presenta el tema sobre el que se deberán realizar estimaciones. Además, en esta fase se deben presentar los antecedentes que se hayan logrado reunir sobre el tema, tales como la condición actual en la que se encuentra, posibles factores externos que pueden influir en él, los recursos con los que se dispone, entre otros. Las especificaciones presentadas tienen un símil en el modelo “dialogue mapping”, correspondiendo a los archivos anexos que se adjuntan en el mapa para entregar información respecto al tema que se está tratando.

Aquí, el elemento comunicacional que se maneja es información argumentada, por ejemplo “Se solicitó implementar la solución en este lenguaje de programación ya que la empresa sólo

trabaja en él". Tomando como base la metodología Wideband Delphi, estos argumentos presentados no son objeto de una discusión acerca de su veracidad, pues se considera que han sido investigados con anterioridad por los participantes y por lo tanto poseen un respaldo válido. Con esto, se estaría hablando de una aseveración del tipo "justifica p con j" del modelo de Quignard/Baker, a la unión de "Informe" y "Porque" de "Beliefs, reasons and moves", o una aseveración con evidencia y garantía en el modelo de argumentación de Toulmin.

Para poder cumplir con el desarrollo de esta fase en la aplicación, es necesario otorgar a los alumnos un lugar donde ellos puedan expresar su conocimiento referente a la tarea sobre la que se debe realizar la estimación. Este conocimiento debe estar presente al momento en que los alumnos realicen sus estimaciones, de forma que se puedan realizar estimaciones informadas. El conocimiento ingresado por los alumnos debe ser anónimo, de forma que éste sea lo más neutral posible.

9.6.2 Fase de discusión del contexto de estimación

En esta fase los miembros del equipo de estimación conversan entre sí referente a los parámetros que deberán tomarse en cuenta al momento de realizar la estimación. En el modelo Wideband Delphi original, esta fase se lleva a cabo mediante una conversación verbal entre los participantes, por lo que no es posible realizar opiniones anónimas.

Debido a que la presentación por parte de los participantes de los parámetros que definen el contexto del tema a resolver puede variar de persona a persona, es difícil establecer que tipo de comunicación se realizará en esta fase. Sin embargo, el modelo seguido debiera ser similar al establecido por "Beliefs, reasons and moves", donde uno de los miembros entrega una información mediante la unión de "Informe" y "Porque", mientras los demás pueden mostrar su acuerdo o entregar otra información de mayor peso que deba ser considerada mediante el uso de "Si pero". A su vez, siguiendo el modelo Quignard/Baker la información se entregaría mediante "Justifica p con q", a lo que se puede responder con "Aceptar p" si se está de acuerdo, o con "Niega p" para decir que no se está de acuerdo con la información propuesta, tras lo cual se debe plantear una nueva información mediante "Justifica p con q". Puede tomarse por ejemplo la conversación presentada a continuación.

Usuario 1: "Esta parte del proyecto debe estar lista en tres semanas más"

Usuario 2: "¿Por qué?"

Usuario 1: "Porque en tres semanas se debe presentar el prototipo del producto"

Usuario 2: "Si pero podemos renegociar esta parte con el cliente para dar prioridad a otras más importantes para él"

Usuario 1: "Tienes razón, no es necesario tener lista esta parte en tres semanas más"

Para representar esta fase en la aplicación puede utilizarse la herramienta de discusión o Chat.

9.6.3 Fase de respuesta a formularios

En esta fase los miembros del grupo responden de forma anónima los cuestionarios que han sido preparados con anterioridad por el coordinador de la reunión, tomando como base el contexto y supuestos establecidos en las fases anteriores.

Al igual que la primera fase de Wideband Delphi, la comunicación que se presenta en esta etapa corresponde a la de un miembro al resto del equipo, donde no es posible solicitar justificación adicional a la solución que plantea debido al anonimato en que ésta es realizada. Por esto, se las afirmaciones efectuadas corresponden al tipo “justifica p con j” del modelo Quignard/Baker, a la unión de “Informo” y “Porque” de “Beliefs, reasons and moves”, o una aserción con evidencia y garantía en el modelo de argumentación de Toulmin..

Como el software pretende eliminar la diferenciación entre los alumnos otorgándoles a todos las mismas responsabilidades, el rol de coordinador no se ha considerado, por lo que se debe crear un formulario de estimación que permita a los alumnos ingresar su estimación referente al tema y pueda avalarla mediante argumentos.

9.6.4 Fase de resumen de estimaciones realizadas

En esta fase, el coordinador de la reunión prepara un resumen con las estimaciones realizadas por los demás miembros, indicando los puntos donde las estimaciones variaron considerablemente, en donde estuvieron cercanas unas con otras y otros posibles puntos de interés. Por esto, en esta fase no hay acciones comunicativas entre los usuarios.

Debido a que no existe un coordinador en la aplicación, esta fase debe realizarla la misma herramienta sin tomar en cuenta de quién vino cada estimación.

9.6.5 Fase de discusión de estimaciones

En esta fase los participantes analizan los resultados y puntos de interés obtenidos de sus estimaciones, recolectados por el coordinador de la reunión. Discuten acerca de las discrepancias obtenidas y exponen sus ideas referentes a los argumentos entregados por los demás. Esta fase puede romper el anonimato en las estimaciones realizadas debido a que un participante que desee que su estimación sea la elegida, puede mostrar demasiado interés en avalar sus argumentos. Además este mismo comportamiento puede llevar a imponerse frente a los demás en el llamado efecto vagoneta.

Las acciones dialécticas que aparecen en esta etapa corresponden a aquellas del tipo “¿Es/no es así?” del modelo “Beliefs, reasons and moves”, donde los demás usuarios piden la aprobación de sus argumentaciones a los demás miembros. Estos a su vez responden con un “Si/no” para mostrar acuerdo o desacuerdo con el argumento entregado, o con un “Si pero” para reemplazar el argumento entregado con otro que tenga mayor fuerza. En el modelo de Quignard/Baker también se pueden apreciar estas relaciones. Al buscar la confirmación de los demás miembros del equipo se está utilizando la acción “Propone”, justificándola con los argumentos mediante “Justifica con”. La aceptación o rechazo de los demás participantes puede realizarse con “Afirmar” o “Negar”, tras lo cual podrán plantear sus propios argumentos con “Propone” y “Justifica con”. Finalmente el mismo esquema se puede apreciar en el modelo de Toulmin. Para presentar una tesis se debe entregar una aserción junto a su evidencia y

garantía, mientras que para reemplazarla con otra de mayor peso se debe utilizar una reserva.

En el contexto del diseño de la aplicación, donde se desea que cada participante tenga el mismo derecho de opinión que los demás, se debe crear una forma de que los participantes expresen su agrado o desagrado frente a las estimaciones y argumentos entregados por los otros miembros, sin exponer cuál fue su propia estimación.

9.6.6 Fase de análisis de éxito

Posterior al análisis de los resultados alcanzados, se debe discutir si se realizará una nueva ronda de estimación, o si se tomará algún acuerdo que termine la sesión. En caso de decidir realizar una nueva ronda, se repiten las fases explicadas en los capítulos 9.6.3 a 9.6.6. En caso de llegar a un acuerdo, la estimación termina eligiendo como valor final el acuerdo al que se llegó. En este caso, las acciones que mueven el diálogo corresponderán entonces a “Solicita cierre” y “Acepta cierre” en el modelo de Quignard/Baker, o a “Desacuerdo/acuerdo” de “Beliefs, reasons and moves”.

Para poder elegir cuando una estimación ha llegado a su término dentro de la aplicación, se debe ofrecer al alumno la posibilidad de deliberar junto a los demás participantes cuándo se cree que se ha llegado a una estimación que cumple con los requerimientos expuestos en las especificaciones y el contexto del problema.

9.7 Análisis de las acciones comunicativas encontradas

De la descomposición de las fases de Wideband Delphi se puede apreciar que no todas las acciones encontradas en los modelos estudiados pueden ser aplicadas a una sesión de estimación. Considerando la naturaleza del problema según lo expuesto por Rittel [29], en las estimaciones no se está buscando una aproximación buena y una mala, sino que una mejor y una peor. Debido a esto, acciones comunicativas del tipo “Niega” (con el objetivo de anular el argumento entregado por una persona) o “Retractar” (para anular un argumento expresado por quién lo propuso) del modelo Quignard/Baker no se muestran como acciones útiles al buscar de forma colaborativa una respuesta.

“Conceder”, del mismo modelo, implica que el participante puede aceptar una estimación sin aceptar o comprender la argumentación de la misma, lo que puede llevar a que esté ignorando un factor explicado en dicha argumentación que lo podría llevar a modificar su propia estimación, por lo que tampoco se considera una acción válida.

“Informo”, del modelo “Beliefs, reasons and moves” se considera carente de sentido en una discusión basada en argumentos. Esto dado que corresponde a una información variable que debido a su naturaleza no puede ser incluida correctamente en una estimación sin poseer una argumentación mayor que explique su comportamiento. En ese caso, se estaría hablando de la presentación de una idea seguida de su argumentación, lo que puede ser representado de mejor manera por una mezcla de “Informo porque” o “Justifico con” de Quignard/Baker. Eliminando la presentación de ideas sin justificación, puede omitirse la acción “¿Por qué?” de “Beliefs, reasons and moves”, que solicita argumentos para una aseveración que no los incluya.

Finalmente, “Acuerdo/desacuerdo” de “Beliefs, reasons and moves” es aplicable al modelo Wideband Delphi cuando se intenta decidir si una estimación alcanzado un valor aceptable para los participantes. Lo mismo puede decirse de “Solicita/acepta cierre” del modelo

Quignard/Baker, que será propuesto por los participantes que consideren que la estimación ya ha llegado a su fin ya sea porque se llegó a un acuerdo o porque la sesión se ha extendido más de lo planificado.

9.8 Acciones comunicativas planteadas

Basado en el estudio realizado, se plantean las siguientes acciones aplicables al modelo Wideband Delphi. Se asume que la elección del tema a discutir se realiza fuera del ámbito de la estimación, en una reunión anexa.

Tabla 3: acciones comunicativas planteadas

Acción	Descripción
Informo i_1 Porque a_1	Comunica a los demás participantes la información i_1 , que es avalada por el argumento j_1 . La información i_1 puede corresponder a una especificación del problema o una descripción del contexto de la estimación.
Estimo e_1 Porque a_1	Se estima la variable e_1 para el tema que se discute, basándose en a_1 , donde a_1 puede ser uno o más argumentos de los ya entregados o uno nuevo. Las unidades en que esté la variable e_1 cambiarán según sea el tipo de problema que se esté analizando.
Acepto cierre	Se comunica a los participantes que se desea terminar la estimación actual. Como se trabaja en el contexto de una sesión de estimación, la aceptación corresponderá al caso en que el participante considere que se ha llegado a una estimación aceptable, o que la reunión ya lleva un tiempo considerable de desarrollo. Esta acción se hace llegar a los demás integrantes de forma anónima.
Rechazo cierre	Se comunica a los participantes que no se desea terminar la estimación actual. Como se trabaja en el contexto de una sesión de estimación, el rechazo corresponderá al caso en que el participante considere que el resultado alcanzado en la estimación actual se considera no aceptable. Esta acción se hace llegar a los demás integrantes de forma anónima.
No i_1 Porque a_2	Comunica que no se considera válida la información i_1 , debido a las razones expuestas en el argumento a_2 . En caso que uno de los participantes exprese esta acción, implica que hay un desacuerdo entre los argumentos de los participantes. De aceptarse a_2 , la información soportada por el argumento original, en este caso i_1 , deja de ser válida.
Conuerdo con i_1	Un participante expresa que la información i_1 expresada por otro de los participantes está avalada por suficientes argumentos correctos como para considerar la información presentada válida.

Se ha optado por dar énfasis a las acciones que definen el acuerdo o desacuerdo entre los miembros del equipo de estimación. Esto debido a que una de las partes más importantes del proceso de estimación no corresponde a los valores de la estimación individual que haga cada participante, sino que al acuerdo global que se tome entre estos.

Las acciones “No i1 Porque a2” y “Concuerdo con i1” no pueden ser controladas en este momento por la aplicación, debido a que estas son expresadas solamente al momento de plantear la especificación del problema o el contexto de la estimación. Esto se realiza bajo el entorno de la herramienta de discusión o Chat, donde se mantiene un ambiente de conversación casual. Sin embargo estas acciones representan un ideal del estilo de conversación que debería mantenerse para llevar una estimación ordenada y con un control claro de los argumentos entregados. Dependerá de los alumnos el uso adecuado de estas herramientas.

9.9 Fases planteadas

Para poder utilizar el modelo Wideband Delphi dentro del contexto de la aplicación se proponen las siguientes modificaciones a las fases, de forma de mantener los principios de este modelo a lo largo de toda la estimación.

9.9.1 Fase de discusión

Esta fase se corresponde con las fases de “presentación de las especificaciones” y “discusión del contexto de estimación” del modelo Wideband Delphi original. En esta fase los usuarios que posean mayor conocimiento del tema a tratar serán los encargados de presentarlo a los demás usuarios, de forma de poder crear una discusión acerca de todos los elementos que como grupo consideren se deban tener en cuenta al iniciar el proceso de estimación. Además, se deberá analizar los recursos con los que cuentan, como tiempo disponible, y el conocimiento y experiencia en el desarrollo de temas similares. Si ya se han realizado iteraciones anteriores, en esta fase se puede discutir acerca de los resultados obtenidos en ellas.

Datos de salida: la salida esperada de esta fase corresponde a las ideas presentadas por los usuarios y los acuerdos que se hayan alcanzado, lo que constituirá una primera aproximación al contexto en el que se desarrollará la estimación. La conversación será almacenada de forma de crear una memoria organizacional que pueda ser vista por los usuarios para analizar que factores fueron considerados y descartados.

9.9.2 Fase de planteamiento del contexto

Nuevamente dentro del ámbito de la fase de “discusión del contexto de estimación”, los usuarios en esta parte podrán traspasar las ideas y acuerdos que hayan considerado más importantes de la fase anterior y presentarlos formalmente como parte del contexto de la estimación que se realizará. Cada usuario podrá ingresar estos acuerdos desde su punto de vista, creando la posibilidad de otorgar una nueva visión del problema a otro miembro del equipo. Es importante que los factores que sean ingresados por los usuarios sean escritos de la forma más concisa posible, para que puedan ser revisados por los demás usuarios de forma rápida y eficiente.

Datos de entrada: la entrada de esta fase corresponde a las ideas y acuerdos obtenidos en la fase anterior.

Datos de salida: la salida de esta fase corresponde a la especificación de estas ideas por parte de cada usuario. El conjunto de estas especificaciones corresponderá al contexto de la estimación.

9.9.3 Fase de estimación

Al igual que en la fase de “respuesta a formularios” del modelo original, en esta fase se presentará a los usuarios un cuestionario en el que ellos deberán ingresar los valores que consideren correctos para la situación que se esté intentando resolver. Como la aplicación se enfoca a la estimación de períodos de tiempo para el desarrollo de una tarea, el cuestionario presentado solicitará a los alumnos que realicen individualmente el proceso de estimación de las fechas de inicio y término de este proceso, y las ingresen en los campos correspondientes. Además, cada estimación realizada deberá poseer argumentos que la justifiquen o no será aceptada. Junto con el formulario se mostrará a los alumnos los distintos puntos de vista alcanzados referentes al contexto de la estimación, según hayan sido expresados por los participantes en la fase anterior. Esto con el objetivo de otorgar la mayor cantidad de información y puntos de vista sobre el problema al momento de realizar la estimación.

Datos de entrada: contexto de la estimación desde el punto de vista de los diferentes miembros del grupo.

Datos de salida: las estimaciones realizadas por todos los alumnos acerca de las fechas de inicio y término del desarrollo del tema a resolver, junto con los argumentos que las avalen.

9.9.4 Fase de votación de estimaciones

Esta fase, si bien correspondería en la aplicación a la de “discusión de estimaciones” en Wideband Delphi, se diferencia de ella en la forma en la que se lleva a cabo. Aquí los participantes pueden ver las estimaciones realizadas por los demás usuarios y los argumentos que entregaron para respaldarlas. Sin embargo para evitar la pérdida del anonimato y el efecto vagoneta presentes en esta fase del modelo original, se utiliza un modo de votación anónima donde los participantes asignan un puntaje a cada estimación entregada, según consideren que se adapta a la realidad del problema que se esté resolviendo. Un participante que haya propuesto ciertas fechas en su estimación puede otorgar un puntaje mayor a otra si considera que ésta se adapta de mejor manera al problema, o si las argumentaciones que la apoyan poseen mayor peso que las presentadas personalmente. Con estos puntajes, se podrá ver si existe una mayoría en las votaciones o si el grupo está dividido entre dos o más opciones propuestas.

Datos de entrada: las estimaciones y argumentos entregados por los usuarios en la fase anterior.

Datos de salida: el puntaje asignado por los usuarios a las estimaciones obtenidas de la fase anterior.

9.9.5 Fase de acuerdo en estimación

Similar a la fase de “análisis de éxito”, en esta fase los usuarios deben decidir si están de acuerdo con la estimación alcanzada, o si se realizará una nueva ronda. La estimación presentada como la alcanzada por el grupo será la que obtenga la mayoría de los votos entregados en la fase anterior. Si los usuarios se muestran conformes con esta decisión pueden aceptar el acuerdo y solicitar el término de la reunión de estimación. Por el contrario, si consideran que el acuerdo alcanzado no es el correcto, pueden elegir realizar una nueva ronda de estimación. En ambos casos, la reunión de estimación puede ser retomada con posterioridad, y se podrán ver las fechas propuestas, sus argumentos y el contexto presentado para las iteraciones anteriores

Datos de entrada: la estimación que obtuvo la mayoría de votos en la fase anterior.

Datos de salida: la decisión por mayoría de votos de si continuar la reunión con una nueva iteración o elegir la estimación que haya obtenido la mayoría de votos.

9.10 Acciones comunicativas en cada fase

Con el conocimiento de las actividades que se presentarán en cada fase, se adjunta la siguiente tabla que muestra que opciones de diálogo tienen los participantes en cada una de ellas.

Tabla 4: acciones comunicativas en cada fase

Fase	Acciones comunicativas esperadas
Fase de discusión	“Informo i1 porque a1” “No i1 porque a2” “Conuerdo con i1”
Fase de planteamiento del contexto	“Informo i1 porque a1”
Fase de estimación	“Estimo e1 porque a1”
Fase de votación de estimaciones	“Conuerdo con a1”
Fase de acuerdo en estimación	“Acepto cierre” “Rechazo cierre”

A continuación se explica el uso de estas acciones, y se presentan breves ejemplos basados en una reunión donde éstas son utilizadas.

9.10.1.1 Fase de discusión

Los usuarios que quieran compartir sus conocimientos del tema o quienes deseen enviar su aporte a la discusión pueden realizarlo mediante “Informo i1 porque a1”. Los que no concuerden con los argumentos entregados en a1 pueden hacerlo con “No i1 porque a2”. Si a2 es lo suficientemente fuerte, es posible incluso modificar i1 para transformarla en una nueva información, que puede ser transmitida mediante “Informo i2 porque a2”. Cuando los usuarios consideren que cierta información es útil o importante, pueden expresarlo mediante “Conuerdo con i1”.

9.10.1.2 Ejemplo de fase de discusión con las acciones comunicativas planteadas

Usuario1: Creo que deberíamos considerar que nadie sabe el lenguaje en el que se programa la aplicación que tenemos que hacer, nos quitará mucho tiempo el aprender a utilizarlo (“Informo i1 porque a1”).

Usuario2: Yo opino lo mismo. Además tenemos que conseguir un lugar que permita alojar la aplicación, lo que tomara más tiempo (“Conuerdo con i1” e “Informo i2 porque a2”).

Usuario3: Conuerdo con lo de que tenemos que aprender el lenguaje, pero no necesitamos buscar un lugar donde alojar la aplicación porque el equipo docente nos va a entregar uno. (“Conuerdo con i1” y “No i2 porque a3”).

Usuario2: Tienes razón, entonces no nos tenemos que preocupar de eso (“Conuerdo con i2” y “Informo i2 porque a3”).

Usuario1: Entonces por ahora el factor a considerar es que nadie conoce el lenguaje (“Informo i1 porque a1”). Propongo que tengamos reuniones semanales de tutoría entre nosotros mismos para solucionar este problema (“Informo i4 porque a4”).

Usuario2: Estoy de acuerdo (“Concuerdo con i4”).

También es posible que los usuarios realicen preguntas en esta fase. En este contexto, las preguntas podrían entenderse como “Informo que quiero saber i1 porque considero que sería útil para a1”, lo que puede resumirse en “Informo i1 porque a1”. Las respuestas a estas preguntas también poseen esta forma.

9.10.2.1 Fase de planteamiento del contexto

Como los usuarios expondrán su visión de las ideas y acuerdos obtenidos en la fase anterior de forma anónima y sin intervención de los demás participantes, la acción comunicativa utilizada en esta fase es “Informo i1 porque a1”. Mediante la argumentación es posible que surjan nuevos factores que no hayan sido considerados por los demás usuarios en la discusión.

9.10.2.2 Ejemplo de fase de planteamiento del contexto con las acciones comunicativas planteadas

Usuario2: Nadie maneja el lenguaje en el que se debe programar la aplicación, por lo que tendremos reuniones semanales de tutoría. Además, no es necesario que busquemos un lugar donde hospedar la aplicación pues eso lo realiza el equipo docente (“Informo i1 porque a1” y “Informo i2 porque a2”).

El Usuario1 podría agregar la misma idea con mayor detalle o desde otro punto de vista.

Usuario1: El lenguaje en el que hay que desarrollar la aplicación es desconocido, por lo que tenemos que asignar un horario en el que nos reuniremos para tener tutorías. Sé que se pueden encontrar manuales en la página oficial del lenguaje (“Informo i1 porque a1”).

9.10.3.1 Fase de estimación

Nuevamente, en esta fase no hay intervención de los demás participantes, por lo que el usuario simplemente le informa a los demás usuarios de los resultados de su estimación mediante “Estimo e1 porque a1”.

9.10.3.2 Ejemplo de fase de estimación con las acciones comunicativas planteadas

Usuario3: Creo que una fecha de inicio correcta sería d1/m1/a1, debido a que para esa fecha la mayoría de nosotros no tendremos pruebas y nos podremos dedicar completamente a esto. Una fecha de término podría ser d2/m2/a2 porque se nos solicita un avance la semana siguiente a la que propuse, y así tenemos unos cuantos días para realizar una buena revisión de la aplicación (“Estimo e1 porque a1” y “Estimo e2 porque a2”).

9.10.4.1 Fase de votación de estimaciones

En esta fase los usuarios emitirán anónimamente su grado de acuerdo con las estimaciones presentadas, por lo que la acción a utilizar es “Concuero con i1”. Si un usuario no concuerda con una de las estimaciones presentadas, puede decirse que concuerda en un grado nulo con ella, lo que se puede ver reflejado al asignarle un puntaje bajo.

9.10.4.2 Ejemplo de fase de votación con las acciones comunicativas planteadas

La comunicación en esta fase corresponde exactamente a “Concuero con i1” para todos los usuarios, expresando su grado de acuerdo mediante un puntaje.

9.10.5.1 Fase de acuerdo en estimación

Cuando los usuarios vean los resultados de la votación de las estimaciones, ellos podrán comunicar su deseo del término de la reunión de estimación mediante “Acepto cierre”, o podrán pedir una nueva iteración mediante “Rechazo cierre”.

9.10.5.2 Ejemplo de fase de acuerdo en estimación con las acciones comunicativas planteadas

La comunicación en esta fase corresponde exactamente a “Acepto cierre” o “Rechazo cierre”, para todos los usuarios.

9.11 Diagrama resumen del modelo presentado

Este diagrama presenta la relación que existe entre cada una de las fases del modelo. Si se llega a la decisión de terminar la estimación en la fase de acuerdo, se da por terminada la estimación. Si se decide realizar una nueva ronda, se comienza desde la fase de discusión. Es importante notar que toda salida obtenida en cada una de las fases es utilizada en la fase siguiente. Además, intentando seguir el modelo Wideband Delphi, se trabajará utilizando cuestionarios que se adapten lo mejor posible a la tarea de realizar estimaciones.

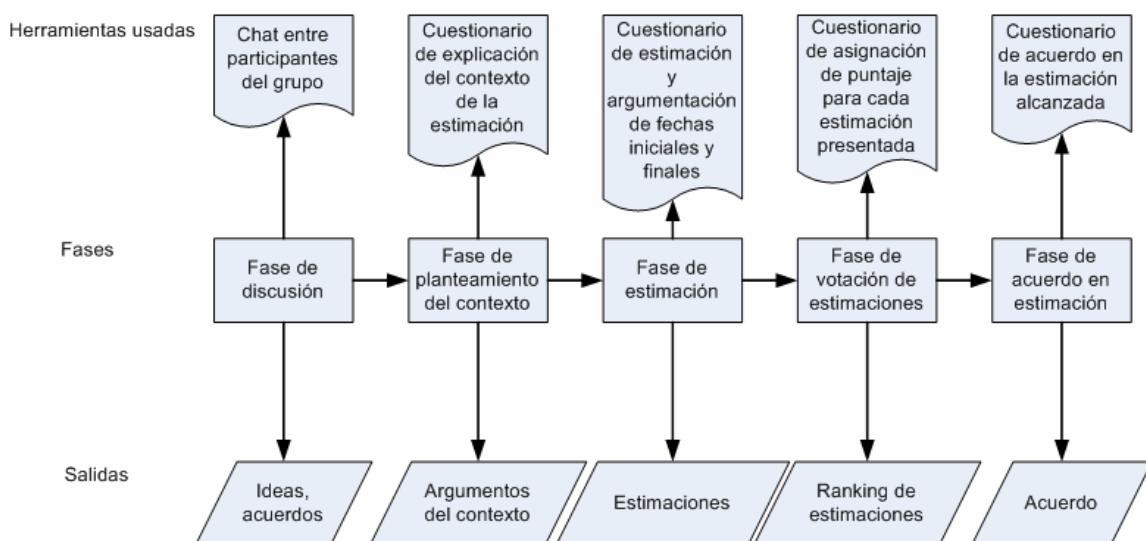


Ilustración 9: diagrama del modelo presentado

9.12 Beneficios y problemas del modelo

Se propuso en el documento “Electronic meeting systems to support group work” [37] la siguiente tabla, que contiene una lista de posibles factores que pueden beneficiar el transcurso de una reunión (ganancias) o perjudicarla de alguna forma (pérdidas) al trabajar en grupo, y cuyo balance determina el éxito o fracaso de ésta. Estas ganancias y pérdidas se consideran relativas al esfuerzo que realizarían estas mismas personas trabajando solas, o a los equipos que no pueden experimentarlas debido a la forma en que se celebran sus reuniones.

9.12.1 Ganancias en una reunión grupal

Tabla 5: ganancias de una reunión grupal

Ganancias en el proceso de una reunión	Descripción
Mayor información	Un grupo tiene mayor cantidad de información que cualquiera de sus miembros por separado.
Sinergia	Un miembro usa una información de una manera que el portador no pudo, debido a que un miembro tiene conocimientos o habilidades diferentes.
Evaluaciones más objetivas	Los grupos pueden detectar errores más rápidamente que los individuos que proponen las ideas.
Estimulación	Trabajar como parte de un grupo puede estimular y animar a los individuos a tener un mejor desempeño.
Aprendizaje	Miembros de un equipo pueden aprender e imitar a otros miembros más aventajados para mejorar su desempeño.

9.12.2 Pérdidas en una reunión grupal

Tabla 6: pérdidas en una reunión grupal

Pérdidas en el proceso de una reunión	Descripción
Fragmentación del tiempo	El grupo debe dividir el tiempo disponible para discutir entre todos los participantes.
Bloqueo por atenuación	Éste (junto con bloqueo por concentración y bloqueo por atención descritos más abajo) es un sub-elemento del “bloqueo de producción”. El bloqueo por atenuación ocurre cuando algunos miembros que no pueden emitir sus comentarios apenas se les ocurren, los olvidan o los omiten por considerarlos poco originales, relevantes o importantes.
Bloqueo por concentración	Se hacen menos comentarios debido a que los miembros se concentran en recordar comentarios (en lugar de pensar unos nuevos) hasta que puedan contribuirlos.

Bloqueo por atención	Nuevos comentarios no son generados porque los miembros deben estar constantemente escuchando a otros hablar y no pueden detenerse para pensar.
Fallo al recordar	A los miembros les falta enfocarse más en la comunicación, ignorando u olvidando los aportes de otros.
Presión conformista	Los miembros tienen temor a criticar los comentarios de otros por buenos modales o temor a represalias.
Aprehensión a evaluaciones	Miedo de evaluaciones negativas causa que algunos miembros repriman sus ideas y comentarios.
Free riding	Algunos miembros dependen de otros para cumplir los objetivos, debido a la inercia cognitiva, a la necesidad de competir por la fragmentación del tiempo o porque creen que su aporte es innecesario.
Inercia cognitiva	La discusión se mueve a lo largo de una línea de pensamiento sin desviarse, debido a que los miembros del grupo no se atreven a realizar comentarios que no están directamente relacionados con la discusión actual.
Socializar	La discusión de temas no relacionados con la tarea reduce el desempeño, aunque se requiere socializar un poco para funcionar de manera efectiva.
Dominación	Algunos miembros del grupo pueden monopolizar o utilizar el tiempo del grupo de una manera poco productiva.
Sobrecarga de información	La información es presentada más rápidamente de lo que puede ser procesada.
Problemas de coordinación	Las dificultades para integrar las contribuciones de un miembro porque el grupo no tiene una estrategia apropiada para ello, puede llevar a ciclos disfuncionales o a discusiones incompletas que terminan en decisiones prematuras.
Uso incompleto de la información	Uso y acceso incompleto a la información necesaria para completar la tarea exitosamente.
Análisis incompleto del tema	Análisis y entendimiento incompleto del tema resultan en discusiones superfluas.

9.12.3 Ganancias en una reunión grupal utilizando Wideband Delphi y la metodología propuesta

Se presenta la siguiente tabla que muestra como afectan las ganancias presentadas en el capítulo anterior a una reunión de estimación utilizando Wideband Delphi y la metodología presentada anteriormente.

Tabla 7: ganancias en una reunión grupal con Wideband Delphi y la metodología propuesta

Ganancias en el proceso de una reunión	Efecto sobre Wideband Delphi	Efecto sobre metodología presentada
Mayor información	Positivo: el grupo comparte sus conocimientos en las fases de presentación de especificaciones, discusión del contexto, discusión de estimaciones y análisis de éxito.	Positivo: el grupo comparte sus conocimientos en la fase de discusión, planteamiento del contexto y estimación.
Sinergia	Positivo: el grupo utiliza los conocimientos comunes en las fases de discusión del contexto, discusión de estimaciones y análisis de éxito.	Positivo: el grupo comparte sus conocimientos en las fases de discusión, planteamiento del contexto y estimación.
Evaluaciones más objetivas	Positivo: el grupo discute las ideas propuestas en las fases de discusión del contexto y discusión de estimaciones.	Positivo: el grupo discute las ideas propuestas en las fases de discusión y votación, al asignar un puntaje bajo a las estimaciones deficientes.
Estimulación	Positivo: puede apreciarse o no según sea el grupo de trabajo.	Positivo: puede apreciarse o no según sea el grupo de trabajo.
Aprendizaje	Positivo: puede apreciarse o no según sea el grupo de trabajo.	Positivo: puede apreciarse o no según sea el grupo de trabajo.

Se ve de esta tabla que ambas metodologías se pueden beneficiar de distinta manera de las posibles ganancias de trabajar en grupo.

9.12.4 Pérdidas en una reunión grupal utilizando Wideband Delphi y la metodología propuesta

Tabla 8: pérdidas en una reunión grupal con Wideband Delphi y la metodología propuesta

Pérdidas en el proceso de una reunión	Efecto sobre Wideband Delphi	Efecto sobre metodología presentada
Fragmentación del tiempo	Negativo: se aprecia en las fases de presentación de especificaciones, discusión del contexto, discusión de estimaciones y análisis de éxito.	Nulo: no se aprecia.

Bloqueo por atenuación	Negativo: se aprecia en las fases de presentación de especificaciones, discusión del contexto, discusión de estimaciones y análisis de éxito.	Nulo: no se aprecia.
Bloqueo por concentración	Negativo: se aprecia en las fases de presentación de especificaciones, discusión del contexto, discusión de estimaciones y análisis de éxito.	Nulo: no se aprecia.
Bloqueo por atención	Negativo: se aprecia en las fases de presentación de especificaciones, discusión del contexto, discusión de estimaciones y análisis de éxito.	Nulo: no se aprecia.
Fallo al recordar	Incierto: según sea la tecnología con la que se registren los eventos, el efecto de este problema podría ser nulo.	Nulo: no se aprecia.
Presión conformista	Negativo: se aprecia en las fases de discusión del contexto, discusión de estimaciones y análisis de éxito.	Negativo: puede presentarse en la fase de discusión.
Aprehensión a evaluaciones	Negativo: se aprecia en las fases de discusión del contexto, discusión de estimaciones y análisis de éxito.	Negativo: puede presentarse en la fase de discusión.
Free riding	Nulo: no se aprecia.	Negativo: puede presentarse en todas las fases, principalmente en la fase de planteamiento del contexto y estimación.
Inercia cognitiva	Negativo: puede apreciarse en todo el proceso, según la forma en la que se lleve la estimación.	Negativo: se aprecia a lo largo de todo el proceso debido a la naturaleza rígida de la aplicación.
Socializar	Incierto: no todos los grupos utilizarán el tiempo de reunión para conversar acerca de temas ajenos a ésta.	Incierto: si bien es cierto que la herramienta utilizada en la fase de discusión puede utilizarse para socializar, no es éste su principal objetivo.

Dominación	Negativo: puede presentarse principalmente en las fases de discusión de estimaciones y análisis de éxito, donde se puede llevar la desviarse la discusión del problema principal para convencer a los participantes de elegir una estimación particular.	Nulo: si un usuario no está realizando aportes referidos al tema principal a resolver, es fácil ignorarlos.
Sobrecarga de información	Nulo: debido a la naturaleza del diálogo de una reunión, no se presentará mayor información que la velocidad a la que hablen los participantes de ésta.	Negativo: dado que los usuarios no necesitan respetar el orden en que los demás participantes hablan para omitir sus juicios, pueden observarse problemas en la fase de discusión, sobre todo para un grupo con un número grande de usuarios.
Problemas de coordinación	Incierto: dependerá de la naturaleza del grupo la forma en que aborden el ingreso de nuevos conocimientos a la reunión.	Incierto: dependerá de la naturaleza del grupo la forma en que aborden el ingreso de nuevos conocimientos a la aplicación.
Uso incompleto de la información	Incierto: dependerá de la naturaleza del grupo la forma en que hayan preparado el acceso a la información y el uso que le den a esta.	Incierto: dependerá de la naturaleza del grupo la forma en que utilicen la información ingresada en la aplicación.
Análisis incompleto del tema	Incierto: dependerá de la naturaleza del grupo la forma en que aborden la información ingresada antes de comenzar la reunión.	Incierto: dependerá de la naturaleza del grupo la forma en que aborden la información ingresada en la fase de discusión.

Si bien ambas metodologías tienen problemas con algunas de estas pérdidas, el impacto que estas tienen para la metodología presentada es menor al que se presenta en el modelo Wideband Delphi original.

10- Funcionamiento del sistema

En este capítulo se explicará en mayor detalle las acciones que podrán realizar los usuarios dentro del sistema mediante casos de uso, las páginas que se crearán para satisfacerlos y la relación entre éstas con los diferentes componentes del sistema.

10.1 Actores del sistema

Son actores del sistema las personas que deben interactuar con la aplicación. Los actores del sistema actual corresponden a los siguientes.

Administrador: este rol corresponde a los miembros del equipo docente del curso en el que se utilice la aplicación, y deben poseer permisos de administrador en el equipo en el que se ejecute la aplicación con el fin de poder llevar a cabo las modificaciones necesarias para el correcto funcionamiento del sistema. Deben poseer un listado con los datos de los alumnos del curso y el grupo de desarrollo al que pertenecen.

Alumno: este rol corresponde a los alumnos que estén cursando el ramo en el que se utilice la aplicación. No necesitan tener permisos especiales asignados en el equipo donde esté instalada la aplicación. Deben tener asignado un grupo de desarrollo.

Usuario: se hablará de usuario cuando se hable de acciones que puedan ser realizadas tanto por un alumno como por un administrador.

10.2 Diagramas de casos de uso

Los siguientes son los diagramas de los principales casos de uso para los diferentes actores del sistema.

10.2.1 Diagrama de casos de uso para administración

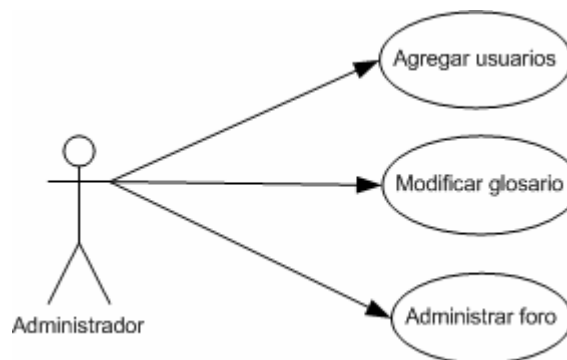


Ilustración 10: diagrama de casos de uso para administrador

Agregar usuarios: se refiere a la acción de ingresar al sistema los datos de las personas que utilizarán la aplicación. Esto puede hacerse uno a uno mediante el panel de control del foro, o de forma masiva con un formulario especial de la aplicación, el cual solicitará el ingreso de un archivo en formato CSV con los datos de los usuarios. Por cada usuario que se desee agregar,

el archivo deberá contener una línea con los siguientes datos: identificador del usuario dentro del sistema, nombre real del usuario, identificador del grupo al que pertenece, dirección de correo electrónico y la contraseña que utilizará para ingresar al sistema. Este archivo CSV deberá traer estos campos separados por un punto y coma. De estos campos, el único que puede omitirse corresponde al nombre real, en cuyo caso se utilizará el identificador del usuario cuando sea necesario mostrar su nombre. La primera línea de este archivo no será revisada, pues se asume que contendrá los nombres de los campos a listar. Esto se describe en mayor detalle en el capítulo 12.2.3.6: “Ingreso de usuarios al sistema”.

Modificar glosario: corresponde a la modificación de los conceptos y sus definiciones que estarán contenidos en el glosario. Los términos que deseen agregarse deberán estar en un archivo de texto, el cual contendrá por cada término que se desee agregar un paréntesis de corchete de apertura en una línea en blanco, seguido del concepto en la siguiente línea. En las líneas siguientes al concepto irá la definición. Una vez finalizada la definición, debe ir un paréntesis de corchete de cierre en una línea en blanco. Esto se describe en mayor detalle en el capítulo 12.2.3.5: “Actualización de glosario”.

Administrar foro: incluye las actividades relativas a la administración del foro de mensajes. Estas incluyen la creación y modificación de categorías y subforos, cambio de apariencia del foro y moderación de mensajes. Además como la aplicación utiliza las cuentas de usuario definidas en el foro, la creación de cuentas de usuario y la asignación de estas a grupos de trabajo se realizan desde el mismo foro.

10.2.2 Diagrama de casos de uso para proyectos

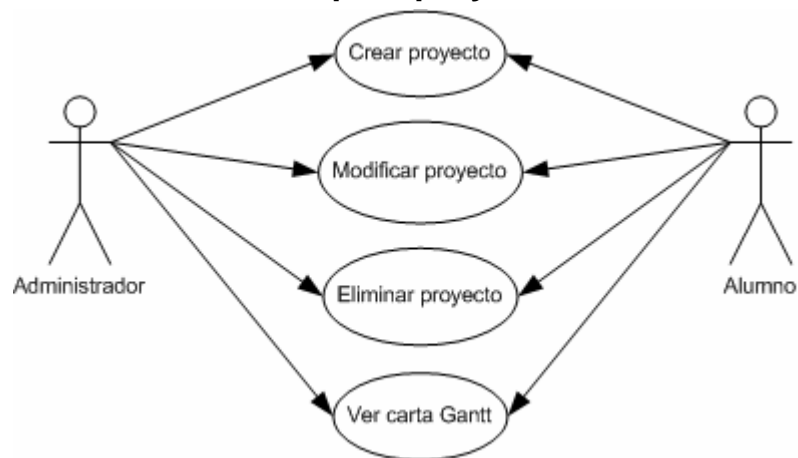


Ilustración 11: diagrama de casos de uso para proyectos

Crear proyecto: un usuario del sistema puede crear proyectos indicando los siguientes datos: nombre del proyecto, fecha de inicio, fecha de término y descripción del proyecto.

Modificar proyecto: una vez que se tenga un proyecto creado, se puede modificar su fecha de inicio, término y su descripción, para actualizar la información frente a cambios en las condiciones de desarrollo.

Eliminar proyecto: cuando se tenga un proyecto creado, es posible eliminarlo, junto con todas las tareas que desprendan de él.

Ver carta Gantt: cuando se tenga un proyecto creado, debe ser posible ver la duración ingresada del proyecto en una carta Gantt, así como la duración de todas las tareas que ya hayan sido creadas y se desprendan de él.

10.2.3 Diagrama de casos de uso para tareas

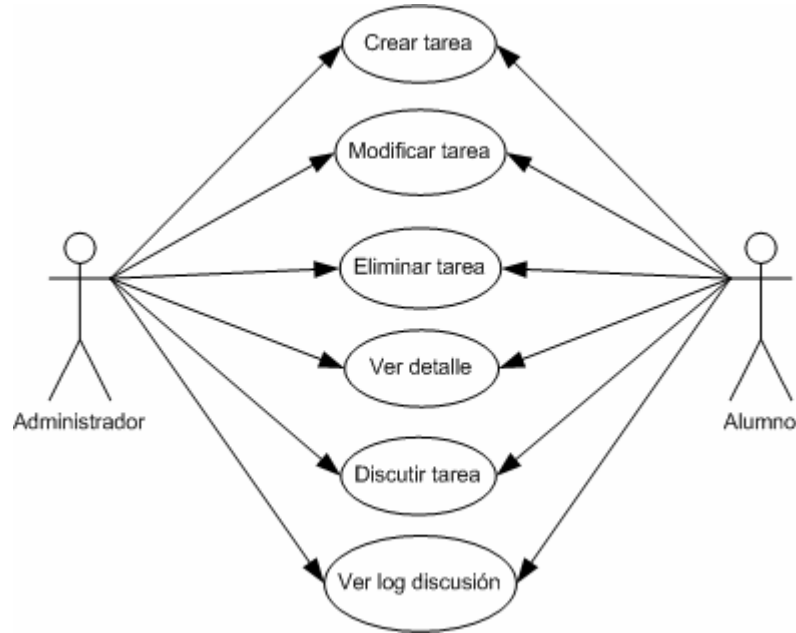


Ilustración 12: diagrama de casos de uso para tareas

Crear tarea: un usuario del sistema puede crear tareas indicando los siguientes datos: nombre, miembros del grupo responsables de la tarea, restricciones entre tareas ya creadas, fecha de inicio y fecha de término.

Modificar tarea: cuando exista una tarea creada se pueden modificar los responsables de esa tarea, las restricciones entre tareas ya creadas, la fecha de inicio, de término, el estado en el que se encuentra la tarea y el porcentaje de avance que se lleva en ella hasta el momento.

Eliminar tarea: cuando exista una tarea creada, esta puede eliminarse de la aplicación.

Ver detalle: cuando se haya creado una tarea, será posible ver la información ingresada de ésta, así como una representación gráfica de su avance e información adicional.

Discutir tarea: al momento de crear una tarea, se creará una sala de Chat donde se podrán realizar discusiones referentes a ella.

Ver log discusión: una vez que se haya iniciado una discusión en una sala de Chat de una tarea, será posible ver un registro con las conversaciones realizadas entre sus participantes.

10.2.4 Diagrama de casos de uso para herramientas



Ilustración 13: diagrama de casos de uso de herramientas

Responder cuestionario: un usuario podrá responder un cuestionario con preguntas referentes a la temática del trabajo en grupo, y le asignará un puntaje a distintas áreas de habilidad basado en sus respuestas.

Revisar glosario: un usuario podrá revisar la terminología básica utilizada en labores de desarrollo de aplicaciones de manera grupal y en ingeniería de software.

Calcular indicador de complejidad: un usuario podrá obtener información acerca de los estimadores algorítmicos de complejidad y del modelo COCOMO, así como calcular el valor de este indicador basados en los factores de complejidad que el mismo usuario estime.

Ir a foro: los usuarios podrán acceder al foro del sistema, donde podrán crear temas de conversación públicos. Para poder participar de un foro privado para el grupo al que pertenece el usuario, el administrador del sistema debe haber previamente creado este foro con los permisos pertinentes.

10.2.5 Diagrama de casos de uso para estimaciones

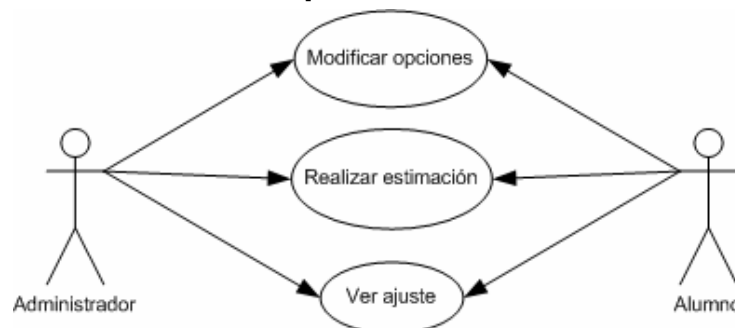


Ilustración 14: diagrama de casos de uso de estimaciones

Modificar opciones: un usuario que pertenezca a un grupo que haya creado anteriormente una tarea podrá alterar algunos parámetros de la sala de estimación de esta tarea, como son los tiempos de duración de cada fase.

Realizar estimación: un usuario que pertenezca a un grupo que haya creado anteriormente una tarea, podrá participar en una reunión de estimación en la que se discutirá la duración del desarrollo de la tarea, mediante la argumentación de fechas de inicio y término.

Ver ajuste: un usuario que pertenezca a un grupo que haya creado anteriormente una tarea podrá ver como se han ido ajustando las estimaciones realizadas previamente por el grupo. Se mostrarán las estimaciones entregadas por los alumnos y el puntaje que alcanzaron en la iteración en la que fueron registradas, los argumentos que las avalaban y el contexto en el que se desarrollo la estimación.

10.3 Estructura de la aplicación

Para cumplir con los casos de uso propuestos en el capítulo anterior, se distribuirán los contenidos de la aplicación de la siguiente forma, agrupados según la funcionalidad que ofrezcan al alumno.

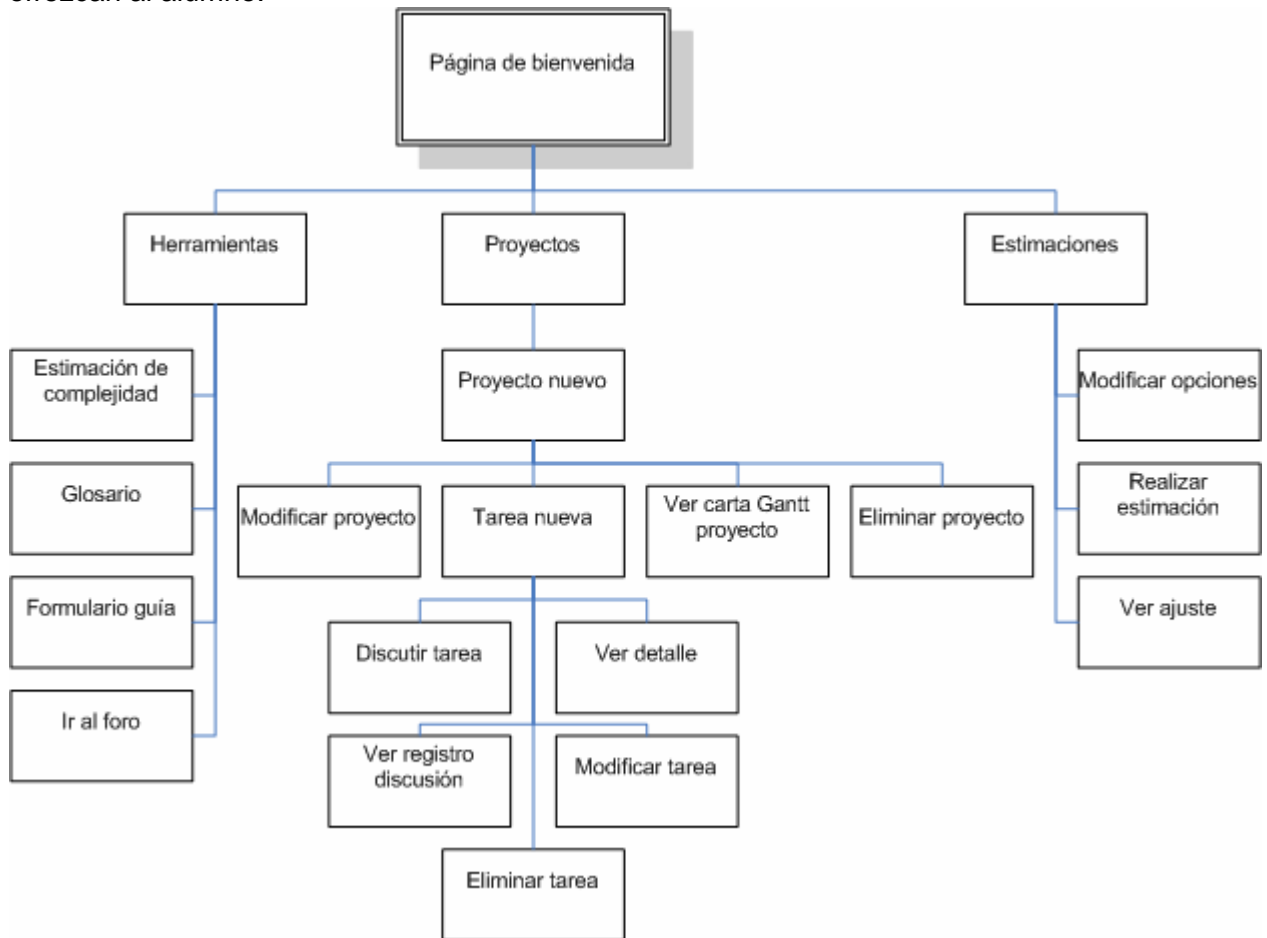


Ilustración 15: estructura de la aplicación

10.4 Diagramas de secuencia

Se utilizarán los diagramas de secuencia para mostrar los pasos que sigue el sistema en los casos de uso más representativos de la aplicación.

10.4.1 Diagrama de secuencia para creación de proyectos

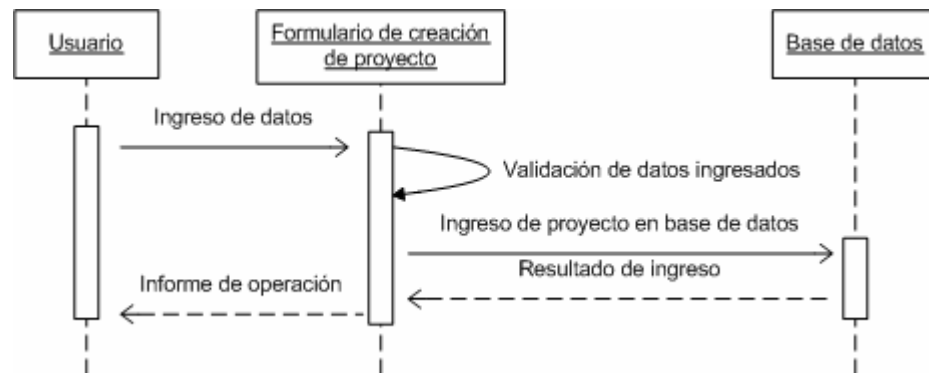


Ilustración 16: diagrama de secuencia para creación de proyectos

Este diagrama muestra la secuencia de operaciones que representan al caso de uso “Crear proyecto”. Para que la secuencia comience a ejecutarse, se debe tener un usuario que ha ingresado al sistema previamente con su nombre de usuario y contraseña. Este usuario puede corresponder a un alumno o a un administrador. Posteriormente el usuario debe acceder a la página que contenga el formulario de creación de proyecto nuevo, donde los campos solicitados serán validados para asegurar que hayan sido ingresados correctamente. La página llamará entonces a la base de datos, solicitando la información del usuario necesaria para formar la consulta que creará el nuevo proyecto. Si esta consulta se ejecuta correctamente, el nuevo proyecto quedará creado para todos los miembros del grupo al que pertenece el usuario actual. Si el usuario pertenece a más de un grupo, el proyecto se creará para el grupo que haya seleccionado. Desde ese momento estarán disponibles las opciones de modificación y eliminación del proyecto, así como la visualización de su carta Gantt. Además, se podrán comenzar a agregar tareas nuevas para ese proyecto.

10.4.2 Diagrama de secuencia para creación de tareas



Ilustración 17: diagrama de secuencia para creación de tareas

Este diagrama muestra la secuencia de operaciones que representan al caso de uso “Crear tarea”. Para que la secuencia comience a ejecutarse, se debe tener un usuario que ha ingresado al sistema previamente con su nombre de usuario y contraseña. Este usuario puede corresponder a un alumno o a un administrador. Posteriormente el usuario debe acceder a la página que contenga el formulario de creación de tarea nueva, donde los campos solicitados serán validados para asegurar que hayan sido ingresados correctamente. La página llamará entonces a la base de datos, solicitando la información del usuario necesaria para formar la consulta que creará la nueva tarea. Con esta información se ejecutarán tres funciones de acceso a la base de datos, que se encargarán de crear la tarea, así como su respectiva sala de Chat y la sala donde se realizarán las estimaciones. Cuando una tarea es creada para un determinado proyecto, ésta se crea sólo para los miembros del grupo creador de dicho proyecto.

10.4.3 Diagrama de secuencia para eliminar proyectos o tareas

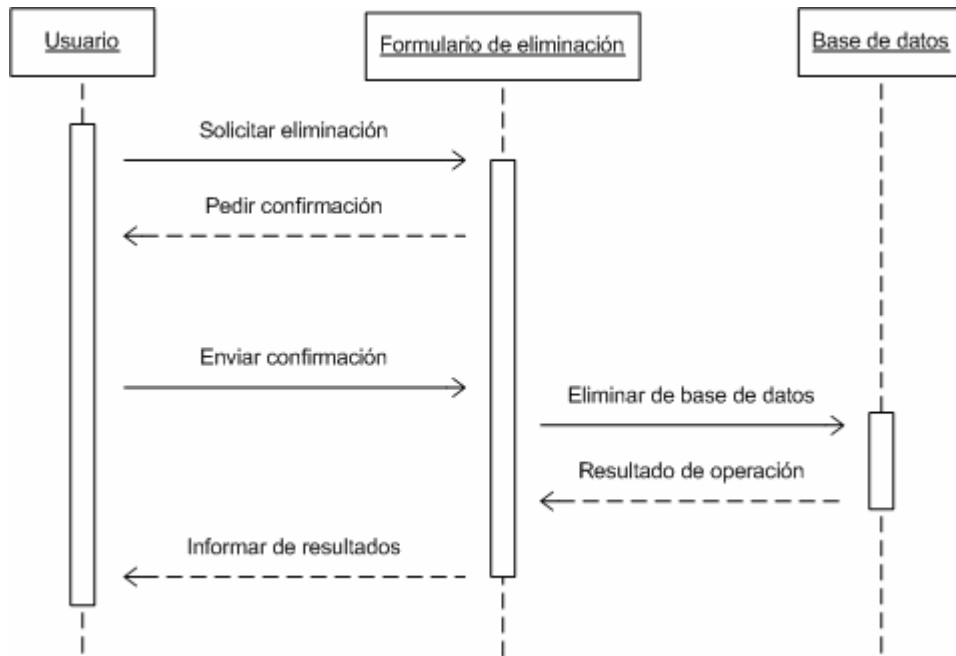


Ilustración 18: diagrama de secuencia para eliminar proyectos o tareas

El diagrama de secuencia aquí presentado corresponde a los casos de uso “Eliminar proyecto” y “Eliminar tarea”. Para eliminar un proyecto o una tarea se necesita que el usuario haya ingresado al sistema con su nombre de usuario y contraseña, y que haya creado con anterioridad un proyecto o una tarea, según corresponda. Una vez que el usuario ingrese al formulario de eliminación, se le pedirá una confirmación para borrar el proyecto o tarea. En caso que el usuario confirme su decisión, el formulario llamará a la función correspondiente para quitar el proyecto o tarea de la base de datos.

En el caso en que se elimine un proyecto, se eliminarán todas las tareas asociadas a ese proyecto, así como todo registro que haya sido generado en el Chat de discusión o en las salas de estimación de cada tarea del proyecto. En el caso que se elimine una tarea, se borrará su sala de Chat y todo registro que haya sido generado en ella, junto con la sala de estimación.

10.4.4 Diagrama de secuencia para ver carta Gantt

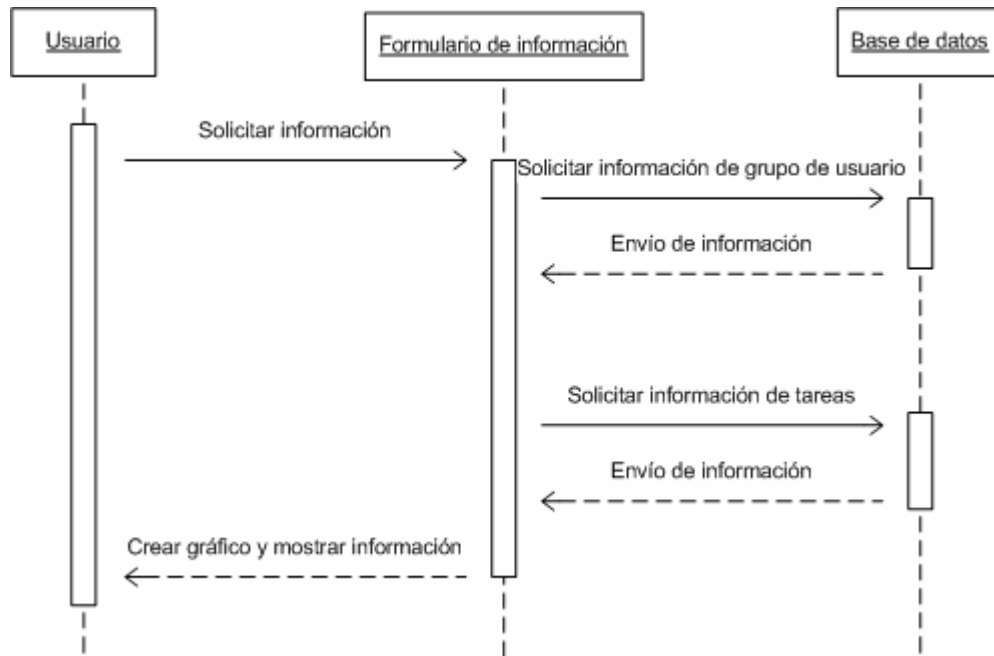


Ilustración 19: diagrama de secuencia para ver carta Gantt

Este diagrama de secuencia corresponde a los casos de uso “Ver carta Gantt” para un proyecto, “Ver detalle” para una tarea o “Ver ajuste” para una sala de estimación. En todos estos casos se requiere que el usuario se haya identificado en el sistema con su nombre de usuario y contraseña, y que haya creado un proyecto o una tarea según corresponda.

Si el usuario accede al formulario “Ver carta Gantt” para un proyecto, éste solicitará a la base de datos la información acerca de las fechas de inicio y término del proyecto, las fechas de inicio y término de las tareas que dependan de él, las restricciones que existan entre estas, el avance en cada tarea y los miembros del grupo asignados a cada tarea. Con esta información generará el gráfico que será mostrado en pantalla.

Si el usuario accede al formulario “Ver detalle” para una tarea, éste solicitará a la base de datos la información acerca de las fechas de inicio y término de la tarea elegida, sus responsables y el avance alcanzado hasta el momento, generando con esto el gráfico correspondiente.

Si se accede a “Ver ajuste”, el formulario solicitará a la base de datos las fechas de inicio y término estimadas por los alumnos para la tarea seleccionada, para cada iteración que se haya realizado, junto con el puntaje obtenido y los argumentos entregados. Con esta información, el formulario podrá solicitar la generación del gráfico correspondiente.

10.4.5 Diagrama de secuencia para modificar tarea

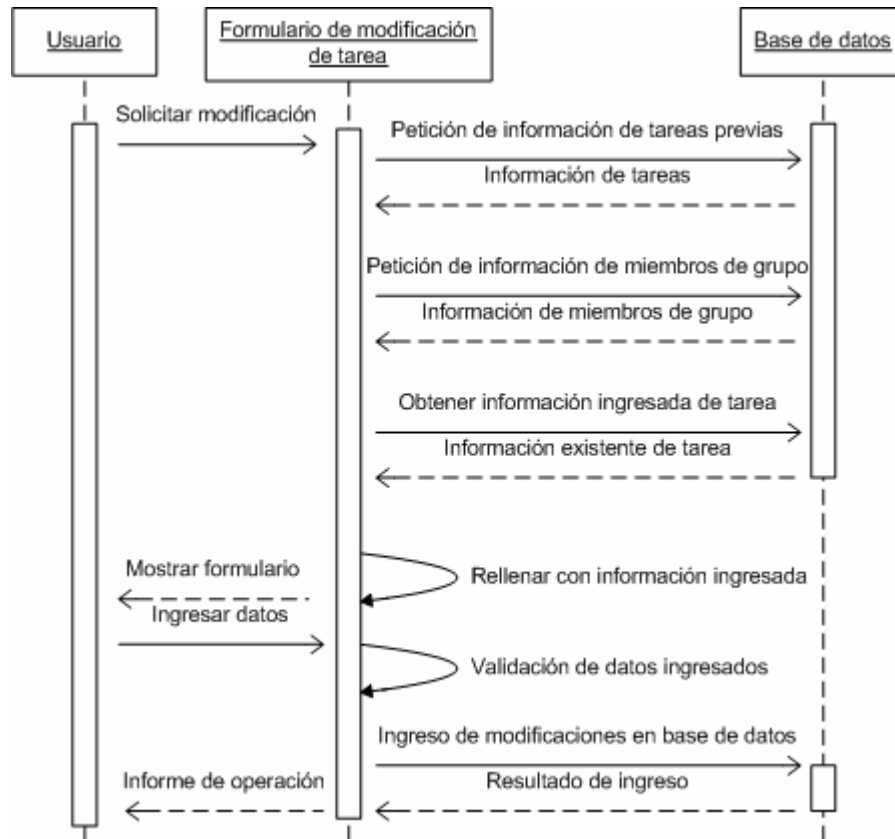


Ilustración 20: diagrama de secuencia para modificar tarea

Este diagrama muestra las llamadas que se realizan cuando el usuario desea modificar una tarea ya existente. Para poder llevar a cabo esto se requiere que el usuario se haya identificado en el sistema con su nombre de usuario y contraseña, y que haya creado con anterioridad una tarea perteneciente a un proyecto. Cuando el usuario solicite ver el formulario de modificación de tarea, se obtendrá de la base de datos la información que haya sido ingresada con anterioridad y se cargarán como las opciones por defecto. Una vez que el usuario haya modificado la información deseada, ésta será revisada y almacenada en la base de datos, informándole al usuario del resultado de esta operación.

Este diagrama es similar al utilizado para modificar proyectos o para modificar las opciones de una sala de estimación. Se debe modificar el llamado a la base de datos de forma de poder obtener y modificar la información correspondiente según sea el caso.

10.4.6 Diagrama de secuencia para ingreso a estimación

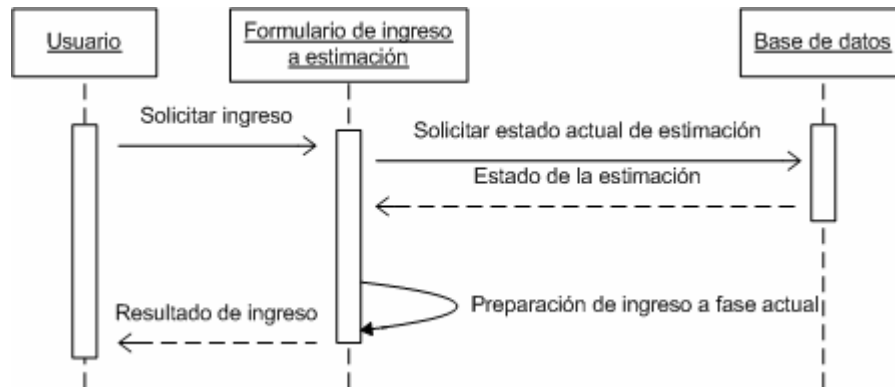


Ilustración 21: diagrama de secuencia para ingreso a estimación

Este diagrama muestra la secuencia de operaciones que representan a la primera parte del caso de uso “Realizar estimación”. Para que la secuencia comience a ejecutarse, se debe tener un usuario que haya ingresado al sistema previamente con su nombre de usuario y contraseña. Además, este usuario debe pertenecer a un grupo que haya creado anteriormente un proyecto y al menos una tarea perteneciente a él. De esta forma podrá acceder a la sala de estimación para la tarea creada.

Una vez que el usuario elija entrar a la sala de estimación, el formulario de ingreso solicitará a la base de datos la información del estado actual en el que está la sesión de estimación, así como la fase en la que se encuentra. Según sea la información entregada por la base de datos, el formulario se preparará para redirigir al usuario a la fase correspondiente.

10.4.7 Diagrama de secuencia para fase de estimación

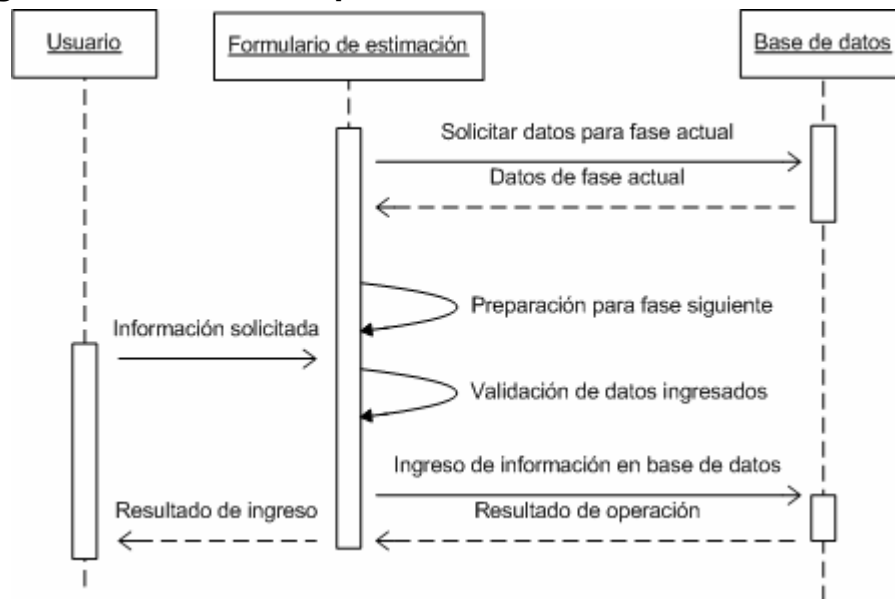


Ilustración 22: diagrama de secuencia para fase de estimación

Los formularios en los que se necesita que el usuario ingrese información realizan una secuencia similar a la mostrada en la ilustración 22. El formulario solicitará a la base de datos la información que necesite para preparar la redirección a la fase siguiente de la estimación. Luego, el usuario deberá ingresar una información determinada, según sea la fase en la que se encuentre la estimación actual. Estos datos serán validados e ingresados a la base de datos.

10.5 Diagramas de flujo de datos

A continuación se explica el flujo de datos que se produce entre la aplicación y la base de datos para los casos de uso más representativos de los presentados en el capítulo 10.2, con el objetivo de explicar el funcionamiento interno del sistema.

10.5.1 Flujo de datos para creación de proyectos



Ilustración 23: flujo de datos para creación de proyectos

Una vez que el usuario ha seleccionado crear un proyecto nuevo, deberá ingresar los campos solicitados por el formulario, representados en la ilustración 23. Estos datos serán validados por el formulario, y en caso de ser incorrectos se pasarán nuevamente al formulario, para mostrar los errores al usuario. Cuando sean ingresados correctamente, estos serán enviados desde el mismo formulario a la tabla que almacena los proyectos en la base de datos.

10.5.2 Flujo de datos para creación de tareas

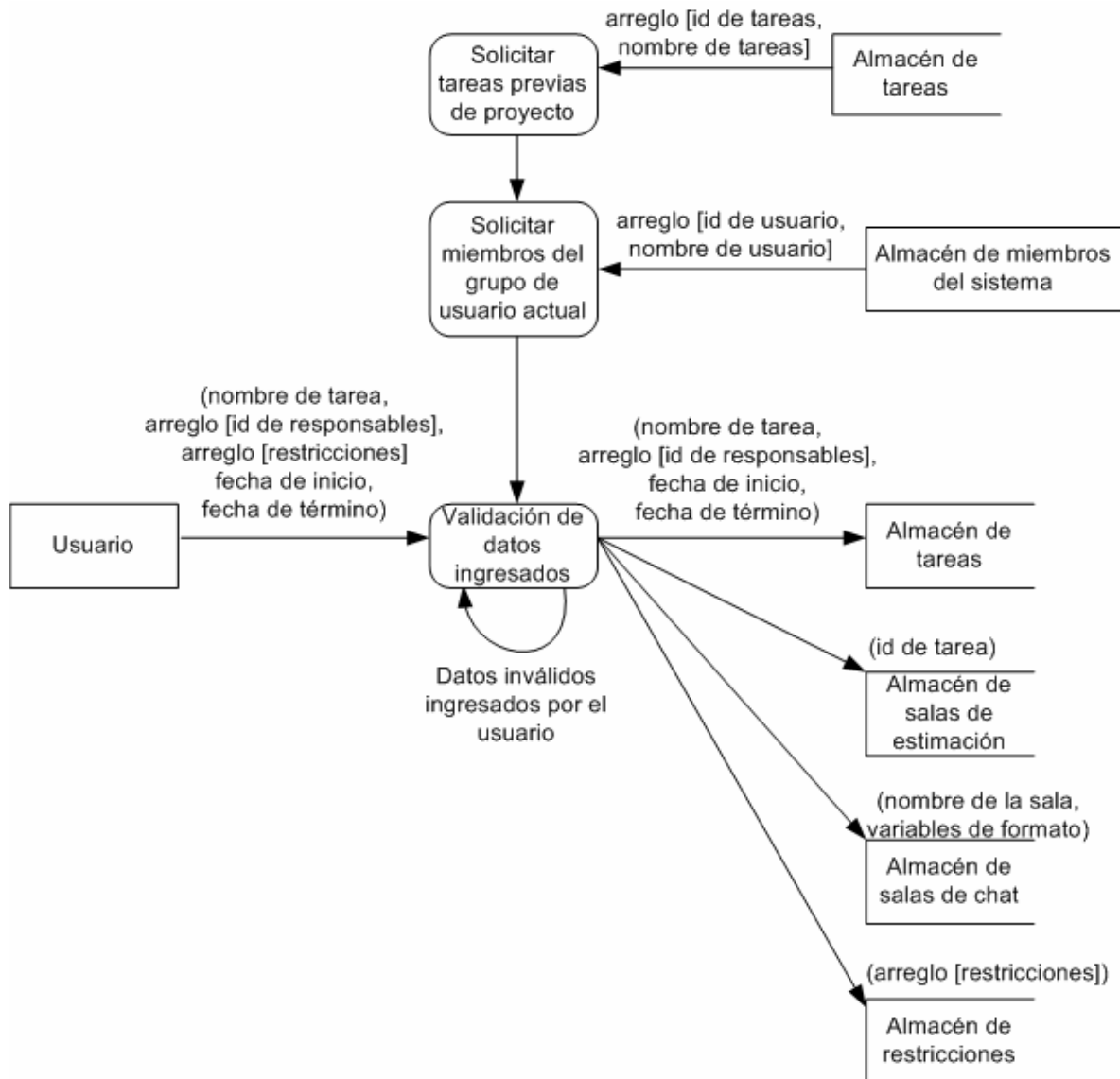


Ilustración 24: flujo de datos para creación de tareas

Antes de poder mostrar el formulario de creación de tareas al usuario, es necesario extraer información de la base de datos para tener la información relacionada con tareas previas del proyecto y los miembros de su grupo con el fin de preparar las posibles restricciones entre tareas y los responsables. Con esto, el formulario es mostrado y el usuario debe ingresar los datos solicitados, según se ve en esta ilustración. Si son ingresados de forma incorrecta, los datos se volverán a pasar al formulario para indicar los errores ocurridos. Cuando sean ingresados correctamente, estos serán insertados en las tablas que almacenan la información de las tareas, salas de estimación, salas de Chat y restricciones entre tareas.

10.5.3 Flujo de datos para eliminar proyectos

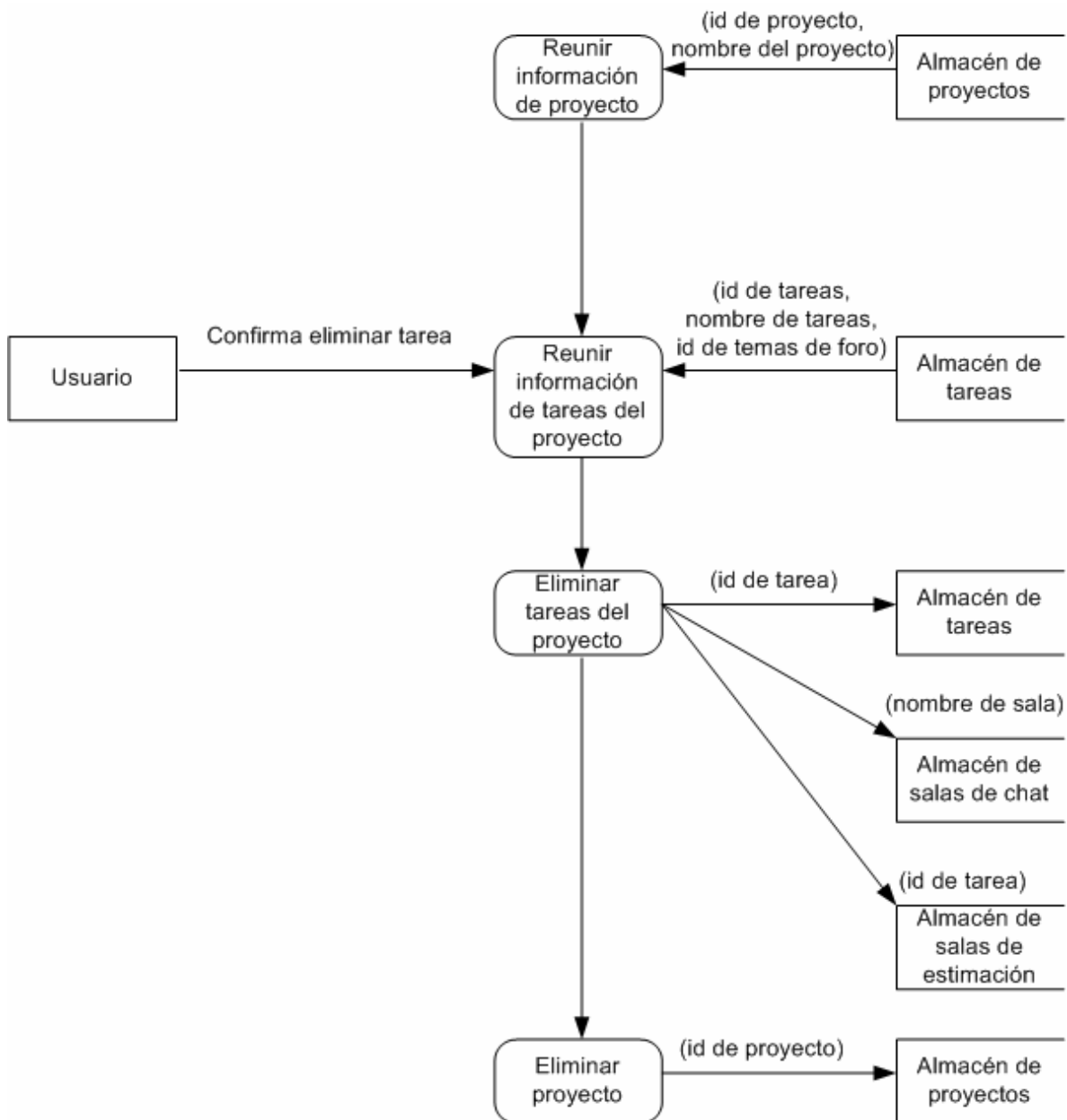


Ilustración 25: flujo de datos para eliminar proyectos

Previo al despliegue del formulario, se obtiene información del proyecto a eliminar. Una vez que el usuario confirme que desea eliminar el proyecto, se obtiene información de las tareas que lo componen, y se procede a eliminarlas de la base de datos, junto con sus salas de Chat y de estimación. Una vez que esta operación ha concluido, se elimina el proyecto de la tabla de proyectos.

10.5.4 Flujo de datos para eliminar tareas

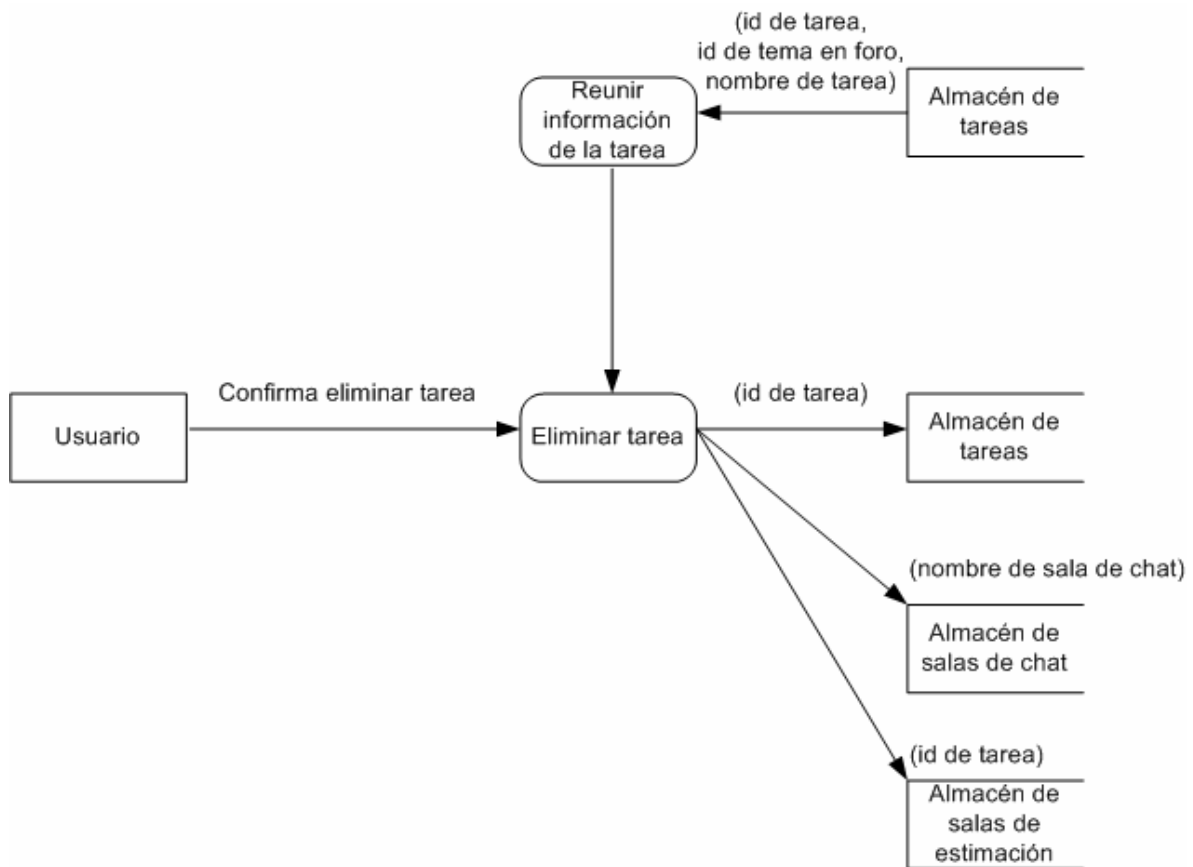


Ilustración 26: flujo de datos para eliminar tareas

Antes de mostrar el formulario al usuario, se debe reunir información de la tarea, según se muestra en la ilustración 26. Si el usuario confirma la eliminación, la tarea se elimina de la tabla que mantiene las tareas. Además, se eliminará su sala de estimación y su correspondiente sala de Chat.

10.5.5 Flujo de datos para ver carta Gantt

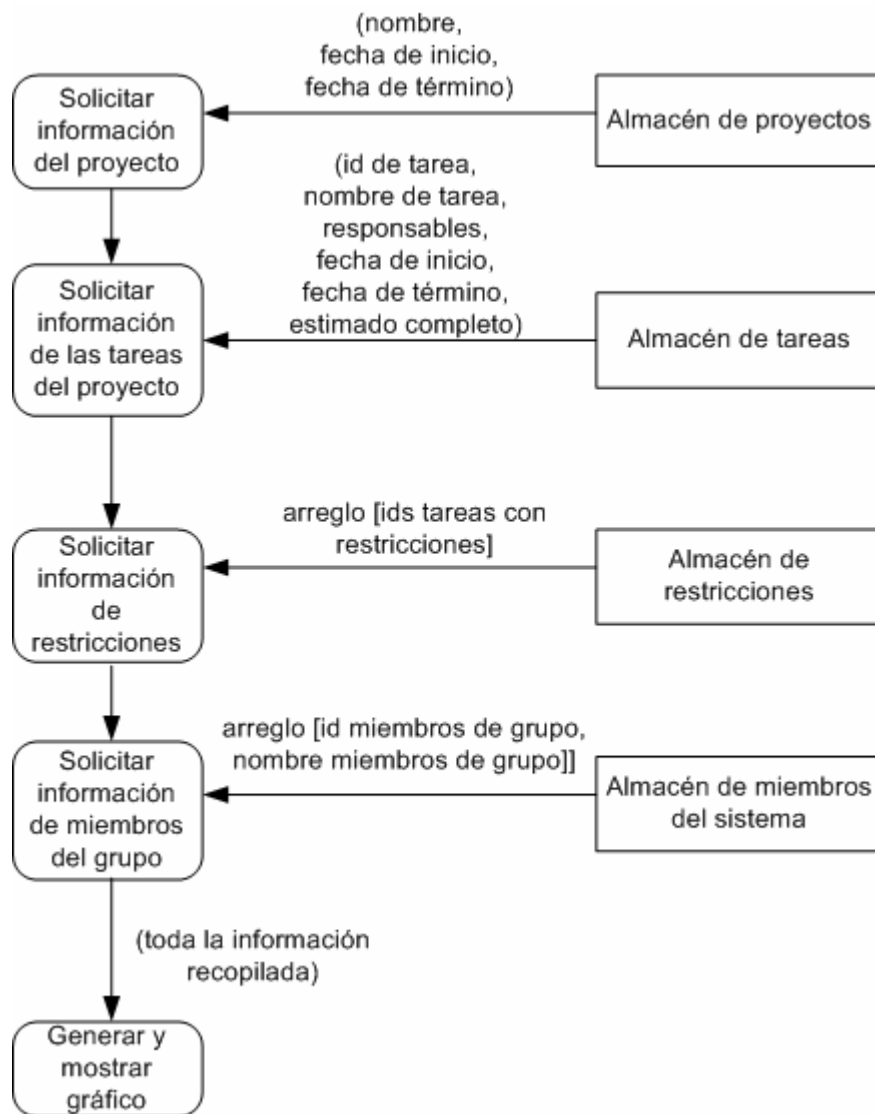


Ilustración 27: flujo de datos para ver carta Gantt

Antes de poder mostrar la carta Gantt de un proyecto, es necesario reunir la información de su duración, así como la de sus tareas y los responsables de estas. También se debe solicitar las restricciones de orden que existan entre tareas. Toda esta información es pasada como parámetro a la función que generará el gráfico.

10.5.6 Flujo de datos para modificar tarea

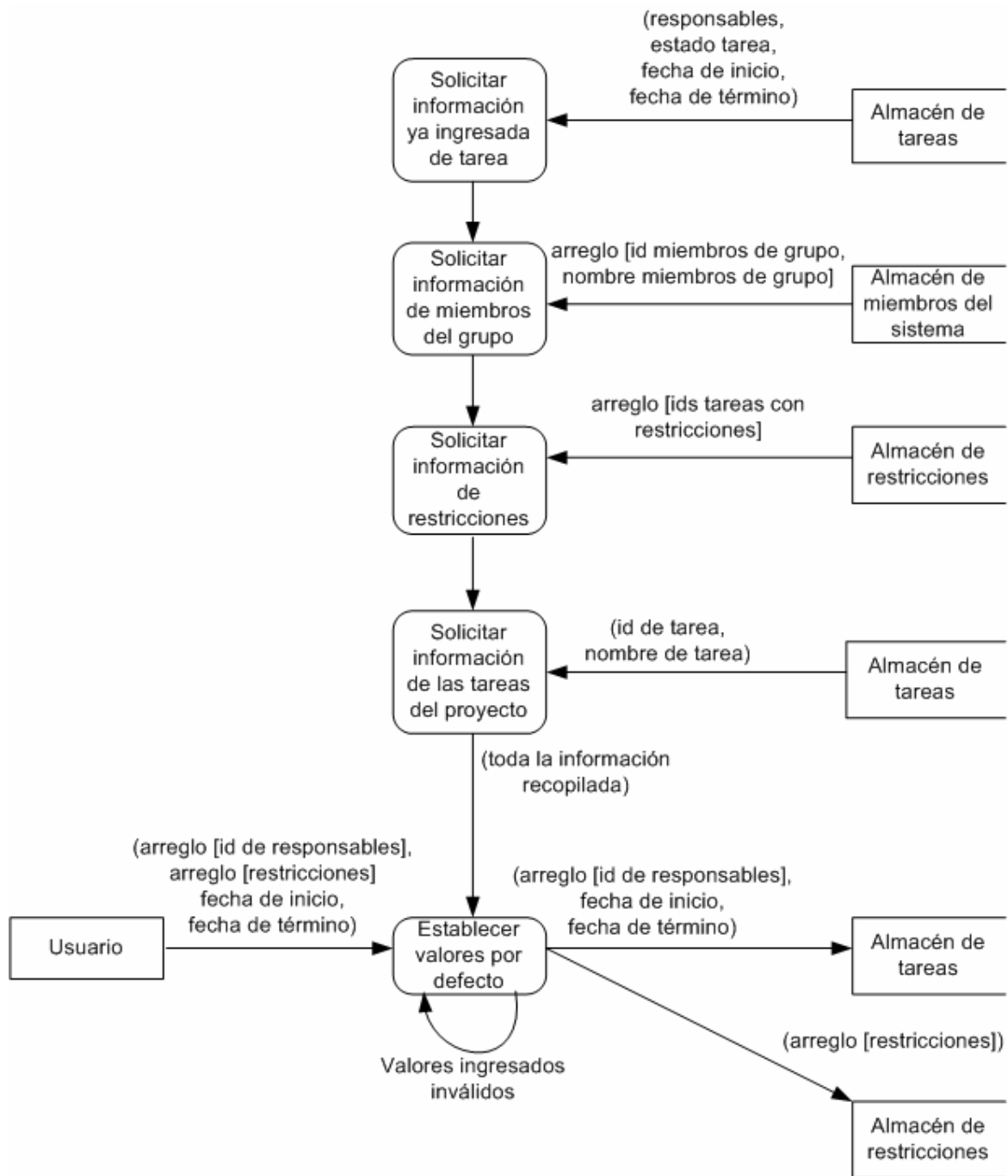


Ilustración 28: flujo de datos para modificar tareas

Cuando el usuario desea modificar la información que ha sido ingresada para una tarea, esta se debe obtener de las tablas que almacenan las tareas y las restricciones, junto con la información de los miembros del grupo creador de la tarea. Con estos datos, se presenta el formulario al usuario, llenado con la información previamente ingresada de forma que el usuario modifique sólo lo necesario. Si la información ingresada por el usuario es correcta, ésta es almacenada en las tablas de tareas y restricciones.

11- Implementación de la aplicación

11.1 Análisis costo efectividad de herramientas

Se comenzó el trabajo eligiendo las herramientas con las cuales se llevaría a cabo la construcción de los servicios que ofrecería la aplicación. Estos servicios, como se vio en el capítulo 8.4, están relacionados con ciertos recursos que ya fueron elegidos, por lo que la búsqueda de estas herramientas estará centrada en esas áreas. Para facilitar la elección se realizó un análisis costo-efectividad entre las alternativas encontradas, tomando en cuenta las prestaciones que éstas deben ofrecer. A estas herramientas se les asignará un puntaje según cumplan o no con las características deseadas. Este puntaje obtiene una clasificación dependiendo del rango en el que se encuentre, según se muestra a continuación.

Tabla 9: rangos de puntaje para características de herramientas

Rango de puntaje	Clasificación
[0,50]	Insuficiente
[51,70]	Regular
[71,85]	Buena
[86,100]	Excelente

Es importante mencionar que todas las herramientas que se elijan deben ser de código abierto y licencia gratuita. Es necesario que sean de código abierto para poder modificarla y adecuarla al sistema que se desea implementar. Se exige que su licencia sea gratuita pues no es posible incurrir en costos extras para adquirir la licencia de una herramienta comercial. El recurso de implementación de código fue elegido en el capítulo 8.4.2, por lo que no será incluido nuevamente aquí. Las características que se desea tengan las herramientas varía según el objetivo que deban cumplir. Se presentan a continuación estas características separadas según el grupo de recursos al que pertenezcan.

11.1.1 Características deseadas para recursos de comunicación

11.1.1.1 Características deseadas para el Chat

Registro de discusiones: es importante que la herramienta de Chat ofrezca mecanismos que permitan almacenar en el equipo de alguna manera las conversaciones que se realizan entre los usuarios.

Salas separadas: es importante que la herramienta elegida permita crear salas en las que de alguna manera se filtre el acceso de los usuarios, de forma de impedir que personas no autorizadas obtengan información sensible a un grupo determinado.

Seguridad: es importante que se muestre un interés del equipo de desarrollo en lograr que la herramienta sea segura. Además se espera que en las opiniones realizadas por usuarios de la herramienta, ya sean en el sitio Web de ésta o en páginas anexas, sean positivas en este aspecto.

Compatibilidad con otras herramientas: es deseable que la herramienta elegida permita adaptarse para funcionar junto a otras herramientas, que pueden pertenecer incluso a otro grupo de recursos.

Existencia de soporte: es deseable que la herramienta cuente con un foro que sea mantenido por el equipo de desarrollo de la aplicación o personal calificado, donde se entregue apoyo a los usuarios con problemas en su uso.

A estas características se asigna un ponderador que será aplicado al puntaje obtenido en cada área, de forma de aplicar la importancia de cada atributo deseado a la elección. Estos ponderadores se muestran a continuación.

Tabla 10: características para Chat

Característica	Ponderador
Registro de discusiones	25%
Salas separadas	25%
Seguridad	25%
Compatibilidad con otras herramientas	15%
Existencia de soporte	10%

Las características más importantes que se necesitan del Chat son la capacidad de registrar las conversaciones y que ofrezca salas separadas. Además como la aplicación será utilizada por alumnos de computación, es importante que posea un buen nivel de seguridad para evitar posibles ataques. La compatibilidad con otras herramientas es deseable pero no vital, debido a que al ser de código abierto, éste puede modificarse para permitir una mejor adaptación al sistema.

11.1.1.2 Características deseadas para el foro

Acceso a áreas diferenciado: es importante que la herramienta elegida permita crear sub-foros o categorías en las que de alguna manera se filtre el acceso de los usuarios, de forma de impedir que personas no autorizadas obtengan información sensible a un grupo determinado.

Seguridad: es importante que se muestre un interés del equipo de desarrollo en lograr que la herramienta sea segura. Además se espera que en las opiniones realizadas por usuarios de la herramienta, ya sean en el sitio Web de ésta o en páginas anexas, sean positivas en este aspecto.

Compatibilidad con otras herramientas: es importante que la herramienta elegida permita adaptarse para funcionar junto a otras herramientas, que pueden pertenecer incluso a otro grupo de recursos.

Existencia de soporte: es importante que la herramienta cuente con un foro que sea mantenido por el equipo de desarrollo de la aplicación o personal calificado, donde se entregue apoyo a los usuarios con problemas en su uso.

Los ponderadores para estas características son los siguientes.

Tabla 11: características para foro

Característica	Ponderador
Acceso a áreas diferenciado	25%
Seguridad	25%
Compatibilidad con otras herramientas	25%
Existencia de soporte	25%

Debido a que los sistemas de foros son herramientas complejas y de gran tamaño en los que es posible que muchos factores sean críticos, se otorgó a todas las características necesarias el mismo peso.

11.1.2 Características deseadas para recursos de almacenamiento

11.1.2.1 Características deseadas para la base de datos

Velocidad en transacciones: es importante que la velocidad de las transacciones sea alta, debido a que el número de accesos puede aumentar mucho en períodos en que los alumnos deban cumplir con fechas de término de proyectos.

Seguridad: es importante que se muestre un interés del equipo de desarrollo en lograr que la herramienta sea segura. Además se espera que en las opiniones realizadas por usuarios de la herramienta, ya sean en el sitio Web de ésta o en páginas anexas, sean positivas en este aspecto.

Compatibilidad con otras herramientas: es importante que la herramienta elegida permita adaptarse para funcionar junto a otras herramientas, que pueden pertenecer incluso a otro grupo de recursos.

Existencia de soporte: es importante que la herramienta cuente con un foro que sea mantenido por el equipo de desarrollo de la aplicación o personal calificado, donde se entregue apoyo a los usuarios con problemas en su uso. Además debe poseer documentación referente a las partes importantes del sistema.

Los ponderadores para estas características son los siguientes.

Tabla 12: características para base de datos

Característica	Ponderador
Velocidad en transacciones	30%
Seguridad	20%
Compatibilidad con otras herramientas	30%
Existencia de soporte	20%

Debido a que la base de datos almacenará toda la información del sistema, es importante que esta sea compatible con todas las herramientas que se utilicen. La velocidad en las transacciones es otro factor importante, por lo que su ponderador refleja este hecho.

11.1.3 Características deseadas para recursos de representación

11.1.3.1 Características deseadas para la herramienta de representación

Compatibilidad con otras herramientas: es deseable que la herramienta elegida permita adaptarse para funcionar junto a otras herramientas, que pueden pertenecer incluso a otro grupo de recursos.

Existencia de soporte: es importante que la herramienta cuente con un foro que sea mantenido por el equipo de desarrollo de la aplicación o personal calificado, donde se entregue apoyo a los usuarios con problemas en su uso. Además debe poseer documentación referente a las funcionalidades que posea la herramienta.

Los ponderadores para estas características son los siguientes.

Tabla 13: características para herramienta de representación

Característica	Ponderador
Compatibilidad con otras herramientas	50%
Existencia de soporte	50%

11.1.4 Análisis para los recursos de comunicación

11.1.4.1 Análisis para Chat

Entre las opciones encontradas se destacan “phpFreeChat”, “phpOpenChat” y “X7 Chat”.

phpFreeChat: ofrece la posibilidad de poder conversar con otras personas en tiempo real, sin embargo la forma en la que son almacenadas las conversaciones no ofrece funcionalidad ni seguridad alguna, dado que éstas se guardan en formato de texto plano. Además limita la conversación a una sala general para todos los usuarios, lo que no otorga privacidad entre grupos. Soporta conversaciones privadas entre dos usuarios. Su documentación era relativamente simple y su sitio Web posee un foro de soporte con poca actividad. Se puede integrar fácilmente en un sitio Web debido a que no interfiere con otros componentes del sistema.

phpOpenChat: ofrece diferentes canales de conversación y almacenamiento de las discusiones en una base de datos, sin embargo al momento de escribir este informe el código no era mantenido desde hace dos años [38]. Aún así, éste demostraba preocupación por la seguridad del sistema. Posee un foro de discusión en alemán, pero no en la página oficial de la herramienta. La documentación de la herramienta también se encuentra en alemán.

x7 Chat: se encuentra en desarrollo constante. Posee un foro de discusión con gran actividad y una buena cantidad de documentación. Su código demuestra una gran preocupación por la seguridad de la aplicación, así como de las prestaciones que ofrece. Permite discutir en salas separadas, ya sean públicas o privadas. Ofrece la posibilidad de almacenar las conversaciones, además de en una base de datos, en archivos de texto plano. Además es posible integrarla con distintas herramientas como sistemas de foros y administradores de contenido.

En base a estas herramientas se realizó la siguiente asignación de puntaje.

Tabla 14: análisis de características para alternativas de Chat

Aplicación		phpFreeChat		phpOpenChat		x7 Chat	
Característica	Peso	Puntaje	Total	Puntaje	Total	Puntaje	Total
Registro de discusiones	25%	55	13	80	20	90	22
Salas separadas	25%	30	7	85	21	85	21
Seguridad	25%	30	7	80	20	90	22
Compatibilidad	15%	80	12	80	12	95	14
Soporte	10%	60	6	40	4	75	7
Total	100%		45		77		86

Debido al análisis anterior, la herramienta elegida para ser utilizada como Chat corresponde a x7 Chat.

11.1.4.2 Análisis para foro

Las alternativas más sobresalientes por su soporte a los usuarios y prestaciones otorgadas correspondían a “phpBB” y “SMF”.

phpBB: las opiniones encontradas referentes a esta opción la destacan como una alternativa muy insegura y vulnerable a ataques [39]. Posee gran cantidad de documentación y un foro muy activo. Es posible integrarla con la herramienta de Chat elegida. Permite crear acceso diferenciado a secciones del foro.

SMF: es visto como una herramienta de foro muy segura [40]. Posee gran cantidad de documentación y un foro muy activo. Es posible integrarla con la herramienta de Chat elegida. Permite crear acceso diferenciado a secciones del foro.

En base a estas herramientas se realizó la siguiente asignación de puntaje.

Tabla 15: análisis de características para alternativas de foro

Aplicación		phpBB		SMF	
Característica	Peso	Puntaje	Total	Puntaje	Total
Acceso a áreas diferenciado	25%	90	22	90	22
Seguridad	25%	60	15	80	20
Compatibilidad con otras herramientas	25%	90	22	90	22
Existencia de soporte	25%	95	23	95	23
Total	100%		82		87

Debido a este análisis, se eligió la herramienta SMF como foro para la aplicación.

11.1.5 Análisis para los recursos de almacenamiento

11.1.5.1 Análisis para la base de datos

Entre las alternativas gratuitas, las que cuentan con mayor soporte corresponden a PostgreSQL y MySQL.

PostgreSQL: se destaca por su gran cantidad de prestaciones [41]. En su sitio Web se puede encontrar gran cantidad de documentación, y se presenta como una base de datos segura y confiable.

MySQL: en el sitio Web oficial es presentada como una aplicación segura y que destaca por su rapidez al realizar transacciones [42]. Posee variada documentación y un foro de mensajes en su sitio oficial. Muchas aplicaciones de las encontradas para la elección de recursos de los capítulos anteriores la utilizan como base de datos.

En base a estas herramientas se realizó la siguiente asignación de puntaje.

Tabla 16: análisis de características para alternativas de base de datos

Aplicación		PostgreSQL		MySQL	
Característica	Peso	Puntaje	Total	Puntaje	Total
Velocidad en transacciones	30%	80	24	90	27
Seguridad	20%	90	18	90	18
Compatibilidad con otras herramientas	30%	30	9	90	27
Existencia de soporte	20%	95	19	95	19
Total	100%		70		91

En base al análisis de estas características, MySQL se presenta como un mejor sistema de base de datos para la aplicación que PostgreSQL.

11.1.6 Análisis para los recursos de representación

11.1.6.1 Análisis para las herramientas de representación

Se encontraron dos herramientas para esto, “Gantt Chart Class” y “jpGraph”.

Gant Chart Class: ofrece una función, escrita en php, a la cual se le entregan los parámetros necesarios para dibujar el gráfico, el cual es generado en formato png. Esta alternativa no ofrece ningún tipo de documentación y no es mantenida en la actualidad [43].

jpGraph: corresponde a un conjunto de clases y funciones escritas en php que permiten crear distintos tipos de gráficos aparte del de carta Gantt. En su sitio oficial se puede encontrar la documentación de la aplicación, así como un foro de mensajes con actividad media. Es mantenido en la actualidad.

Tabla 17: análisis de características para herramientas de representación

Aplicación		Gant Chart Class		jpGraph	
Característica	Peso	Puntaje	Total	Puntaje	Total
Compatibilidad con otras herramientas	50%	90	45	90	45
Existencia de soporte	50%	0	0	70	35
Total	100%		45		80

Por el análisis aquí presentado, se eligió la herramienta jpGraph para la realizar la representación de los gráficos.

11.1.7 Herramientas de apoyo a la aplicación

Junto con las herramientas elegidas en los capítulos anteriores, se trabajará con las siguientes aplicaciones debido a su relación con los recursos elegidos, y a las ventajas que éstas ofrecen comparadas con un ambiente de programación que no las incluye.

Tabla 18: herramientas de apoyo a la aplicación

Nombre herramienta	Versión	Descripción
Apache	2.2.3	Servidor HTTP.
smarty	2.6.16	Herramienta que permite crear páginas html con código propio de smarty, creando con esto plantillas de páginas para facilitar la muestra de las consultas realizadas a la base de datos.
PEAR	1.4.14	Conjunto de paquetes desarrollados en php que ofrecen características listas para ser ingresadas en otras aplicaciones.
HTML_Common	1.2.3	Base para otras clases HTML, perteneciente a PEAR.
HTML_Quickform	3.2.7	Permite la creación de formularios de forma eficiente, utilizando reglas de validación predefinidas. Perteneciente a PEAR. Dependiente de HTML_Common.
phpMyAdmin	2.9.2	Herramienta de administración gráfica para base de datos MySQL.
Windows XP	Profesional, Versión 2002, Service Pack 2	Sistema operativo
Dtree	2.05	Herramienta para representar vistas de contenidos en forma de árbol.

11.1.7.1 Justificación de elección de herramientas de apoyo a aplicación

Apache 2.2.3: se eligió esta herramienta debido a que su uso está bien documentado tanto de su página oficial [44] como en la página del lenguaje que se utilizará [45]. Además existen tutoriales en la red acerca de las diferentes herramientas que posee php que consideran como servidor Web a Apache. Es una herramienta gratuita, a diferencia de otras alternativas como

Microsoft IIS y obtiene actualizaciones de forma más frecuente que servidores como OmniHTTPd [46].

smarty 2.6.16: se utilizará para lograr separar el código de la aplicación de la capa de presentación, de forma que el proceso de diseño no afecte las funcionalidades ya creadas. Además, los errores en el código de la aplicación serán reportados por el intérprete de php, mientras que los errores de la capa de presentación serán reportados por smarty, facilitando su detección y resolución.

PEAR 1.4.14: dado que se trabajara con php como lenguaje de programación, se utilizará este repositorio de herramientas para obtener paquetes que puedan facilitar la programación de ciertas funcionalidades de la aplicación.

HTML_Quickform 3.2.7: para facilitar la implementación de los cuestionarios se consideró utilizar este paquete de PEAR, que ofrece métodos para crear, validar, y procesar formularios html [47].

phpMyAdmin 2.9.2: dado que se utiliza MySQL, el administrador gráfico de bases de datos recomendado en diferentes sitios corresponde a phpMyAdmin. Este administrador ofrece una visualización clara de las tablas y los datos que contiene, a diferencia de otras alternativas para bases de datos PostgreSQL como pgAdmin III.

Windows XP Profesional, Versión 2002, Service Pack 2: se utilizará este sistema debido a la facilidad que ofrece el instalar y mantener algunas de las herramientas en esta plataforma, a pesar de las ventajas en cuanto a uso de recursos que presentan otros sistemas como las plataformas basadas en Unix. Además ofrece ventajas con los procesos que se ejecutan en diferentes hilos al utilizar el servidor HTTP Apache [48].

Dtree 2.05: para facilitar la representación de los proyectos y sus tareas dentro del sistema, se optó por representarlos mediante una estructura de árbol desplegable mediante esta herramienta. Fue elegida frente a otras alternativas debido a su sencilla API y la facilidad con que se puede personalizar.

Para cada una de las herramientas listadas anteriormente, se eligió la última versión estable liberada y recomendada por sus creadores, debido a que estas contienen mejoras en la seguridad y ejecución de su código.

Con la selección de herramientas inicial, el siguiente paso fue la instalación de las mismas en la máquina de desarrollo.

11.2 Instalación de herramientas iniciales

Para instalar Apache 2.2 se descargó el archivo “apache_2.2.3-win32-x86-no_ssl.msi” del sitio del servidor Web Apache [44], instalándolo como un servicio para todos los usuarios, escuchando en el puerto 80. Los datos necesarios referentes al nombre del servidor se completaron con el nombre “localhost”, dado que el sistema se mantendrá en un equipo de desarrollo. Se debió especificar una dirección de correo electrónico para el administrador del sistema, en este caso “me@localhost”, dado que si no se especificaba Apache no podía inicializarse. Se efectuó una instalación personalizada que incluyera todos los componentes del servidor en el directorio “c:/Apache2/” para evitar problemas que pueda ocasionar tener

“PEAR”. La versión 5.2.0 de php incluía una versión defectuosa de este archivo [51], por lo que se debió descargar una versión corregida desde el sitio oficial de PEAR [50]. Luego de esta descarga, se procedió a ejecutar “go-pear.bat”, el cual solicitó que se confirmaran una serie de directorios en los que se instalarían los archivos de PEAR. Se mantuvieron los directorios especificados por defecto, modificando el directorio de instalación del archivo “pear.ini” desde la carpeta “c:\Windows” a la carpeta donde se encuentra instalado php. El instalador de PEAR agregó a la variable “include_path” especificada en el archivo “php.ini” la ruta a la carpeta de instalación de los archivos de pear “php\pear”. Además el instalador creó un archivo llamado “PEAR_ENV.reg”, que debía ser ejecutado para agregar variables de ambiente al sistema que indican los directorios de instalación de PEAR. Luego de su ejecución, el archivo “pear.bat” que debería haberse creado no se encontraba, lo que ha sido detectado como un bug que es solucionable renombrando el nombre de un archivo creado llamado “pear.bat_old” a “pear.bat” [51].

Luego de haber instalado PEAR, se instaló el paquete “html_quickform” mediante el comando “pear install html_quickform”, incluyendo la dependencia “html_common” necesaria.

La instalación del foro se realizó tras descomprimir los contenidos del archivo “smf_1-1-2_install.zip” en el directorio “foro” ubicado en el directorio “htdocs” de Apache. Junto con este archivo, se descomprimió también el archivo “smf_1-1-2_spanish-utf8.zip” en la carpeta “Themes” del directorio “foro”, que contiene los archivos necesarios para que el foro sea mostrado en español. Tras esto, se abrió desde un navegador Web la página de configuración “install.php”, que permitió configurar los parámetros bajo los cuales funcionará el foro y sus tablas en la base de datos. Estas tablas fueron almacenadas en la base de datos llamada “smf”, y sus nombres están acompañados del prefijo “smf_”.

Para instalar el Chat se creó una carpeta llamada “Chat” dentro del directorio que contiene al foro. Dentro de esa carpeta se descomprimieron los contenidos del archivo “x7chat2_0_5.zip”, y se abrió desde el navegador Web la página de configuración “install.php”. Desde esta página se configuró la aplicación de forma que utilizara la misma base de datos que el foro, donde todas las tablas que tienen relación con la herramienta de Chat comienzan su nombre con el prefijo “x7chat2_”.

Con los pasos recién mencionados, quedaron instaladas todas las aplicaciones necesarias para comenzar la implementación del sistema.

Un detalle importante es que inicialmente se había considerado el uso de la librería ADOdb, que permite realizar llamadas a la base de datos sin importar que tipo de base de datos se esté utilizando, abstrayéndose de los métodos propios de cada sistema. Sin embargo todas las herramientas consideradas (salvo phpOpenChat) utilizaban como motor de base de datos a MySQL, y el traslado de sus implementaciones a la librería ADOdb, si bien no imposible de realizar para algunas de sus funcionalidades básicas, traería graves problemas para llamadas a funciones exclusivas de MySQL que no puedan ser abstraídas de su implementación. Debido a esto, se decidió eliminar la librería ADOdb de la aplicación y utilizar MySQL como motor de base de datos, considerando que muchas de las aplicaciones encontradas a lo largo de esta búsqueda para las herramientas de discusión y otras partes de la aplicación lo utilizan [52].

11.3 Implementación del sistema

11.3.1 Modelo de datos

El modelo de datos a utilizar se diseño inicialmente tomando en cuenta los siguientes principios.

- Un alumno debe tener un identificador único en la base de datos, su primer nombre y sus dos apellidos, y una contraseña que será utilizada para ingresar a la aplicación.
- Un proyecto debe tener un identificador único en la base de datos, un nombre, una descripción de su objetivo, así como una fecha de inicio y una fecha de término para calcular el tiempo de desarrollo restante y otros indicadores.
- Las distintas tareas que se desprendan de un proyecto deben tener cada una un identificador único, el identificador del proyecto al que pertenece, un nombre, una descripción del estado de avance de la tarea, la fecha de inicio, la fecha de término, y el porcentaje estimado de avance de la tarea.

Cada uno de los principios recién mencionados puede ser resumido en una tabla independiente. Para poder relacionar a un alumno a un proyecto se crearon llaves foráneas en la tabla alumno, de modo que se pudiera obtener directamente desde esa tabla la información de que alumnos pertenecen a un determinado proyecto.

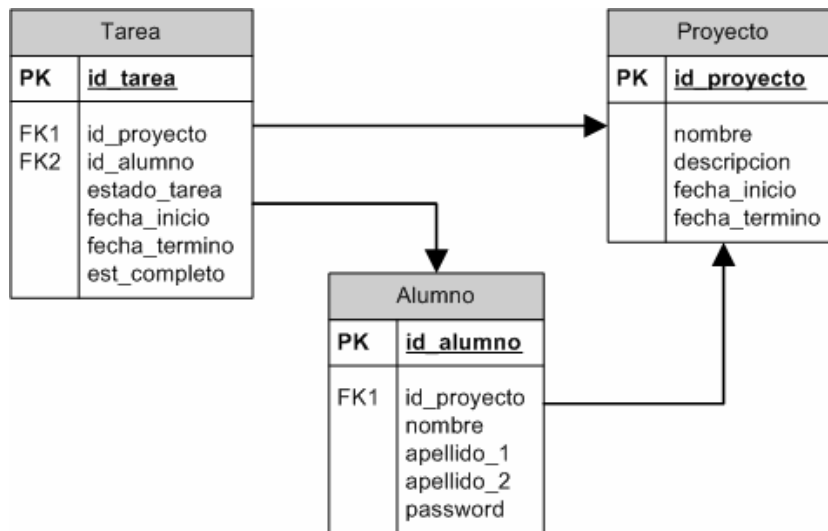


Ilustración 29: Modelo inicial de datos

La creación de llaves foráneas en MySQL se logra creando un índice en el campo que se desee utilizar como llave foránea e indicando a que elemento se desea referenciar. A continuación se muestra una tabla con las llaves foráneas utilizadas.

Tabla 20: llaves foráneas iniciales

Tabla con llave foránea	Llave foránea	Tabla referenciada	Referencia a	Acción a tomar "ON UPDATE"	Acción a tomar "ON DELETE"
alumno	id_proyecto	proyecto	id_proyecto	CASCADE	NO ACTION
tarea	id_alumno	smf_members	id_member	CASCADE	CASCADE

tarea	id_proyecto	proyecto	id_proyecto	CASCADE	NO ACTION
-------	-------------	----------	-------------	---------	-----------

Un posterior análisis al modelo de datos mostró que incorporar la llave foránea “id_proyecto” a la tabla “alumno”, con “id_proyecto” utilizando un valor entero auto incremental, generaba muchas filas con datos repetidos sobre los alumnos en el caso de que un alumno perteneciera a varios proyectos a la vez.

Para evitar este comportamiento, se optó por eliminar el campo “id_proyecto” de la tabla “alumno”. Al realizar esta modificación, la tabla “alumno” terminó con campos similares a los contenidos por la tabla “smf_members”, que almacena la información acerca de los alumnos y equipo docente registrados en el foro de discusión. Debido a esto se decidió utilizar la tabla “smf_members” para controlar las cuentas de los usuarios del sistema y eliminar la tabla “alumno”, agregando a la tabla “proyecto” el identificador del grupo que creó la tarea. De esta forma, un proyecto no estará asociado exclusivamente a un alumno sino al grupo al que pertenece. Es importante notar que con esto el sistema de autenticación de usuarios a utilizar correspondería al del foro. Además se debe recordar que dentro del foro es posible crear grupos de usuarios, y que se puede asignar foros exclusivos a estos grupos de forma que sólo puedan ser vistos por los grupos autorizados por los administradores. Así, el registro de los grupos de usuarios del sistema queda también dentro de las tablas contenidas por el foro. Esto otorga un gran beneficio al no tener que mantener información repetida dentro de la base de datos, y simplificar el proceso de registro y autenticación de los usuarios. Sin embargo, con esto se genera mayor cohesión entre las componentes del sistema, lo que no siempre es deseable. Sin embargo los beneficios recién presentados se consideran mayores a los posibles problemas que podría traer, por lo que se modificó el modelo de datos para representar estos cambios.

Junto con el cambio anterior, se agregó a la tabla tarea el identificador del grupo al que pertenece, junto con una lista que contiene los identificadores de los miembros responsables de la tarea y un registro que almacene las fechas de inicio y término que fueron propuestas inicialmente para esa tarea.

Para poder almacenar las restricciones entre tareas se creó una tabla que contendrá los identificadores de las tareas que están relacionadas, junto con el identificador del tipo de restricción que las une, según se especifica en la documentación de jpGraph para la representación de cartas Gantt.

Para poder almacenar los cambios que se realizaran en las tareas y proyectos se creó una tabla para cada caso respectivamente, que contiene el identificador de la tarea o proyecto, el nombre del alumno que realizó la modificación, la fecha en la que éste se realizó y la descripción del cambio realizado según sea ingresado por el alumno.

Para almacenar las estimaciones realizadas por los alumnos se crearon 4 tablas diferentes. La primera almacenará el identificador de la sala de estimación, las opciones de la sala, el número de iteraciones que se han realizado en esa sala y el tipo de estimación que se realizará ahí (estimación de fechas o de propuestas). La siguiente tabla almacenará el contexto de la estimación expresado por los alumnos para cada iteración que se realice. La tercera y cuarta tabla almacenarán las estimaciones y argumentaciones realizadas por los alumnos para cada iteración de la estimación, según sean estas estimaciones de fechas o de propuestas.

Finalmente, también fue necesario crear una tabla que almacenará las distintas definiciones que formarán parte del glosario.

Con todos los cambios recién mencionados, el modelo de datos final tiene la siguiente forma (no se incluyen tablas ni campos utilizados internamente por las herramientas de foro o Chat).

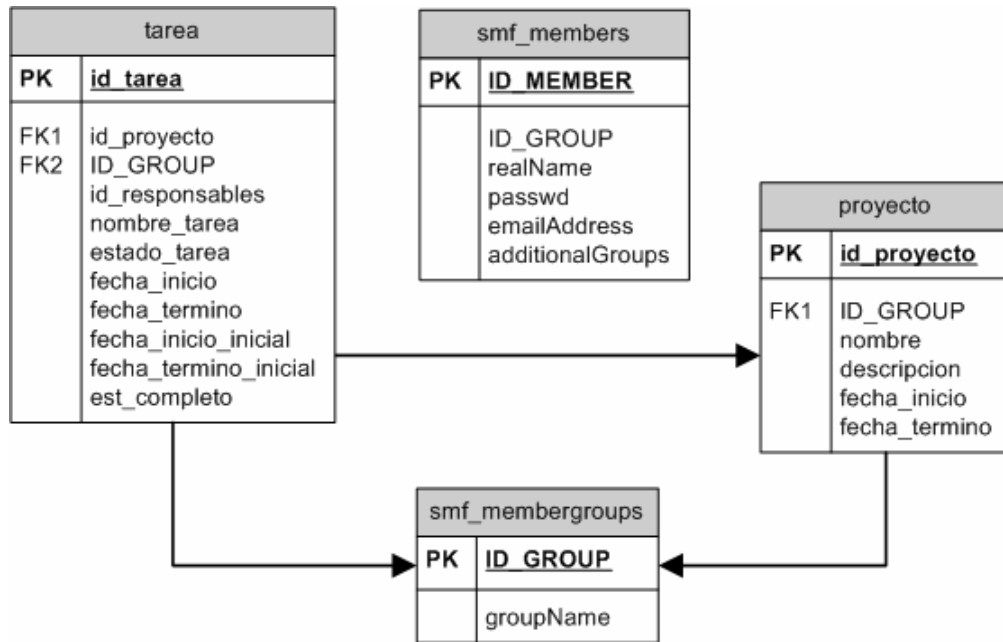


Ilustración 30: modelo de datos para tablas relacionadas con cuentas de usuario

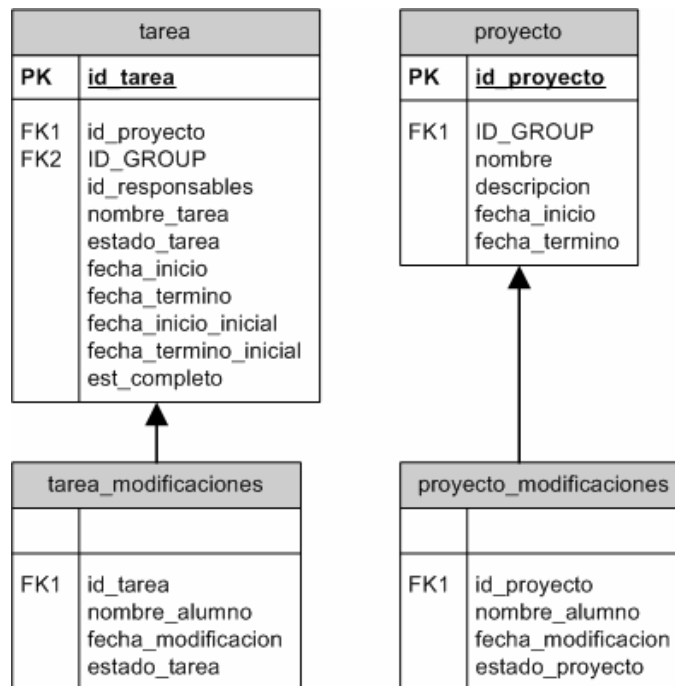


Ilustración 31: modelo de datos para tablas relacionadas con modificaciones

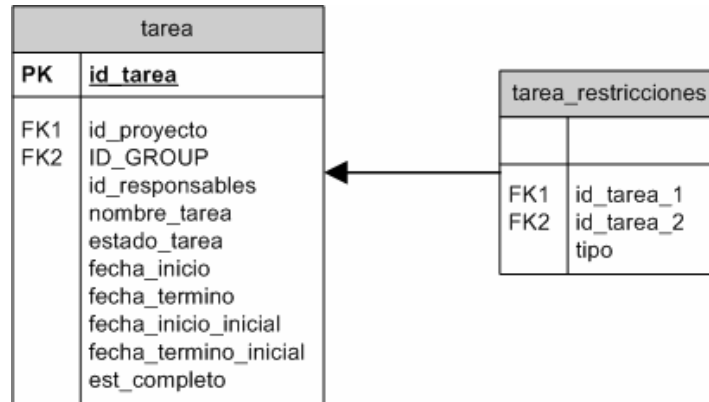


Ilustración 32: modelo de datos para tablas relacionadas con modificaciones

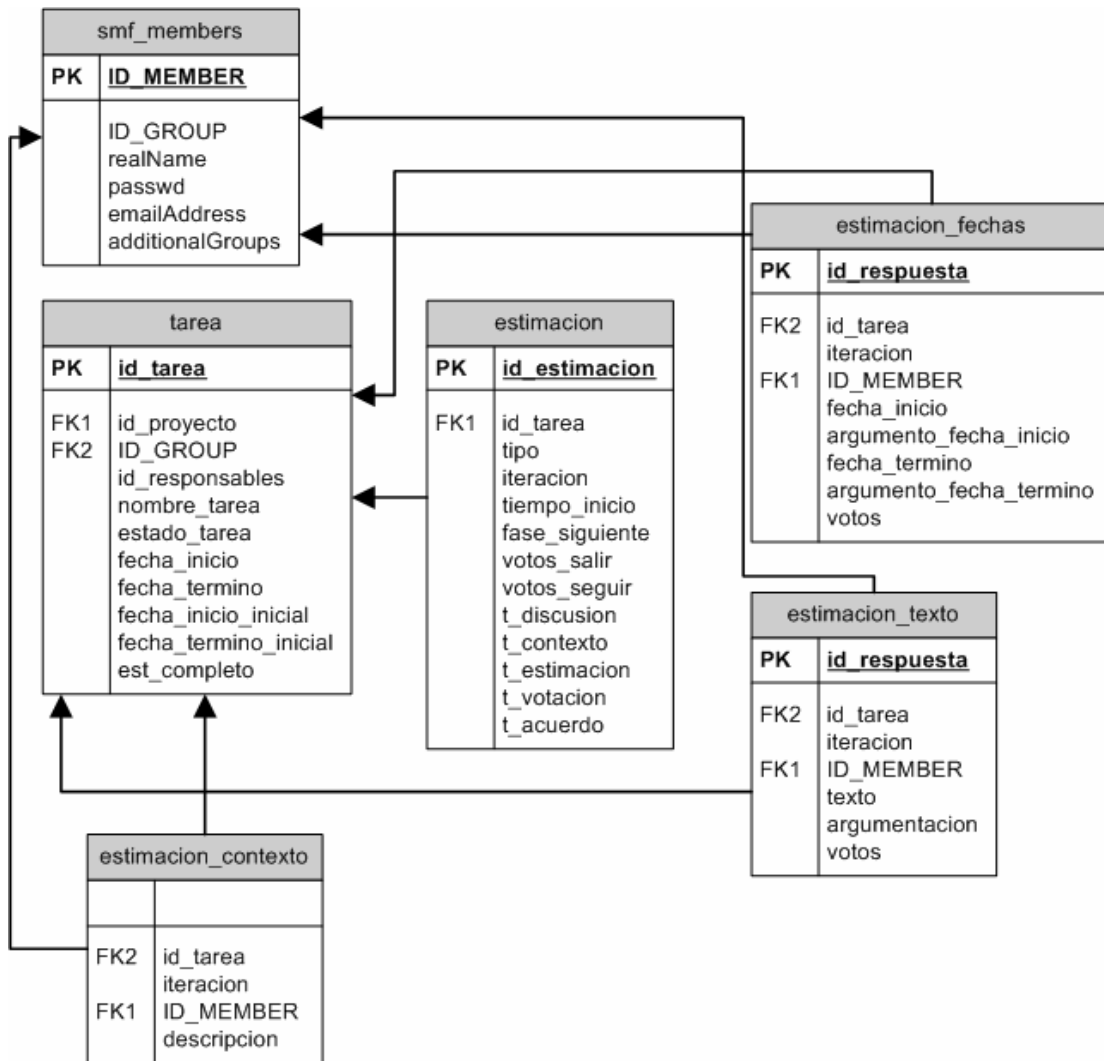


Ilustración 33: modelo de datos para tablas relacionadas con estimaciones

glosario	
PK	<u>id_concepto</u>
	concepto definicion

Ilustración 34: modelo de datos para tablas relacionadas con el glosario

Las llaves foráneas en el modelo final corresponden a las expresadas en la siguiente tabla.

Tabla 21: llaves foráneas del modelo

Tabla con llave foránea	Llave foránea	Tabla referenciada	Referencia a	Acción a tomar "ON UPDATE"	Acción a tomar "ON DELETE"
estimacion	id_tarea	tarea	id_tarea	CASCADE	CASCADE
estimacion_contexto	id_tarea	tarea	id_tarea	CASCADE	CASCADE
estimacion_contexto	ID_MEMBER	smf_members	ID_MEMBER	CASCADE	CASCADE
estimacion_fechas	id_tarea	tarea	id_tarea	CASCADE	CASCADE
estimacion_fechas	ID_MEMBER	smf_members	ID_MEMBER	CASCADE	CASCADE
estimacion_texto	id_tarea	tarea	id_tarea	CASCADE	CASCADE
estimacion_texto	ID_MEMBER	smf_members	ID_MEMBER	CASCADE	CASCADE
proyecto	id_group	smf_members	ID_GROUP	CASCADE	CASCADE
proyecto_modificaciones	id_proyecto	proyecto	id_proyecto	CASCADE	CASCADE
Tarea	id_grupo	smf_members	ID_GROUP	CASCADE	CASCADE
Tarea	id_proyecto	proyecto	id_proyecto	CASCADE	CASCADE
tarea_modificaciones	id_tarea	tarea	id_tarea	CASCADE	CASCADE
tarea_restricciones	id_tarea_1	tarea	id_tarea	CASCADE	CASCADE
tarea_restricciones	id_tarea_2	tarea	id_tarea	CASCADE	CASCADE

11.3.1.1 Consideraciones al modelo de datos

Se tomaron las siguientes consideraciones al implementar el modelo de datos.

11.3.1.1.1 Llaves foráneas

Los campos que son llaves foráneas poseen como acción a realizar en caso de borrado o cambio en los datos la actualización en cascada. Esto con el fin de evitar problemas de consistencia en los datos. Así por ejemplo, al borrar una tarea, inmediatamente serán eliminados todos los registros de las estimaciones o modificaciones relacionadas con ella.

11.3.1.1.2 Tipos de datos

Se modificó el tipo inicial de la tabla "smf_members" al tipo InnoDB, dado que éste es el único formato que soporta llaves foráneas en MySQL. Esto se hizo inicialmente para poder relacionar las cuentas existentes de los usuarios y sus grupos en el foro con los creadores de las tareas. A

todas las tablas creadas que no pertenecen ni al foro ni al Chat se les dio este formato.

11.3.1.1.3 Consultas a la base de datos

Para realizar las consultas a la base de datos se optó por utilizar la función “db_query” incluida en la implementación del foro. Esta realiza un trabajo de limpieza sobre el string que recibe como consulta para evitar posibles ataques, además de utilizar una conexión única con la base de datos.

11.3.2 Permisos

Se deshabilitó el registro de cuentas nuevas en el foro y Chat para evitar que los alumnos creen cuentas de forma maliciosa. El registro de alumnos al foro debe realizarlo una persona del equipo docente del curso.

11.3.3 Modificaciones a código de herramienta de Chat

Fue necesario modificar el código de la herramienta “X7 Chat” para adaptarla a las necesidades de la aplicación. La siguiente tabla muestra una lista con las modificaciones realizadas.

Tabla 22: modificaciones a herramienta de Chat

Archivo modificado	Razones de la modificación
sources\ frame.php	Este archivo contiene la información necesaria para mostrar la sala de Chat, los usuarios que están en ella y el control de los mensajes enviados. Se modificó de forma que la interfaz ocultara las opciones que permitirían a un usuario ingresar a las salas de conversación de otros grupos u obtener control sobre las que tenga permiso de acceso.
sources\ newroom.php	Inicialmente se almacenaba sólo el identificador de la persona que creó una sala de Chat. Se modificó para permitir almacenar el grupo al que pertenece la persona que creó la sala de Chat.
lib\ auth\ smf.php	Inicialmente la herramienta de Chat y el foro asignan a los usuarios grupos que no son los mismos en ambas aplicaciones. Se modificó para que al agregar un usuario a un grupo desde la pizarra de mensajes, al momento de ingresar al Chat se le asigne ese mismo grupo automáticamente.
lib\ db\ mysql.php	Inicialmente el programa no muestra los errores internos que ocurran con MySQL. Se modificó para que los errores aparezcan en pantalla mientras dure el período de desarrollo.
themes\ x7chat2_alpha\ theme.data	Contiene la información de presentación del Chat, como son fuentes a utilizar y los colores de pantalla. Se modificó este archivo para unificar su apariencia con el sitio.
sources\ roomlist.php	Se modificó para que no se mostrara un link de salida del sistema dentro del Chat, dado que al estar dentro de la aplicación es el foro quien controla el ingreso y salida de los usuarios del sistema. También se eliminaron los links que permiten ver el listado de salas o la lista de miembros.

sources\ usercp.php	Se modificó para evitar que aparezca el listado de salas aparezca en el panel de control del usuario.
sources\ admin.php	Se modificó para evitar que aparezca el listado de salas aparezca en el panel de control del administrador.
sources\ roomcp.php	Se modificó para evitar que aparezca el listado de salas aparezca en el panel de control de la sala.
lang\ espanol.php	Se cambiaron las Ñ y ñ por sus códigos html para evitar tener problemas al usarlas en el Chat.
lib\ message.php	Se modificó para almacenar correctamente los mensajes en la base de datos codificados según UTF-8.

11.3.4 Modificaciones a código de herramienta de foro

Dado que la herramienta de foro controla las cuentas de usuario y el ingreso de estos a la aplicación, fue necesario modificarlo de forma que se integrara correctamente con los demás componentes. El archivo SSI.php incluido ofrecía distintas funciones que facilitaron la unión de estas herramientas.

Tabla 23: modificaciones a herramienta de foro

Archivo modificado	Razones de la modificación
Themes\ default\ index.template.php	Se eliminó el link del template principal del foro que permite salir de éste. Se realizó esto dado que el foro se ve dentro de la aplicación, por lo que se podría haber abandonado el foro y haber cerrado su sesión sin haber salido de la aplicación correctamente.
Sources\ Subs- Members_2.php	Se creó este archivo para manejar el registro de los usuarios de forma masiva al sistema, incluyendo en él una versión modificada de la función de registro de usuarios contenida en el archivo Subs-Members.php. Esta versión modificada permite el ingreso del nombre real de los usuarios directamente de un archivo CVS, sin necesidad de que el usuario lo modifique posteriormente desde su panel de control.
Sources\ LogInOut.php	Se modificó el archivo para que al ingresar un usuario al sistema se carguen un conjunto de variables asociadas al usuario, que posteriormente serán utilizadas por la aplicación. Al momento de salir estas variables son eliminadas.
Settings.php	Se modificó para asignar a algunas variables los valores correspondientes a la configuración del sistema.
SSI.php	Se modificó la forma en la que se muestra el link de salida de la aplicación de forma que sea igual al del resto del sitio.

11.3.5 Modificaciones a herramienta de representación

Según el manual de la herramienta, para graficar las restricciones entre dos tareas se debe incluir el tipo de restricción que se desea representar como un string. Sin embargo esto generaba restricciones que no eran graficadas correctamente. Para solucionar esto, se reemplazo el string por el identificador numérico correspondiente entregado en la documentación, generándose así correctamente la línea que une las tareas.

Tabla 24: modificaciones a herramienta de representación

Archivo modificado	Razones de la modificación
jpgraph\src\ jpgraph_gantt.php	Fue modificado para aumentar el ancho y largo máximo por defecto que pueden poseer las cartas Gantt generadas, pues los valores iniciales eran insuficientes.
jpgraph\src\jpg- config.inc.php	Se modificó para evitar que se mostraran los mensajes de error generados por php en caso de que el gráfico no pudiera ser generado.

11.3.6 Modificaciones a herramienta de representación de estructura de árbol

Se cambió su conjunto de íconos iniciales por los desarrollados por los desarrollados por Mark James incluidos en el paquete “Silk icon set 1.3” [53]. Para esto se modificó el archivo “dtree.js” y así como parte de “dtree.css” para darle una apariencia similar a la del resto del sitio.

12- Uso de la aplicación

12.1 Modo administración

Usuarios de este modo: Tendrán acceso a este modo los usuarios identificados como administradores del sistema. Estos deberán corresponder al equipo docente del curso en el que se esté utilizando la herramienta.

Permisos disponibles: Los usuarios que ingresen en este modo poseerán permisos para agregar y eliminar usuarios del sistema, crear y eliminar grupos de trabajo y realizar labores de creación, mantención y moderación de temas en el foro de mensajes. Podrán acceder además a los proyectos y tareas de los demás grupos del sistema para realizar revisiones o modificaciones. Además tendrán acceso a las funciones del modo alumno.

12.1.1 Registro del equipo docente

Los primeros usuarios que deben poseer acceso al sistema corresponden a los miembros del equipo docente. Durante el proceso de instalación del foro de discusión se solicitará que se ingrese una cuenta que corresponderá a la de “Administrador” del foro. Esta cuenta debe asignarse a un miembro del equipo docente que tenga la disponibilidad de ser el responsable de la aplicación durante el semestre. Con esta cuenta, el usuario administrador podrá entonces ingresar a la sección “Herramientas” de la aplicación y desde ahí al foro. Con esto podrá crear las cuentas de los demás miembros del equipo docente desde el panel de administración del foro. Todos ellos deberán ser ingresados al grupo “Administrador”, existente desde la instalación del foro. Este grupo no debe ser eliminado, puesto que es necesario para el correcto funcionamiento del foro, así como de la aplicación en general.

Ilustración 35: registro del equipo docente

12.1.2 Creación de categorías

Una categoría corresponde a un conjunto de foros que poseen un tema en general. Es recomendable crear al menos una categoría, que contenga los foros privados de conversación de cada grupo. Esto puede hacerse en el foro, desde el menú “Foros” del panel de control de un usuario con permisos de administrador.



Ilustración 36: creación de categorías

12.1.3 Creación de foros

Es posible crear foros privados para cada grupo. Esto se puede hacer desde el menú “Modificar foros” del menú “Foros” del panel de control de un usuario con permisos de administrador. Al crear los foros privados para los grupos de alumnos, se recomienda asignarles la categoría creada exclusivamente para contener este tipo de foros. Al momento de crear el foro se debe asignar permiso de acceso sólo al grupo correspondiente a ese foro (en caso de que ya haya grupos existentes), salvo que se desee crear un foro de acceso general. El grupo administrador puede crear un foro de acceso exclusivo para él al no asignar permisos de acceso para ningún grupo, ya que por defecto el grupo administrador tiene acceso a todos los foros.



Ilustración 37: creación de foros

12.1.4 Creación de grupos de usuarios

Antes de ingresar a los alumnos a la aplicación, es necesario crear los grupos de usuarios a los que pertenecerán. Esto puede realizarse en el foro desde la opción “Grupos de usuarios” desde el panel de control de “Administrador”. En este formulario se le solicitará que ingrese el nombre del grupo que desea crear y los foros a los que tendrá acceso. Una vez creados los grupos estos aparecerán listados en el menú “Grupos de usuarios”. Si un administrador desea tener acceso en la aplicación a los proyectos y tareas de los grupos de los alumnos, debe asignarse como miembro de esos grupos.



Ilustración 38: creación de grupos de usuarios

12.1.5 Registro de alumnos

Existen dos formas de ingresar los alumnos al sistema. Desde el foro, se puede ingresar a los alumnos desde la opción “Registro de usuarios” del panel de control de un usuario administrador. Aquí se debe especificar el nombre de usuario, dirección de correo electrónico, una contraseña y el grupo al que pertenece. Si no se ingresa el grupo de usuario al que pertenece el alumno, éste no podrá utilizar la aplicación.

The screenshot shows the 'Registro de Usuarios' (User Registration) form within a forum administration interface. The form is titled 'Registrar nuevo usuario' and contains the following fields and options:

- Nuevo Nombre de usuario:** A text input field for the user's name.
- Nueva Dirección de email:** A text input field for the user's email address, with a note that it is required if the 'Enviar por email' option is selected.
- Contraseña:** A text input field for the user's password.
- Grupo primario:** A dropdown menu currently set to 'Grupo proyecto 1'.
- Enviar por email la nueva contraseña al usuario:** A checked checkbox.
- Pedirle al usuario que active la cuenta:** An unchecked checkbox.
- Registrarse:** A button to submit the registration form.

Ilustración 39: registro de alumnos desde foro

Otra forma de registrar a los alumnos es a través del enlace “Actualizar usuarios” de la sección “Herramientas” de la aplicación. Desde esta página se solicitará a un usuario administrador que ingrese un archivo CSV con los siguientes datos para cada usuario que se desee agregar.

- Nombre de usuario para el alumno a ingresar: por ejemplo, la primera letra de su primer nombre junto a su primer apellido.
- Nombre completo del alumno: el nombre completo es un solo campo en el archivo CSV.
- Identificador de grupo: si no se han asignado los grupos de trabajo al momento de ingresar los alumnos al sistema, se puede especificar un grupo genérico al que pertenezcan todos los alumnos. Todo grupo al que se desee agregar alumnos debe haber sido creado anteriormente desde el panel de control del “Administrador”. El identificador de cada grupo creado puede extraerse del campo “ID_GROUP” de la tabla “smf_membergroups” en la base de datos del foro.
- Correo electrónico del alumno, en caso de no poseer los correos de los alumnos puede ingresarse una dirección genérica modificable más adelante por los alumnos desde su panel de control.

- Contraseña: debe ser asignada por el equipo docente y puede ser repartida inicialmente a cada grupo indicando que deben modificarla. Tras esta modificación la contraseña pasará a ser personal.

Posterior al registro, los alumnos podrán ingresar al sistema.

The screenshot shows a web application interface with a top navigation bar containing 'Estimaciones', 'Proyectos', 'Herramientas', and 'Salir'. On the left, there is a sidebar with 'Vista de herramientas' and links for 'Cuestionario fortalezas', 'Glosario', 'Estimador de complejidad', and 'Ir al foro'. Below this is 'Opciones de administración' with links for 'Actualizar glosario' and 'Actualizar usuarios'. The main content area contains instructions for uploading a CSV file to add users. It states that the first line is for field names and provides an example: 'id_de_usuario;nombre_real_de_usuario;id_grupo_del_usuario;email;password'. A '>Descargar ejemplo<' link is provided. Below the instructions is a form with a label '*Elija el archivo a subir:', a text input field, and an 'Examinar...' button. A 'Subir el archivo' button is also present. A note at the bottom indicates '* denota campo requerido'.

Ilustración 40: registro de alumnos desde aplicación

12.2 Modo alumno

Usuarios de este modo: Tendrán acceso a este modo los usuarios identificados como alumnos del curso en el que se esté utilizando la herramienta. Estos deben haber sido previamente agregados al sistema por un usuario administrador. Los administradores del sistema también podrán acceder a las opciones presentadas en este modo.

Permisos disponibles: Los usuarios que ingresen en este modo poseerán permisos para crear o eliminar proyectos y tareas, modificar la información existente acerca de éstas, realizar estimaciones sobre la duración de las tareas creadas, discutir sobre ellas en la herramienta de Chat, ver registros de conversaciones pasadas y realizar discusiones en el foro.

12.2.1 Vista de mantenimiento

Corresponden a la vista de mantenimiento todas las opciones que guardan relación con la creación, modificación y eliminación de proyectos y tareas. Se accede a este modo desde el enlace "Proyectos" del menú superior derecho.

12.2.1.1 Creación de proyectos

Para agregar un proyecto nuevo se debe elegir la opción "Agregar proyecto nuevo" desde el menú "Proyectos" de la aplicación. Esta opción les permitirá agregar un nuevo proyecto a su lista. Este formulario les solicitará los siguientes datos.

- Nombre del proyecto, con el que será identificado en el menú. Debe ser un nombre corto y comprensible para el grupo.
- Descripción del proyecto, que permitirá recordar al grupo el objetivo real y no desviarse implementando soluciones a problemas inexistentes.

- Fecha tentativa de inicio, puede ser una fecha ya discutida con el grupo, o que será tratada en una reunión posterior.
- Fecha tentativa de término, puede ser una fecha ya discutida con el grupo, o que será tratada en una reunión posterior.
- En caso que el usuario pertenezca a más de un grupo, se le solicitará que elija a que grupo desea asignar el proyecto. Esta decisión es muy importante, puesto que toda tarea que se cree para este proyecto quedará automáticamente asignado al grupo encargado del proyecto.

The screenshot shows a web application window with a title bar containing 'Estimaciones', 'Proyectos', 'Herramientas', and 'Salir'. On the left, there is a sidebar titled 'Vista de mantenimiento' with a tree view containing 'Mis proyectos' and 'Crear proyecto nuevo'. The main area is titled 'Ingreso de nuevo proyecto' and contains the following form elements:

- *Nombre del proyecto: [Text input field]
- Descripción: [Text area]
- Fecha inicio (d/m/a): [01] [01] [2007] (dropdowns)
- Fecha término (d/m/a): [01] [01] [2007] (dropdowns)
- Elija al grupo al que desea asignar el proyecto:
 - Administrador
 - grupo1
 - grupo final
 - grupo2
- [Crear proyecto] button
- * denota campo requerido

Ilustración 41: creación de proyectos

12.2.1.2 Creación de tareas

Una vez que los usuarios hayan creado proyecto, podrán comenzar a agregar tareas a él mediante el elemento del menú “Agregar tarea nueva” que se desprende del proyecto elegido. Este elemento los llevará a un formulario donde se deben especificar los siguientes datos.

- Nombre de la tarea con el que será identificada en el listado de proyectos y tareas. Debe ser un nombre corto y comprensible para el grupo.
- Fecha tentativa de inicio, puede ser una fecha ya discutida con el grupo, o que será tratada en una reunión posterior.
- Fecha tentativa de término, puede ser una fecha ya discutida con el grupo, o que será tratada en una reunión posterior.
- Responsables para la tarea a crear. Estos deben elegirse entre los miembros que pertenezcan al grupo responsable del proyecto al que pertenece la tarea.
- Restricciones de término para tareas que hayan sido creadas anteriormente. Esto modificará la representación de las tareas en las cartas Gantt, indicando que tareas deben estar terminadas para poder comenzar con la tarea actual.

Al crear una tarea se crea su entrada en la lista de proyectos y tareas, una sala de Chat y una sala donde realizar estimaciones referentes a ella.

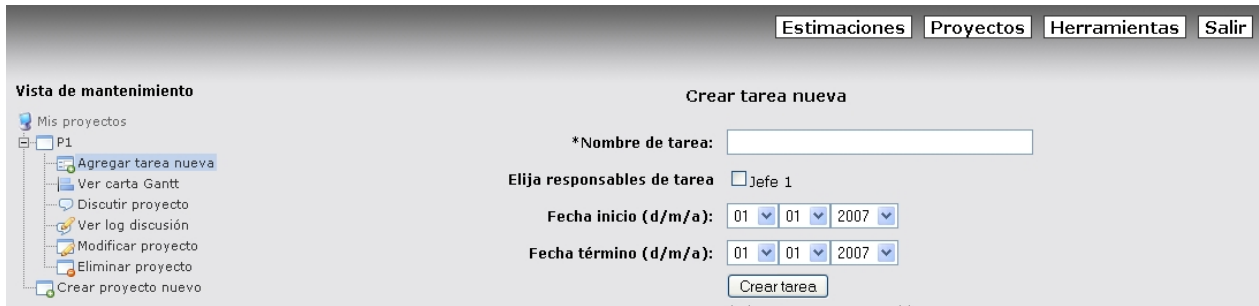


Ilustración 42: creación de tareas

12.2.1.3 Carta Gantt

La opción de carta Gantt mostrará para un proyecto seleccionado un gráfico con todas las tareas que hayan sido ingresadas, indicando su fecha de inicio y término, el porcentaje avanzado, las restricciones entre tareas y las personas encargadas en caso de haberlas.

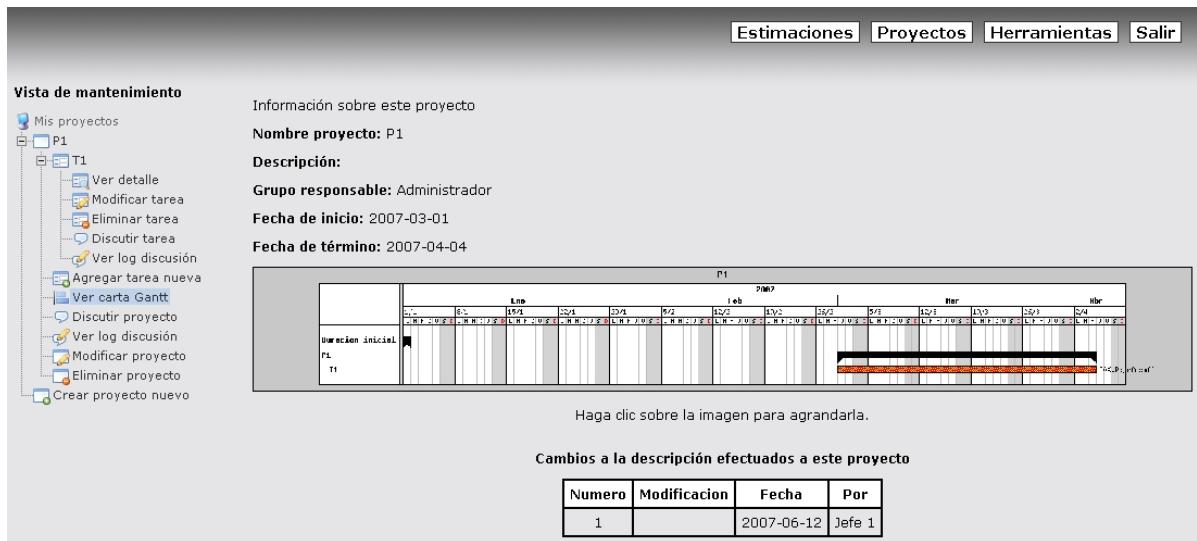


Ilustración 43: vista de carta Gantt

12.2.1.4 Discutir proyecto

La opción “Discutir proyecto” llevará a los alumnos a una sala de Chat donde se podrán realizar discusiones o reuniones acerca del proyecto elegido, así como realizar acuerdos o tomar decisiones.



Ilustración 44: sala de discusión de proyectos

12.2.1.5 Ver log discusión

Cuando los usuarios hayan realizado discusiones en la sala de Chat de un proyecto o tarea, este enlace mostrará el registro de la conversación transcurrida para el proyecto o tarea elegida.



Ilustración 45: vista de log de discusión

12.2.1.6 Modificar proyecto

Si cambia la descripción del proyecto tras una negociación con el cliente, o las fechas de inicio y término debido a un ajuste en la duración de las tareas, es posible modificar los parámetros que definen el proyecto desde esta opción. En este formulario se cargarán por defecto las opciones ingresadas en la última modificación realizada al proyecto.

Ilustración 46: formulario de modificación de proyecto

12.2.1.7 Eliminar proyecto

Si un proyecto deja de ser desarrollado y se desea eliminar toda opción referente a él, puede elegirse esta opción, que borrará del sistema toda tarea relacionada con él, así como las estimaciones que se hayan realizado.

Ilustración 47: formulario para eliminar un proyecto

12.2.1.8 Ver detalle de tarea

Cuando se haya creado una tarea, desde este enlace podrán verse todas las características de la tarea seleccionada, junto con un gráfico que muestre sus fechas de inicio y término, el avance realizado y sus responsables.

Estimaciones Proyectos Herramientas Salir

Vista de mantenimiento

- Mis proyectos
 - P1
 - T1
 - Ver detalle
 - Modificar tarea
 - Eliminar tarea
 - Discutir tarea
 - Ver log discusión
 - Agregar tarea nueva
 - Ver carta Gantt
 - Discutir proyecto
 - Ver log discusión
 - Modificar proyecto
 - Eliminar proyecto
 - Crear proyecto nuevo
 - Sala espera Administrador

Información sobre esta tarea (con jornada de Lunes a Viernes)

Nombre tarea: T1
Estado tarea: Sin iniciar
Responsables: jefesmf
Fecha de inicio: 08-07-2007
Fecha de término: 08-08-2007
Días de trabajo asignados: 23
Dís de trabajo restantes: 23
Estimado completo: 0%
Completo esperado: 0%

T1

		2007																	
		Jul							Ago										
		2/7	9/7	16/7	23/7	30/7	6/8												
		L	M	J	V	S	D	L	M	J	V	S	D	L	M	J	V	S	D
T1																			

Haga clic sobre la imagen para agrandarla.

Modificaciones efectuadas a esta tarea

Numero	Modificacion	Fecha	Por
1	Sin iniciar	2007-07-08	Jefe 1

Ilustración 48: vista de detalle para una tarea

12.2.1.9 Modificar tarea

Desde este enlace es posible modificar las fechas de inicio y término de una tarea, así como sus responsables y las restricciones que posea con otras tareas. Se cargarán por defecto las opciones que hayan sido ingresadas en la última modificación de la tarea.

Estimaciones Proyectos Herramientas Salir

Vista de mantenimiento

- Mis proyectos
 - P1
 - T1
 - Ver detalle
 - Modificar tarea
 - Eliminar tarea
 - Discutir tarea
 - Ver log discusión
 - Agregar tarea nueva
 - Ver carta Gantt
 - Discutir proyecto
 - Ver log discusión
 - Modificar proyecto
 - Eliminar proyecto
 - Crear proyecto nuevo
 - Sala espera Administrador

Modificar tarea

Elija responsables de tarea Jefe 1

Fecha inicio (d/m/a): 01 01 2007

Fecha término (d/m/a): 01 01 2007

***Estado de tarea:** Sin iniciar

***Porcentaje estimado completo:** 0

* denota campo requerido

Ilustración 49: formulario de modificación de tarea

12.2.1.10 Eliminar tarea

Si una tarea no formará más parte de un proyecto, es posible eliminarla. Al eliminarla se borrará también todo registro de las discusiones que se hayan realizado en su sala de Chat, su sala de estimación, así como toda estimación efectuada.



Ilustración 50: formulario para eliminar una tarea

12.2.1.11 Discutir tarea

Este enlace redirigirá al usuario a la sala de discusión creada para la tarea elegida.

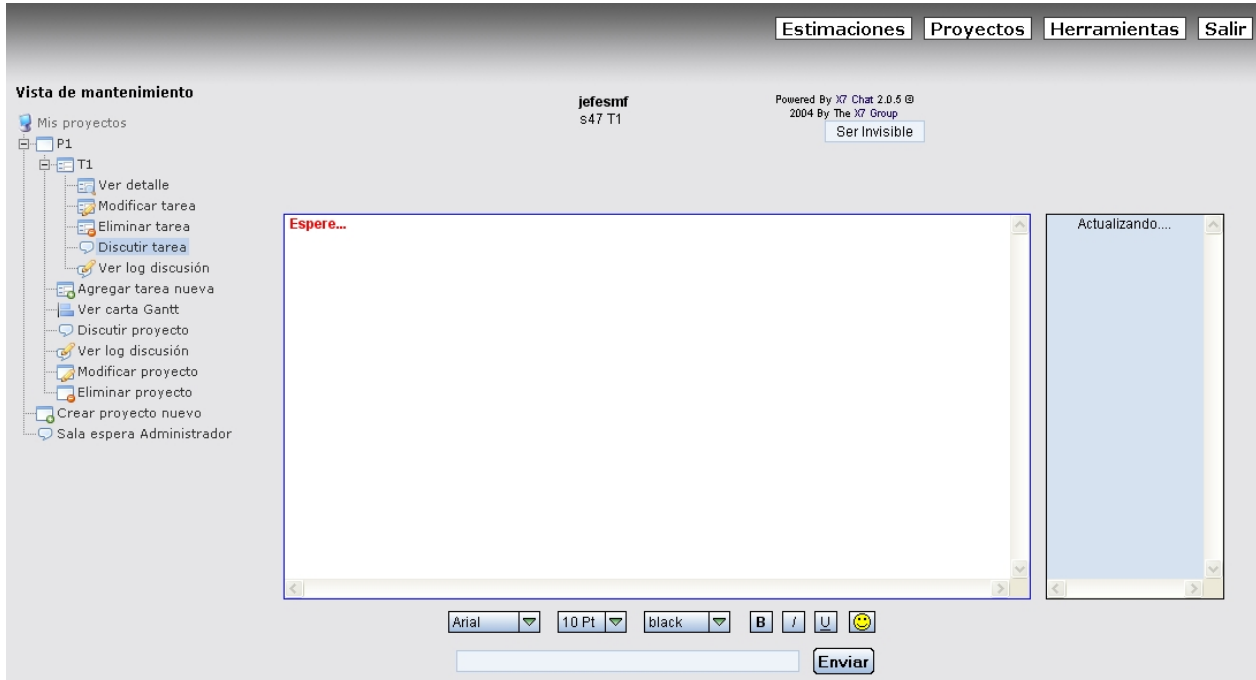


Ilustración 51: sala de discusión para una tarea

12.2.2 Vista de estimación

La vista de estimación incluye las salas donde los usuarios pueden realizar estimaciones referentes a las tareas que han creado, modificar las opciones de estas salas y ver un registro de las estimaciones realizadas.

12.2.2.1 Modificar opciones

Desde aquí es posible modificar las opciones de la sala de estimación para la tarea seleccionada, como son los tiempos que se desea estar en cada fase de estimación, y el tipo de estimación que se realizará.

Ilustración 52: modificar opciones de estimación

12.2.2.2 Realizar estimación

Los usuarios que ingresen a este enlace serán llevados a la sala de estimación para la tarea seleccionada. Esta sala está formada por un conjunto de formularios en los cuales se realizan las acciones expuestas en el capítulo 9.9: “Acciones comunicativas en cada fase”. El tiempo que se ocupe en cada fase variará según lo especificado en las opciones para cada sala. Los formularios que componen una sesión de estimación son los siguientes.

Formulario de ingreso a estimación

Dado que la herramienta pretende ser utilizada aún cuando todos los miembros de un equipo no se encuentren en el mismo lugar físico, se debe considerar que no todos los miembros pueden llegar al inicio de una sesión de estimación. Además, como éstas son un proceso cíclico, es posible algunos miembros lleguen cuando ya se esté en una fase avanzada de la estimación. Para controlar esto se ha creado un formulario que redirige a los usuarios al formulario correspondiente, según sea la fase de la estimación en la que se encuentre el grupo.

Ilustración 53: formulario de ingreso a estimación

Formulario de discusión

Este formulario contiene la herramienta de Chat, que será utilizada para discutir acerca de los parámetros que puedan influir en el desarrollo de una tarea. Además se puede conversar acerca del resultado de iteraciones de estimación anteriores, en caso de haberlas. Este formulario se corresponde con la “fase de discusión” del modelo presentado, según se describe en el capítulo 9.9.1: “Fase de discusión”.



Ilustración 54: formulario de discusión

Formulario de contexto de estimación

Para informar a los demás usuarios del conocimiento que se tiene acerca de la tarea sobre la cual se realizara la estimación, así como de los factores externos que podrían influir en su desarrollo, se creó un formulario en el que se pueden ingresar estos datos mediante texto escrito. Esta información será almacenada para crear una memoria organizacional de la tarea, y también para mostrarla al momento de llevar a cabo una sesión de estimación. Este formulario se corresponde con la “fase de planteamiento del contexto”, presentada en el capítulo 9.9.2: “Fase de planteamiento del contexto”.

Ilustración 55: formulario de contexto de estimación

Formulario de estimación

Según sea el tipo de estimación especificado en las opciones de la sala, este formulario podrá ser utilizado para estimar las fechas de inicio y término de la tarea elegida, o para estimar alguna propuesta entre todos los participantes. En ambos casos, los participantes deberán ingresar los argumentos que avalen su propuesta. Este formulario representa la “fase de estimación” del modelo presentado, según fue descrita en el capítulo 9.9.3: “Fase de estimación”.

Ilustración 56: formulario de estimación

Formulario de opinión de estimaciones

Una vez que los usuarios han realizado sus estimaciones e ingresado sus argumentos, se presentará un formulario donde podrán opinar acerca de que estimación de las entregadas les parece más adecuada para el problema planteado, otorgándole un puntaje de 1 a 6. El rango del puntaje fue establecido de esta forma para evitar tener notas con un carácter neutral. Este formulario representa a la “fase de votación de estimaciones”, según se presento en el capítulo

9.9.4: “Fase de votación de estimaciones”.

The screenshot shows a web interface titled 'Vista de estimación'. At the top, there are navigation tabs: 'Estimaciones', 'Proyectos', 'Herramientas', and 'Salir'. On the left, there is a sidebar with 'Mis proyectos' containing 'P1' and 'T1'. Under 'T1', there are three options: 'Modificar opciones', 'Realizar estimación', and 'Ver ajuste'. The main content area has the instruction: 'Califique las siguientes estimaciones de 1 (no concuerda) a 6 (concuerda)'. Below this is a table with the following data:

Puntaje	Fecha inicio	Argumentos fecha inicio	Fecha término	Argumentos fecha término
<input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 <input checked="" type="radio"/> 6	2007-03-01	Argumento 1	2007-04-04	Argumento 2

Below the table is an 'Enviar' button. At the bottom, it says 'Quedan 2 segundos para la siguiente fase.'

Ilustración 57: formulario de opinión de estimaciones

Formulario de acuerdo en la estimación

Con los votos enviados y el puntaje contabilizado por la aplicación, se muestra a los participantes las diferentes estimaciones y el puntaje alcanzado por ellas. Desde este formulario, el usuario podrá elegir si aceptar la estimación con mayor puntaje, o realizar una nueva ronda de estimación en caso que no concuerde con la votación de la mayoría. Se realizará una nueva ronda de estimación sólo si la mitad de los participantes así lo desea. Este formulario corresponde a la “fase de acuerdo en estimación”, que fue presentada en el capítulo 9.9.5: “Fase de acuerdo en estimación”.

The screenshot shows the same 'Vista de estimación' interface, but now displaying 'Resultados de votación'. The table shows the results of the vote:

Fecha inicio	Argumentos fecha inicio	Fecha término	Argumentos fecha término	Puntaje obtenido
2007-03-01	Argumento 1	2007-04-04	Argumento 2	6

Below the table are two buttons: 'Aceptar mayoría' and 'Solicitar nueva estimación'. At the bottom, it says 'Quedan 3 segundos para la siguiente fase.'

Ilustración 58: formulario de acuerdo en la estimación

12.2.2.3 Ver ajuste

Cuando se hayan realizado estimaciones para alguna tarea, será posible ver desde este enlace un gráfico con la representación de las fecha de inicio y término estimadas por los alumnos en cada iteración para esta tarea, junto con el puntaje que alcanzaron. Además del gráfico, se mostrará una tabla con los argumentos entregados para cada estimación realizada. Si se han realizado estimaciones de propuestas para la tarea elegida, estas serán listadas a continuación de las estimaciones de tiempo de desarrollo.

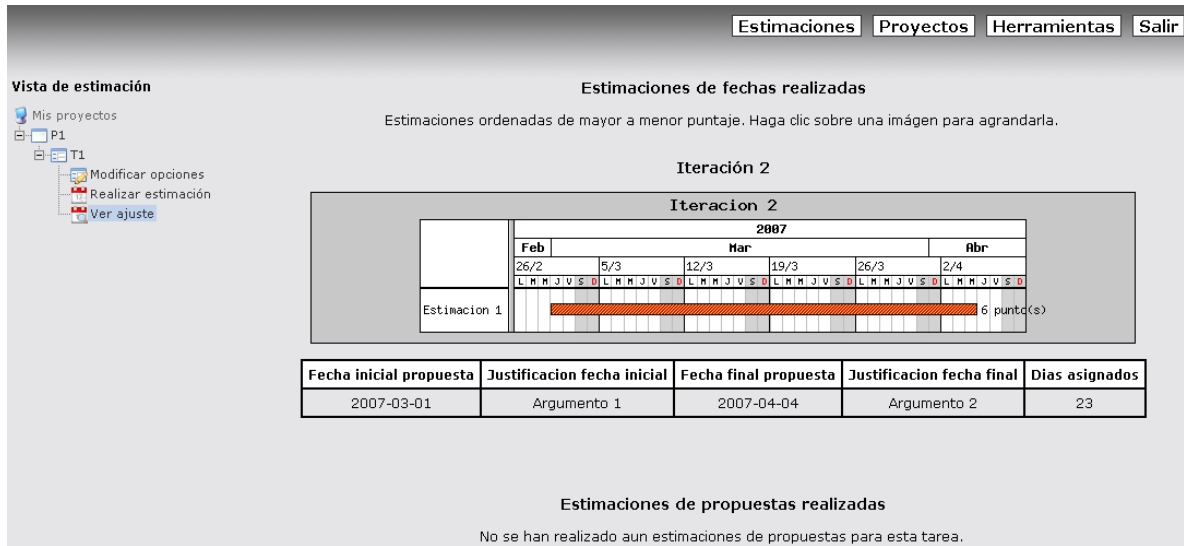


Ilustración 59: vista de ajuste de estimaciones

12.2.3 Vista de herramientas

En esta vista van las herramientas que no poseen una relación directa con los proyectos y tareas.

12.2.3.1 Cuestionario de fortalezas

Se creó un formulario utilizando un cuestionario presente en las diapositivas del curso CC51A, dictado por el profesor Sergio Ochoa el semestre de otoño del año 2006. El objetivo de este cuestionario es presentarle al alumno las fortalezas que posee en distintas áreas de un equipo de desarrollo, indicándole en las que se encuentra más capacitado según las respuestas que haya entregado. El cuestionario consta de 48 preguntas divididas en 6 grupos de 8 preguntas. En cada pregunta, el alumno debe seleccionar como respuesta su grado de conformidad respecto a la aseveración que acompaña a las posibles respuestas. Estas son “Muy de acuerdo”, “De acuerdo”, “Neutro”, “En desacuerdo” y “Muy en desacuerdo”. A cada una de las cinco alternativas existentes por pregunta se le asigna un puntaje, dependiendo de la pregunta en la que se encuentre el alumno. Al terminar de responder el cuestionario, se contabiliza el puntaje total obtenido por pregunta y por sección, y en base a eso se otorga a cada cargo posible un lugar de preferencia según las respuestas entregadas.

Ilustración 60: cuestionario de fortalezas

12.2.3.2 Glosario

Se presenta al alumno un glosario con los términos más comunes de la ingeniería de software, extraídos de las diapositivas del curso CC51A, dictado por el profesor Sergio Ochoa el semestre de otoño del año 2006.

Ilustración 61: glosario

12.2.3.3 Estimador de complejidad

Esta sección de la aplicación corresponde a un formulario que solicita al alumno evalúe un conjunto de características relativas al software que debe implementar, indicándole en base a

estos datos la complejidad del problema que debe resolver basado en el modelo COCOMO.

Estimaciones Proyectos Herramientas Salir

Vista de herramientas

- [Cuestionario fortalezas](#)
- [Glosario](#)
- [Estimador de complejidad](#)
- [Ir al foro](#)

Opciones de administración

- [Actualizar glosario](#)
- [Actualizar usuarios](#)

Elija el nivel de importancia o valor que tienen las siguientes características

	Muy bajo	Bajo	Normal	Alto	Muy alto	Extra alto
Atributos del producto						
Confiabilidad requerida del software	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tamaño de la base de datos de la aplicación	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Complejidad del producto	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Atributos del hardware						
Restricciones en el uso de la CPU	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Restricciones en el uso de memoria	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Volatilidad del hardware	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Requisitos del tiempo de respuesta	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Atributos del personal						
Capacidad del analista	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Experiencia con el tipo de aplicación	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Calidad de los programadores	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Experiencia con el hardware	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Experiencia con el lenguaje de programación	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Atributos del proyecto						
Empleo de herramientas de software	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Empleo de métodos de ingeniería de software	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Restricciones a duración del proyecto	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Ilustración 62: formulario para estimador de complejidad

12.2.3.4 Foro

Este enlace llevará al usuario al foro de mensajes, donde podrá discutir con los demás usuarios del sistema si el administrador así lo permite. Además, el foro presentará opciones extras a los usuarios como modificar su perfil en caso de tratarse de alumnos; o crear usuarios, grupos, foros y categorías en el caso de los administradores del sistema.



Ilustración 63: herramienta de foro

12.2.3.5 Actualización de glosario

Sólo los administradores del sistema verán este enlace en la sección herramientas. Éste les permitirá subir un archivo con los conceptos y definiciones que deseen formen parte del glosario. El archivo de texto debe poseer el siguiente formato.

```
{
Concepto1
Definición...
...
...
}

{
Concepto2
Definición...
...
...
}
```

El nombre del concepto debe ir en la primera línea inmediatamente después del paréntesis de corchete de apertura. El nombre del concepto sólo puede ocupar la primera línea, pues el resto se considerara parte de la definición. Para indicarle al programa que viene un concepto, éste va precedido por una línea que contiene sólo el paréntesis de corchete de apertura. Todo lo que venga luego de la línea que contiene el concepto, hasta la línea que contiene sólo el paréntesis

de corchete de cierre, es la definición. Puede también incluir sin problemas los símbolos “{“ y “}” en la definición, siempre que no vengan éstos solos en una línea, pues se considerarían como un error o el término de la definición respectivamente. Para terminar una definición debe escribirse una línea que contenga sólo un paréntesis de corchete de cierre.

Si se detecta algún error en la sintaxis del archivo, el glosario no será modificado. Si el archivo está correctamente escrito, las definiciones almacenadas serán reemplazadas por las contenidas en el nuevo archivo.

Estimaciones Proyectos Herramientas Salir

Vista de herramientas

- Cuestionario fortalezas
- Glosario
- Estimador de complejidad
- Ir al foro

Opciones de administración

- Actualizar glosario
- Actualizar usuarios

ATENCIÓN: ¡Se reemplazarán las definiciones existentes por las contenidas en el nuevo archivo de definiciones. Este debe venir en formato ANSI, pues el formato UTF-8 genera errores relativos al carácter especial BOM.

>Descargar ejemplo<

Si el archivo es abierto por su navegador, pruebe a hacer clic derecho sobre el enlace y elija la opción "Guardar como..."

*Elija el archivo a subir: Examinar...

Subir el archivo

* denota campo requerido

Ilustración 64: actualización de glosario

12.2.3.6 Ingreso de usuarios al sistema

Este enlace será visible sólo para los administradores de sistema, y les permitirá subir un archivo CSV con la información de los alumnos del curso de forma de ingresarlos rápidamente al sistema. Este archivo debe contener líneas en el formato que se indica a continuación, una por cada usuario que se desee agregar.

Identificador_de_usuario; Nombre_real_del_usuario; Identificador_del_grupo_del_usuario; Email_del_usuario; Contraseña_de_usuario
--

Si se detecta un error en el formato del archivo, los usuarios no serán agregados al sistema. Si el archivo es correcto, los usuarios serán agregados al listado de usuarios ya existentes. Es importante recordar que, como se ha dicho anteriormente, la aplicación utiliza el sistema de usuarios del foro.

13- Discusión

En este capítulo se opina acerca de diferentes razones que afectaron el diseño del modelo de estimación presentado en el capítulo 9: “Modelo de estimación”, y se explica la actividad realizada en la aplicación con los alumnos de uno de los cursos de la facultad, junto con un resumen de sus opiniones.

13.1 Discusión sobre el modelo de estimación presentado

En algunas versiones del modelo Wideband Delphi se considera que en las primeras reuniones de estimación el equipo debe preocuparse de dividir el proyecto en todas las tareas que lo conformen, realizando a continuación una estimación del tiempo necesario para completar cada tarea y con esto, la duración del proyecto [4]. La idea de la herramienta, considerando el contexto académico para el que fue diseñada, no es crear todas las tareas al momento de dividir el proyecto. Esto debido a la dificultad que puede significar para los alumnos que no han desarrollado anteriormente proyectos de gran tamaño el obtener una visión global del problema a resolver.

Además, a lo largo del semestre es muy posible que se generen atrasos en las entregas del producto, por la falta de experiencia en estimar tiempos de desarrollo ajustados a la realidad. Estos atrasos llevarán a los alumnos a repactar las condiciones de entrega con los clientes, eliminando algunas partes del producto que sean complejas y que no agreguen mucho valor por otras que sean algo más simples de implementar o que el cliente requiera con mayor urgencia. Debido a esto, el listado de tareas a cumplir variará constantemente, por lo que se prefirió un enfoque en el que los alumnos puedan agregar tareas según lo consideren necesario, y teniendo la posibilidad de pactar reuniones de estimación para cuando el grupo posea tiempo disponible. Según cambien las estimaciones de tiempo asignadas a cada tarea, cambiará la carta Gantt del proyecto completo, por lo que se podrá saber que tareas requieren mayor atención para poder completarse a tiempo, o cuales deberán ser repactadas con el cliente.

Al factor recién mencionado se agrega el hecho de que para poder realizar una sesión de estimación donde se pueda dividir el proyecto en cada tarea posible y además estimar la duración de todas ellas en forma correcta, se requiere que el equipo disponga de un gran bloque de tiempo disponible para llevarla a cabo. El enfoque dado a la aplicación, donde la estimación se realiza para cada tarea por separado, permite realizar reuniones de estimación de menor duración para las que es más probable que el equipo tenga tiempo de juntarse.

Un punto en contra de la metodología propuesta es que si se modifica una estimación ya realizada al estar el proyecto en una etapa más avanzada, es posible que se deban realizar nuevas sesiones de estimación por separado para cada tarea que dependa de ella, al no poder estimar varias tareas en conjunto como en otras versiones del modelo Wideband Delphi. Sin embargo, no se pierde la noción de que cada tarea forma parte de un todo, ya que la aplicación muestra al momento de estimar todas las tareas que conforman el proyecto. Además se espera que los alumnos cometan este tipo de errores, de forma que puedan mejorar a través de la práctica. El realizar nuevas estimaciones para tareas cuyas fechas iniciales propuestas no pudieron cumplirse permitirá a los alumnos aprender de sus errores y realizar mejores estimaciones en el futuro.

Otro elemento que no se encuentra en el modelo propuesto es el uso de un moderador en la

discusión. Esto debido a que se deseó otorgar a todos los miembros del grupo el mismo nivel de participación en la discusión o creación de tareas, eliminando la posible apropiación de la discusión por parte del moderador. Aún así, si el grupo lo desea puede otorgar a un usuario este cargo fuera de la aplicación, de forma que él sea el encargado de agregar al sistema las tareas que el grupo genere o de regular las discusiones.

13.2 Discusión sobre la aplicación

13.2.1 Experiencia de uso de la aplicación

Se realizó una experiencia de uso de la aplicación junto con los alumnos del ramo CC52N: “Computación para el trabajo grupal” del semestre de otoño del 2007, con el objetivo de probar la herramienta en el entorno para el que fue diseñada. Para esta actividad se dividió el curso en grupos de tres a cuatro alumnos, donde a cada grupo se le encargó un proyecto distinto. La actividad consistió en dividir el proyecto asignado en las tareas que lo componen, y realizar estimaciones para la duración de cada tarea creada. Del total de alumnos del curso, once completaron esta tarea, tras lo cual respondieron un cuestionario dónde se les preguntaba su opinión acerca de la experiencia y la aplicación. A continuación se incluye un resumen de las opiniones más destacadas.

¿Cambió el razonamiento de tu estimación tras ver los argumentos de los demás participantes en la fase de estimación? Si es así, ¿en qué cambió? ¿Por qué?

En esta pregunta, seis alumnos opinaron que sus estimaciones cambiaron luego de ver los argumentos de los demás participantes, debido a que esto les entrega una visión más general del problema a resolver. Cinco afirman que mantuvieron las duraciones iniciales establecidas. La principal razón entregada por los alumnos para mantener sus estimaciones es que habían discutido previamente las fechas, por lo que al momento de estimar ya tenían una idea de que propuestas entregar.

¿Crees que se llegó a un acuerdo de forma más rápida o lenta que en una reunión convencional? ¿Por qué?

Siete alumnos consideraron que se llegó a una respuesta de forma más rápida que en una reunión convencional, debido al límite preestablecido de tiempo que impone cada fase. Según ellos, esto es difícil de controlar en una reunión cara a cara. Los cuatro alumnos que consideran que se llegó a un acuerdo de forma lenta atribuyen esto a los “tiempos muertos” en cada fase, cuando se ha enviado su respuesta y se espera el inicio de la fase siguiente. Además, dos de ellos consideran un factor importante en la lentitud el tiempo que fue necesario utilizar para comprender el modo de uso de la herramienta.

¿Crees que la solución alcanzada es mejor o peor a la alcanzada en una reunión convencional? ¿Por qué?

Cuatro alumnos piensan que la herramienta los ayudó a lograr mejores resultados debido a que poseen la visión anónima de los demás participantes, lo que les otorga nuevos puntos de vista. También consideraron importante que la aplicación permita que la opinión de personas que no se encuentran físicamente presentes en la reunión tenga la misma validez que las que están, y el orden que impone el proceso de estimación. Cuatro alumnos consideran que la solución alcanzada fue peor que en una reunión convencional, debido a que consideran que el poder discutir cara a cara entrega mayores posibilidades de debate y que el contador que indicaba el tiempo restante para la siguiente fase era una fuente de presión. Dos personas consideraron que la solución alcanzada es igual a la que se lograría en una reunión convencional, sin

embargo agregan que para grupos con más participantes la calidad del acuerdo podría mejorar. Una de las respuestas no fue válida para la pregunta efectuada.

¿Consideras que la interfaz de la aplicación fue adecuada para realizar las actividades de esta tarea? ¿Qué consideraste inadecuado?

Seis alumnos consideraron adecuada la interfaz. Se destacó la forma en que se muestra el listado de proyectos y tareas. Cinco alumnos consideran que la interfaz no fue adecuada, debido a que no existe una forma dentro del programa de coordinar a los miembros de un grupo, a que no existen mayores explicaciones sobre las acciones que se pueden realizar, y a los colores utilizados en ella.

¿Consideras que esta herramienta es útil para llevar un registro del avance de un proyecto y sus tareas?

Seis alumnos piensan que la herramienta permite cumplir este objetivo, debido a que almacena toda conversación realizada, las fechas pactadas inicialmente por los integrantes, y es posible visualizar el proyecto de forma global con sus tareas en los gráficos. Tres alumnos opinaron que con las actividades realizadas no es posible determinar la utilidad de la herramienta en este ámbito. Dos de las respuestas dadas fueron poco claras como para clasificarlas de positivas o negativas.

¿Crees que esta herramienta podría ayudarte a mejorar las estimaciones de tiempo que realices al trabajar en grupo y a aprender de estimaciones pasadas?

Seis alumnos opinan que la herramienta podría ayudarles, sin embargo consideran que deben mejorarse los puntos en contra expresados en las respuestas anteriores. Los principales factores considerados son la rapidez con la que se puede llegar a un acuerdo en la estimación, el sistema de votación de las propuestas entregadas por los integrantes y el poder realizar reuniones sin que todos estén en el mismo lugar. Tres alumnos consideran que puede ayudar a mejorar las estimaciones si se le agrega funcionalidad adicional como estadísticas más detalladas. Un alumno opina que la herramienta no mejora las estimaciones realizadas debido a que no ve forma de incluir las capacidades de los alumnos, el nivel de funcionamiento del grupo o la incertidumbre en los requerimientos. Un alumno entregó una respuesta que no puede ser clasificada de positiva o negativa.

13.2.2 Discusión sobre experiencia de uso

Uno de los elementos de la aplicación mencionado en distintas respuestas como problema por los alumnos fue la forma en que se ingresa a las distintas fases de estimación. En lugar de asignar a cada fase una duración predefinida, algunos alumnos preferirían que se pasara a la fase siguiente en cuanto todos los usuarios del grupo dentro de la estimación hayan terminado de responder la fase actual, evitando así perder el tiempo entre que se responde una fase y comienza la siguiente. Si bien esta opción fue estudiada al momento de decidir la forma en que se implementarían las reuniones de estimación, se consideró que generaría muchos accesos a la base de datos, al ser necesario revisar cada cierto tiempo y por cada miembro que esté estimando, si los demás usuarios ya respondieron su parte y ver quienes siguen dentro de la estimación y quienes la abandonaron. Es posible modificar esta característica de la aplicación, pero se consideraría necesario realizar un estudio con un gran número de usuarios de forma de verificar el impacto real que tendría este comportamiento sobre la carga en el servidor.

Otro factor mencionado corresponde a que no existe una guía clara sobre que acciones se deben realizar dentro de la aplicación. Para solucionar esto es posible agregar mensajes al

sistema que sean más explícitos en cuanto a los pasos a seguir para poder crear un proyecto, sus tareas, y el como realizar las estimaciones.

Relacionado con el punto anterior, los alumnos expresaron que no existe forma de coordinar a los miembros del grupo dentro de la aplicación. Una solución a esto podría ser el crear una sala de Chat para cada grupo, de forma que al ingresar a las vistas de mantención de proyectos o estimación, los miembros sean redirigidos a ella y puedan decidir que acciones seguir.

13.2.3 Discusión sobre modificaciones a la aplicación

Uno de los puntos de la aplicación que puede ser mejorada corresponde a la utilización de información anexa al realizar estimaciones. Esto puede realizarse implementando un sistema de subida de archivos en la fase de planteamiento del contexto y un sistema de descarga de los mismos en la fase de estimación. Sin embargo, esto podría significar una carga muy grande para el servidor. Una adaptación a esta idea podría ser crear una sección cuyo único objetivo sea subir y bajar este tipo de archivos, sin embargo esto sería similar a enviar la información por correo electrónico o mecanismos similares, por lo que no se considera una necesidad inmediata. La mayor ventaja de implementar el sistema de subida y bajada de archivos es el poder tener la información almacenada en un lugar común. Esto, para los proyectos que deben desarrollarse en la duración del semestre académico, no corresponde a un factor crítico, siendo útil en los casos en que los proyectos deban ser desarrollados por varios grupos a lo largo de diferentes semestres.

Por otra parte, debió decidirse al comienzo de la implementación la división temporal que se le daría al desarrollo de los proyectos y tareas. Uno de los puntos que se consideró en la decisión a tomar fue que permitiera representar adecuadamente la duración de los proyectos y tareas en el recurso de representación de progreso elegido. Debido a que se trabajaría con cartas Gantt, una división muy pequeña generaría problemas en la representación, al tener que ampliar el espacio necesario para permitir una visualización clara de la información. Otro factor que se consideró fue el nivel de precisión que se pretende obtener de las estimaciones de los alumnos. Fijar duraciones de horas o minutos para tareas es justificable cuando estas son simples o de corta duración. Sin embargo tareas más complejas que deban ser desarrolladas en varios días por los alumnos, donde no todos pueden asistir diariamente al lugar de trabajo, dificulta el realizar una estimación ajustada de la duración en horas que se le dedicará a cada actividad. Al aumentar el intervalo de tiempo en que se puede dividir una tarea, se baja el nivel de precisión que los alumnos deben tener al realizar sus estimaciones. Sin embargo, si se aumenta mucho el intervalo, por ejemplo a una semana, se pierde precisión en el nivel de representación que se puede alcanzar para la duración de tareas más cortas. Finalmente, para alcanzar un equilibrio entre los factores recién mencionados, se utilizó la división en días para las tareas y proyectos. Esto puede ser modificado para adaptarse mejor al curso en el que sea utilizada la herramienta.

Para medir el avance realizado en cada tarea se decidió utilizar porcentajes, otorgando a los alumnos la posibilidad que ellos mismos ingresen el avance que consideren correcto. Para medir el avance de un proyecto se decidió no implementar ninguna representación gráfica, sino que el alumno pueda inferir esto basándose en el avance de todas las tareas que conformen el proyecto. Es posible modificar esto de forma que aparezca un indicador que, basado en el avance de cada tarea y en el tiempo restante de cada una de ellas, permita mostrar el avance total del proyecto según los datos ingresados por los alumnos. Sin embargo, esto sería sólo una representación ideal, que no consideraría la diferencia en la dificultad que pueda existir entre las tareas.

A fin de que los miembros de un grupo puedan ver la forma en que han convergido las estimaciones que han realizado para una determinada tarea, y como éstas se han ido ajustando a la realidad con el paso del tiempo, se decidió implementar una página que muestre a los usuarios un gráfico que refleja para cada iteración de una estimación el valor ingresado por cada miembro que haya participado, seguido de una tabla que muestra de forma anónima los argumentos entregados. La idea tras esta página es que los alumnos puedan aprender de los errores que se cometieron en estimaciones anteriores. Inicialmente esta página solo comparaba las primeras estimaciones elegidas por el grupo con la última realizada para una determinada tarea, pero fue modificada por considerar el enfoque actual más adecuado para el aprendizaje.

Uno de los puntos del modelo que no pudo ser implementado como se hubiera deseado fue la fase de discusión. Para que los usuarios se vieran obligados a exponer sus puntos de vista según el modelo indicado se podría haber modificado la herramienta de Chat, de forma de que incluyera dos campos de texto, uno para la idea a presentar, y otro para los argumentos. A estos campos se les debe agregar un sistema de validación, de forma que a menos que ambos campos hayan sido ingresados, el mensaje no se ingrese al Chat. Sin embargo, dado que ésta es la única parte de todo el proceso de estimación en la que el grupo puede hablar libremente sobre los resultados de las estimaciones de iteraciones anteriores, se prefirió dejar la herramienta como se incluyó en la aplicación.

14- Conclusiones

En el trabajo presentado se creó una herramienta basada en la metodología Wideband Delphi que pretende ayudar a los alumnos de forma que estos puedan mejorar las estimaciones que realicen acerca de la duración del desarrollo de un proyecto de software, así como de las tareas que lo componen. Sin embargo para poder realizar esto de forma adecuada, fue necesario presentar a los alumnos el problema completo, siendo ellos los encargados de dividir el proyecto en sus tareas, establecer un orden entre ellas y asignarles responsables. Para esto se les ofreció a los alumnos una forma de representar sus proyectos y tareas dentro de la aplicación, y se les entregó la posibilidad de ver una representación gráfica del período de tiempo disponible para el desarrollo. Mediante este proceso se refuerzan los conocimientos que poseen los alumnos, motivándolos a ser objetivos con las tareas que crean y a mantener siempre una visión global del problema a resolver. Además, para poder diseñar correctamente la herramienta de estimación, fue necesario crear un modelo de discusión basada en argumentos que se adaptara a una sesión de estimación. Este modelo fue incluido en la aplicación, de forma que los alumnos puedan realizar estimaciones donde la información entregada por otros les permita mejorar sus capacidades y considerar nuevos factores que puedan llegar a modificar su modelo mental.

De las opiniones recogidas de los alumnos que probaron la aplicación es posible ver que se valora en las sesiones de estimación la eliminación de algunos de los factores negativos que se presentan en una reunión convencional, como es el no poder realizar comentarios de forma completamente anónima o la dificultad que puede representar el intentar terminar una discusión cuando hay integrantes que no consideran correcta la estimación alcanzada y desean seguir discutiendo pasado el límite de la reunión. Además fue considerada útil para llevar un registro de un proyecto y sus tareas, y se opinó que permitiría mejorar las estimaciones a través del aprendizaje mediante el estudio de estimaciones anteriores. Si bien la experiencia realizada no permitió completar algunos objetivos propuestos como verificar en un ambiente real de desarrollo si las estimaciones iniciales realizadas por los grupos que utilizan la herramienta son más ajustadas que los grupos que no la utilizan, o si en un mismo grupo la aplicación mejora las estimaciones finales en comparación con las primeras realizadas, su primer acercamiento con los alumnos fue positiva y se consideró útil frente al tema que se pretende resolver.

Para permitir el trabajo grupal, en el desarrollo de la herramienta se debieron considerar distintos factores que no se aprecian en una herramienta de trabajo asíncrono, como es el ordenar a todos los participantes y guiarlos en la forma en que tienen que realizar sus aportes, de forma de facilitar la lectura de estos por parte de los demás usuarios y aprovechar lo mejor posible el tiempo que se dedica a las reuniones. Debido a esto, el establecer un modelo de comunicación fue sumamente útil, ya que sirvió como base para la forma en la que fueron implementadas las distintas fases de estimación, así como para definir la información que se espera que entreguen los usuarios.

La herramienta fue creada de forma que fuera posible su uso sin importar el ramo en el que sea utilizada. Sin embargo, debido a que sólo utiliza herramientas de código abierto, es posible modificarla de forma de adaptarla a las necesidades que presente cada ramo en caso de que se desee personalizar alguna de las prestaciones que posee, como cambiar la representación de las cartas Gantt para utilizar horas en lugar de días en la duración de las tareas. De esta forma, la aplicación puede ajustarse de mejor manera a las necesidades de los estudiantes, ofreciéndoles mejores herramientas con las cuales aprender.

15- Referencias

- [1] Boehm, Barry. "Software Engineering Economics". Upper Saddle River, NJ. Prentice Hall PTR. 1981.
- [2] Jorgensen, Magne. "A review of studies on expert estimation of software development effort". Journal of Systems and Software (70): pp. 37-60. 2004.
- [3] Linstone, Harold y Turoff, Murray. "The Delphi Method, Techniques and Applications". Reading, Ma. Addison-Wesley. 1975. pp. 3.
- [4] Wiegers, Karl E. "Stop Promising Miracles" [en línea]. <<http://www.processimpact.com>> [Consulta: agosto 2006]
- [5] Leibenstein, Harvey. "Bandwagon, Snob, and Veblen Effects in the Theory of Consumers' Demand". Quarterly Journal of Economics, 64 (2): pp. 183. 1950.
- [6] Stellman, Andrew y Greene, Jennifer. "Applied software project management". O'Reilly Media. 2005. Capítulo 3: Estimation, pp. 34.
- [7] Sommerville, Ian. "Software Engineering". Addison-Wesley. 1996. Capítulo 29: Software cost estimation, pp. 599.
- [8] Bellasai, G., Borges, M., Fuller, D., Pino, J. A., Salgado, A. C., "An IBIS-based model to support group discussions". En: B. Glasson, D. Vogel, P. Bots, J. Nunamakes (eds). "Information Systems and Technology in the International Office of the Future". Londres. Champman & Hall. 1996. pp. 49-62.
- [9] Wikipedia [en línea]. <http://en.wikipedia.org/wiki/Wideband_delphi> [Consulta: agosto 2006]
- [10] Leung, Hareton y, Fan, Zhang. "Software Cost Estimation" [en línea]. Department of Computing, The Hong Kong Polytechnic University. pp. 1-2. <<http://paginaspersonales.deusto.es/cortazar/doctorado/articulos/leung-handbook.pdf>> [Consulta: agosto 2006]
- [11] Wikipedia [en línea]. <<http://en.wikipedia.org/wiki/Cocomo>> [Consulta: agosto 2006]
- [12] Barry Boehm, Bradford Clark, Ellis Horowitz, Chris Westland, Ray Madachy, Richard Selby, "THE COCOMO 2.0 SOFTWARE COST ESTIMATION MODEL" [en línea]. pp. 15-20. <<http://sunset.usc.edu/research/COCOMOII/Docs/ispa95.pdf>> [Consulta: agosto 2006]
- [13] Wikipedia [en línea]. <<http://en.wikipedia.org/wiki/SEER-SEM>> [Consulta: agosto 2006]
- [14] SEER-SEM Software Estimation and Control [en línea]. <http://www.galorath.com/tools_sem.html> [Consulta: agosto 2006]
- [15] Rule, Grant. "Guest Editors' Introduction: Software Estimation Perspectives". IEEE Software. Nov/Dic 2000, 17 (6): pp. 22-26.
- [16] Pert Chart [en línea]. <<http://www.netmba.com/operations/project/pert/>> [Consulta: agosto 2006]
- [17] Williams, Fred E. "PERT Completion Times Revisited". University of Michigan-Flint. Julio 2005.
- [18] Wikipedia [en línea]. <http://en.wikipedia.org/wiki/Extreme_Programming_Practices#Planning_game> [Consulta: julio 2007]

- [19] Steindl, Christoph y Krogdahl, Pal. "Estimation in agile projects". En Best Practices in Project Estimation Conference. IBM Academy of Technology. 2005.
- [20] Sánchez I., Jaime. "Aprendizaje visible, tecnología invisible". Dolmen ediciones. Santiago. 2001. Capítulo "Constructivismo e informática educativa", pp. 72.
- [21] Ormrod, J. E., "Educational Psychology: Developing Learners". Cuarta edición. 2003. pp. 232.
- [22] Belbin, Meredith. "Nobody's perfect, but a team can be" [en línea]. <<http://www.belbin.com>> [Consulta: agosto 2006]
- [23] "ESA software engineering standards, Issue 2" [en línea]. <<http://styx.esrin.esa.it/premfire/Docs/PSS050.pdf>> [Consulta: mayo 2007]
- [24] "Guide to applying the ESA software engineering standards to small software projects" [en línea]. <<http://styx.esrin.esa.it/premfire/Docs/Bssc962.pdf>> [Consulta: mayo 2007]
- [25] Wikipedia [en línea]. <http://en.wikipedia.org/wiki/Xtreme_Programming> [Consulta: mayo 2007]
- [26] Marchesi, Michele. "The new XP". Revista Web "XProgramming.com" [en línea]. <<http://www.agilexp.org/downloads/TheNewXP.pdf>> [Consulta: mayo 2007]
- [27] Beck, Kent. "extreme Programming explained: Embrace change". Addison-Wesley Professional. Octubre 1999. pp. 7-9.
- [28] Desmond, Celia. "Project Management for Telecommunications Managers". Springer. Diciembre 2003. pp. 71-84.
- [29] Rittel, Horst y Webber, Melvin. "Dilemmas in a general theory of planning", en "Policy Sciences", 4. Ámsterdam. Elsevier Scientific Publishing Company. 1973. pp. 155-169.
- [30] Wikipedia [en línea]. <http://en.wikipedia.org/wiki/Wicked_problem> [Consulta: mayo 2007]
- [31] Conklin, Jeff. "Dialogue Mapping: Building Shared Understanding of Wicked Problems". Wiley. Noviembre 2005.
- [32] CogNexus Institute [en línea]. <<http://cognexus.org/id41.htm>> [Consulta: mayo 2007]
- [33] Toulmin, Stephen. "The uses of argument". Cambridge University Press. 2003.
- [34] Rodríguez, Luisa. "El Modelo argumentativo de Toulmin en la escritura de artículos de investigación educativa" [en línea]. Revista Digital Universitaria. Enero 2004. <<http://www.revista.unam.mx/vol.5/num1/art2/art2.htm>> [Consulta: mayo 2007]
- [35] Zabala, Sandra, Lara, Ines y Gener, Hector. "Beliefs, Reasons and Moves in a Model for Argumentative Dialogues". Universidad Simón Bolívar, Departamento de Computación. Venezuela. 1999.
- [36] Quignard, Matthieu y Baker, Michael. "Modelling Argumentation and Belief Revision in Agent Interactions". Manchester, U.K. 2nd European Conference on Cognitive Science (ECCS 97). Abril 1997.
- [37] Nunamaker, J.F., Dennis, Alan, Valacich, Joseph, Vogel, Douglas y George, Joey. "Electronic meeting systems to support group work". Communications of the ACM, 34 (7): pp. 40-61. Julio 1991.
- [38] PHPOpenChat [en línea]. <<http://phpopenchat.org/download.php>> [Consulta: febrero 2007]

- [39] OpenSourceCMS [en línea]. <http://www.opensourcecms.com/index.php?option=com_content&task=view&id=143> [Consulta: febrero 2007]
- [40] OpenSourceCMS [en línea]. <http://www.opensourcecms.com/index.php?option=com_content&task=view&id=462> [Consulta: febrero 2007]
- [41] PostgreSQL [en línea]. <<http://www.PostgreSQL.org/about/>> [Consulta: febrero 2007]
- [42] MySQL [en línea]. <<http://www.mysql.com/why-mysql/>> [Consulta: febrero 2007]
- [43] PHP Classes [en línea]. <<http://www.phpclasses.org/browse/file/11924.html>> [Consulta: febrero 2007]
- [44] Apache [en línea]. <<http://httpd.apache.org/>> [Consulta: febrero 2007]
- [45] PHP [en línea] <<http://www.php.net>> [Consulta: febrero 2007]
- [46] Omnicron Technologies Corporation [en línea]. <<http://www.omnicron.ca/>> [Consulta: febrero 2007]
- [47] PEAR [en línea]. <http://pear.php.net/package/HTML_QuickForm> [Consulta: febrero 2007]
- [48] PHP [en línea]. <<http://cl.php.net/manual/en/faq.installation.php#faq.installation.apache2>> [Consulta: febrero 2007]
- [49] PHP [en línea]. <<http://cl.php.net/manual/en/install.windows.apache1.php>> [Consulta: febrero 2007]
- [50] PEAR [en línea]. <<http://pear.php.net/go-pear.phar>> [Consulta: febrero 2007]
- [51] PEAR [en línea]. <<http://pear.php.net/bugs/bug.php?id=1156>> [Consulta: febrero 2007]
- [52] MySQL [en línea]. <<http://www.mysql.com/why-mysql/marketshare/>> [Consulta: marzo 2007]
- [53] famfamfam.com [en línea]. <<http://www.famfamfam.com/lab/icons/silk/>> [Consulta: abril 2007]

Anexos

A- Respuestas de alumnos a cuestionario de evaluación

Las siguientes fueron las respuestas entregadas por los alumnos del curso CC52N: "Computación para el trabajo grupal" al cuestionario que debían realizar una vez finalizado su trabajo con la aplicación.

¿Cambió el razonamiento de tu estimación tras ver los argumentos de los demás participantes en la fase de estimación? Si es así, ¿en qué cambió? ¿Por qué?

Alumno 1: "No cambió demasiado ya que los tiempos creo que los definimos bastante bien al principio."

Alumno 2: "Cambió en su duración, como corresponde a la última etapa del proyecto, en esta podríamos encontrarnos con los errores arrastrados de las 2 fases anteriores, entonces sería necesario una semana mas para incluir esta posibilidad, quedando en 3 semanas."

Alumno 3: "En el caso de la tarea que escogí, no cambio el razonamiento."

Alumno 4: "No cambió mucho porque todos somos computines, pero podría cambiar si esto se hace con un equipo de trabajo que tenga distintos pensamientos."

Alumno 5: "Me di cuenta de la disponibilidad de tiempo de los otros integrantes del grupo de trabajo. La percepción de su estimación me ayudo a refinar la mía, pues el otro integrante puede tener un mejor conocimiento de su habilidad/capacidades."

Alumno 6: "Se dejó un poco de tiempo para hacer la integración del sistema y hacer pruebas."

Alumno 7: "No cambiaron mucho porque ya habíamos hablado el tema antes de usar la aplicación."

Alumno 8: "La colaboración entre los miembros fué muy importante para ir ajustando las estimaciones de los tiempos. Al ver otros puntos de vista se pueden apreciar elementos que antes pasaban desapercibidos."

Alumno 9: "Si eso es verdad, creo que cambió porque mis compañeros tenían otro punto de vista."

Alumno 10: "Creo que uno estaba mas preocupado del tiempo que de los argumentos. En lo personal, no los vi mucho y me dejaba influenciar más por mis creencias."

Alumno 11: "Sí, ya que ahora no son una fecha dictada por alguien, sino que también hay una discusión detrás de la fecha determinada y además queda registrada esa discusión"

¿Crees que se llegó a un acuerdo de forma más rápida o lenta que en una reunión convencional? ¿Por qué?

Alumno 1: “Fue bastante más rápida pero no necesariamente mejor.”

Alumno 2: “Más lenta. La principal problemática es ponerse de acuerdo en cuando entrar a las distintas etapas, ya que si no entras cuando corresponde a la estimación te pierdes de la primera etapa. Si estuviéramos todos en la misma sala y pudiéramos comunicarnos verbalmente todo el tiempo, sería de mayor utilidad este software. El otro problema es que cuando se acaba el tiempo y no apretaste enviar se pierde tu respuesta, lo mismo si tienes un error en las fechas y no alcanzas a arreglarlo. Haciendo desagradable el asunto. Tampoco tienes la capacidad de detener la estimación si algún error sucede y tienes que esperar a que termine.”

Alumno 3: “En nuestro caso particular, creo que nos tomó MÁS tiempo del usual, pues nos tomó un tiempo acostumbrarnos al software. Sin duda con más uso podría sacarse más provecho.”

Alumno 4: “Ayuda n y creo que se lograron avances rápidos, sin embargo hay muchos tiempos muertos.”

Alumno 5: “Lenta, mucho tiempo muerto, entre que se votan o se escribe se pueden hacer otras cosas. Lo que se hace en una reunión.”

Alumno 6: “Creo que de forma más rápida, debido a las limitaciones de tiempo en cada etapa.”

Alumno 7: “De forma más rápida, ya que solo nos juntamos una vez, completamos los pasos y terminamos.”

Alumno 8: “El método utilizado es sumamente rápido, ya que todos los tiempos se encuentran acotados, lo que en una reunión convencional rara vez ocurre y es difícil de controlar.”

Alumno 9: “De todas maneras fue más rápido, dado que me da la impresión que me estaba apurando mucho, pero esto a veces puede ser malo, dado que, al no tener mucho tiempo o un límite, no dan mucho tiempo para pensar.”

Alumno 10: “El tiempo efectivo gastado yo creo que fue menor, pero porque la cuenta regresiva del tiempo era estresante. Pero no sé de qué tan buena calidad o representatividad sea el acuerdo.”

Alumno 11: “Más lenta por problemas que hubo con la aplicación y los tiempos de cada persona.”

¿Crees que la solución alcanzada es mejor o peor a la alcanzada en una reunión convencional? ¿Por qué?

Alumno 1: “Yo creo que no es la mejor ya que por tiempo muchas veces no se pueden tener todas las ideas.”

Alumno 2: “Dado lo anterior, creo q la solución es tan buena como la que se podría alcanzar en una reunion ordenada y con las herramientas informaticas ad-hoc. Pero claramente eso no siempre se tiene, entonces este software es de utilidad siempre que se mejore la interfaz con el usuario en lo expuesto en el punto anterior.”

Alumno 3: “Creo que, siguiendo un procedimiento similar, en una reunión puede llegarse a una solución igual, pero probablemente, en un tiempo mayor.”

Alumno 4: “Puede ser mejor, porque hay un feedback inmediato al leer todas las opiniones de manera anónima. Se evita el prejuicio y uno se enfoca en lo que quiere: estimar. Aún así si las ideas no son claras se pueden malinterpretar para proyectos más complejos.”

Alumno 5: “Más ordenada tal vez, menos eficiente si. Falta dinamismo, un ping pong mas veloz entre los participantes, se podría apoyar con otras herramientas o tener esto en una habitación con todos reunidos.”

Alumno 6: “Creo que la solución alcanzada es peor que en una reunión convencional, debido a la tremenda presión de tener el contador de segundos mientras se trabaja, sumado a la poca riqueza de los medios de intercambio y discusión del sistema. De hecho, la reunión fue cara a cara.”

Alumno 7: “La solución es igual a la que se habría llegado normalmente, porque sólo éramos 3 personas. Para grupos más grandes las soluciones son mejores.”

Alumno 8: “La solución es mejor si la participación de todos los miembros es apreciada, en especial cuando permite que algunos integrantes puedan interactuar en forma remota. No siempre es posible reunir a todo el equipo de trabajo para discutir problemas y estimar tiempos.”

Alumno 9: “Creo que depende mucho del tiempo y la paciencia, pero creo que el cara a cara, es realmente importante, porque en Chat no siempre es fácil poder expresarse.”

Alumno 10: “Creo que es una buena aproximación. Creo que en las reuniones normales uno puede generar más debate y por ende a lo mejor una solución mejor.”

Alumno 11: “Más lenta por problemas que hubo con la aplicación y los tiempos de cada persona.”

¿Consideras que la interfaz de la aplicación fue adecuada para realizar las actividades de esta tarea? ¿Qué consideraste inadecuado?

Alumno 1: “La interfaz está bien pero se podría mejorar sobre todo las distintas actualizaciones de las ventanas cuando otro las modifica (creación de proyectos).”

Alumno 2: “No lo fue:

- No tienes control de la estimación, si algún error sucede tienes q esperar a q termine.
- Falta diferenciar entre empezar una estimación y entrar a la estimación q ya empezó.
- Falta una forma de poner de acuerdo a todos los usuarios, eso siempre fue un problema para saltar del Chat a la estimación.”

Alumno 3: “En algunos casos, me pareció poco clara, quizás con más acciones visibles de las necesarias, pero en parte puede ser porque era la primera vez que la usaba.”

Alumno 4: “Sip, ta re wena. Súper usable.”

Alumno 5: “La estimación de los tiempos, de repente no carga las estimaciones realizadas. Estas preguntas podrían tener el formato de votación. El que se mostró en la interfaz no era el más adecuado, puede apoyado por colores y aparecer mas ordenado. Un poco tosco lo encontré.”

Alumno 6: “El menú de la izquierda, con la separación de tareas por proyecto y las actividades relacionadas es adecuado. El sistema de Chat no aporta valor a la herramienta, ya que es incómodo, y uno está acostumbrado a otros programas de Chat, por lo que no veo la necesidad de re-inventar la rueda.”

Alumno 7: “Se pueden hacer todas las tareas que se piden, pero en Explorer algunas páginas se ven mal.”

Alumno 8: “El aspecto visual de la interfaz puede ser mejorado, tal vez solucionando los problemas del Chat con los acentos. El resto me pareció adecuado.”

Alumno 9: “Si fue adecuada, pero falta mas señales que indiquen, por ejemplo, cuando una estimación se está haciendo.”

Alumno 10: “No. La aplicación no es clara en cuanto a las “vistas” de las herramientas, no es intuitiva en ese sentido. Además la información que uno ingresa durante la evaluación, queda como texto plano. Creo que sería mejor una estructura por punteos o conceptos, más que dejar que el usuario ingrese lo que piensa en forma de un texto tedioso de leer.”

Alumno 11: “La interfaz estructuralmente es buena, sin embargo, visualmente no lo es. Para empezar los colores hacen que la aplicación parezca antigua. Además, no es fácil de identificar visualmente los distintos aspectos (los botones superiores).”

¿Consideras que esta herramienta es útil para llevar un registro del avance de un proyecto y sus tareas?

Alumno 1: “Yo creo que sirve para un registro pero no es muy útil, ya que un video también lo haría, lo que si se puede rescatar es que queda una especie de acta de lo que se habló y acordó.”

Alumno 2: “No tengo idea si este software permite llevar este tipo de registro. Con lo que hicimos, diría que no lo permite.”

Alumno 3: “Creo que no hice uso de suficientes capacidades del sistema para opinar al respecto.”

Alumno 4: “Lo dije anteriormente, ayuda para proyectos pequeños en que los trabajadores se conocen, pero en proyectos grandes con distintos intereses (por ejemplo que estime un abogado y un médico) puede generar caos. Además si muchos piden reevaluar, se puede

nunca terminar de hacerlo.”

Alumno 5: “Si muchísimo, se pueden hacer controles mas a fondo de lo que se lleva y tener el respaldo de las cosas que fueron prometidas por los integrantes del grupo. Muy útil.”

Alumno 6: “Sí, la herramienta permite llevar un registro del avance del proyecto, ya que maneja las fechas (que suele ser el principal problema en los proyectos), pero no es suficiente por sí sola.”

Alumno 7: “Como queda todo registrado es bueno porque se puede volver a leer lo que se dijo antes.”

Alumno 8: “Completamente. Es importantísimo disponer de registros históricos para poder cotejar a futuro las opiniones emitidas en el pasado por los integrantes del equipo. Además, se visualiza claramente el tiempo invertido en cada una de las tareas constituyentes del proyecto.”

Alumno 9: “El registro del avance si puede ser, pero falta destacar mas los hitos y con más detalles.”

Alumno 10: “No entendí cómo se lleva un registro del avance del proyecto. No vi que abarcara el desarrollo del proyecto, me quedó la impresión de que es solo de la parte de evaluación del proyecto.”

Alumno 11: “Pese a que las herramientas son buenas, se hubiese preferido una reunión en persona, y en este caso, la aplicación se convierte en dispensable”

¿Crees que esta herramienta podría ayudarte a mejorar las estimaciones de tiempo que realices al trabajar en grupo y a aprender de estimaciones pasadas?

Alumno 1: “Yo creo que esta herramienta podría ayudar si se tuviese una especie de patrones de tiempo ya que así es casi lo mismo que se ve en la reuniones”

Alumno 2: “Podría si se mejora en todos los puntos anteriores.”

Alumno 3: “Si, creo que el programa ayuda a estimar mejor en grupo, y creo que ahorra tiempo, pero para eso debe hacerse uso más extensivo del sistema.”

Alumno 4: “La misma respuesta anterior. Puede ayudar, bajo ciertas condiciones. Puede ser catastrófica bajo otras.”

Alumno 5: “Si de todas maneras, le falta pulirse un poco, pero es un diamante en bruto. La idea es buenísima, la forma de hacer cada procedimiento no es tan bueno, y la interfaz la verdad no me agrado.”

Alumno 6: “No. No veo qué parte de la herramienta ayudaría a aprender de las estimaciones pasadas. Además, la estimación de tiempo requiere mucha información adicional no considerada (experiencias previas, capacidades de los individuos, nivel de funcionamiento del equipo, dificultad técnica del proyecto, incertidumbre en los requerimientos, etc.).”

Alumno 7: “Me gustó el sistema de votación, ayuda a llegar a un acuerdo más fácil, y con mejores estimaciones.”

Alumno 8: “Las estimaciones colaborativas siempre son mejores que las estimaciones individuales, ya que se pueden sopesar las diferentes opiniones y llegar a un acuerdo que es más certero.”

Alumno 9: “Puede ser, pero creo que eso de los tiempo es raro, a veces hay personas que terminan rápido, y tienen que esperar mucho y hay otras que se demoran, y no alcanzan.”

Alumno 10: “Creo que para aprender debería mostrar una información en algo así como estadísticas, más que en un log.”

Alumno 11: “En el caso en que hayan muchas personas involucradas en el proyecto, es muy probable que sea útil, ya que hay un respaldo de rectitud intransable en ella. Por otro lado, ya sea porque el proyecto tiene pocas personas y/o éstas puedan juntarse en persona, la aplicación llega a ser dispensable.”

Comentarios positivos de la experiencia o la aplicación

Alumno 1: “Está bastante bien pensada pero se puede mejorar sobre todo, se podría poner un modulo de ejemplos.”

Alumno 2: “Es interesante trabajar con una aplicación para este tipo de problemas. He estado en reuniones de este tipo y siempre es complicado ponerse de acuerdo. Si este software se mejora en su interfaz y manejo, creo q seria de gran utilidad.”

Alumno 3: “”

Alumno 4: “Es súper usable, interfaz muy fácil de entender, colores agradables, auto explicativa.”

Alumno 5: “Siempre es un gusto utilizar nuevos y originales sistemas. El software tiene la esencia de apoyar colaborativamente. Buena la idea de asignarle tiempos a cada tópico (un arma de doble filo). Es un diamante en bruto.”

Alumno 6: “El menú de la aplicación muestra claramente lo proyectos y sus componentes, y permite acceder claramente a las tareas relacionadas.”

Alumno 7: “El sistema es muy rápido para hacer las estimaciones, pero hay que regularle bien los tiempos para que no se alargue mucho.”

Alumno 8: “Me gustó principalmente la visualización del avance de los proyectos mediante franjas de colores.”

Alumno 9: “Es una buena idea, pero habría que ver los resultados en la práctica de un proyecto, dado que se necesita más tiempo de uso, y ver si los usuarios se adecuan a esta nueva práctica.”

Alumno 10: “Sirve para acotar las reuniones y evitar que se alarguen (aunque igual uno podría hacer la reunión varias veces hasta que salga bien).”

Alumno 11: “Consistente. Bastante estable.”

Comentarios negativos de la experiencia o la aplicación

Alumno 1: “Se puede mejorar ya que a veces ingresábamos a la estimación y no siempre podíamos entrar todos juntos.”

Alumno 2: “Súper complicado ponerse acuerdo, sólo una ventana de Chat q cuando sales de ella pierdes el log de lo q ha pasado. Nos costó seguir la ayuda dada en los pasos de la estimación. Qué pasa con la información que uno ingresa, donde quedan los logs, sería bueno poder leer todas esas cosas al momento de tomar una decisión. También el tiempo de espera cuando todos estábamos listo y teníamos que esperar a que termine el tiempo fijado.”

Alumno 3: “Los contadores no deberían comenzar de inmediato al ingresar a la sección, deberían esperar a que todos los participantes estén dentro.”

Alumno 4: “Problemas de concurrencia en el envío de mensajes o con respecto a los logeos de los usuarios.”

Alumno 5: “Los tiempos asignados para cada fase son determinantes en el correcto desarrollo de la reunión. Si estos están mal asignados se torna una lata. Lo primero antes de empezar la sesión es cranean bien el funcionamiento de los tiempos para que la experiencia del usuario se buena. Lo otro sería que el usuario pudiera hacer más de una cosa a la vez.”

Alumno 6: “El contador de segundos es enervante. No me permitió trabajar tranquilo con la aplicación. Sugeriría cambiarlo por un contador de minutos para minimizar el impacto.”

Alumno 7: “Las actividades no terminan inmediatamente cuando ya todos los participantes terminaron y hay que esperar hasta que se acabe el tiempo.”

Alumno 8: “Pienso que se podría mejorar el sistema de comentarios, de manera de poder tener una base de conocimiento para ser usada en proyectos futuros.”

Alumno 9: “Creo que todavía falta hacerle más pruebas de usuario, ojala que con éstas, pueda mejorar.”

Alumno 10: “Creo que a la interfaz le falta bastante, tanto en apariencia como en usabilidad. El flujo dentro de la página no es intuitivo.”

Alumno 11: “Problemas con las horas entre personas, revisar log. Interfaz poco llamativa. Encuesta larga.”