

Título: Creación de una aplicación para el portal de la red social de facebook

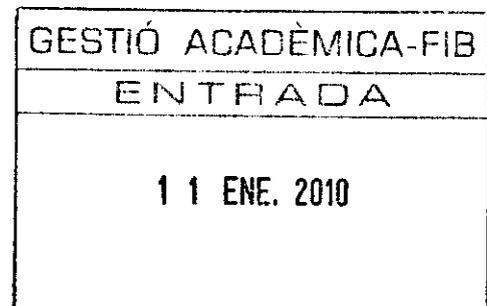
Volumen: 1/1

Alumno: Rubén Vasallo González

Director/Poniente: Carlos Villavieja Prados

Departamento: AC

Data: 15 de enero de 2010





Índice

1. Introducción	6
1.1 Definición del proyecto	6
1.2 Organización de la memoria	6
1.3 Descripción del proyecto.....	7
1.4 ¿Que es Facebook?	8
1.5 Motivaciones	9
1.6 Lugar de trabajo y herramientas	10
2. Análisis Previo	11
2.1 Objetivos del proyecto	11
2.1.1 <i>Implicaciones de los objetivos y del análisis general</i>	11
2.2 Metodologías	12
2.3 Procesos a realizar.....	14
2.4.1 <i>Ficha resumen del proyecto</i>	15
2.4.2 <i>Descomposición en actividades: Orden de precedencia</i>	15
2.4.3 <i>Descomposición en actividades: Duración en días</i>	18
2.4.4 <i>Calendario de actividades</i>	20
2.4.5 <i>Estudio económico</i>	23
2.4.6 <i>Coste temporal real</i>	27
3. Análisis de requerimientos.....	28
3.1 Requerimientos funcionales	28
3.1.1 <i>Tipos de usuarios</i>	28
3.1.2 <i>Funciones disponibles para cada usuario</i>	29
3.2 Requerimientos no funcionales	31
3.2.1 <i>Plataformas de desarrollo</i>	32
3.2.2 <i>Usuarios</i>	34
3.2.3 <i>Eficiencia</i>	34
3.2.4 <i>Seguridad y LOPD</i>	35
4. Especificación	36
4.1 Modelo Conceptual.....	37
4.2 Especificación de los Casos de Uso	39
4.3 Modelo UML.....	49
4.3.1 <i>Capa del modelo dominio y datos</i>	50

4.3.2	<i>Capa del modelo vista</i>	51
4.3.3	<i>Capa del Controlador</i>	52
4.4	Contratos de las operaciones	53
5.	Diseño	77
5.1	La arquitectura utilizada en la aplicación	77
5.1.1	<i>Normalización</i>	79
5.1.2	<i>Diseño del Modelo de Datos</i>	79
5.1.3	<i>Modelo Físico</i>	82
5.1.4	<i>Diagramas de secuencia</i>	83
5.2	Diseño de la interfaz	107
5.2.1	<i>Diseño de la interfaz del portal de El preguntometro</i>	108
6.	Implementación	112
6.1	Introducción	112
6.2	Tecnologías empleadas	112
6.2.1	<i>La tecnología web</i>	112
6.2.2	<i>Cliente / Servidor</i>	113
6.2.3	<i>El servidor web</i>	114
6.2.4	<i>Navegadores (browsers)</i>	114
6.2.5	<i>El SGBD: Sistema Gestor de Bases de Datos</i>	115
6.3	Lenguajes utilizados. Explicación y ejemplos.	115
6.3.1	<i>HTML</i>	116
6.3.2	<i>JavaScript</i>	117
6.3.3	<i>PHP</i>	118
6.3.4	<i>SQL</i>	119
6.3.5	<i>Plataforma Facebook</i>	121
6.4	Pruebas	128
6.5	Diseño Gráfico	129
7.	Acerca de Facebook	130
7.1	¿Exactamente qué datos podemos obtener de los usuarios de Facebook?	130
7.2	¿Por qué puedo acceder a los datos privados de un usuario de Facebook?	131
7.3	¿Qué puedo hacer si quiero utilizar una aplicación de Facebook pero no deseo que esta pueda acceder a ciertos datos privados?	133
8.	Desarrollo en Facebook ¿es un negocio rentable?	134
8.1	Reglas a seguir para realizar una aplicación con Facebook	135

8.2 ¿Realmente es útil la presencia en Facebook para una empresa?	135
8.3 ¿Pero realmente es seguro?.....	136
8.4 Además de la web, otros beneficios de desarrollar en Facebook.....	137
9. Conclusiones.....	138
9.1 Objetivos alcanzados.....	138
9.2 Posibles ampliaciones y mejoras (el trabajo futuro).....	139
Bibliografía	140
ANEXO A: Diagrama de GANTT del proyecto realizado en Microsoft Project	143

ÍNDICE DE FIGURAS

FIG. 1	8
FIG. 2: CICLO DE VIDA CLÁSICO DE UN SISTEMA SOFTWARE.	13
FIG. 3: INTERFAZ DE LA HERRAMIENTA DE DESARROLLO DE FACEBOOK	32
FIG. 4 : INTERFAZ DE LA HERRAMIENTA VISUAL PARADIGM FOR UML.....	33
FIG. 5: INTERFAZ DE LA HERRAMIENTA MYSQL ADMINISTRATOR. PROGRAMA PARA ADMINISTRAR BASES DE DATOS DE MYSQL SERVER	33
FIG. 6: MODELO CONCEPTUAL DEL SISTEMA GLOBAL	38
FIG. 7: DIAGRAMA DE CASOS DE USO USUARIO, EXPERTO, ADMINISTRADOR.....	39
FIG. 8: UN DIAGRAMA SENCILLO QUE MUESTRA LA RELACIÓN ENTRE EL MODELO, LA VISTA Y EL CONTROLADOR.	49
FIG. 9: MODELO CONCEPTUAL DE LA CAPA DEL DOMINIO Y DATOS	50
FIG. 10: MODELO CONCEPTUAL DE LA CAPA DE VISTA	51
FIG. 11: MODELO CONCEPTUAL DE LA CAPA DEL CONTROLADOR	52
FIG. 12: EJEMPLO DE UNA OPERACIÓN DEL SISTEMA.....	53
FIG. 13: MODELO DE DATOS NORMALIZADO.....	81
FIG. 14 : EJEMPLO DE TRADUCCIÓN DE UNA SENTENCIA SQL A LENGUAJE NATURAL.....	83
FIG. 15: PAGINA PRINCIPAL DE LA APLICACIÓN	108
FIG. 16: LISTADO DE PREGUNTAS DE UN TEMA.	108
FIG. 17: VISOR DE UNA PREGUNTA Y LISTADO DE RESPUESTAS DE UNA PREGUNTA.	109
FIG. 18: VISOR DE RESPUESTAS.....	109
FIG. 19: EDITOR DE PREGUNTAS Y RESPUESTAS.....	110
FIG. 20: GESTIÓN DE TEMAS.....	110
FIG. 21: GESTIÓN DE USUARIOS	111
FIG. 22: FUNCIONAMIENTO DE LA ARQUITECTURA CLIENTE / SERVIDOR	113
FIG. 23: USO DE NAVEGADORES (BROWSERS) EN SEPTIEMBRE 2006.....	114
FIG. 24: HISTÓRICO DE USO DEL NAVEGADOR INTERNET EXPLORER	115
FIG. 25: EJEMPLO SENCILLO DE CÓDIGO HTML	116
FIG. 26: EJEMPLO SENCILLO DE CÓDIGO JAVASCRIPT	117
FIG. 27: EJEMPLO SENCILLO DE CÓDIGO PHP	119
FIG. 28: COMO DAR DE ALTA UNA APLICACIÓN EN FACEBOOK, PASO 0.....	122
FIG. 29: COMO DAR DE ALTA UNA APLICACIÓN EN FACEBOOK, PASO 1.....	122
FIG. 30: COMO DAR DE ALTA UNA APLICACIÓN EN FACEBOOK, PASO 2.....	123
FIG. 31: COMO DAR DE ALTA UNA APLICACIÓN EN FACEBOOK, PASO 3.....	124
FIG. 32: IMAGEN DE LA APLICACIÓN WEB DE TESTEO CREADA PARA PODER PROBAR MUCHAS DE LAS FUNCIONALIDADES QUE OFRECE FACEBOOK.....	128
FIG. 33: PARTE DE LA LICENCIA DE CONDICIONES DE USO DE FACEBOOK	131
FIG. 34: AVISO DE FACEBOOK DE QUE LA APLICACIÓN QUE VA A UTILIZAR, PODRÍA ACCEDER A SUS DATOS PERSONALES.	132
FIG. 35: IMAGEN DE LA PANTALLA DE CONFIGURACIÓN DE LA PRIVACIDAD DE UNA CUENTA DE FACEBOOK.	133
FIG. 36: LOGO QUE APARECE EN UNA APLICACIÓN QUE HA SIDO VERIFICADA POR FACEBOOK.....	136
FIG. 37: MOVISTAR OFRECE EN CASI TODOS SUS TERMINALES, LA POSIBILIDAD DE CONECTAR CON FACEBOOK Y UTILIZAR SUS APLICACIONES.	137
FIG. 38: XBOX OFRECE LA POSIBILIDAD DE VISITAR VARIAS REDES SOCIALES COMO TWITTER O FACEBOOK.	137

ÍNDICE DE TABLAS

TABLA 1: FICHA RESUMEN DEL PROYECTO	15
TABLA 2 : ORDEN DE PRECEDENCIA DE LAS TAREAS	18
TABLA 3: DURACIÓN EN DÍAS DE LAS TAREAS.....	20
TABLA 4: CALENDARIO DE ACTIVIDADES	22
TABLA 5: TABLA DE COSTE DE LOS RECURSOS HUMANOS SI SE CONTRATASE UNA EMPRESA DE SOFTWARE	23
TABLA 6: TABLA DE PRECIOS DEL HARDWARE UTILIZADO	24
TABLA 7: TABLA DE PRECIOS DEL SOFTWARE UTILIZADO	24
TABLA 8: TABLA DE COSTE INICIAL.....	27
TABLA 9: TABLA DE COSTE TEMPORAL REAL	27
TABLA 10: HERRAMIENTAS UTILIZADAS PARA CADA CAPA	78
TABLA 11 : EJEMPLO DE DIAGRAMA DE SECUENCIA UML	83
TABLA 12: TECNOLOGÍAS EMPLEADAS	112
TABLA 13: CÓDIGO DEL ARCHIVO XD_RECEIVER.HTM	123
TABLA 14: CÓDIGO A AÑADIR EN TU WEB PARA CONECTAR TU APLICACIÓN CON FACEBOOK	125
TABLA 15: CÓDIGO QUE MUESTRA EL BOTÓN DE CONECTAR CON FACEBOOK	125
TABLA 16: CÓDIGO QUE SE HA DE AÑADIR AL COMIENZO DEL ARCHIVO DE TU WEB	125
TABLA 17: MODIFICACIÓN PARA QUE UNA VEZ LOGADO EL USUARIO, REFRESQUE LA WEB.....	125
TABLA 18: EJEMPLO COMPLETO PARA CONECTAR TU APLICACIÓN A FACEBOOK.....	126
TABLA 19: EJEMPLO DE FUNCIÓN QUE OBTIENE EL NOMBRE DE USUARIO DEL USUARIO LOGADO ACTUALMENTE.....	127
TABLA 20: TRADUCCIÓN AL CASTELLANO DE PARTE DE LA DECLARACIÓN DE DERECHOS Y RESPONSABILIDADES DE FACEBOOK (DEL 28 DE AGOSTO DEL 2009). EL ARTÍCULO COMPLETO LO PUEDE ENCONTRAR EN LA WEB HTTP://WWW.FACEBOOK.COM/TERMS.PHP?REF=PF	135

1. Introducción

Esta introducción pretende acercar al lector a los objetivos y contenidos del proyecto guiándolo por lo que será el trabajo realizado para la obtención de las metas marcadas con el cliente después de las diversas reuniones mantenidas antes del inicio del proyecto.

1.1 Definición del proyecto

Los objetivos del proyecto son los siguientes:

- Realización de un estudio de las funcionalidades que ofrece la plataforma de desarrollo Facebook.
- Evaluar las posibilidades de negocio que esta plataforma puede ofrecer.
 - Empresas que utilizan actualmente esta plataforma.
 - Beneficios que aporta el utilizar esta plataforma.
- Desarrollar una aplicación de demostración que utilice algunas de estas funcionalidades.
 - Diseñar, implementar y publicar la aplicación.
 - Analizar los resultados obtenidos con la aplicación.

1.2 Organización de la memoria

Para organizar esta memoria he tenido en cuenta dos factores, primero la opinión de mi director de proyecto, el cual me ayudo mucho a organizar la memoria y, por otra parte, la que personalmente he considerado más intuitiva, fácil de entender y de utilizar para encontrar cualquier tipo de información. En mi opinión, creo que el objetivo de la documentación de un PFC debe ser, además de contener toda la información necesaria y de interés, hacer inteligible el tema tratado para cualquier persona que la lea, aunque ésta no tenga nociones especialmente avanzadas sobre el tema del que trata. A continuación paso a explicar el por qué de la estructura seguida en la elaboración del presente documento, que está directamente relacionado con los pasos seguidos para la consecución del proyecto.

Lo primero que hice al afrontar este proyecto, una vez confirmado por parte del cliente, fue planificar, pero para hacer esto se necesita previamente definir claramente los objetivos a alcanzar. Estos, en mi caso, surgieron de las diversas reuniones con el cliente. Seguidamente llega una etapa que es, en gran parte, un trabajo individual que

necesita de ciertos: apuntes y matices que no se habían tratado suficiente en primera instancia: el análisis de requerimientos y la especificación formales (define el qué quiere el cliente). Todos los puntos de esta fase se hicieron bajo la supervisión del cliente, es decir, cada análisis y especificación del proyecto no se podía realizar si el análisis o especificación anterior no había sido aprobado por el cliente. Por ello, se realizaron muchas reuniones con el cliente para que este diera el visto bueno. Quería destacar que la parte de especificación se ha realizado siguiendo el modelo **UML(Unified Modeling Language)** visto en la facultad en la asignatura de “Enginyeria del software”.

Una vez tuve todo el trabajo anterior perfectamente definido y asimilado, es decir, no dudé en nada de lo que quería el cliente, comencé el proceso de diseño (el cual define el cómo se dará solución al qué requerido por el cliente, formalizando en el punto anterior) tanto de procesos como de la Base de Datos que albergaría el sistema de información a crear.

Cuando terminé la etapa de diseño pasamos a traducirlo al lenguaje de programación correspondiente que se eligió en la etapa de requerimientos. A esta etapa se le llama etapa de implementación.

Los siguientes puntos serían los que documenta el proceso de prueba y verificación del buen funcionamiento del sistema ya creado.

El último punto es la conclusión donde, entre otras cosas, se han comparado los objetivos planteados con los logrados y se han estudiado posibles ampliaciones y mejoras futuras.

NOTA: La explicación de las decisiones de cada elección sobre las tecnologías empleadas en el desarrollo se han realizado en el apartado de implementación, donde además de explicar en qué consiste cada una y se detallan sus características principales. Se ha pensado que esta es la forma más adecuada y fácil de asimilar a la hora de exponer estos conceptos.

1.3 Descripción del proyecto

Consiste en la creación de una aplicación para el portal de la red social de Facebook y una herramienta de gestión y administración de éste, usando las funcionalidades que ofrece la plataforma “**Facebook**”.

Esta aplicación debe permitir la posibilidad de crear preguntas relacionadas con varias temáticas y que estas puedan ser respuestas añadiendo puntuaciones a los usuarios que participen en estas respuestas. Además de esto, se debe implementar una

herramienta capaz de permitir la gestión, administración y mantenimiento de contenidos de una manera extremadamente sencilla.

La iniciativa de este proyecto surgió por parte del cliente con el objetivo de querer saber más acerca de las posibilidades que ofrece la plataforma de desarrollo de **"Facebook"** y hasta qué punto se le podría sacar beneficio comercial.

También se evaluarán las posibilidades de negocio que ofrece esta plataforma para las empresas.

1.4 ¿Que es Facebook?

Facebook es un sitio web de **redes sociales** creado por **Mark Zuckerberg**. Originalmente era un sitio para estudiantes de la Universidad de Harvard, pero actualmente está **abierto a cualquier persona** que tenga una cuenta de correo electrónico. Los usuarios pueden participar en una o más redes sociales, en relación con su situación académica, su lugar de trabajo o región geográfica.

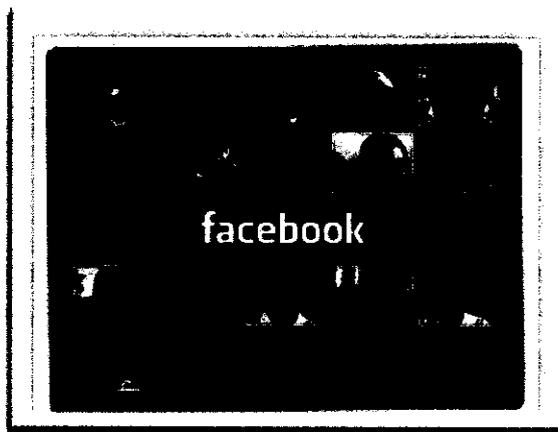


Fig. 1

Ha recibido mucha atención en la blogosfera y en los medios de comunicación al convertirse en una **plataforma** sobre la que **terceros pueden desarrollar aplicaciones y hacer negocio** a partir de la red social. A pesar de ello, existe la preocupación acerca de su posible modelo de negocio, dado que los resultados en publicidad se han revelado como muy pobres.

En noviembre de 2008 ha llegado a tener la mayor cantidad de usuarios registrados¹ en comparación con otros sitios web orientados a estudiantes de nivel superior, teniendo más de 200 millones de usuarios activos en todo el mundo, pese a tener un curioso contrato de "condiciones de uso" del que hablare más adelante en el capítulo 7 (Acerca de Facebook). Es probable que uno de los motivos de la crecida de usuarios sea que, a principios de ese mismo año (febrero del 2008), Facebook lanzó su versión en francés, alemán y español para impulsar su expansión fuera de Estados Unidos, ya que la mayoría de sus usuarios se concentraban en Estados Unidos, Canadá y Gran Bretaña.

¹ Las estadísticas de esta red social pueden consultarse en su propia web de estadísticas <http://www.facebook.com/press/info.php?statistics>

1.5 Motivaciones

Formamos parte de un mundo en constante cambio en el cual siempre se intenta evolucionar, tecnológicamente hablando para, a priori, hacer la vida más agradable a las personas. Invento tras invento, experimento tras experimento, éxito, fracaso,.. un mundo, guste o no, cada vez más deshumanizado donde la tecnología va difuminando las antiguas tradiciones.

En el siglo XXI, el que para muchos se le ha dado el nombre de "Era de la comunicación" o "*Era de la información*", casi cualquier persona (lamentablemente hablo básicamente de países desarrollados) pude tener acceso desde prácticamente cualquier lugar, por recóndito que sea, a un MUNDO, literalmente hablando. Y Cuando digo MUNDO lo hago usando mayúsculas por la grandeza que esto supone: toda la información, servicios y en definitiva, posibilidades habidas y por haber. El principal artífice de este hecho es un "coloso" llamado *Internet*. En muchos ámbitos y sectores se suele decir que no estar en Internet es como no existir. Es por ese motivo que una gran parte de los negocios está optando actualmente, y desde hace ya algún tiempo, por tener su espacio también en la *gran red*, con los objetivos principales de ampliar su negocio, darse a conocer a nivel mundial.

Dentro de este MUNDO mencionado antes, podríamos, quizás, destacar un SUBMUNDO creado por las redes sociales donde, *gente de cualquier rincón del mundo* se da a conocer y comparte su información y experiencias con cualquiera que desee recibirlas. En ese submundo las empresas están haciendo su lugar, creando "*seguidores fieles*" y en definitiva, creando otro nivel de negocio en el que obtener beneficios. En especial, la red Facebook es una de las redes en las que más empresas están captando su atención por el número de usuarios que la utilizan. En ella ven una oportunidad de negocio y una forma literalmente nueva de acercarse a sus clientes y ofertar exactamente lo que el cliente demanda sin intermediarios ni puentes que distorsionen la realidad.

La verdad es que una de las cosas que más me motivó a nivel personal fue la idea de participar en el crecimiento de este nuevo submundo creado por las redes sociales que yo sólo conocía como navegante; el hecho de participar en la construcción de algo que bien utilizado sería una gran herramienta para todos en la construcción de nuevas aplicaciones (generando más oportunidades de negocio). El hecho de que mi proyecto haga referencia a una de las más grandes redes sociales me hace sentir más útil ya que, creo que colaboro en algo positivo para los navegantes.

No obstante no puedo ocultar la satisfacción que me produce el hecho de enfocar mi trabajo desde una perspectiva profesional ofreciendo con él lo que será mi primer servicio enfocado al ámbito profesional, ganando así una experiencia que me será de gran utilidad para mi futuro, espero, más inmediato.

1.6 Lugar de trabajo y herramientas

El presente proyecto comenzó a realizarse en la facultad, aprovechando que mi trabajo estaba a tan solo 5 minutos de ella.

Debido a los requerimientos técnicos (tanto software como hardware) que necesitaba en esa primera etapa no tuve ningún problema ya que mi objetivo era analizar la necesidad del cliente e intentar buscar una solución que satisficiera al cliente final dentro del plazo que tenía para finalizar el proyecto.

Llegado el punto en que tenía que comenzar a implementar la solución que había estado elaborando (esto significaba tener que disponer de una máquina especialmente configurada a mis necesidades –básicamente un servidor Apache, un Sistema Gestor de Bases de Datos MySQL y el intérprete del lenguaje de programación que se iba a usar: PHP), tuve que decidir dónde proseguiría con el trabajo. La decisión estaba básicamente entre la facultad y casa. Finalmente, teniendo en cuenta que, por motivos de movimiento de personas que tiene una facultad tan grande como es la UPC, decidí que lo mejor sería trabajar, para esta fase del proyecto, en casa, por practicidad y comodidad.

2. Análisis Previo

2.1 Objetivos del proyecto

Uno de los objetivos del proyecto, como ya hemos mencionado anteriormente, es la creación de una aplicación para el portal de la red social de Facebook en el que, un usuario pueda participar realizando preguntas sobre temáticas concretas, que podrán responder cualquier usuario y a las que personas expertas darán el visto bueno conforme si la respuesta es válida o no.

Los usuarios también podrán votar según crean si la pregunta y las respuestas que se han ido dando le han parecido interesantes con lo que se podrá comprobar el nivel de participación de la gente en este tipo de plataforma.

Por otro lado añadiré mis objetivos personales que son los siguientes:

- Aprender a fondo un lenguaje de programación de alto nivel en entorno web, ya que es una de las tecnologías más demandadas para el mercado laboral actualmente.
- Aprender también a fondo HTML y JavaScript.
- Obtención de todos los conocimientos relacionados con el entorno de seguridad de servidores web, funcionamiento interno de los browsers (o navegadores) más usados, etc.
- Aprender sobre la legislación vigente y la seguridad de los datos en la web.

2.1.1 Implicaciones de los objetivos y del análisis general

Para iniciar un análisis formal habría primero que introducir ciertos aspectos generales sobre los conceptos y herramientas de las que se va a disponer.

Las páginas web sencillas o estáticas son simples textos ASCII escritos con un sistema de marcas llamado **HTML (HyperText Markup Language)** que se transfieren entre servidores y navegadores (de los usuarios solicitantes de la página) mediante el protocolo **HTTP (HyperText Transfer Protocol)**.

Las páginas estáticas únicamente muestran la información contenida en su fichero de origen escrito con HTML. Esto las hace poco útiles para ciertas cosas, puesto que no ofrece prácticamente interacción con el usuario. En los orígenes de Internet todas las páginas eran estáticas hasta que llegó un punto en que se comenzó a pensar en ampliar las posibilidades de la gran red dotando las páginas de la interacción necesaria

para, entre otras cosas, ofrecerle servicios personalizados on-line. Estas páginas, que en realidad estaban pensadas como aplicaciones (programas) para entorno web, reciben hoy día el nombre de páginas dinámicas, ya que los ficheros que las forman no muestran siempre la misma información, sino que ésta varía en función de lo que solicite el usuario; además tienen la posibilidad de **interaccionar con bases de datos**, lo cual supone una gran ventaja para almacenar cantidades considerables de información. Se comenzaron a programar mediante los lenguajes tradiciones existentes como C y Perl (CGI –Common Gateway Interface), aunque más tarde surgieron nuevos lenguajes específicos para programar en entornos web como **PHP**, ASP, JSP, etc. (En el capítulo 6 implementación se proporciona información mucho más detallada sobre estos temas, los motivos de cada elección y el papel que desempeña el servidor web en el uso de estas tecnologías).

Después de este pequeño análisis concluiremos con lo siguiente:

Nuestro portal no mostrará información estática únicamente, sino que ofrecerá diversas funcionalidades a los usuarios, por tanto, aquí vemos la necesidad de **dotar a nuestro portal de dinamismo**. A esto le debemos añadir el hecho de que deberá tener acceso a una gran cantidad de información, como es un listado de preguntas y respuestas hechas para cada pregunta; por lo tanto tendremos que añadirle la creación de una base de datos que nos permita almacenar y acceder a la información de una manera ordenada, sencilla y eficiente.

2.2 Metodologías

La metodología a seguir para el desarrollo de un sistema software es uno de los factores más importantes a tener en cuenta. La metodología es una de las cosas más visibles que diferencia a un ingeniero o ingeniero técnico en informática, de una persona cualquiera con conocimientos informáticos. Seguramente aunque ambas personas sean capaces de desarrollar una aplicación con las mismas funcionalidades, el ingeniero, posiblemente tarde incluso más tiempo, pero el resultado, por norma general, tiene una calidad muy superior con respecto a lo requerido por el cliente.

Las ventajas de seguir la metodología de una ingeniería, en nuestro caso la ingeniería del software, estudiada en la asignatura de ES (Enginyeria del Software) son muchas:

- ✓ Se preocupa de la fiabilidad y del rendimiento
- ✓ Trata de reducir costes y complejidad
- ✓ Basa sus modelos en teorías matemáticas sólidas
- ✓ Utiliza diagramas formales
- ✓ Se utilizan técnicas probadas que dan resultados precisos
- ✓ ...

En este proyecto se va a tratar de desarrollar un producto basado en el denominado **ciclo de vida clásico de un sistema software**, el cual trata de determinar las diferentes fases por las que se tiene que pasar a fin de conseguir un buen producto software final. A continuación, en un esquema, las etapas de este ciclo de vida:

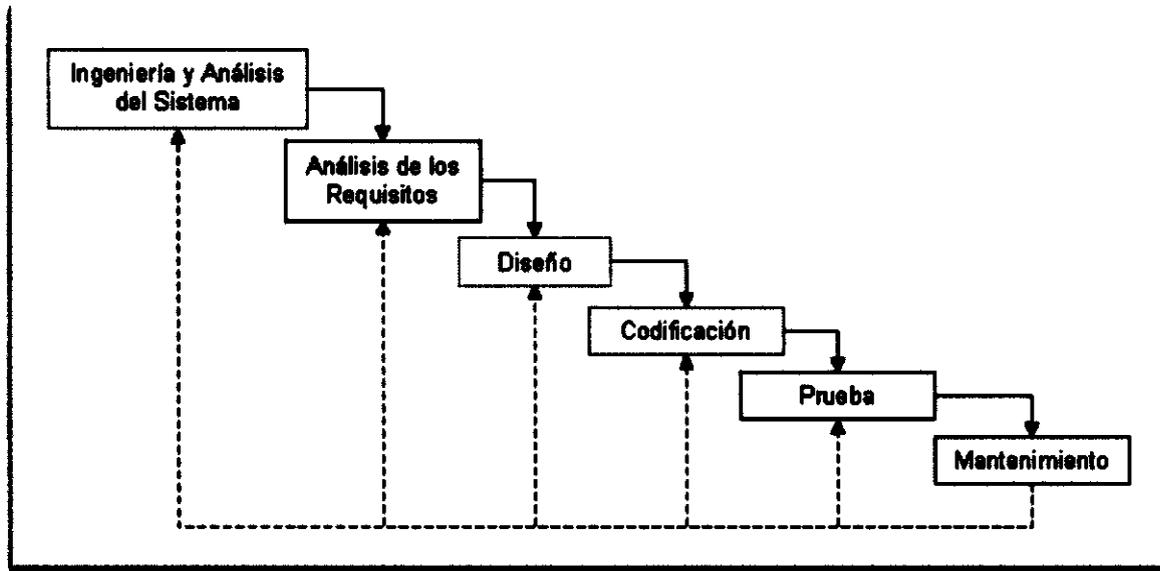
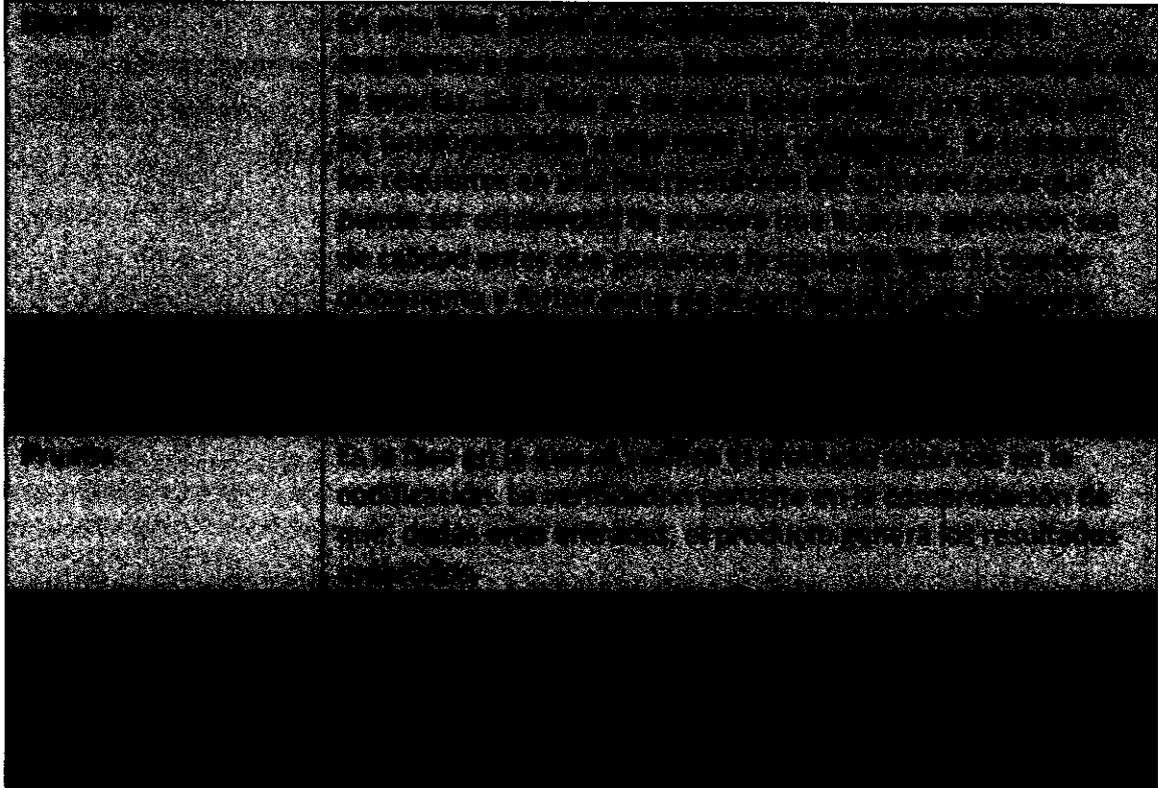


Fig. 2: Ciclo de vida clásico de un sistema software.

Lo más destacable sería la realimentación que se da en cada fase. Ésta consiste en lo siguiente: cuando se detecta un error o carencia, sea en la etapa que sea, se debe rehacer todo lo relacionado con la posible causa desde la fase origen del problema. El objetivo de esta realimentación es conseguir un producto sólido y libre de “parches” de funcionamiento, pues no es lo mismo resolver un error con una pequeña “chapuza”, puesto que en un futuro es probable que genere otro fallo, que retroceder a la causa y subsanarlo por completo. Además, cada vuelta a una fase anterior aumenta los costes en tiempo con lo que se debe atribuir la misma importancia a cada una de las etapas, intentando evitar fallos que nos harán volver atrás.

A continuación veremos una breve descripción de cada etapa:

ETAPA	DESCRIPCIÓN
Ingeniería y Análisis del Sistema	En esta etapa se mantendrán reuniones con los usuarios para determinar los requerimientos del sistema. Es de gran importancia puesto que unos requerimientos bien definidos suponen una base sólida sobre la que construir la solución.



Con respecto a las fases, destacar que uno de los problemas que causa más fracasos en los proyectos de ingeniería es el hecho de no atribuir la suficiente importancia a las dos primeras etapas (**Ingeniería del sistema y Análisis**) que son la base sobre las que sucederán las demás. La causa principal es que se pretenden tener resultados de manera inmediata (por parte de ingenieros y/o clientes) por diferentes motivos, y esto hace que se tengan prisas por comenzar a programar, craso error. En este proyecto se pretende dar a estas dos etapas la importancia que merecen para evitar tener que volver a fases anteriores (realimentación del ciclo de vida).

2.3 Procesos a realizar

A continuación se procederá a describir los **puntos generales** establecidos para la consecución del proyecto:

- 1) Análisis exhaustivo de los requerimientos del cliente y los usuarios finales del producto (administradores y visitantes del portal)
- 2) Análisis de las posibles alternativas de desarrollo (factibilidad de los requerimientos, tecnologías candidatas, etc.)
- 3) Especificación formal del sistema global.
- 4) Revisión de la especificación con el cliente (Para esta etapa se han establecido varios puntos de control)

- 5) Cambios, si proceden, de la especificación hasta que sea la que satisfaga las necesidades del cliente por completo.
- 6) Diseño del sistema global (Para esta etapa se han establecido varios puntos de control)
- 7) Aprendizaje del lenguaje de programación elegido y adquisición de todos los conocimientos específicos del desarrollo de aplicaciones en entorno web (servidores, "arquitectura" cliente/servidor, seguridad, etc.)
- 8) Implementación del sistema (para esta etapa se han establecido un punto de control con el cliente)
- 9) Pruebas exhaustivas del sistema desarrollado
- 10) Depuración del diseño gráfico acorde con las preferencias del cliente.
- 11) Implantación (incluye la generación de un manual para el usuario final).

2.4.1 Ficha resumen del proyecto

- Fecha de inicio:	15/06/2009
- Fecha prevista de fin:	16/02/2010
- Fechas de control previstas:	<ul style="list-style-type: none"> • Durante la etapa de la especificación • Finalización de la etapa de diseño • Finalización de la etapa de implementación
- Esfuerzo previsto:	<ul style="list-style-type: none"> • Horas de análisis: 475 • Horas de programación: 225
- Número de programadores:	1
- Número de analistas:	1
- Total de personas trabajando en el proyecto:	1
- Coste previsto del proyecto (en horas):	<p style="text-align: center;">Horas totales: 140 días * 5 horas/día = 700 horas</p>

Tabla 1: Ficha resumen del proyecto

2.4.2 Descomposición en actividades: Orden de precedencia

Num	Nombre	Predecesoras
1.	Inicio del proyecto	
3.	Reunión con el cliente para especificar el proyecto	1
4.	Análisis previo y de requerimientos	3
5.	Revisar las Condiciones de uso de Facebook	
6.	Hacer un estudio del funcionamiento de las cases de Facebook	3
7.	Hacer un estudio del LOPD	3
8.	Buscar un servidor donde hacer pruebas	3
9.	Hacer un Gantt	3
10.	Redactar el Análisis previo y los análisis de requerimientos en la	3

	memoria	
11.	Margen de maniobra	10
12.	Inscribir el proyecto	3
13.	Margen de maniobra	11
14.	Realización de la especificación (en la Memoria)	
15.	Realización del Modelo de Entidad Relación	3
16.	Margen de maniobra	15
17.	Reunión con el cliente para aprobación del modelo anterior (Punto de control)	15
18.	Aplicar posibles cambios al modelo acordados en la reunión anterior	17
19.	Realización del Diagrama de la capa dominio + datos	18
20.	Margen de maniobra	19
21.	Reunión con el cliente para aprobación del modelo anterior (Punto de control)	19
22.	Aplicar posibles cambios al modelo acordados en la reunión anterior	21
23.	Realización de un boceto de las vistas	22
24.	Margen de maniobra	23
25.	Reunión con el cliente para aprobación de las vistas anteriores (Punto de control)	23
26.	Aplicar posibles cambios a las vistas acordadas en la reunión anterior	25
27.	Realización del Diagrama de la capa de vistas	26
28.	Margen de maniobra	27
29.	Reunión con el cliente para aprobación del modelo anterior (Punto de control)	27
30.	Aplicar posibles cambios al modelo acordados en la reunión anterior	29
31.	Realización del Diagrama de Casos de Uso	30
32.	Realización de un boceto de un diagrama de estados para facilitar la explicación de las funcionalidades	31
33.	Margen de maniobra	32
34.	Reunión con el cliente para aprobación de los modelos anteriores y refrescar el proyecto (Punto de control)	32
35.	Aplicar posibles cambios a los modelos acordados en la reunión anterior	34
36.	Realización de los contratos de las operaciones	35
37.	Reunión con el cliente para aprobación de los contratos anteriores (Punto de control)	36
38.	Aplicar posibles cambios a los contratos acordados en la reunión anterior	37
39.	Realización de los Diagramas de Secuencia del sistema	38
40.	Margen de maniobra	39
41.	Reunión con el cliente para aprobación de los diagramas anteriores (Punto de control)	39
42.	Aplicar posibles cambios a los contratos acordados en la reunión anterior	41
43.	Realizar los contratos de las operaciones	42
44.	Margen de maniobra	43
45.	Reunión con el cliente para aprobación de los diagramas anteriores (Punto de control)	43
46.	Aplicar posibles cambios a los contratos acordados en la reunión anterior	45
47.	Realización del manual de usuario	
48.	Cómo identificarse, tipos de usuarios que hay	46
49.	Cómo crear preguntas	48
50.	Cómo modificar preguntas	49
51.	Cómo consultar preguntas	50

52.	Cómo dar puntuación a un usuario	51
53.	Usuario Experto	
54.	Como seleccionar una respuesta como buena	52
55.	Como quitar una respuesta buena de una pregunta	54
56.	Usuario Administrador	
57.	Como eliminar una pregunta	55
58.	Como eliminar una respuesta	57
59.	Cómo crear temas	58
60.	Cómo modificar un tema	59
61.	Cómo eliminar un tema	60
62.	Cómo cambiar el tipo de usuario	61
63.	Cómo quitar puntos a un usuario	62
64.	Redactar una Introducción para la memoria	46
65.	Explicar los análisis de requerimientos	64
66.	Redactar un estudio económico	65
67.	Redactar un estudio temporal	66
68.	Redactar las conclusiones	67
69.	Redactar las líneas de futuro	68
70.	Redactar la bibliografía	69
71.	Realizar el índice de la memoria	70
72.	Realización del Diseño (en la memoria)	
73.	Redactar el diseño de la Arquitectura utilizada en la aplicación	71
74.	Redactar el diseño de la Interfaz	73
75.	Margen de maniobra	74
76.	Realización de la programación	
77.	Redactar una breve introducción y las tecnologías utilizadas	75
78.	Transformar el diagrama de dominio + datos en sentencias SQL	74
79.	Programar los controladores de Dominio	78
80.	Programar los controladores de Vistas	79
81.	Programar las Vistas	80
82.	Margen de error	81
83.	Crear la base de datos en el servidor	81
84.	Realizar juegos de pruebas en el servidor	83
85.	Reunión con el cliente para demostración de la aplicación (Punto de control)	84
86.	Aplicar posibles cambios a la aplicación acordados en la reunión anterior	85
87.	Redactar las vistas finales en la memoria	86
88.	Redactar las conclusiones	83
89.	Realización de la introducción de la memoria	
90.	Redactar la definición del proyecto	88
91.	Redactar la organización de la memoria	88
92.	Redactar una breve descripción del proyecto	88
93.	Redactar una breve explicación de que es FACEBOOK	88
94.	Redactar las motivaciones	88
95.	Redactar el lugar de trabajo y herramientas	88
96.	Redactar el Glosario	90;91;92;93;94;95
97.	Redactar la Bibliografía	90;91;92;93;94;95
98.	Realizar la preparación de la presentación	97;96
99.	Margen de maniobra	98

100.	Presentación final	98
101.	Fin del proyecto	100

Tabla 2 : Orden de precedencia de las tareas

2.4.3 Descomposición en actividades: Duración en días

La planificación se ha efectuado en días puesto que es una medida mucho más intuitiva a la hora de tomar conciencia del esfuerzo estimado. Para conseguir la estimación en horas simplemente debemos recordar que la jornada laboral de este proyecto está establecida en 5 horas, por tanto se establece esta simple ecuación:

$$\text{Nº horas} = \text{nº días} \times 5$$

Num	Nombre	ANALISTA	PROGRAMADOR
1.	Inicio del proyecto		
3.	Reunión con el cliente para especificar el proyecto	1	
4.	Análisis previo y de requerimientos		
5.	Revisar las Condiciones de uso de Facebook	1	
6.	Hacer un estudio del funcionamiento de las casas de Facebook	1	
7.	Hacer un estudio del LOPD	1	
8.	Buscar un servidor donde hacer pruebas	1	
9.	Hacer un Gantt	2	
10.	Redactar el Análisis previo y los análisis de requerimientos en la memoria	3	
11.	Margen de maniobra		
12.	Inscribir el proyecto		
13.	Margen de maniobra		
14.	Realización de la especificación (en la Memoria)		
15.	Realización del Modelo de Entidad Relación	2	
16.	Margen de maniobra		
17.	Reunión con el cliente para aprobación del modelo anterior (Punto de control)	1	
18.	Aplicar posibles cambios al modelo acordados en la reunión anterior	1	
19.	Realización del Diagrama de la capa dominio + datos	2	
20.	Margen de maniobra		
21.	Reunión con el cliente para aprobación del modelo anterior (Punto de control)	1	
22.	Aplicar posibles cambios al modelo acordados en la reunión anterior	1	
23.	Realización de un boceto de las vistas	2	
24.	Margen de maniobra		
25.	Reunión con el cliente para aprobación de las vistas anteriores (Punto de control)	1	
26.	Aplicar posibles cambios a las vistas acordadas en la reunión anterior	1	
27.	Realización del Diagrama de la capa de vistas	2	
28.	Margen de maniobra		

29.	Reunión con el cliente para aprobación del modelo anterior (Punto de control)	1	
30.	Aplicar posibles cambios al modelo acordados en la reunión anterior	1	
31.	Realización del Diagrama de Casos de Uso	1	
32.	Realización de un boceto de un diagrama de estados para facilitar la explicación de las funcionalidades	1	
33.	Margen de maniobra		
34.	Reunión con el cliente para aprobación de los modelos anteriores y refrescar el proyecto (Punto de control)	2	
35.	Aplicar posibles cambios a los modelos acordados en la reunión anterior	1	
36.	Realización de los contratos de las operaciones	3	
37.	Reunión con el cliente para aprobación de los contratos anteriores (Punto de control)	1	
38.	Aplicar posibles cambios a los contratos acordados en la reunión anterior	1	
39.	Realización de los Diagramas de Secuencia del sistema	3	
40.	Margen de maniobra		
41.	Reunión con el cliente para aprobación de los diagramas anteriores (Punto de control)	1	
42.	Aplicar posibles cambios a los contratos acordados en la reunión anterior	1	
43.	Realizar los contratos de las operaciones	3	
44.	Margen de maniobra		
45.	Reunión con el cliente para aprobación de los diagramas anteriores (Punto de control)	1	
46.	Aplicar posibles cambios a los contratos acordados en la reunión anterior	1	
47.	Realización del manual de usuario		
48.	Cómo identificarse, tipos de usuarios que hay		1
49.	Cómo crear preguntas		1
50.	Cómo modificar preguntas		1
51.	Cómo consultar preguntas		1
52.	Cómo dar puntuación a un usuario		1
53.	Usuario Experto		
54.	Como seleccionar una respuesta como buena		1
55.	Como quitar una respuesta buena de una pregunta		1
56.	Usuario Administrador		
57.	Como eliminar una pregunta		1
58.	Como eliminar una respuesta		1
59.	Cómo crear temas		1
60.	Cómo modificar un tema		1
61.	Cómo eliminar un tema		1
62.	Cómo cambiar el tipo de usuario		1
63.	Cómo quitar puntos a un usuario		1
64.	Redactar una Introducción para la memoria	2	
65.	Explicar los análisis de requerimientos	4	
66.	Redactar un estudio económico	2	
67.	Redactar un estudio temporal	1	
68.	Redactar las conclusiones	1	
69.	Redactar las líneas de futuro	2	
70.	Redactar la bibliografía	1	

71.	Realizar el índice de la memoria	2	
72.	Realización del Diseño (en la memoria)		
73.	Redactar el diseño de la Arquitectura utilizada en la aplicación	5	
74.	Redactar el diseño de la Interfaz	1	
75.	Margen de maniobra		
76.	Realización de la programación		
77.	Redactar una breve introducción y las tecnologías utilizadas		1
78.	Transformar el diagrama de dominio + datos en sentencias SQL		1
79.	Programar los controladores de Dominio		10
80.	Programar los controladores de Vistas		10
81.	Programar las Vistas		5
82.	Margen de error		
83.	Crear la base de datos en el servidor		1
84.	Realizar juegos de pruebas en el servidor		2
85.	Reunión con el cliente para demostración de la aplicación (Punto de control)	1	
86.	Aplicar posibles cambios a la aplicación acordados en la reunión anterior		1
87.	Redactar las vistas finales en la memoria	1	
88.	Redactar las conclusiones	1	
89.	Realización de la introducción de la memoria		
90.	Redactar la definición del proyecto	1	
91.	Redactar la organización de la memoria	1	
92.	Redactar una breve descripción del proyecto	1	
93.	Redactar una breve explicación de que es FACEBOOK	1	
94.	Redactar las motivaciones	1	
95.	Redactar el lugar de trabajo y herramientas	1	
96.	Redactar el Glosario	1	
97.	Redactar la Bibliografía	2	
98.	Realizar la preparación de la presentación	15	
99.	Margen de maniobra		
100.	Presentación final	1	
101.	Fin del proyecto		

Tabla 3: Duración en días de las tareas

2.4.4 Calendario de actividades

Num	Nombre	FECHA DE INICIO
1.	Inicio del proyecto	15/06/09
3.	Reunión con el cliente para especificar el proyecto	15/06/09
4.	Análisis previo y de requerimientos	
5.	Revisar las Condiciones de uso de Facebook	16/06/09
6.	Hacer un estudio del funcionamiento de las casas de Facebook	17/06/09
7.	Hacer un estudio del LOPD	18/06/09
8.	Buscar un servidor donde hacer pruebas	22/06/09
9.	Hacer un Gantt	22/06/09
10.	Redactar el Análisis previo y los análisis de requerimientos en la memoria	22/06/09
11.	Margen de maniobra	
12.	Inscribir el proyecto	01/07/09

13.	Margen de maniobra	
14.	Realización de la especificación (en la Memoria)	
15.	Realización del Modelo de Entidad Relación	03/07/09
16.	Margen de maniobra	
17.	Reunión con el cliente para aprobación del modelo anterior (Punto de control)	08/07/09
18.	Aplicar posibles cambios al modelo acordados en la reunión anterior	09/07/09
19.	Realización del Diagrama de la capa dominio + datos	10/07/09
20.	Margen de maniobra	
21.	Reunión con el cliente para aprobación del modelo anterior (Punto de control)	15/07/09
22.	Aplicar posibles cambios al modelo acordados en la reunión anterior	16/07/09
23.	Realización de un boceto de las vistas	17/07/09
24.	Margen de maniobra	
25.	Reunión con el cliente para aprobación de las vistas anteriores (Punto de control)	22/07/09
26.	Aplicar posibles cambios a las vistas acordadas en la reunión anterior	23/07/09
27.	Realización del Diagrama de la capa de vistas	24/07/09
28.	Margen de maniobra	
29.	Reunión con el cliente para aprobación del modelo anterior (Punto de control)	29/07/09
30.	Aplicar posibles cambios al modelo acordados en la reunión anterior	30/07/09
31.	Realización del Diagrama de Casos de Uso	01/09/09
32.	Realización de un boceto de un diagrama de estados para facilitar la explicación de las funcionalidades	02/09/09
33.	Margen de maniobra	
34.	Reunión con el cliente para aprobación de los modelos anteriores y refrescar el proyecto (Punto de control)	04/09/09
35.	Aplicar posibles cambios a los modelos acordados en la reunión anterior	08/09/09
36.	Realización de los contratos de las operaciones	09/09/09
37.	Reunión con el cliente para aprobación de los contratos anteriores (Punto de control)	14/09/09
38.	Aplicar posibles cambios a los contratos acordados en la reunión anterior	15/09/09
39.	Realización de los Diagramas de Secuencia del sistema	16/09/09
40.	Margen de maniobra	
41.	Reunión con el cliente para aprobación de los diagramas anteriores (Punto de control)	22/09/09
42.	Aplicar posibles cambios a los contratos acordados en la reunión anterior	23/09/09
43.	Realizar los contratos de las operaciones	24/09/09
44.	Margen de maniobra	
45.	Reunión con el cliente para aprobación de los diagramas anteriores (Punto de control)	30/09/09
46.	Aplicar posibles cambios a los contratos acordados en la reunión anterior	01/10/09
47.	Realización del manual de usuario	
48.	Cómo identificarse, tipos de usuarios que hay	02/10/09
49.	Cómo crear preguntas	05/10/09
50.	Cómo modificar preguntas	06/10/09
51.	Cómo consultar preguntas	07/10/09
52.	Cómo dar puntuación a un usuario	08/10/09
53.	Usuario Experto	
54.	Como seleccionar una respuesta como buena	09/10/09

55.	Como quitar una respuesta buena de una pregunta	12/10/09
56.	Usuario Administrador	
57.	Como eliminar una pregunta	13/10/09
58.	Como eliminar una respuesta	14/10/09
59.	Cómo crear temas	15/10/09
60.	Cómo modificar un tema	16/10/09
61.	Cómo eliminar un tema	19/10/09
62.	Cómo cambiar el tipo de usuario	20/10/09
63.	Cómo quitar puntos a un usuario	21/10/09
64.	Redactar una Introducción para la memoria	22/10/09
65.	Explicar los análisis de requerimientos	26/10/09
66.	Redactar un estudio económico	30/10/09
67.	Redactar un estudio temporal	03/11/09
68.	Redactar las conclusiones	04/11/09
69.	Redactar las líneas de futuro	05/11/09
70.	Redactar la bibliografía	09/11/09
71.	Realizar el índice de la memoria	10/11/09
72.	Realización del Diseño (en la memoria)	
73.	Redactar el diseño de la Arquitectura utilizada en la aplicación	12/11/09
74.	Redactar el diseño de la Interfaz	19/11/09
75.	Margen de maniobra	20/11/09
76.	Realización de la programación	
77.	Redactar una breve introducción y las tecnologías utilizadas	23/11/09
78.	Transformar el diagrama de dominio + datos en sentencias SQL	24/11/09
79.	Programar los controladores de Dominio	25/11/09
80.	Programar los controladores de Vistas	09/12/09
81.	Programar las Vistas	23/12/09
82.	Margen de error	
83.	Crear la base de datos en el servidor	04/01/10
84.	Realizar juegos de pruebas en el servidor	05/01/10
85.	Reunión con el cliente para demostración de la aplicación (Punto de control)	11/01/10
86.	Aplicar posibles cambios a la aplicación acordados en la reunión anterior	12/01/10
87.	Redactar las vistas finales en la memoria	15/01/10
88.	Redactar las conclusiones	18/01/10
89.	Realización de la introducción de la memoria	
90.	Redactar la definición del proyecto	19/01/10
91.	Redactar la organización de la memoria	19/01/10
92.	Redactar una breve descripción del proyecto	19/01/10
93.	Redactar una breve explicación de que es FACEBOOK	19/01/10
94.	Redactar las motivaciones	19/01/10
95.	Redactar el lugar de trabajo y herramientas	19/01/10
96.	Redactar el Glosario	20/01/10
97.	Redactar la Bibliografía	20/01/10
98.	Realizar la preparación de la presentación	22/01/10
99.	Margen de maniobra	
100.	Presentación final	16/02/10
101.	Fin del proyecto	16/02/10

Tabla 4: Calendario de actividades

2.4.5 Estudio económico

El estudio económico hace referencia al precio total real que tendría para el cliente un proyecto como éste. Se ha supuesto que se tuviese que comprar todo lo necesario (básicamente la contratación de un host para la publicación de una web).

Basándonos en la anterior planificación de actividades se han añadido las siguientes suposiciones:

- Distinción entre el sueldo por hora de analista y programador.
- Distinción entre los costes para analista y programador en función del coste de mercado (precio al que “según creo” se pagan las horas de análisis y programación) y el coste real que ha tenido.

El coste del proyecto se va a dividir en el coste de los recursos humanos y el coste de los recursos técnicos (básicamente los gastos derivados en otros conceptos como el alojamiento).

❁ **Coste de los recursos humanos según planificación inicial**

Éste es el coste de los recursos humanos que supondría para el cliente el hecho de contratar una empresa software para el desarrollo de este producto. Aquí se define “según creo” a cuanto se paga en el mercado actualmente la hora de analista y programador (en media) cuando se subcontratan por medio de una empresa especializada en desarrollo de aplicaciones software.

Para este apartado vamos a tener en cuenta que el precio por hora medio estándar de un analista se está pagando actualmente a 40 €, mientras que la hora de programador está a 20 €. Después de esto sólo tenemos que hacer los cálculos siguientes:

Recurso	Núm.	Importe estándar/hora	Número de horas	Importe total
Analista	1	40 €	475	19.000 €
Programador	1	20 €	225	4.500 €
TOTAL	2	-	700	23.500 €

Tabla 5: Tabla de coste de los recursos humanos si se contratase una empresa de software

❖ Coste de los recursos técnicos según planificación inicial

Hardware

Para el desarrollo de este proyecto no se ha tenido que destinar dinero a la adquisición de un equipo informático específico que haga las funciones de servidor web, ya que para las pruebas de la aplicación, he configurado mi equipo personal como servidor web para alojar la web y hacer los juegos de prueba.

De todos modos vamos a describir que hubiese pasado con el concepto de costes de hardware (además del de software) en el caso de que el cliente hubiese contratado los servicios de una empresa dedicada al desarrollo software.

Servidor HP ProLiant ML350 G6	1500 €
COSTE TOTAL HARDWARE	1500 €

Tabla 6: Tabla de precios del Hardware utilizado

Software

Tipo	Nombre	Precio(IVA incluido)
Operativo	Windows XP Profesional	139 €
Servidor Web	Apache	0 €
BD	MySQL	0 €
Oficina	MS Office Standard 2007	569 €
Navegador	MS Internet Explorer 8	0 €
Planificación	Microsoft Project Standard 2007	779 €
Especificación y diseño	Visual Paradigm for UML 6.4	256 €
TOTAL		1743 €

Tabla 7: Tabla de precios del Software utilizado

Otros conceptos

Servicios de hosting:

En un principio, se había acordado con el cliente la contratación de un hosting gratuito para la publicación final de la web pero, debido a que estos servidores no ofrecen ningún tipo de garantía ni seguridad, se decidió contratar los servicios de hosting a la empresa namecheap². A continuación detallamos los costes de la contratación de dicho servicio.

El pack "hosting básico"³ que incluye 5 GB de espacio en disco, 250 GB de transferencia mensual, 50 buzones de correo, 5 cuentas FTP, 15 Bases de datos MySQLServer, Posibilidad de tener la web programada en PHP, ASP,Perl, Coste del alquiler 3,33 \$/mes (2,28 €/mes Tipo de cambio 1 € = 1,457 USD) si se contrata durante un año.

Total Servicios Internet	2,28 €/mes
---------------------------------	-------------------

❖ *Coste total del proyecto según planificación inicial*

Llegados aquí nos encontramos ante uno de los puntos más conflictivos por los motivos que vamos a describir a continuación.

Tras haber realizado un estudio exhaustivo en el que he mantenido contactos con informáticos profesionales que han trabajado como autónomos y como empleados de empresas de desarrollo software he llegado a las siguientes conclusiones sobre lo que al precio presupuestado final de un sistema software se refiere:

El coste de los recursos informáticos para el desarrollo (básicamente en lo que a hardware y software se refiere), en general, puede convertirse en uno de los más significativos dentro de un proyecto. **Los precios de los productos para el desarrollo son caros**, pero además no se pueden atribuir exclusivamente a un proyecto concreto puesto que, **generalmente, lo podemos emplear para posteriores proyectos**, por tanto, un cliente, de entrada, no va a consentir que se le cobren todos estos recursos, al menos de manera total. Soluciones:

1. Contarlo y cobrarlo como un gasto basado en la amortización del recurso en cuestión.
2. Incluirlo parcialmente y de manera indirecta en el sueldo de los recursos humanos.
3. Considerarlo como una inversión a medio plazo cargando el valor del software en el beneficio del proyecto.

² www.namecheap.com

³ Más información en <http://www.namecheap.com/learn/webhosting/web-hosting.asp>

El primero consistiría en calcular la amortización del recurso de desarrollo en función de su vida útil (que en el caso del software acostumbra a ser muy corto debido a que continuamente se sacan nuevas versiones más completas de los productos). Es decir, **calculando el periodo de vida útil y la amortización resultante por días (por ejemplo), podríamos cobrarle al cliente los días que se usa el software para su proyecto.** Así podríamos hacer con todos los proyectos para los que se usase dicho recurso consiguiendo repartir el peso del coste y no cargándolo todo a uno.

El problema principal es la complejidad y la precisión que requiere, amén que el cálculo del periodo de vida útil y, por tanto, de la amortización es un tanto dependiente del mercado (lo que exija cada proyecto puede ser totalmente diferente) y del ritmo al que se actualice el recurso, y si nos fijamos, ninguno de los dos está en nuestras manos.

El segundo consistiría en usar el factor recursos humanos para cubrir el coste de los recursos técnicos incrementando el precio de los primeros. En el presupuesto solamente se hace referencia a las horas trabajadas y el precio por hora (tanto de analista como de programador) parece un tanto más caro pero en realidad es porque incluye los otros costes de manera indirecta.

El tercero es el más utilizado y extendido y consiste en considerar la compra de software como una inversión a medio plazo, es decir, que el coste de la compra de software repercutirá directamente en el beneficio nuestro. Pero, como es probable que ese software lo utilices en otros proyectos (y más en este caso en el que el software de pago es bastante común y usable en casi cualquier tipo de proyecto informático), el precio del software no repercutiría directamente en el cliente final y lo recuperaremos seguramente si unimos el beneficio de dos o más proyectos juntos. Es una jugada arriesgada pero puede resultar efectiva si tenemos en cuenta que en la mayoría de proyectos, al precio final de este, hay que sumarle un 15% por posibles problemas que puedan surgir en el “que no tienen por qué surgir”, por lo que es probable que el gasto se recupere antes.

Finalmente ponemos una tabla resumen del presupuesto inicial:

Concepto	Coste
Recursos Humanos	23.500,00 €
Recursos Tecnológicos	27,36 €
Total Inicial	23.527,36 €
Margen del 16%	(16%) 3.764,38 €
TOTAL	27.291,64 €

Tabla 8: Tabla de coste inicial

NOTA: La cifra anterior es ficticia ya que para la realización de este proyecto, se me ha facilitado todo el software necesario de manera gratuita por la facultad, el hardware necesario ya lo tenía para mi propio uso y el servicio de hosting lo he contratado.

2.4.6 Coste temporal real

Finalmente, solo ha sido necesario realizar el pago del hosting para tener la web online y poder observar que aceptación tiene una web realizada con esta plataforma. A continuación paso a detallar el coste de temporal real del proyecto.

Concepto	Coste
Hosting	27,36 €
TOTAL	27,36 €

Tabla 9: Tabla de coste temporal real

3. Análisis de requerimientos

Ésta es la parte de un proyecto encargada de determinar los requerimientos o condiciones deseadas por el cliente que debe cumplir el sistema a desarrollar para conseguir el objetivo o solucionar el problema planteado.

Este apartado es el resultado de diversas reuniones con el cliente (etapa de Ingeniería de Sistema), alguno de los datos de las cuales aparecen reflejadas ya en el anterior punto (Análisis previo), y de un posterior proceso de formalización llevado a cabo por mí en calidad de analista.

Llegados este punto hay que destacar que debido a la necesidad de hacer el mantenimiento del portal extremadamente sencillo para cualquier administrador y teniendo en cuenta, como ya hemos dicho, que tendrá que gestionar una base de datos con datos de usuarios y preguntas con respuestas catalogadas en varias temáticas, se ha llegado a la conclusión que lo adecuado sería la creación de una parte administrativa (de uso privado por los administradores del portal) en la propia web donde se podrán administrar dichas tareas de una manera muy clara e intuitiva.

3.1 Requerimientos funcionales

Aquí se van a determinar absolutamente todas las funcionalidades que deberá cubrir el sistema software. Para hacerlo más estructurado se han definido primero los diferentes tipos de usuarios y, posteriormente, para cada uno se han descrito todas las funcionalidades (casos de uso) a las que podrás acceder.

3.1.1 Tipos de usuarios

Se han entendido básicamente 3 tipos de usuario diferentes que intervendrán en las acciones del portal y del sistema de administración:

- **Usuario normal:** Con este nombre nos referiremos al navegante que está visitando el portal y se ha identificado anteriormente en el sistema de identificación del portal de Facebook.
- **Usuario experto:** Este será el usuario que tendrá (además de las mismas funcionalidades que el usuario normal) el privilegio de dar el visto bueno una respuesta de una pregunta en concreto.
- **Usuario administrador:** Este usuario es el encargado final de gestionar y administrar los contenidos que ofrece el portal. Es decir, se encarga de la

gestión de la temática, de los privilegios de los usuarios y de la gestión de preguntas y respuestas del portal.

NOTA: Es importante recordar que el usuario administrador no será informático. Sus conocimientos de administración de web serán muy limitados, es por tanto una de las principales metas, como ya se ha comentado en los objetivos, el hecho de hacer extremadamente sencillo el mantenimiento del portal.

3.1.2 Funciones disponibles para cada usuario

🕒 Usuario Normal

1. Consultar Preguntas: Esta función permite al usuario visualizar todas las preguntas de una temática concreta que se hallen introducidas en el portal.
2. Consultar Respuestas: Esta función nos permitirá visionar todas las respuestas de una pregunta concreta que se hallen introducidas en el sistema. Si algún usuario experto ha seleccionado alguna de las respuestas como buena, esta saldrá destacada entre las demás.
3. Nueva Pregunta: Con esta funcionalidad, el usuario podrá introducir una pregunta de una de las temáticas concretas que tenga el sistema en el portal para que otros usuarios, (o el mismo) puedan introducir respuestas a dicha pregunta.
4. Nueva Respuesta: Al igual que la anterior, esta funcionalidad tendrá por objetivo dar de alta una respuesta a una pregunta en concreto.
5. Modificar Pregunta: Dada una pregunta hecha por el usuario que esta logado en el portal, podrá ser modificada por el mismo dentro de los 15 minutos después de que se halla introducido la pregunta.
6. Modificar Respuesta: Al igual que la anterior, el usuario podrá modificar una respuesta hecha por el dentro del plazo de los 15 minutos posteriores de haberla introducido en el sistema.
7. Dar puntuación a un usuario: Un usuario podrá dar puntos a otro usuario en concreto (que no sea el mismo) como máximo una vez al día respecto a una pregunta/respuesta en concreto. Es decir, un usuario podrá dar puntuación a muchos usuarios siempre y cuando al usuario al que se vaya a dar esta puntuación no sea el mismo que esta logado y no le haya dado puntos con anterioridad dentro del mismo día respecto a una pregunta/respuesta en concreto.

✪ Usuario Experto

Además de las funcionalidades del usuario normal, el experto podrá hacer las siguientes acciones:

1. Validar una respuesta: Dada una respuesta concreta de una pregunta concreta, este podrá dar por buena dicha pregunta para que salga destacada entre las otras que pueda haber.

✪ Usuario Administrador

Además de las funcionalidades del usuario experto, el administrador podrá hacer las siguientes acciones:

1. Modificar una Pregunta: Sera la misma funcionalidad que la explicada en el usuario normal pero sin la restricción de tiempo.
2. Modificar una respuesta: Sera la misma funcionalidad que la explicada en el usuario normal pero sin la restricción de tiempo.
3. Ocultar una Pregunta: Con esta función, pretendemos que la pregunta junto con todas las respuesta que pueda tener, dejen de ser visible en la web. La pregunta y sus respuestas no se borrarán de la Base de datos para tener un histórico de preguntas y respuestas.
4. Ocultar una Respuesta: Al igual que la anterior, esta funcionalidad no pretende eliminar la respuesta sino que esta deje de ser visible en la web.
5. Dar de alta un Tema: Esta función dará de alta una nueva temática en el portal para que los demás usuarios puedan introducir nuevas preguntas y respuestas en el.
6. Dar de baja un Tema: Con esta funcionalidad, lo que se pretende es que un tema con todas sus preguntas dejen de ser visibles en la web. De la Base de datos no se borrarán ya que así quedara constancia de las preguntas y respuestas que se hicieron de aquella temática.
7. Listar los usuarios: Esta función permitirá listar todos los usuarios que se han identificado alguna vez en la web. Cuando un usuario inicie sesión por primera vez, este asumirá el rol de usuario normal.
8. Cambiar el tipo de usuario: Dado un usuario concreto, se le podrá cambiar el tipo de rol que asume dentro de la web, (normal, experto o administrador).
9. Quitar puntos a un usuario: Esta funcionalidad permitirá, dado un usuario concreto, quitar puntos (por el motivo que sea).

3.2 Requerimientos no funcionales

Este tipo de requerimientos podríamos decir que son aquellas *cualidades* generales (económicas, interfaces, plataforma donde funcionará, etc.) que debe cumplir el software desarrollado y que son independientes de las funcionalidades específicas que se crearán.

- La principal que se mencionó fue la económica: para el cliente este proyecto es importante, pero principalmente quiere comprobar su verdadera utilidad con respecto a los objetivos que se marcaron, es por eso que no quiere que su inversión sea excesivamente alta. En caso que le reporte una gran utilidad en muchos aspectos no le importara aumentar su inversión y ampliar las funcionalidades de la web.
- La plataforma donde estará alojado el portal puede ser de entorno Windows o Linux: éste es un requerimiento no del cliente, sino que proviene de la empresa de hosting, también llamado ISP (Proveedor de Servicios de Internet), que será la encargada de alojar la aplicación. Solamente destacar que en función de la plataforma usada el lenguaje también tendrá que variar, siendo PHP lo dispuesto para trabajar en Linux y Windows y ASP en Windows solamente (más adelante se concretarán estos temas).
- La interfaz gráfica como ya se ha comentado, ha de ser muy intuitiva y fácil de utilizar en el apartado 3.2.2 se expondrán las razones de este requerimiento ya que tiene que ver básicamente con los usuarios finales.
- La interfaz gráfica del portal ha de ser fácilmente intercambiable (Modelo controlador-vista). Esto se refiere a que debe dejar abierta la posibilidad de cambiar en un futuro y de una manera sencilla el diseño gráfico del portal si se cree oportuno.
- Debe ser sencillo cambiar la plataforma sobre la que se trabaja (portabilidad) por si en un futuro se necesitase ampliar, por ejemplo, el SGBD (Sistema Gestor de Bases de Datos) o cualquier otro elemento.

3.2.1 Plataformas de desarrollo

Facebook Developer

La herramienta que proporciona Facebook permite configurar la conexión entre Facebook y el servidor donde está alojado el portal. Es bastante sencillo de utilizar y existe un manual que ofrece Facebook por si tienes alguna duda. Esta herramienta es gratuita.

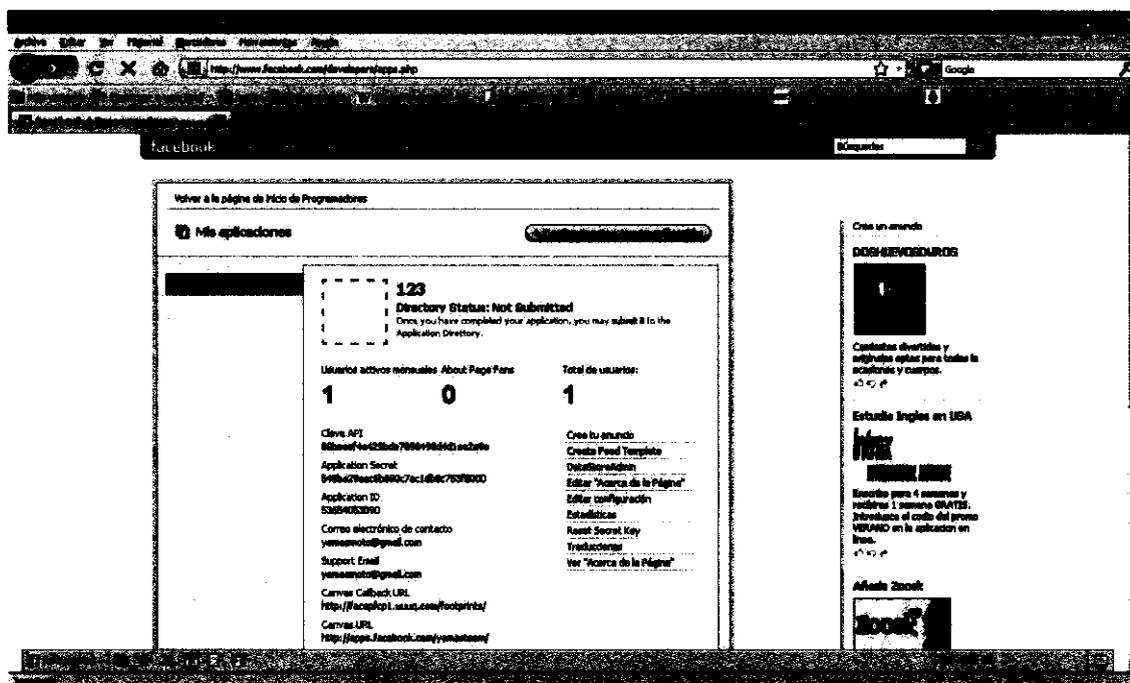


Fig. 3: Interfaz de la herramienta de desarrollo de Facebook

Visual Paradigm for UML

Visual Paradigm es una empresa China que vende software destinado a empresas de desarrollo que facilita el diseño de sus aplicaciones y, en concreto, *Visual Paradigm for UML* es una herramienta de esa casa pensada para hacer más agradable y sencillo el proceso de especificación y diseño del sistema a crear. Utiliza notación UML (UNIFIED MODELING LANGUAGE), la cual permite de una forma muy intuitiva crear los diagramas descriptivos que explican el QUÉ y el CÓMO de lo que se quiere crear.

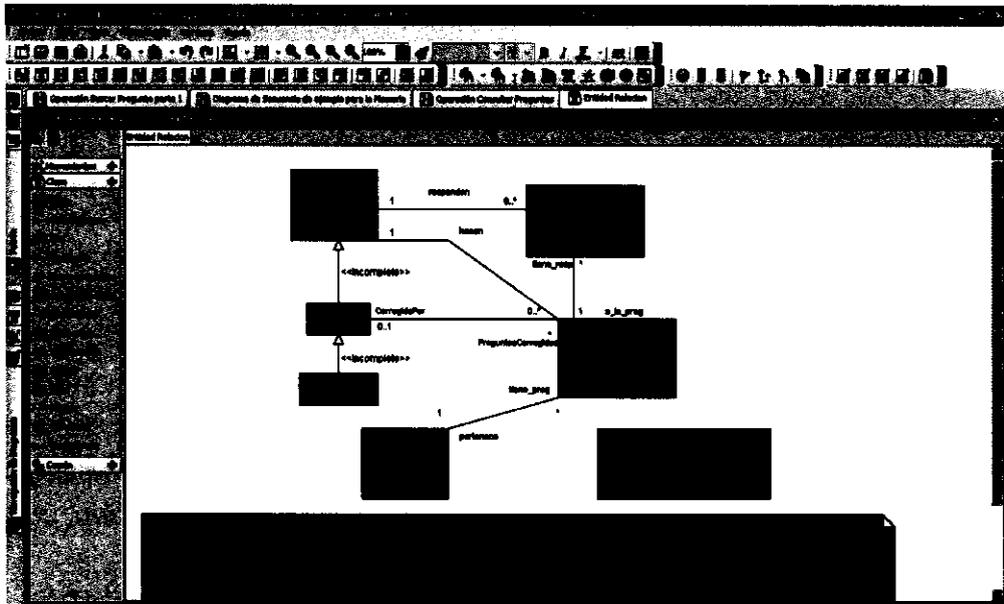


Fig. 4 : Interfaz de la herramienta Visual Paradigm for UML

MySQL

MySQL server es uno de los más utilizados sistema Gestor de Bases de Datos en la actualidad. Es uno de los más útiles y rápidos de utilizar para crear pequeñas aplicaciones de bases de datos como es nuestro caso. Más adelante en el apartado de implementación del presente documento se mostrara el estudio del porqué se ha elegido esta plataforma en detrimento de otras, además de exponer las características más importantes de cada una.

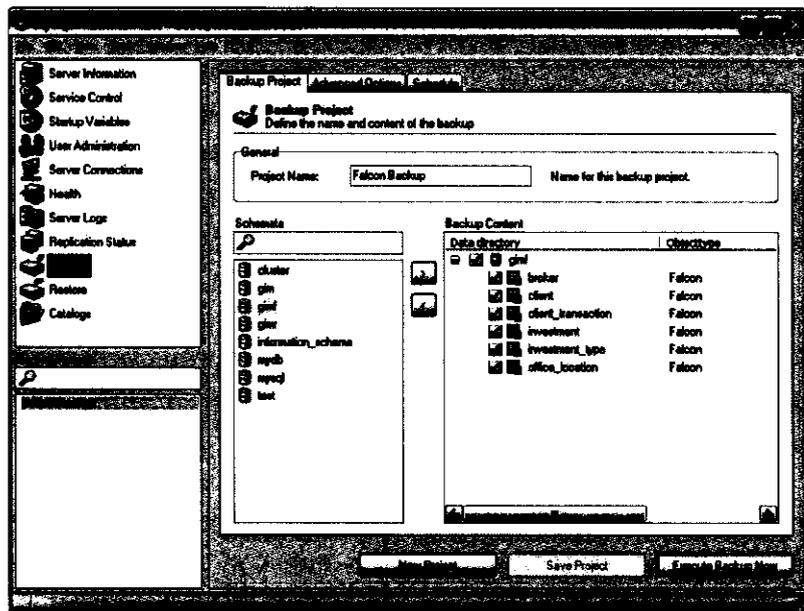


Fig. 5: Interfaz de la herramienta MySQL Administrator. Programa para administrar Bases de Datos de MySQL Server

3.2.2 Usuarios

El cliente desea que el portal sea muy sencillo de manejar también para la gente que no posee una gran experiencia navegando por internet, pudiendo llegar así a más personas. También considera que el hecho de cargar excesivamente de información el portal puede contribuir a que los usuarios se cansen antes de comenzar a leer; muchas veces el ver un texto enorme invita a no leerlo pudiendo así perder la oportunidad de comunicar el mensaje que se pretendía, es por eso que la información principal será breve, concisa y directa. Dejando los textos grandes para informaciones de carácter legal (como las explicaciones sobre el uso de los datos al cual obliga la ley) y similares. Por otro lado, existe un rasgo común entre la gran mayoría de navegantes de todo el mundo: la impaciencia. Es decir, la gente no soporta quedarse esperando demasiado a que una página se visualice, esto será un tema a tratar en el siguiente apartado, puesto que tiene que ver con la eficiencia del sistema.

Por otro lado tenemos la parte de administración del portal. El usuario final de esta parte del proyecto puede ser cualquiera de los usuarios que se conecten a la web (designados por el cliente), por lo que al igual que en la anterior parte, será muy sencillo de manejar. Resumiendo, el objetivo de esta parte del proyecto será que el resultado logre ser intuitivo y sencillo de manejar. Para facilitar aún más el uso de la parte de administración del portal, se facilitará a los usuarios de la misma un manual donde se describirá todo lo necesario para su correcto manejo (Ver punto 7.7).

3.2.3 Eficiencia

A partir del conocimiento de los usuarios finales de los sistemas a desarrollar se puede llegar a las normas que deben regir la eficiencia de las aplicaciones.

Si tenemos en cuenta, por ejemplo, el hecho, comentado justo en el anterior apartado, de la “impaciencia” que caracteriza a los navegantes, ello nos puede llevar a establecer que en el portal se debe buscar, ante todo, velocidad. Especialmente en las operaciones más cotidianas, que serán las de consulta de preguntas y respuestas.

En el caso de que esto no fuese así y el usuario del portal tuviese que esperar demasiado la descarga de una página, esto seguramente propiciaría que se cansasen y dejaran de visitar la web y, lo que es peor, no volviesen a entrar en el portal por considerarlo lento y poco práctico.

En lo que respecta a la parte de administración podemos concluir afirmando que no requiere de tanta eficiencia temporal puesto que son operaciones que se no se efectuarán con tanta asiduidad como las comentadas anteriormente.

No sería correcto terminar este apartado sin comentar que la eficiencia temporal de una aplicación web no sólo depende de la eficiencia del código en sí o de la cantidad

de información que se deposite en las páginas, sino también de factores ajenos a estos como, principalmente, la carga del servidor (la cantidad de peticiones que esté atendiendo la máquina donde se aloja la aplicación), la carga de la plataforma de Facebook que tenga en ese momento (peticiones de login y acceso a datos) la carga del servidor que proporciona la conexión a Internet al usuario, el tipo de conexión que utilice), y otros factores en menor medida.

3.2.4 Seguridad y LOPD

El hecho de trabajar en aplicaciones para Internet supone un riesgo extra y obliga a tener más cuidado aún en el tema de la seguridad, ya que pueden existir personas que quieran corromper los contenidos (información) del portal con fines varios.

Se ha de tener en cuenta, por tanto, los siguientes conceptos:

- ✓ **La integridad:** Los datos enviados por un emisor deben de llegar al receptor sin cambios.
- ✓ **Autenticación:** Comprobar que quien intenta acceder a un apartado privado puede hacerlo. Por ejemplo en el caso de la administración: no puede acceder cualquier persona, si no podría modificar datos del portal y provocar un caos. Sólo deben acceder los administradores que deben conocer tanto el nombre de usuario como la contraseña que los identifica de forma única.
- ✓ **Privacidad:** Nadie puede tener acceso a información privada por tanto se debe garantizar que aún en caso de alguien capturase información, esta no pudiese ser inteligible, y por tanto útil.

En nuestro caso, la única información privada que manejaremos es en la parte del login en el que el usuario introducirá su usuario y contraseña para entrar en el portal. La parte de identificación, la trata la plataforma de Facebook, ya que al logarse, la web redirige la conexión al servidor de Facebook y es en ese momento donde el usuario facilita sus datos privados. La conexión se realiza íntegramente en el servidor de Facebook por lo que nosotros no tenemos en ningún momento conocimiento de los datos privados del usuario y son ellos los encargados de proteger esa información.

4. Especificación

La especificación de un sistema software nos permite básicamente, formalizar los *requerimientos funcionales y no funcionales* definidos en la anterior etapa de *Análisis de requerimientos*. Los requerimientos funcionales describen todas las entradas y salidas y las relaciones entre ellas (de datos y de procesos) y los no funcionales definen las cualidades generales que debe tener el sistema al realizar su función. En otras palabras, podríamos decir que consiste en la formalización de la descripción del comportamiento que ha de tener la aplicación desde el punto de vista del usuario o del entorno. En el caso de este proyecto las fuentes para determinar estos requerimientos han sido principalmente el cliente, que tenía muy claro desde el principio cuáles eran los factores más importantes a tener en cuenta, y la plataforma a desarrollar.

La notación que se ha empleado tanto en esta etapa (la de especificación) como en la posterior (diseño) es UML (Unified Modeling Language), lenguaje que nos permite desarrollar con una metodología orientada a objetos. La orientación a objetos es una forma de desarrollo de sistemas software que aporta una serie de ventajas y características que son muy beneficiosas a nivel de eficiencia en algunos casos. La característica principal, es que toda buena especificación debe poseer la **no-redundancia** (no repetición) entre el modelo conceptual, los diagramas de secuencia y los contratos de las operaciones, ya que la redundancia hace que la especificación sea difícilmente modificable, puesto que si varía un aspecto de los repetidos innecesariamente se deberá modificar en cada uno de los modelos que lo utilizan o referencian.

4.1 Modelo Conceptual

El modelo conceptual nos permite representar gráficamente los conceptos (objetos) significativos en el dominio del problema. En él está representada toda la información que tendrá el sistema software.

- **Objetos:** Entidades que existen en el mundo real. Tienen identidad y son distinguibles entre ellos. Como ejemplo podríamos tomar un usuario en concreto de la aplicación.

- **Clase de objetos:** Define un conjunto de objetos con
 - Las mismas propiedades
 - Comportamiento común
 - Idéntica relación con los otros objetos
 - Semántica comúnPor ejemplo, la clase usuarios sería el conjunto de todos los usuarios y se representa por una caja (ver diagrama más abajo) donde además del nombre de la clase aparecen todos sus atributos.

- **Atributo:** Es una propiedad compartida por los objetos de una clase. Por ejemplo, en el caso de la clase usuario: id, id_facebook, nombre y puntos. Todos los usuarios están definidos por el conjunto de estas propiedades llamadas atributos.

- **Asociación:** Define la manera de relacionar los objetos. Es la representación de relaciones entre dos o más objetos. En el diagrama son las líneas que unen unas clases con otras.

Restricciones de integridad textuales (Restricciones escritas): Las restricciones de integridad textuales son aquellas restricciones que se deben imponer en el sistema y que por las características del UML no pueden ser expresadas en el diagrama del modelo conceptual de forma gráfica, pero aún así se deben de tener en cuenta y deben reflejarse. (Véase un ejemplo en la figura 6)

Las restricciones de integridad se encuentran en la página siguiente, debajo del Modelo conceptual del sistema global.

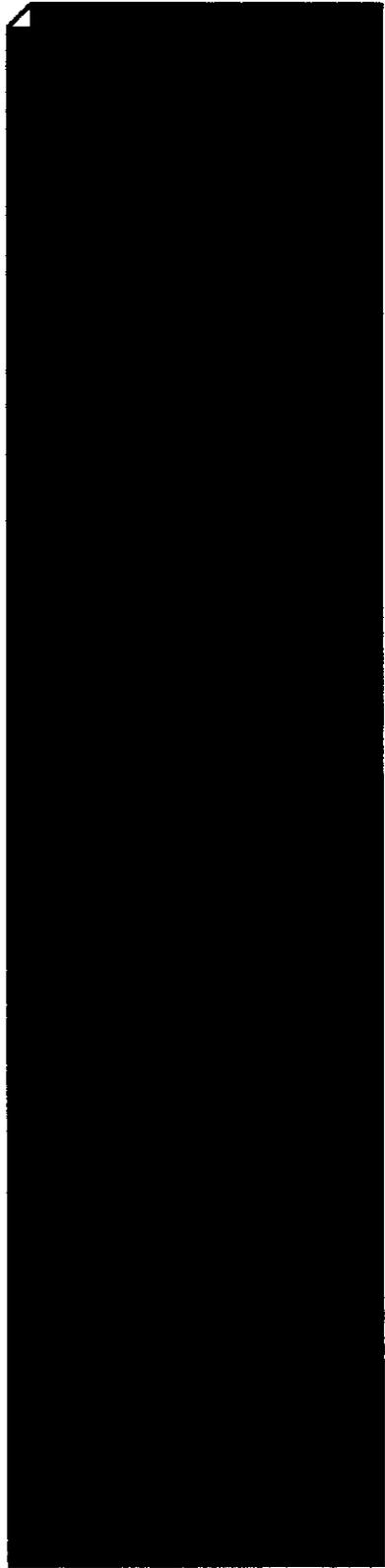
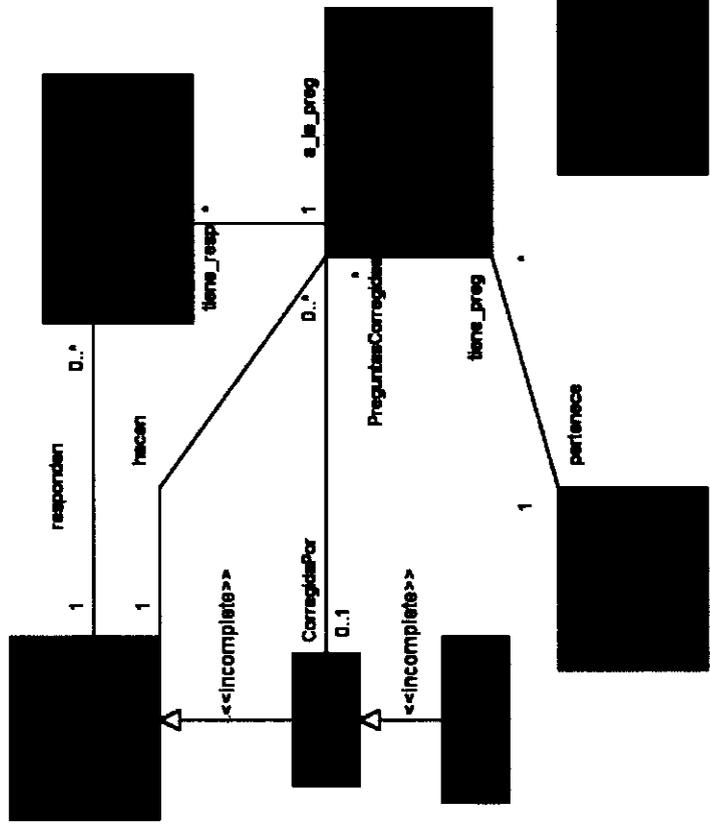


Fig. 6: Modelo conceptual del sistema global

4.2 Especificación de los Casos de Uso

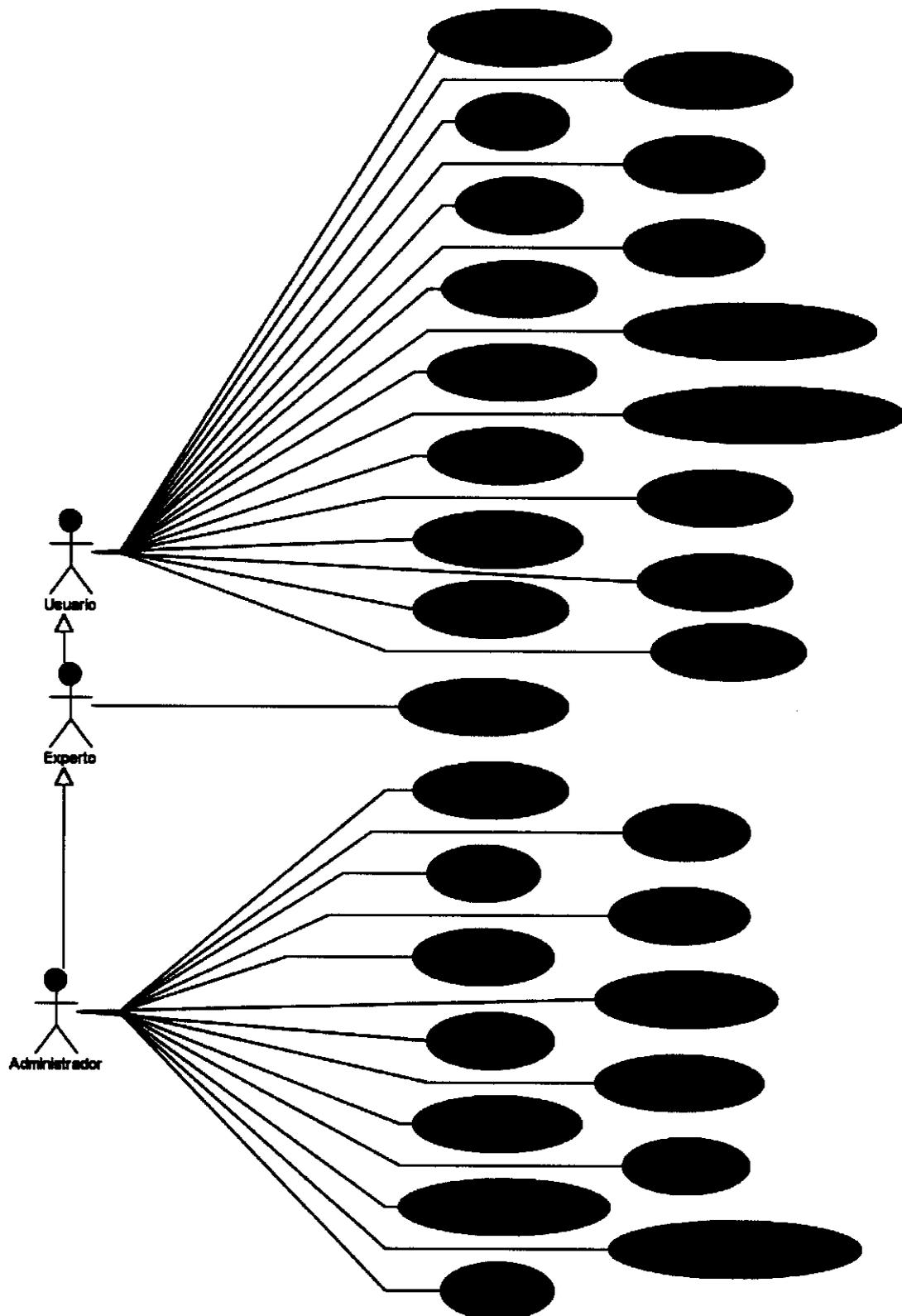


Fig. 7: Diagrama de casos de uso Usuario, Experto, Administrador

- 1. Caso de uso:** Consultar Datos Usuario
Actores: Usuario (iniciador)
Objetivo: Obtiene información del usuario indicado.
Resumen: Esta función, tiene por objetivo obtener la información del sistema del usuario pasado por parámetro. Retornara el nombre de usuario, los puntos que tiene, el tipo de usuario que es y el identificador en Facebook.
Tipo: Primario y esencial.
- 2. Caso de uso:** Consultar Datos Tema
Actores: Usuario (iniciador)
Objetivo: Obtiene información del tema indicado.
Resumen: Esta función, tiene por objetivo obtener la información del sistema del tema pasado por parámetro. Retornara el nombre del tema, el identificador del tema padre y un booleano indicando si está oculto o no.
Tipo: Primario y esencial.
- 3. Caso de uso:** Iniciar Sesión
Actores: Usuario (iniciador)
Objetivo: Inicia sesión en el sistema.
Resumen: Esta función, tiene por objetivo buscar en la base de datos si el usuario se ha logado alguna vez. En el caso en el que no sea cierto, registrara su nombre y el id que tiene asignado en Facebook. En el caso en el que si este registrado pero su nombre no coincida, actualizara el nombre. También se encargara de limpiar los votos del sistema que tengan una antigüedad superior a un día desde la fecha actual del sistema.
Tipo: Primario y esencial.
- 4. Caso de uso:** Buscar Pregunta
Actores: Usuario (iniciador)
Objetivo: Buscar preguntas que contengan texto indicado por el usuario.
Resumen: Esta función, tiene por objetivo buscar en la base de datos las preguntas que contengan en su enunciado exactamente el texto introducido por el usuario (pero no tiene porque ser el texto entero del enunciado, el pasado por el usuario). Una vez localizadas, las mostrara por pantalla ordenadas de manera cronológica, de la más reciente a la más anterior.
Tipo: Primario y esencial.

5. **Caso de uso:** Crear Pregunta
Actores: Usuario (iniciador)
Objetivo: Introducir una pregunta en el sistema
Resumen: Con esta funcionalidad, se permitirá dar de alta nuevas preguntas en el sistema. El usuario introducirá el texto de la pregunta y seleccionara la temática a la que corresponde y esta función la dará de alta en el sistema para que otros usuarios (o el mismo) puedan consultarla y ofrecer posibles soluciones.
Tipo: Primario y esencial.
6. **Caso de uso:** Modificar Pregunta
Actores: Usuario (iniciador)
Objetivos: Modificar una pregunta hecha por un usuario.
Resumen: Con esta funcionalidad, se permitirá modificar una pregunta introducida por el usuario. El usuario introducirá el texto nuevo y seleccionara la temática a la que pertenece (si no quiere cambiar la pregunta de temática seleccionara la que ya tenía) y esta función guardara los cambios en el sistema.
Nota: solo se podrán modificar las preguntas hechas por el mismo usuario que inicia la funcionalidad (o si es un usuario de tipo administrador), y además, solo se dispondrá de 15 minutos para poder realizar estas modificaciones desde el momento en que se dio de alta por primera vez la pregunta en el sistema. (Esta limitación de tiempo no la tendrá un usuario de tipo administrador)
Tipo: Primario y esencial.
7. **Caso de uso:** Consultar Preguntas
Actores: Usuario (iniciador)
Objetivos: Listar las preguntas de un tema en concreto.
Resumen: Esta función permite consultar las preguntas que hay relacionadas con una temática en concreto. El usuario seleccionara la temática que quiere consultar y el sistema mostrara un listado de preguntas relacionadas con esa temática ordenadas cronológicamente empezando por la más reciente (de la fecha en la que se dio de alta en el sistema) y terminando por la más antigua. También mostrara el numero de respuestas que hay asociadas a esta pregunta.
Tipo: Primario y esencial

8. Caso de uso: Consultar las 10 Últimas Preguntas

Actores: Usuario (iniciador)

Objetivos: Listar las 10 últimas preguntas introducidas en el sistema.

Resumen: Esta función permite consultar las 10 últimas preguntas que se introdujeron en el sistema. El sistema mostrara un listado de las 10 últimas preguntas ordenadas cronológicamente empezando por la más reciente (de la fecha en la que se dio de alta en el sistema) y terminando por la más antigua, que tengan estado visible. También mostrara el numero de respuestas que hay asociadas a esta pregunta.

Tipo: Primario y esencial

9. Caso de uso: Responder a Pregunta

Actores: Usuario (iniciador)

Objetivos: Responder a una pregunta concreta.

Resumen: Con esta funcionalidad, un usuario podrá responder a una pregunta que haya realizado otro usuario (también puede responder a sus mismas preguntas). Para ello tendrá que seleccionar la pregunta concreta que quiere responder y a continuación, introducir el texto de la respuesta. El sistema publicara la respuesta y la asociara a la pregunta seleccionada.

Tipo: Primario y esencial.

10. Caso de uso: Modificar Respuesta

Actores: Usuario (iniciador)

Objetivos: Modificar una respuesta hecha por un usuario.

Resumen: Con esta funcionalidad, se permitirá modificar una respuesta introducida por el usuario. El usuario introducirá los cambios a realizar y esta función los modificara en el sistema.

Nota: solo se podrán modificar las respuestas hechas por el mismo usuario que inicia la funcionalidad (o si es un usuario de tipo administrador), y además, solo se dispondrá de 15 minutos para poder realizar estas modificaciones desde el momento en que se dio de alta por primera vez la pregunta en el sistema. (Esta limitación de tiempo no la tendrá un usuario de tipo administrador)

Tipo: Primario y esencial.

11. **Caso de uso:** Consultar Respuestas de una Pregunta

Actores: Usuario (iniciador)

Objetivos: Listar las respuestas que tiene una pregunta en concreto

Resumen: Esta funcionalidad permite consultar las respuestas que tiene asociada una pregunta. Cuando el usuario seleccione una pregunta, el sistema mostrara por pantalla el texto de la pregunta, el nombre del usuario de Facebook que la realizo y la fecha en la que se dio de alta en el sistema. A continuación, mostrara todas las respuestas asociadas a esta pregunta. En concreto, para cada respuesta, mostrara el texto de la respuesta, el nombre de usuario de Facebook que la realizo y la fecha en la que la contesto. Las respuestas saldrán ordenadas cronológicamente de la más actual a la más antigua por fecha de inserción del sistema. Además si algún experto ha seleccionado alguna de las respuestas como correcta, esta saldrá en primer lugar resaltada. En concreto mostrara además de lo mencionado anteriormente en las otras respuestas, el nombre de usuario de Facebook que dio por válida esa respuesta.

Tipo: Primario y esencial.

12. **Caso de uso:** Consultar Pregunta

Actores: Usuario (iniciador)

Objetivo: Obtiene información de la pregunta indicada.

Resumen: Esta función, tiene por objetivo obtener la información del sistema de la pregunta pasada por parámetro. Retornara el texto de la pregunta, la fecha en la que se introdujo en el sistema, el identificador del usuario que crea la pregunta, (si tiene respuesta correcta) el identificador de la respuesta correcta y el identificador del usuario que certifico esa respuestas como válida, un booleano que indica si la pregunta esta visible o oculta y un identificador del tema al que pertenece la pregunta.

Tipo: Primario y esencial.

13. **Caso de uso:** Consultar Respuesta

Actores: Usuario (iniciador)

Objetivo: Obtiene información de la respuesta indicada.

Resumen: Esta función, tiene por objetivo obtener la información del sistema de la respuesta pasada por parámetro. Retornara el texto de la respuesta, la fecha en la que se introdujo en el sistema, el identificador del usuario que crea la respuesta y un booleano que indica si la respuesta esta visible o oculta.

Tipo: Primario y esencial.

14. **Caso de uso:** Menú Temas
- Actores:** Usuario (iniciador)
- Objetivos:** Listar los temas que están registrados en el sistema.
- Resumen:** Esta funcionalidad permite listar todos los temas que hay registrados en el sistema. El orden en el que lo mostrara será primero el tema padre y a continuación todos sus subtemas. En concreto retornara para cada tema/subtema, su identificador, el nombre del tema, el identificador del tema padre, y el estado en el que se encuentra.
- Tipo:** Primario y esencial.
-
15. **Caso de uso:** Dar Puntos a Usuario
- Actores:** Usuario (iniciador)
- Objetivos:** Añadir puntos al usuario indicado
- Resumen:** Esta funcionalidad permite dar puntuación a un usuario. El usuario seleccionara el usuario al que quiere dar puntuación y seleccionara en un nivel el número de puntos que quiere darle (mínimo 1; máximo 5). El sistema añadirá la puntuación al usuario.
- Nota: Un usuario no podrá darse puntuación a sí mismo. Además, un usuario en concreto solo podrá dar puntuación a otro en concreto una vez al día para una pregunta o respuesta en concreto. (Para realizar esta labor, se guardara en el momento de dar puntuación, la IP del ordenador que está realizando esta gestión, el identificador de usuario al que se ha votado y el identificador de la pregunta o respuesta que se ha votado. Entonces, esa IP solo podrá dar puntos a ese usuario una vez al día, para esa pregunta/respuesta)*
- Tipo:** Primario y esencial.
-
16. **Caso de uso:** Listar Temas Padre
- Actores:** Usuario (iniciador)
- Objetivos:** Listar los temas que tiene el tema actual como padre.
- Resumen:** Esta funcionalidad permite listar los temas que tiene el tema actual como padre. Sirve para mostrar un menú de navegación del estilo General-->Deportes--->Baloncesto. La función listara los temas desde el actual hasta el primer tema que no tiene padre en orden ascendente. (En el ejemplo anterior (Baloncesto)->(Deportes)->(General))
- Tipo:** Primario y esencial.

17. **Caso de uso:** Seleccionar Respuesta Correcta
Actores: Experto (iniciador)
Objetivos: Dar una respuesta concreta de una pregunta concreta como valida
Resumen: Esta funcionalidad permitirá seleccionar una respuesta como valida de todas las que hay asociadas a una pregunta en concreto. El usuario experto tendrá que seleccionar una respuesta de la lista de respuestas que hay asociadas a una pregunta en concreto e indicar que esa respuesta es válida. El sistema guardara los cambios para que quede reflejado en el listado de respuestas.
Tipo: Primario y esencial.
18. **Caso de uso:** Crear Tema/Subtema
Actores: Administrador (iniciador)
Objetivos: Dar de alta una temática.
Resumen: Esta funcionalidad tiene por objetivo dar de alta una temática en el sistema para que los usuarios puedan asociar preguntas a dicha temática. El administrador introducirá el nombre de la temática en el sistema y, en caso de querer crear una sub temática, seleccionara la temática padre, y el sistema la dará de alta.
Nota: El sistema no tendrá en cuenta si ya existe alguna temática que tenga el mismo significado que la temática creada por la complejidad que podría tener la operación (por ejemplo, en la temática Cine, podría darse el caso de existir dos sub temáticas "de terror" y "de miedo" que realmente es la misma). Queda a cargo del administrador el cerciorarse de que no existe otra temática o sub temática con el mismo significado de temática.
Tipo: Primario y esencial.
19. **Caso de uso:** Modificar Tema/Subtema
Actores: Administrador (iniciador)
Objetivos: Modificar el texto de una temática o de una sub temática
Resumen: Con esta funcionalidad, se permitirá modificar el texto que hay introducido de una temática o una sub temática en concreto y se podar mover la temática a una sub temática o viceversa. El Administrador seleccionara la temática o la sub temática de la lista de temáticas que desea modificar e introducirá los cambios (el texto que quiere cambiar y, en el caso de querer modificar la temática para que sea una sub temática de otro tema, seleccionara la nueva temática padre para este tema. En el caso de que quiera que esta temática sea una temática sin padre, no seleccionara ninguna temática de la lista). El sistema guardara los datos.
Nota: No se podrá hacer que una temática padre pase a ser sub temática de una de sus sub temáticas. El sistema controlara este error y lo mostrara por pantalla en el caso de que se produzca. Hay que tener en cuenta que lo que se está realizando es modificar el Tema, por lo que

todas las preguntas que estaban asociadas a la antigua temática, se asociaran a esta nueva temática. Al igual que en las operaciones anteriores, el sistema no tendrá en cuenta si ya existe alguna temática con ese nombre por la complejidad que podría tener la operación. Queda a cargo del administrador el cerciorarse de que no existe otra temática o sub temática con el mismo significado de la temática.

Tipo: Primario y esencial.

20. **Caso de uso:** Ocultar Tema

Actores: Administrador (iniciador)

Objetivos: Ocultar una temática y todas sus sub temáticas. También oculta todas las preguntas y respuestas asociadas a las temáticas mencionadas.

Resumen: Con esta funcionalidad, se permite que una temática no sea listada en la aplicación y los usuarios no puedan acceder a sus preguntas. El administrador seleccionara la temática de la lista e indicara que quiere que se oculte. El sistema guardara los datos en el sistema para que no se liste esta temática.

Tipo: Primario y esencial.

21. **Caso de uso:** Des Ocultar Tema

Actores: Administrador (iniciador)

Objetivos: Cambia el estado a visible de una temática y de la temática padre asociada a esta (recursivamente). También cambia el estado a visible de todas las preguntas y respuestas asociadas a las temáticas mencionadas.

Resumen: Con esta funcionalidad, se permite que una temática sea listada en la aplicación y los usuarios puedan acceder a sus preguntas. El administrador seleccionara la temática de la lista e indicara que quiere que sea visible. El sistema guardara los datos en el sistema para que se liste esta temática.

Tipo: Primario y esencial.

22. **Caso de uso:** Ocultar Pregunta

Actores: Administrador (iniciador)

Objetivos: Ocultar una pregunta y todas sus respuestas para que no se listen en la lista de preguntas, ni en el buscador.

Resumen: Con esta funcionalidad, se permite que una pregunta no sea listada en la aplicación y los usuarios no puedan acceder a ella ni a sus respuestas. El administrador seleccionara la pregunta de la lista e indicara que quiere que se oculte. El sistema guardara los datos en el sistema para que no se liste la pregunta ni sus respuestas.

Tipo: Primario y esencial.

- 23. Caso de uso:** Des Ocultar Pregunta
Actores: Administrador (iniciador)
Objetivos: Cambia a visible, el estado de una pregunta y de todas sus respuestas para que se liste en la lista de preguntas.
Resumen: Con esta funcionalidad, se permite que una pregunta sea listada en la aplicación y los usuarios puedan acceder a ella y a sus respuestas. El administrador seleccionara la pregunta de la lista e indicara que quiere que sea visible. El sistema guardara los datos en el sistema para que se liste la pregunta.
Tipo: Primario y esencial.
- 24. Caso de uso:** Ocultar Respuesta
Actores: Administrador (iniciador)
Objetivos: Ocultar una respuesta para que no se liste en la lista de respuestas.
Resumen: Con esta funcionalidad, se permite que una respuesta no sea listada en la aplicación y los usuarios no puedan acceder a ella. El administrador seleccionara la respuesta de la lista e indicara que quiere que se oculte. El sistema guardara los datos en el sistema para que no se liste esa respuesta.
Tipo: Primario y esencial.
- 25. Caso de uso:** Des Ocultar Respuesta
Actores: Administrador (iniciador)
Objetivos: Cambia el estado a visible de una respuesta para que se liste en la lista de respuestas.
Resumen: Con esta funcionalidad, se permite que una respuesta sea listada en la aplicación y los usuarios puedan acceder a ella. El administrador seleccionara la respuesta de la lista e indicara que quiere que sea visible. El sistema guardara los datos en el sistema para que se liste esa respuesta.
Tipo: Primario y esencial.
- 26. Caso de uso:** Modificar puntos a usuario
Actores: Administrador (iniciador)
Objetivos: Aumenta o disminuye los puntos a un usuario en concreto
Resumen: Si en algún momento se considera que hay que añadir o retirar puntos a un usuario en concreto, esta funcionalidad permitirá hacerlo. El Administrador seleccionara a un usuario de la lista de usuarios e indicara cuantos puntos quiere que se le incremente/retire. El sistema guardara los datos.
Nota: Un usuario no podrá tener puntos negativos por lo que solo se podrán retirar como máximo el número de puntos que tiene actualmente el usuario. Si se pone una cantidad mayor, el sistema dejara a 0 el contador del usuario seleccionado.
Tipo: Primario y esencial.

- 27. Caso de uso:** Listar Usuarios
Actores: Administrador (iniciador)
Objetivos: Lista todos los usuarios registrados en el sistema.
Resumen: Esta funcionalidad permite listar todos los usuarios registrados en el sistema. Exactamente, listara para cada usuario, el tipo de usuario que es, su nombre registrado en Facebook, y los puntos que tiene actualmente.
Tipo: Primario y esencial.
- 28. Caso de uso:** Cambiar Privilegios A Usuario
Actores: Administrador (iniciador)
Objetivos: Permite cambiar el tipo de usuario
Resumen: Esta funcionalidad permite cambiar el tipo de usuario. El Administrador seleccionara el usuario en la lista de usuarios e indicara que tipo de usuario desea otorgarle. El sistema guardará los datos.
Nota: Un usuario no cambiarse los privilegios de el mismo.
Tipo: Primario y esencial.
- 29. Caso de uso:** Deseleccionar Respuesta Correcta
Actores: Administrador (iniciador)
Objetivos: Deselecciona la respuesta concreta de una pregunta concreta.
Resumen: Esta funcionalidad permitirá deseleccionar una respuesta como valida de todas las que hay asociadas a una pregunta en concreta. El usuario Administrador tendrá que seleccionar la respuesta válida actualmente e indicar que esa respuesta no es válida. El sistema guardara los cambios para que quede reflejado en el listado de respuestas.
Tipo: Primario y esencial.
- 30. Caso de uso:** Listar Temas
Actores: Administrador (iniciador)
Objetivos: Listar los temas que están registrados en el sistema.
Resumen: Esta funcionalidad permite listar todos los temas que hay registrados en el sistema. El orden en el que lo mostrará será primero el tema padre y a continuación todos sus subtemas. En concreto retornara para cada tema/subtema, su identificador, el nombre del tema, el identificador del tema padre, el estado en el que se encuentra y el número de preguntas que contiene el tema.
Tipo: Primario y esencial.

4.3 Modelo UML

Después de evaluar los diferentes tipos de modelos que existen para realizar un diseño óptimo de una aplicación en UML, optamos por el Modelo Vista Controlador (MVC) que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El Modelo controlador vista se usa frecuentemente en aplicaciones web, donde la capa de la vista es la página y el código que provee de datos dinámicos a la página. La capa del modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y la capa del controlador es el responsable de recibir los eventos de entrada desde la vista.

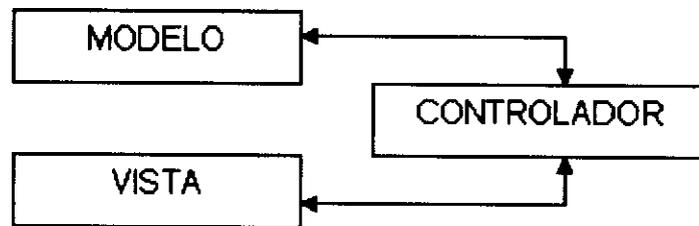


Fig. 8: Un diagrama sencillo que muestra la relación entre el modelo, la vista y el controlador.

A continuación se detallan las capas de los 3 apartados, con las correspondientes operaciones para cada controlador.

4.3.1 Capa del modelo dominio y datos

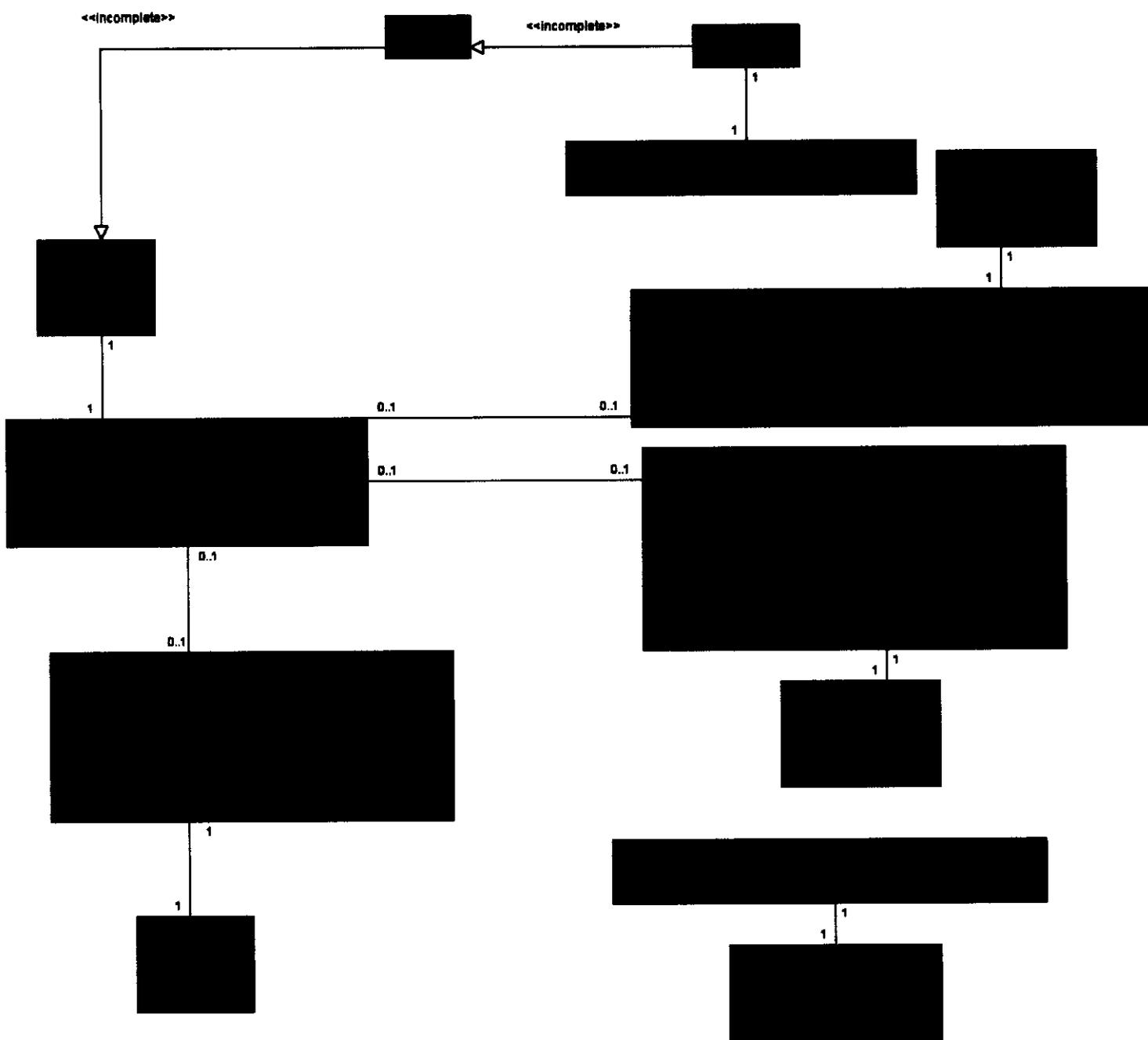


Fig. 9: Modelo conceptual de la capa del dominio y datos

NOTA: A este diagrama se le han de sumar las relaciones del modelo conceptual del sistema global, que se han ocultado para facilitar la comprensión de la capa.

4.3.2 Capa del modelo vista

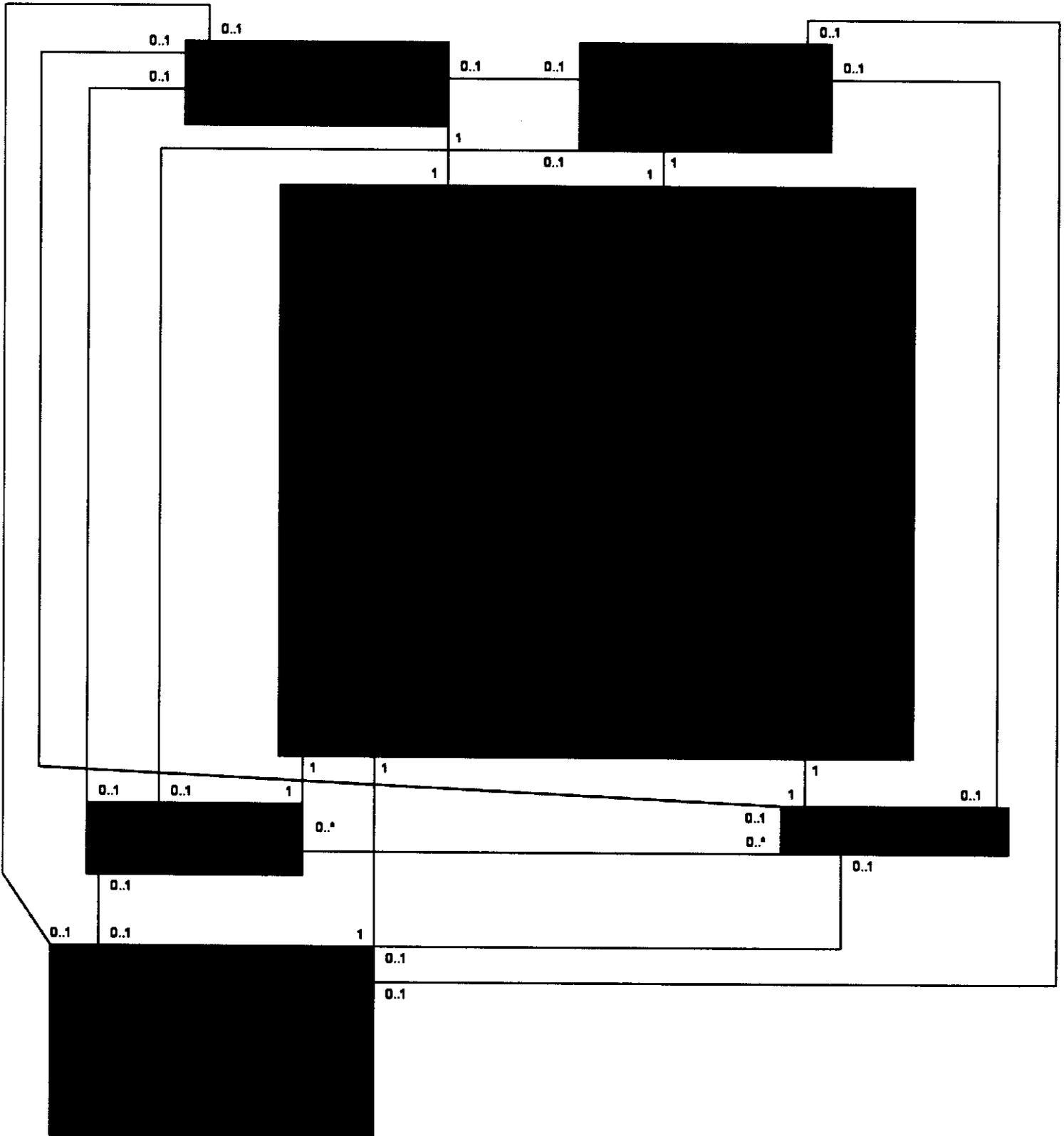


Fig. 10: Modelo conceptual de la capa de Vista

4.3.3 Capa del Controlador

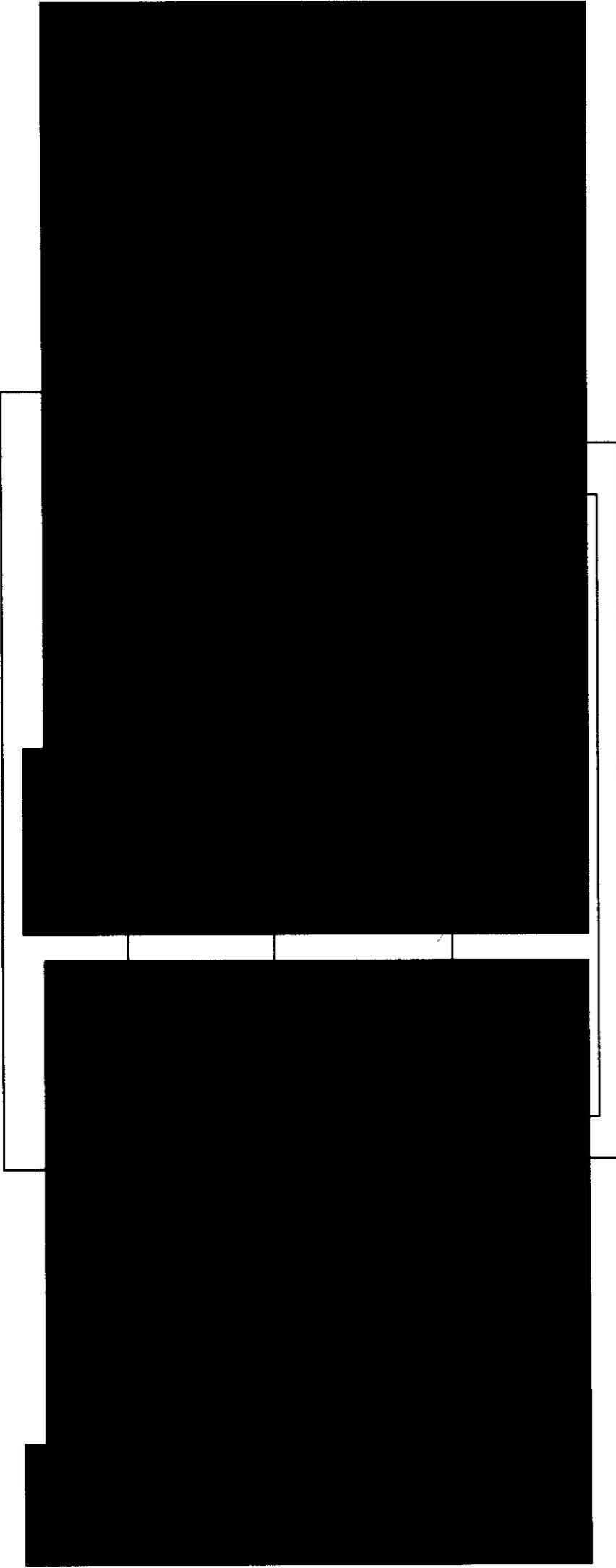


Fig. 11: Modelo conceptual de la capa del Controlador

4.4 Contratos de las operaciones

En la capa del modelo mostrada en la página 50 hemos visto que algunas cajas, contenían algo semejante a esto:



Fig. 12: Ejemplo de una operación del sistema.

Se trata de una **operación del sistema**. Es una operación interna que se ejecuta como respuesta a la comunicación del evento externo producido o generado por un actor, que en nuestro caso puede ser un usuario normal, experto o administrador.

En este apartado para cada una de estas operaciones se va a **describir la especificación concreta de su comportamiento (cuáles son los cambios de estado de la base de la información) y su efecto en el sistema (cuáles son las salidas que el sistema proporciona cuando se invoca la operación)**.

Definiremos todo esto mediante una descripción declarativa, llamada **contrato**, basada en los siguientes conceptos que explicamos previamente a continuación:

- **Nombre:** Define el nombre y los argumentos de la operación a la que se refiere el contrato.
- **Responsabilidades:** Descripción informal (con lenguaje natural) del propósito de la operación.
- **Caso de uso:** Muestra el caso de uso al que pertenece la operación.
- **Excepciones:** Descripción de la reacción del sistema a situaciones excepcionales.
- **Precondiciones:** Hace referencia a las condiciones que se deben de cumplir en el sistema antes de la ejecución de la operación para que no se produzcan errores.
- **Post-condiciones:** Describe los cambios de estado producidos por la ejecución de la operación.
- **Salida:** Descripción de la salida que proporciona la operación.

A continuación se detallan los contratos de todas las operaciones de la capa del modelo del sistema.

Operaciones de ControladorUsuarios

- 1. Nombre:** ConsultarTipo(id) : Int

Responsabilidades: Devolver un entero indicando que tipo de usuario que es usuario con id pasado

Caso de uso: Todos los casos de uso que son llamados por el usuario, Administrador y Experto.

Excepciones:

 - Si el id de usuario pasado no existe, esta función retornara un valor inferior a 0.

Precondiciones: -

Post-condiciones: -

Salida:

 - Retornara un 1 si es un usuario normal, 2 si es un usuario Experto y 3 Si es un usuario Administrador.

- 2. Nombre:** LogadoAlgunaVez(id_facebook, nombre) : Int

Responsabilidades: Comprobar si el usuario esta registrado en la base de datos del sistema. Si no es así, lo registrara. En el caso de que el usuario estuviese ya registrado pero no coincidiese el nombre, lo actualizaría.

Caso de uso: Iniciar Sesión.

Excepciones: -

Precondiciones: -

Post-condiciones:

 - Si no existe, se crea un nuevo registro en la tabla usuarios con los siguientes datos:
 - Id** = Un identificador nuevo no registrado en el sistema.
 - Id_facebook** = El identificador de Facebook pasado.
 - Nombre** = El nombre pasado.
 - Puntos** = 0.
 - Tipo** = 1.
 - Si existe y el nombre de usuario es diferente al pasado;
 - Nombre** = El nombre pasado.

Salida:

 - Retornara el identificador del usuario pasado. Si hay algún problema durante la operación, retornara 0.

3. **Nombre:** ListarUsuarios(order, inicio, limit) : Array [][]
- Responsabilidades:** Devolver un Array con todos los usuarios que se han registrado en el sistema, en el orden indicado. Las variables *inicio* y *limit* se usan para delimitar los resultados.
- Caso de uso:** Listar usuarios.
- Excepciones:** -
- Precondiciones:** -
- Post-condiciones:** -
- Salida:**
- Si order contiene el 1, el array(lista) se ordenara por id_facebook, si contiene el 2 la lista se ordenara por puntuación, si contiene el 3 la lista se ordenara por nombre.
 - Este será el contenido del Array para cada fila:
 - Columna [0]** = El identificador del usuario en el sistema.
 - Columna [1]** = El identificador del usuario en Facebook.
 - Columna [2]** = Los puntos del usuario.
 - Columna [3]** = El tipo de usuario.
4. **Nombre:** DarPuntos(id, puntos) : bool
- Responsabilidades:** Añade la cantidad de puntos indicada al usuario con el identificador indicado.
- Caso de uso:** Dar puntos a usuario
- Excepciones:** -
- Precondiciones:**
- La variable id ha de contener un identificador de un usuario registrado en el sistema.
 - La variable puntos ha de contener un valor entero.
- Post-condiciones:**
- Si puntos es mayor que 0, se sumara la cantidad de puntos pasada al usuario con el identificador pasado.
 - Si puntos es menor o igual que 0, no se realizara ninguna modificación en el sistema y retornara falso.
- Salida:**
- Retorna cierto si se ha realizado la operación con éxito.

5. **Nombre:** ConsultarDatosUsuario(id) : Array []
Responsabilidades: Consulta los datos del usuario con el identificador pasado.
Caso de uso: Consultar Datos Usuario.
Excepciones: -
Precondiciones: -
Post-condiciones: -
Salida:
- Retorna una lista con los siguientes valores:
 - Columna [0]** = Nombre del usuario del id pasado por parámetro.
 - Columna [1]** = Puntos del usuario pasado por parámetro.
 - Columna [2]** = Tipo de usuario.
 - Columna [3]** = El identificador que tiene el usuario en Facebook.
- Si el usuario no está registrado en el sistema, la columna 0 no tendrá valor, la columna 1 tendrá un valor menor que 0, la columna 2 tendrá el valor de 0 y la columna 3 tendrá un valor menor que 0.
6. **Nombre:** NuevoUsuario(id_facebook, nombre) : Int
Responsabilidades: Registra un nuevo usuario en el sistema.
Caso de uso: Iniciar Sesión.
Excepciones: -
Precondiciones:
- La variable id_facebook ha de contener un identificador de usuario que no esté registrado en el sistema.
- Post-condiciones:**
- Se crea un nuevo registro en la tabla usuarios con los siguientes datos:
 - Id** = Un identificador nuevo no registrado en el sistema.
 - Id_facebook** = El identificador de Facebook pasado.
 - Nombre** = El nombre pasado.
 - Puntos** = 0.
 - Tipo** = 1.
- Salida:**
- Retorna el id del usuario nuevo si la operación se ha realizado satisfactoriamente. Si no, retornara 0

Operaciones de ControladorAdministrador

7. **Nombre:** CambiarTipoUsuario(id, tipo, id_administrador) : bool
- Responsabilidades:** Asigna el tipo de usuario pasado al usuario con el id pasado. Esta función solo se podrá realizar si el id_Administrador pasado pertenece al tipo administrador.
- Caso de uso:** Cambiar Privilegios A Usuario.
- Excepciones:** -
- Precondiciones:**
- La variable id ha de contener un identificador de un usuario registrado en el sistema.
 - La variable id_administrador ha de contener un identificador de un usuario registrado en el sistema.
 - La variable tipo ha de ser un valor entero comprendido entre el 1 y el 3
- Post-condiciones:**
- Se modifica el registro con el usuario con id = al id pasado los siguientes campos:
Tipo = El tipo pasado.
- Salida:**
- Retorna cierto si la operación se ha realizado satisfactoriamente.

Operaciones de ControladorTemas

8. **Nombre:** ConsultarDatosTema(id_Tema) : Array [][]
Responsabilidades: Retorna una lista con los datos del tema con el identificador pasado.
Caso de uso: Consultar Datos Tema.
Excepciones: -
Precondiciones:
- La variable id_Tema ha de contener un identificador de un usuario registrado en el sistema.
Post-condiciones: -
Salida:
- Genera un array(lista) con el siguiente contenido para cada fila:
Columna [0] = El nombre del tema.
Columna [1] = El identificador del tema padre.
Columna [2] = El estado del tema.
9. **Nombre:** ListarTemas(id_usuario,numPreg) : Array [][]
Responsabilidades: Retorna una lista con los temas registrados en el sistema, en el siguiente orden: primero el tema y seguidamente, todos sus subtemas. Atención, esta función solo retornara los Temas que tengan el estado = 0 a no ser, que el usuario id_usuario pasado sea un usuario de tipo administrador. En tal caso, no tendrá en cuenta el estado a la hora de retornar los resultados. También tendrá en cuenta el valor de la variable numPreg. Si es cierto retornara el número de preguntas que tiene asociado cada tema. Si es falso, este dato lo obviara.
Caso de uso: Listar Temas, Menú Temas.
Excepciones: -
Precondiciones:
- La variable id_usuario ha de contener un identificador de un usuario registrado en el sistema.
Post-condiciones: -
Salida:
- Genera un array(lista) con el siguiente contenido para cada fila:
Columna [0] = El identificador del tema.
Columna [1] = El nombre del tema.
Columna [2] = El identificador del tema.
Columna [3] = El estado del tema.
Columna [4] = El número de preguntas que contiene el tema, si numPreg tiene el valor a cierto, sino, este campo estará vacío.
Columna [5] = Una lista con los subtemas del tema actual. La lista tendrá la misma estructura que la comentada aquí.

- 10. Nombre:** ModificarTema(id, texto,id_padre,id_usuario) : Int
- Responsabilidades:** Modifica el texto del tema con el identificador pasado. Y lo reasigna al id_padre pasado. Esta función solo se podrá realizar si el id_usuario pasado pertenece al tipo administrador.
- Caso de uso:** Modificar Tema/Subtema.
- Excepciones:**
- Retornara -1 si el id_padre pasado, pertenece a alguno de los subtemas del tema a modificar.
- Precondiciones:**
- La variable id ha de contener un identificador de un tema registrado en el sistema.
 - La variable id_padre ha de contener un identificador de un tema registrado en el sistema.
 - La variable id_usuario ha de contener un identificador de un usuario registrado en el sistema.
- Post-condiciones:**
- Se modifica registro con el id = id pasado en la tabla Tema con los siguientes datos:

Nombre =	El texto pasado.
Id_tema_padre =	El identificador del tema padre pasado.
- Salida:**
- Retorna 1 si la operación se ha realizado satisfactoriamente.
-
- 11. Nombre:** Padre_es_un_hijo(id_tema,id_padre) : Int
- Responsabilidades:** Comprueba si el tema con id_padre pertenece a alguno de los subtemas del tema id_tema. Esta función es utilizada por Modificar Tema para comprobar que no se está intentando asignar un sub tema del id_tema, como padre de este.
- Caso de uso:** Modificar Tema/Subtema.
- Excepciones:**
-
- Precondiciones:**
- La variable id ha de contener un identificador de un tema registrado en el sistema.
 - La variable id_padre ha de contener un identificador de un tema registrado en el sistema.
 - La variable id_usuario ha de contener un identificador de un usuario registrado en el sistema.
- Post-condiciones:**
-
- Salida:**
- Retornara -1 si el id_padre pasado, pertenece a alguno de los subtemas del tema id_tema.

- 12. Nombre:** OcultarTema(id, id_usuario) : bool
- Responsabilidades:** Oculta un tema, todos sus subtemas y todas las preguntas y respuestas asociados a estos temas, para que estos no sean listados por la aplicación. Esta función solo se podrá realizar si el id_usuario pasado pertenece al tipo administrador.
- Caso de uso:** Ocultar Tema.
- Excepciones:** -
- Precondiciones:**
- La variable id ha de contener un identificador de tema registrado en el sistema.
 - La variable Id_usuario ha de contener un identificador de usuario registrado en el sistema.
- Post-condiciones:**
- Si id_usuario es un usuario de tipo Administrador, se modifica registro con el id = id pasado en la tabla Tema con los siguientes datos:
 - estado = 1.**
 - También se modifica el registro anteriormente citado con el mismo valor, en todos los subtemas del tema con el identificador pasado, y en todas las preguntas y respuestas asociadas a tema y su subtemas
- Salida:**
- Retorna cierto si la operación se ha realizado satisfactoriamente.

- 13. Nombre:** DesOcultarTema(id, id_usuario) : bool
- Responsabilidades:** Modifica el valor a visible de un tema y de su tema padre (recursivamente). Tambien modifica el valor a visible en todas las preguntas y respuestas asociados a estos temas, para que estos puedan sean listados por la aplicación. Esta función solo se podrá realizar si el id_usuario pasado pertenece al tipo administrador.
- Caso de uso:** Des Ocultar Tema.
- Excepciones:** -
- Precondiciones:**
- La variable id ha de contener un identificador de tema registrado en el sistema.
 - La variable Id_usuario ha de contener un identificador de usuario registrado en el sistema.
- Post-condiciones:**
- Si id_usuario es un usuario de tipo Administrador, se modifica registro con el id = id pasado en la tabla Tema con los siguientes datos:
estado = 0.
 - También se modifica el registro anteriormente citado con el mismo valor, en todos los subtemas del tema con el identificador pasado, y en todas las preguntas y respuestas asociadas a tema y su subtemas
- Salida:**
- Retorna cierto si la operación se ha realizado satisfactoriamente.

14. **Nombre:** CrearTema(texto, id_padre, id_usuario) : Int
- Responsabilidades:** Crea un tema con el texto pasado y lo asigna al tema padre con el id pasado. Si el id padre es inferior a 1 no se asigna a ningún padre. Esta función solo se podrá realizar si el id_usuario pasado pertenece al tipo administrador.
- Caso de uso:** Crear Tema/Subtema.
- Excepciones:** -
- Precondiciones:**
- La variable id_padre ha de contener un identificador de un tema registrado en el sistema.
 - La variable id_usuario ha de contener un identificador de un usuario registrado en el sistema.
- Post-condiciones:**
- Se crea un registro con el id = (ultimo id disponible) en la tabla Tema con los siguientes datos:
 - Nombre =** El texto pasado.
 - Estado =** 0.
 - Id_padre =** Si (id_padre pasado) < 1, pondrá un 0, sino, pondrá el id_padre pasado.
- Salida:**
- Retorna el id del tema creado. Retornara 0 si la operación no se ha realizado satisfactoriamente.
15. **Nombre:** ListarTemasPadre(id_tema) : Array [][]
- Responsabilidades:** Retorna una lista con los temas que tiene asignado como padre el tema actual. Los retornara en orden ascendente desde el tema actual hasta el tema padre que no tenga ningún padre asignado. En concreto retornara el id del tema y su nombre.
- Caso de uso:** Listar Temas Padre.
- Excepciones:** -
- Precondiciones:**
- La variable id_tema ha de contener un identificador de un tema registrado en el sistema.
- Post-condiciones:** -
- Salida:**
- Genera un array (lista) con el siguiente contenido para cada fila:
 - Columna [0]** = El id del tema.
 - Columna [1]** = El nombre del tema.

Operaciones de ControladorPregunta

- 16. Nombre:** BuscarPregunta(texto, inicio, limit, id_usuario) : Array [][]
- Responsabilidades:** Retorna una lista de preguntas que contengan el texto indicado ordenadas cronológicamente de la mas actual a la más antigua. Las variables *inicio* y *limit* se usan para delimitar los resultados. En concreto retornara para cada fila, el identificador de la pregunta, el texto, el número de respuestas que tiene la pregunta, el estado de la pregunta, el identificador de la persona que realizó la pregunta, la fecha en la que la pregunta se introdujo en el sistema y el identificador del tema al que pertenece. Atención, esta función solo retornara las preguntas que tengan el estado = 0 a no ser, que el usuario id_usuario pasado sea un usuario de tipo administrador. En tal caso, no tendrá en cuenta el estado a la hora de retornar los resultados.
- Caso de uso:** Buscar Pregunta.
- Excepciones:** -
- Precondiciones:** -
- Post-condiciones:** -
- Salida:**
- Se retornara un Array (lista) con el siguiente contenido para cada fila:
 - Columna [0]** = "p" + id de la pregunta.
 - Columna [1]** = El Texto de la pregunta.
 - Columna [2]** = El Número de respuestas que tiene la pregunta.
 - Columna [3]** = El estado de la pregunta.
 - Columna [4]** = El identificador de la persona que ha realizado la pregunta.
 - Columna [5]** = La fecha en la que la pregunta se introdujo en el sistema
 - Columna [6]** = El identificador del tema al que pertenece la pregunta.

17. Nombre: ListarUltimas10Preguntas(id_usuario, inicio, limit) : Array [][]

Responsabilidades: Retorna una lista con las 10 últimas preguntas introducidas en el sistema, que tengan el estado = 0, a no ser que el identificador de usuario pasado sea un usuario de tipo administrador, en cuyo caso no se tendrá en cuenta esta restricción. Las preguntas se retornaran ordenadas cronológicamente de la más actual a la más antigua. En el caso de que haya menos de 10 preguntas se listaran solo las encontradas que cumplan los requisitos indicados anteriormente. Las variables *inicio* y *limit* se usan para delimitar los resultados. En concreto retornara para cada fila, el identificador de la pregunta, el texto, el número de respuestas que tiene la pregunta, el estado de la pregunta, el identificador de la persona que realizó la pregunta, la fecha en la que la pregunta se introdujo en el sistema y el identificador del tema al que pertenece.

Caso de uso: Consultar las 10 Últimas Preguntas.

Excepciones: -

Precondiciones: -

Post-condiciones: -

Salida:

- Se retornara un Array (lista) con el siguiente contenido para cada fila:

- Columna [0]** = "p" + id de la pregunta.
- Columna [1]** = El Texto de la pregunta.
- Columna [2]** = El Número de respuestas que tiene la pregunta.
- Columna [3]** = El estado de la pregunta.
- Columna [4]** = El identificador de la persona que ha realizado la pregunta.
- Columna [5]** = La fecha en la que la pregunta se introdujo en el sistema
- Columna [6]** = El identificador del tema al que pertenece la pregunta.

- 18. Nombre:** ListarPreguntas(id_tema, inicio, limit, id_usuario) : Array [][]
- Responsabilidades:** Retorna una lista de preguntas que pertenezcan al tema con el id indicado ordenadas cronológicamente de la mas actual a la más antigua. Las variables *inicio* y *limit* se usan para delimitar los resultados. En concreto retornara para cada fila, el identificador de la pregunta, el texto, el número de respuestas que tiene la pregunta, el estado de la pregunta, el identificador de la persona que realizó la pregunta, la fecha en la que la pregunta se introdujo en el sistema y el identificador del tema al que pertenece. Atención, esta función solo retornara las preguntas que tengan el estado = 0 a no ser, que el usuario id_usuario pasado sea un usuario de tipo administrador. En tal caso, no tendrá en cuenta el estado a la hora de retornar los resultados.
- Caso de uso:** Consultar Preguntas.
- Excepciones:** -
- Precondiciones:**
- La variable id_tema ha de contener un identificador de un tema registrado en el sistema.
 - La variable id_usuario ha de contener un identificador de un usuario registrado en el sistema.
- Post-condiciones:** -
- Salida:**
- Se retornara un Array (lista) con el siguiente contenido para cada fila:
 - Columna [0]** = "p" + id de la pregunta.
 - Columna [1]** = El Texto de la pregunta.
 - Columna [2]** = El Número de respuestas que tiene la pregunta.
 - Columna [3]** = El estado de la pregunta.
 - Columna [4]** = El identificador de la persona que ha realizado la pregunta.
 - Columna [5]** = La fecha en la que la pregunta se introdujo en el sistema
 - Columna [6]** = El identificador del tema al que pertenece la pregunta.

19. **Nombre:** NumeroPreguntas(id_tema) : Int
Responsabilidades: Retorna el número de preguntas que están relacionadas con el id seleccionado.
Caso de uso: Listar Temas.
Excepciones: -
Precondiciones: - La variable id_tema ha de contener un identificador de un tema registrado en el sistema.
Post-condiciones: -
Salida: - Se retornara un entero que corresponde al número de preguntas que tiene el tema con el id pasado.
20. **Nombre:** ConsultarPregunta(id) : Array []
Responsabilidades: Retorna el contenido de la pregunta pasada por parámetro. En concreto retornara el texto completo de la pregunta, la fecha en la que se introdujo en el sistema, el id de Facebook del usuario que planteo la pregunta, el id de la respuesta correcta, el id del usuario de Facebook que corrigió la respuesta y el estado de la pregunta.
Caso de uso: Consultar Respuestas de una pregunta, Consultar Pregunta
Excepciones: -
Precondiciones: - La variable id ha de contener un identificador de una pregunta registrada en el sistema.
Post-condiciones: -
Salida: - Se retornara un Array (lista) con el siguiente contenido:
Columna [0] = El Texto de la pregunta.
Columna [1] = La fecha en la que se introdujo la pregunta en el sistema.
Columna [2] = El identificador de Facebook de la persona que ha realizado la pregunta.
Columna [3] = "r" + Id de la respuesta correcta. Si no tiene, esta posición estará vacía.
Columna [4] = El id del usuario de Facebook que corrigió la respuesta. Si no tiene, esta posición estará vacía.
Columna [5] = El estado de la pregunta.
Columna [6] = El identificador del tema al que pertenece la pregunta

- 21. Nombre:** SelecRespuestaCorrecta(id_preg, id_resp, id_experto) : bool
- Responsabilidades:** Si el id_experto corresponde a un identificador de un usuario de tipo experto, marca la respuesta id_resp como respuesta válida para la pregunta id_preg. También registra en la pregunta que el usuario id_experto ha sido quien la ha dado por válida.
- Caso de uso:** Seleccionar Respuesta Correcta
- Excepciones:** -
- Precondiciones:**
- La variable id_preg ha de contener un identificador de una pregunta registrada en el sistema.
 - La variable id_resp ha de contener un identificador de una Respuesta registrada en el sistema.
 - La variable id_experto ha de contener un identificador de un usuario registrado en el sistema.
- Post-condiciones:**
- Si id_usuario es de tipo experto, se modifica el atributo de la clase "Pregunta" con id = id_preg, los siguientes valores
Id_respuesta_correcta = id_resp
 - Se crea la relación "Corregida Por" entre "Experto" con id = id_experto y "Pregunta" con id = "id_preg" (Si existía la relación "Corregida Por" en la pregunta con id= "id_preg" se eliminara antes de crear la nueva).
- Salida:**
- Retornara cierto si se ha podido realizar la operación con éxito.

- 22. Nombre:** ModificarPregunta(id, Texto, id_tema, id_Usuario) : bool
- Responsabilidades:** Modifica el texto de una Pregunta y la asigna al tema con id pasado, siempre que el id_usuario Facebook pasado sea el del usuario que ha introducido la pregunta en el sistema y no hayan pasado más de 15 minutos desde que se inserto la pregunta en el sistema, o que el id de usuario pasado corresponda a un usuario de tipo administrador.
- Caso de uso:** Modificar Pregunta
- Excepciones:** -
- Precondiciones:**
- La variable id ha de contener un identificador de una Pregunta registrada en el sistema.
 - La variable id_Usuario ha de contener un identificador de un usuario registrado en el sistema.
 - La variable id_tema ha de contener un identificador de un tema registrado en el sistema.
- Post-condiciones:**
- Se modifica el atributo de la clase "Pregunta" con id = id pasado, los siguientes valores
pregunta = El texto pasado.
 - Además se elimina la relación "pertenece" que había entre la pregunta y el tema y se crea una nueva relación "pertenece" entre la pregunta con id = id pasado y el tema con id = id_tema pasado.
- Salida:**
- Retornara cierto si se ha podido realizar la operación con éxito.
- 23. Nombre:** ModificarEstadoPregunta(id, id_usuario, estado) : bool
- Responsabilidades:** Modifica el estado de la Pregunta con el id pasado con el estado pasado. Esta función solo se podrá realizar si el id_usuario pasado pertenece al tipo administrador.
- Caso de uso:** Ocultar Pregunta, Des Ocultar Pregunta
- Excepciones:** -
- Precondiciones:**
- La variable id ha de contener un identificador de una pregunta registrado en el sistema.
 - La variable id_Usuario ha de contener un id de un usuario registrado en el sistema.
- Post-condiciones:**
- Si id_usuario es un usuario de tipo Administrador, se modifica registro con el id = id pasado en la tabla pregunta con los siguientes datos:
estado = El estado pasado.
- Salida:**
- Retornara cierto si se ha podido realizar la operación con éxito.

- 24. Nombre:** Modificar_Estado_Preguntas(id_tema, id_usuario, estado) : bool
- Responsabilidades:** Modifica el estado de todas las Preguntas del tema con el id_tema pasado con el estado pasado. Esta función solo se podrá realizar si el id_usuario pasado pertenece al tipo administrador.
- Caso de uso:** Ocultar Tema, Des Ocultar Tema
- Excepciones:** -
- Precondiciones:**
- La variable id_tema ha de contener un identificador de un tema registrado en el sistema.
 - La variable id_usuario ha de contener un id de un usuario registrado en el sistema.
- Post-condiciones:**
- Si id_usuario es un usuario de tipo Administrador, se modifica registro con el id = id pasado en la tabla pregunta con los siguientes datos:
estado = El estado pasado.
- Salida:**
- Retornara cierto si se ha podido realizar la operación con éxito.
- 25. Nombre:** QuitarRespuestaCorrecta(id_pregunta, id_usuario) : bool
- Responsabilidades:** Deselecciona la respuesta que tenga la pregunta asignada como válida para la pregunta id_pregunta. También elimina la relación "corregida por". Esta función solo se podrá realizar si el id_usuario pasado pertenece al tipo administrador.
- Caso de uso:** Deseleccionar Respuesta Correcta
- Excepciones:** -
- Precondiciones:**
- La variable id_pregunta ha de contener un identificador de una pregunta registrada en el sistema.
 - La variable id_usuario ha de contener un identificador de un usuario de tipo administrador registrado en el sistema.
- Post-condiciones:**
- Si id_usuario es de tipo administrador. Se elimina el valor del atributo "id_respuesta_correcta" de la clase "Pregunta" con id = id_preg.
 - También se elimina la relación "CorregidaPor" entre el usuario experto y la pregunta con id = id_preg.
- Salida:**
- Retornara cierto si se ha podido realizar la operación con éxito.

- 26. Nombre:** AñadirPregunta(id_usuario, Texto, data, id_Tematica) : Int
- Responsabilidades:** Añade una pregunta con el texto pasado y la fecha pasada. Además se crea la relación entre la pregunta introducida y el usuario que la creo, y la relación entre la pregunta introducida y la temática.
- Caso de uso:** Crear Pregunta
- Excepciones:** -
- Precondiciones:**
- La variable id_usuario ha de contener un identificador de un usuario de tipo administrador registrado en el sistema.
 - La variable id_Tematica ha de contener un identificador de un tema registrado en el sistema.
- Post-condiciones:**
- Se registra en el sistema una pregunta con los siguientes valores:
 - Id** = El Ultimo id introducido en la tabla Preguntas + 1. Si no hay ninguna pregunta introducida se pondrá el valor 1.
 - Pregunta** = El texto pasado por el parámetro Texto.
 - data** = La fecha pasada por el parámetro data.
 - estado** = 0.
 - También se creara la relación "hacen" entre el usuario con id = id_usuario y la pregunta registrada.
 - También se creara la relación "pertenece" entre el tema con id = id_Tematica y la pregunta registrada.
- Salida:**
- Retornara el identificador que ha tenido la pregunta creada. Si no se ha podido realizar la operación con éxito, retornara un 0.

Operaciones de ControladorRespuesta

- 27. Nombre:** AñadirRespuesta(id_Pregunta, Texto, id_usuario, data) : Int
- Responsabilidades:** Añade una respuesta con el texto pasado y la fecha pasada, y la asocia con la pregunta con identificador = id_pregunta Además se crea la relación entre la respuesta introducida y el usuario que la creo.
- Caso de uso:** Responder a Pregunta
- Excepciones:** -
- Precondiciones:**
- La variable id_usuario ha de contener un identificador de un usuario de tipo administrador registrado en el sistema.
 - La variable id_Pregunta ha de contener un identificador de una pregunta registrada en el sistema.
- Post-condiciones:**
- Se registra en el sistema una respuesta con los siguientes valores:
 - Id** = El ultimo id introducido en la tabla respuesta + 1. Si no hay ninguna pregunta introducida se pondrá el valor 1.
 - respuesta** = El texto pasadoTexto.
 - data** = La fecha pasada por la variable data.
 - estado** = 0.
 - Se creara una relación entre la respuesta creada y la pregunta con id_Pregunta.
 - También se creara la relación "responden" entre el usuario con id = id_usuario y la respuesta registrada.
- Salida:**
- Retornara el identificador que ha tenido la respuesta creada. Si no se ha podido realizar la operación con éxito, retornara un 0.

- 28. Nombre:** ListarRespuestas(id_Pregunta, inicio, limit, id_usuario) : Array [][]
- Responsabilidades:** Retorna una lista de respuestas que pertenezcan a la pregunta con el id indicado ordenadas cronológicamente de la mas actual a la más antigua. Las variables *inicio* y *limit* se usan para delimitar los resultados. En concreto retornara para cada fila, el id de la respuesta, el texto, el identificador de Facebook del usuario que registro la respuesta, el estado y la fecha en la que se introdujo en el sistema. Atención, esta función solo retornara las respuestas que tengan el estado = 0 a no ser, que el usuario id_usuario pasado sea un usuario de tipo administrador. En tal caso, no tendrá en cuenta el estado a la hora de retornar los resultados.
- Caso de uso:** Consultar Respuestas De Una Pregunta.
- Excepciones:** -
- Precondiciones:**
- La variable id_pregunta ha de contener un identificador de una pregunta registrada en el sistema.
 - La variable id_usuario ha de contener un identificador de un usuario registrado en el sistema.
- Post-condiciones:** -
- Salida:**
- Se retornara un Array (lista) con el siguiente contenido para cada fila:
 - Columna [0]** = "r" + id de la respuesta.
 - Columna [1]** = El texto de la respuesta.
 - Columna [2]** = El identificador de Facebook del usuario que registró la respuesta.
 - Columna [3]** = El estado de la respuesta
 - Columna [4]** = La fecha en la que se introdujo la respuesta en el sistema.

29. **Nombre:** ConsultarRespuesta(id) : Array []
- Responsabilidades:** Retorna el contenido de la respuesta pasada por parámetro. En concreto retornara el texto completo de la respuesta, la fecha en la que se introdujo en el sistema, el id de Facebook del usuario que planteo la respuesta y el Estado de la respuesta.
- Caso de uso:** Consultar Respuestas de una Pregunta
- Excepciones:** -
- Precondiciones:**
- La variable id ha de contener un identificador de una Respuesta registrada en el sistema.
- Post-condiciones:** -
- Salida:**
- Se retornara un Array (lista) con el siguiente contenido:
 - [0]** = Texto de la respuesta
 - [1]** = Fecha en la que se introdujo la respuesta en el sistema.
 - [2]** = Id Facebook del usuario que planteo la pregunta en el sistema.
 - [3]** = Estado de la respuesta.
30. **Nombre:** ModificarRespuesta(id, Texto, id_Usuario) : bool
- Responsabilidades:** Modifica el texto de una respuesta con el id pasado, siempre que el id_usuario pasado sea el del usuario que ha introducido la respuesta en el sistema y no hayan pasado más de 15 minutos desde que se inserto la respuesta en el sistema, o que el id de usuario pasado corresponda a un usuario de tipo administrador.
- Caso de uso:** Modificar Respuesta
- Excepciones:** -
- Precondiciones:**
- La variable id ha de contener un identificador de una Respuesta registrada en el sistema.
 - La variable id_Usuario ha de contener un identificador de un usuario registrado en el sistema.
- Post-condiciones:**
- Se modifica el atributo de la clase "Respuesta" con id = id, los siguientes valores
 - respuesta** = El texto pasado.
- Salida:**
- Retornara cierto si se ha podido realizar la operación con éxito:

- 31. Nombre:** ModificarEstadoRespuesta(id, id_Usuario, estado) : bool
- Responsabilidades:** Se modifica el estado de una Respuesta con el id pasado, actualizando su estado al estado pasado por parámetro. Esta función solo se podrá realizar si el id_usuario pasado pertenece al tipo administrador.
- Caso de uso:** Ocultar Respuesta, Des Ocultar Respuesta
- Excepciones:** -
- Precondiciones:**
- La variable id ha de contener un identificador de una respuesta registrada en el sistema.
 - La variable id_Usuario ha de contener un id de un usuario registrado en el sistema.
- Post-condiciones:**
- Si id_usuario es un usuario de tipo Administrador, se modifica registro con el id = id pasado en la tabla respuesta con los siguientes datos:
estado = 1.
- Salida:**
- Retornara cierto si se ha podido realizar la operación con éxito.
- 32. Nombre:** Modificar_Estado_Respuestas(id_preg, id_Usuario, estado) : bool
- Responsabilidades:** Se modifica el estado de todas las Respuestas que pertenezcan a la pregunta con el id_preg pasado, actualizando su estado al estado pasado por parámetro. Esta función solo se podrá realizar si el id_usuario pasado pertenece al tipo administrador.
- Caso de uso:** Ocultar Pregunta, Des Ocultar Pregunta, Ocultar Tema, Des Ocultar Tema
- Excepciones:** -
- Precondiciones:**
- La variable id_preg ha de contener un identificador de una pregunta registrada en el sistema.
 - La variable id_Usuario ha de contener un id de un usuario registrado en el sistema.
- Post-condiciones:**
- Si id_usuario es un usuario de tipo Administrador, se modifica registro con el id = id pasado en la tabla respuesta con los siguientes datos:
estado = 1.
- Salida:**
- Retornara cierto si se ha podido realizar la operación con éxito.

33. Nombre: NumeroRespuestas(id_Pregunta, id_usuario) : Int

Responsabilidades: Retorna el numero de respuestas que están asociadas a una pregunta y que tienen como valor de estado=1, a no ser que el id_usuario pasado pertenezca a un usuario de tipo administrador, en cuyo caso, no se tendrá en cuenta esa restricción.

Caso de uso: Buscar Preguntas, Consultar las 10 Últimas Preguntas, Consultar Preguntas

Excepciones: -

Precondiciones:

- La variable id_Pregunta ha de contener un identificador de una Pregunta registrada en el sistema.
- La variable id_Usuario ha de contener un id de un usuario registrado en el sistema.

Post-condiciones: -

Salida:

- Retornara el numero de respuestas asociadas a la pregunta con el id pasado.

Operaciones de ControladorVotos

34. **Nombre:** DarPuntos(id, ip, data, id_Pregunta_o_Respuesta,puntos) : bool
- Responsabilidades:** Si puntos es más grande que 0, registra en el sistema que el pc con la ip pasada ha dado un voto al usuario con id pasado por la pregunta con id_Pregunta pasada, siempre y cuando no le haya votado ya antes por esa pregunta.
Si puntos es menor que 0, quitara los puntos indicados al usuario con el id pasado.
- Caso de uso:** Dar Puntos a usuario
- Excepciones:** -
- Precondiciones:**
- La variable id_Pregunta_o_Respuesta ha de contener un identificador de una Pregunta registrada en el sistema.
 - La variable id ha de contener un id de un usuario registrado en el sistema.
- Post-condiciones:**
- Se registra en el sistema un Voto con los siguientes valores:
ip = La ip pasada.
Id_usuario_votado = El id pasado.
Id_pregunta_o_respuesta_votada = El identificador de la Pregunta pasada.
data = La fecha pasada.
- Salida:**
- Retornara cierto si se ha podido realizar la operación con éxito.
35. **Nombre:** LimpiarVotos() : bool
- Responsabilidades:** Elimina los votos que tengan una antigüedad superior a un día respecto a la fecha actual del sistema.
- Caso de uso:** Iniciar Sesión
- Excepciones:** -
- Precondiciones:** -
- Post-condiciones:**
- Se elimina de Votos todos los registros cuya data sea más antigua de un día respecto a la fecha actual del sistema.
- Salida:**
- Retornara cierto si se ha podido realizar la operación con éxito.

5. Diseño

A diferencia de la anterior etapa en la que se hacía referencia al QUÉ, esta sección será donde se determinará el CÓMO desarrollaremos ese QUÉ.

Esta etapa de diseño es dependiente de las tecnologías que se emplearan, es decir, que aquí sí importa qué (lenguajes de programación, sistema gestor de base de datos, etc.) se van a utilizar en la implementación del producto final puesto que estos factores determinan la manera de diseñar en gran medida.

5.1 La arquitectura utilizada en la aplicación

Tal y como vimos en el apartado 4.3 (Modelo UML), la arquitectura utilizada en la aplicación se basará en el patrón arquitectónico en capas. Concretamente se trata de una arquitectura distribuida en 3 capas que se divide en una capa que gestiona los datos y el dominio y otra capa de presentación.

A continuación se van a describir por encima las funciones y objetivos de cada capa de este patrón arquitectónico:

- ❖ **Capa de gestión de datos y dominio:** Estas dos capas se encarga de acceder a la Base de datos y consultar y actualizar los datos de esta. También se encarga de ejecutar las peticiones de la capa más exterior (**la capa de presentación**), que es la que interacciona con el usuario controlando la validez de las entradas que se proporcionan. Para la ejecución de las órdenes de la capa de presentación a veces se requiere de consultas o modificaciones de los datos de la BD del sistema que esta capa lleva a cabo. Los resultados producidos por sus ejecuciones son comunicados a la capa de presentación para que ésta última los muestre al usuario.
- ❖ **Capa de presentación:** Es la capa que interacciona directamente con el usuario final de la aplicación, por tanto se encarga de captar los eventos externos, de ordenar las acciones oportunas (a la capa de **dominio y datos**) y de mostrar los resultados de estas últimas al usuario.

A continuación se va a analizar, mediante una tabla, la relación existente entre las diferentes capas y las tecnologías usadas en cada una de ellas a fin de lograr una visión práctica de la función de cada una:

Capa	Tecnologías
Gestión de datos y dominio	MySQL, PHP
Presentación	HTML, JavaScript, API Facebook

Tabla 10: Herramientas utilizadas para cada capa.

Por último cabría justificar el hecho de haber elegido esta arquitectura, ya que esto no es algo aleatorio:

Para empezar se tendría que aclarar que no hay una arquitectura perfecta que sea de manera absoluta mejor que el resto, sino que todas tienen ventajas e inconvenientes. Para elegir la mejor para un caso concreto se deben analizar una serie de factores referidos a los requerimientos y objetivos perseguidos que, en nuestro caso, están expuesto en los siguientes puntos:

Después de este inciso podemos decir qué nos aporta entonces la elección de esta arquitectura:

1. **Reusabilidad y fácil mantenimiento:** Poder reutilizar el código en futuras ampliaciones. Por ejemplo, en el caso que se quisiese cambiar toda la interfaz gráfica de la aplicación sólo se tendría que modificar la *capa de presentación* y nada más.
2. **Portabilidad:** Nos permite que cambios en la plataforma (hardware, Sistemas Operativos, SGBD...) no supongan cambios de gran envergadura.
3. **Ensayo y prueba:** La parte de ensayo y prueba del sistema es algo importante ya que sirve para detectar todos los posibles errores o carencias antes de la entrega del producto. Esta arquitectura facilita el "el *testeo*" de las aplicaciones desarrolladas puesto que permite hacer pruebas independientes de cada capa por separado. Con esta propiedad se logra un desarrollo mucho más fiable que desemboca en un producto más robusto a posibles fallos.

5.1.1 Normalización

La tecnología orientada a objetos tiene algunas limitaciones que se manifiestan en la etapa de implementación. Estas limitaciones se basan en la existencia de conceptos (usados en el diseño orientado a objetos) que no son permitidos actualmente por ningún lenguaje de programación. Tales son los casos de las asociaciones n-arias con $n > 2$ (asociaciones de más de dos objetos), las clases asociativas, la información derivada y el control de las restricciones de integridad.

El proceso de normalización es el “filtro” donde se modificarán de nuestro modelo conceptual y de contratos de operaciones (**sin cambiar el significado**) todos aquellos componentes que no sean válidos para la etapa de implementación.

En nuestro caso, nuestro modelo conceptual no tiene ningún objeto que se tenga que normalizar por lo que no es necesario realizar ninguna modificación.

5.1.2 Diseño del Modelo de Datos

En este apartado, explicaremos el diseño de la base de datos relacional. Este, se extrae directamente del **modelo conceptual**, o modelo **UML**, del sistema, que ha sido presentado en el apartado 4.1. El resultado del diseño de la Base de datos a partir del modelo conceptual es el llamado **modelo lógico (modelo relacional clásico)**, que consiste en la conversión de las clases y asociaciones del primero en tablas SQL.

SQL es uno de los lenguajes más utilizados en la programación con Bases de Datos, por su fácil comprensión (muy similar al lenguaje natural) y por su gran portabilidad (si se utilizan el lenguaje nativo). Pero también tiene sus desventajas. Una de ellas es que no acepta ciertos objetos del modelo conceptual como pueden ser los objetos heredados y las relaciones binarias (ya que las relaciones n-arias las hemos eliminado en la anterior normalización).

A continuación detallaremos como solucionar estos dos casos:

- **Objetos heredados:** Dado que en nuestro caso, un usuario puede ser de varios tipos, se ha decidido fusionar las herencias “*Experto*” y “*Administrador*” en la clase padre “*Usuarios*” añadiendo un atributo llamado “*tipo*” que será el encargado de discriminar de que tipo será ese usuario (1 significara que es un usuario normal. 2 significará que es un usuario experto y 3 significa que es un usuario administrador).
- **Relaciones binarias:** En nuestro caso, siempre tendremos una relación que es de tipo $*..1$. Esto significará que en cada una de las relaciones que existe entre dos objetos, cada registro del objeto que tenga en su lado de la relación el valor $*$ significara que como máximo, solo podrá tener ese registro una única relación con un registro del otro objeto. Un ejemplo: en el caso de la relación “*Responden*” entre los objetos “*Usuarios*” y “*Respuesta*”, si aplicamos lo

mencionado anteriormente y nos fijamos en el lado del objeto Respuesta, diremos que una respuesta en concreto es respondida por un único usuario en concreto. Otra forma de leerlo sería, si nos fijamos en el objeto usuarios, diremos que un usuario en concreto puede responder a muchas preguntas o ninguna. Con esto pretendemos que una respuesta se relacione con el usuario que la creó para saber quien fue el que publicó esa respuesta.

Para traducir esto al lenguaje SQL lo que hacemos es crear un atributo con el nombre de la relación en el objeto que tiene en su lado de la relación el *. En el caso comentado anteriormente, añadiremos en el objeto "*Respuesta*" el atributo "*Responden*". De este modo, cuando consultemos una respuesta en concreto, sabremos que usuario realizó esa respuesta.

El resultado de aplicar estos cambios es un diagrama normalizado que pretende reflejar todas las restricciones y conceptos de interés del modelo UML para la Base de Datos.

A continuación se encuentra el modelo de datos normalizado.

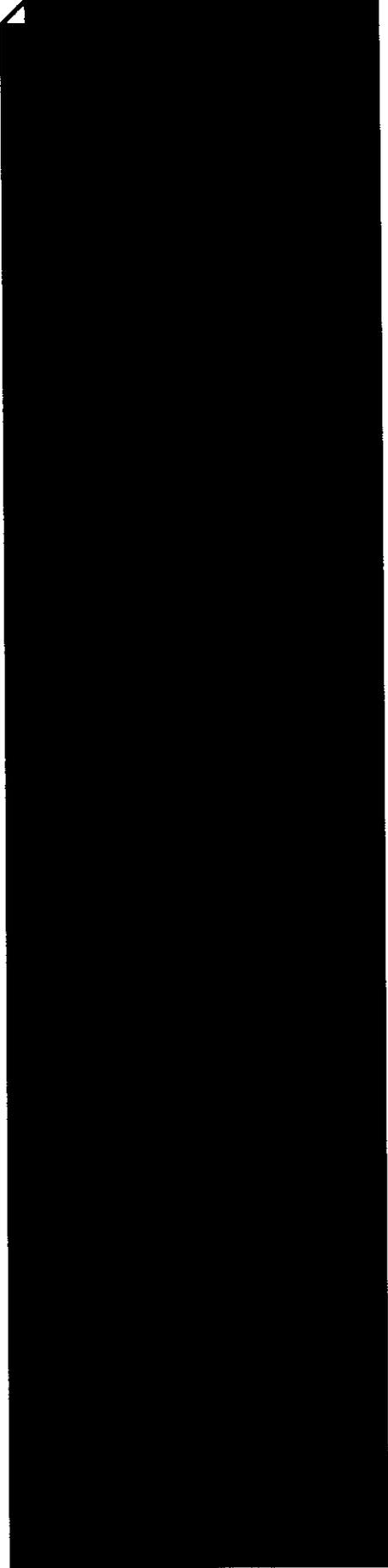
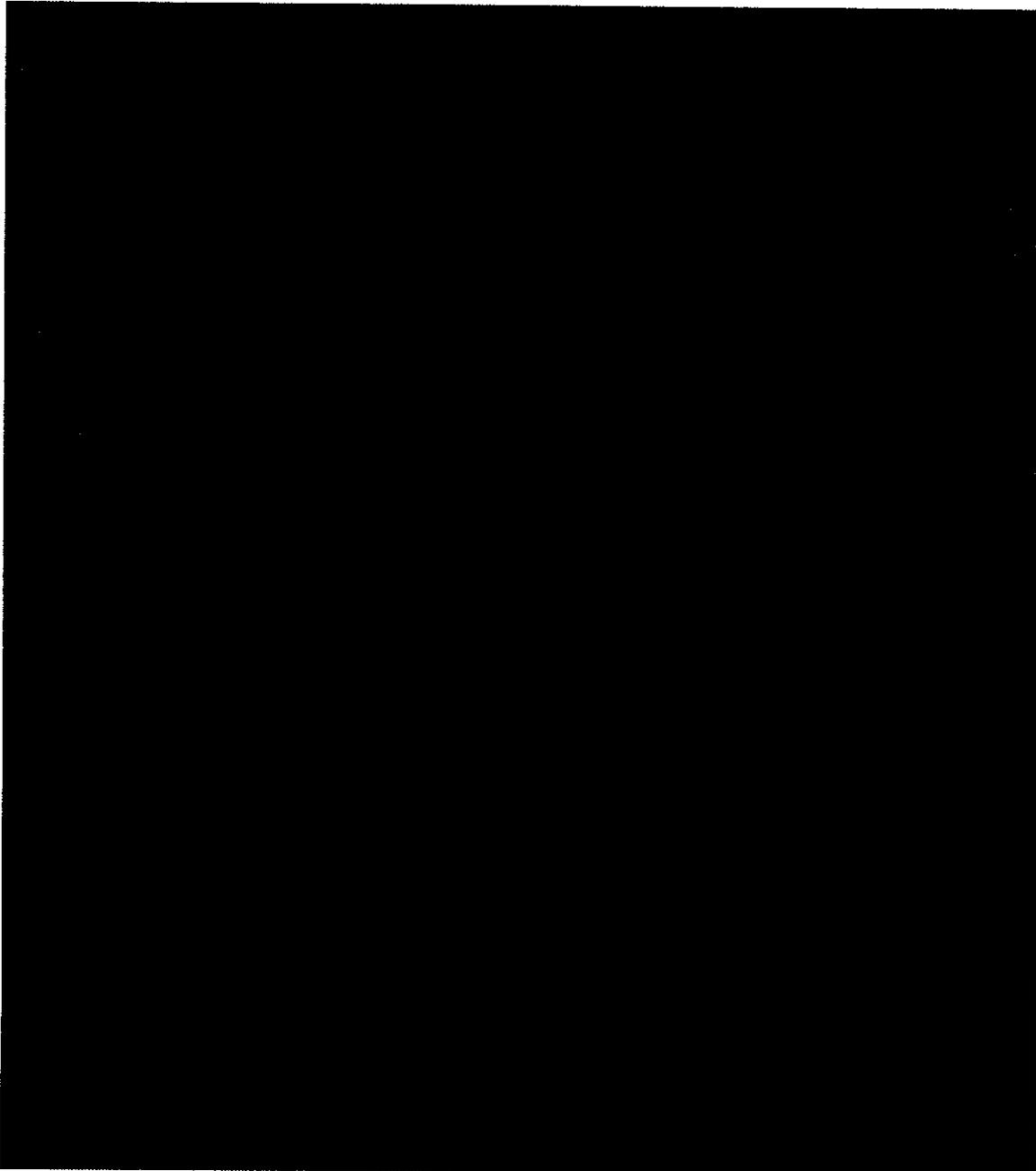


Fig. 13: Modelo de datos normalizado

5.1.3 Modelo Físico

Finalmente, para la creación física de la base de datos, se tiene que traducir el modelo de Datos al SGBD (Sistema gestor de Bases de Datos) del que se dispone. En nuestro caso se ha usado la versión 5.0.84 de MySQL. En los apartados (6.2.5 El SGBD) y (6.3.4 SQL) correspondiente a la implementación y en el apartado (5.1.4 Diagramas de secuencia) hay varios puntos que hacen referencia a las razones de la elección de este SGBD.

A continuación se muestra el fichero de creación de la base de datos para MySQL.



5.1.4 Diagramas de secuencia

Antes de continuar, quería puntualizar una modificación que se han realizado sobre los diagramas de secuencia (respecto a un diagrama de secuencia normal). Para poder facilitar su comprensión, se ha decidido utilizar en los diagramas el lenguaje SQL a la hora de realizar búsquedas en la base de datos. Se ha decidido usar este lenguaje por los motivos expuestos en el apartado 5.1.2 (Diseño del Modelo de Datos), y para facilitar su comprensión. Expongo a continuación un ejemplo detallado para mejor comprensión.

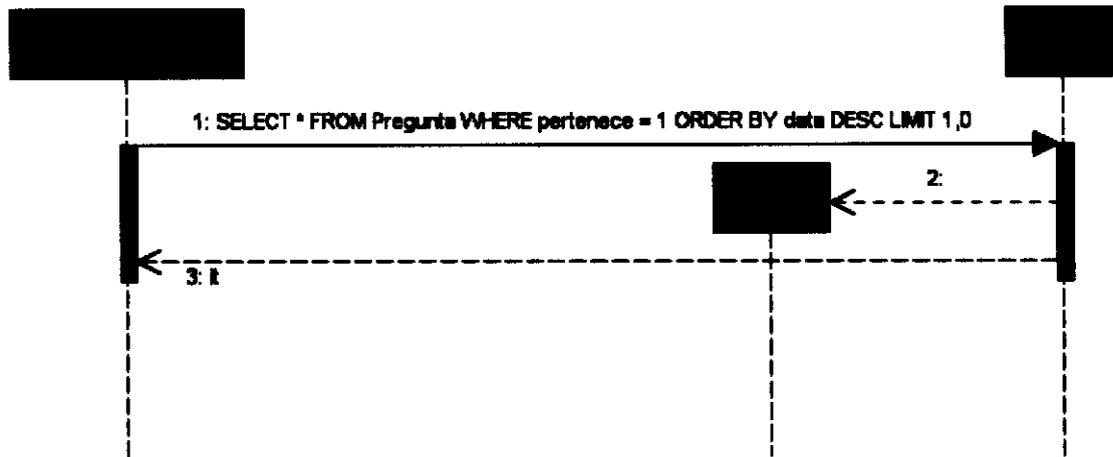


Tabla 11 : ejemplo de diagrama de secuencia UML

En el diagrama de secuencia anterior, el ControladorPregunta está realizando una consulta a la Base de Datos (en concreto a la tabla Pregunta). El SGBD le retornará un iterador para recorrer los resultados.

Si hacemos una traducción al lenguaje natural de un ejemplo de sentencia SQL, obtenemos la siguiente expresión:

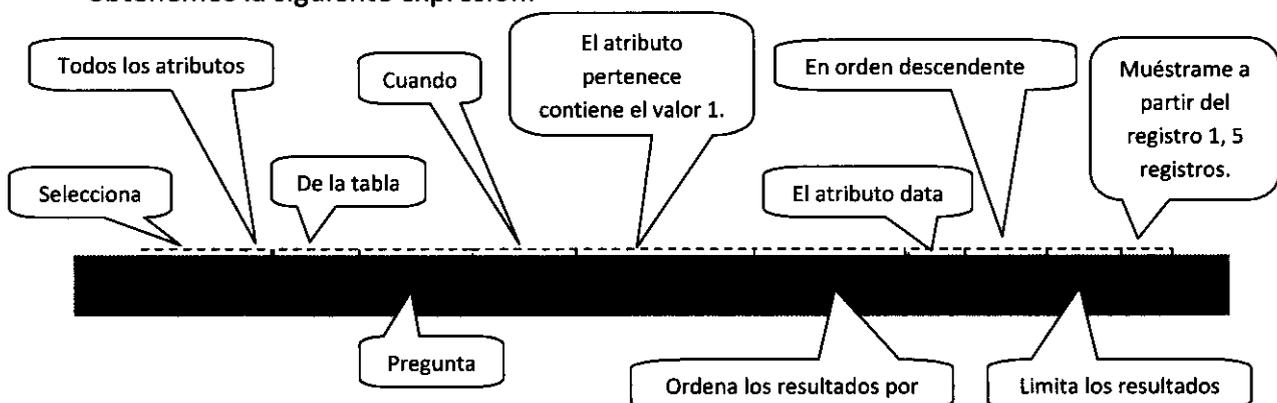


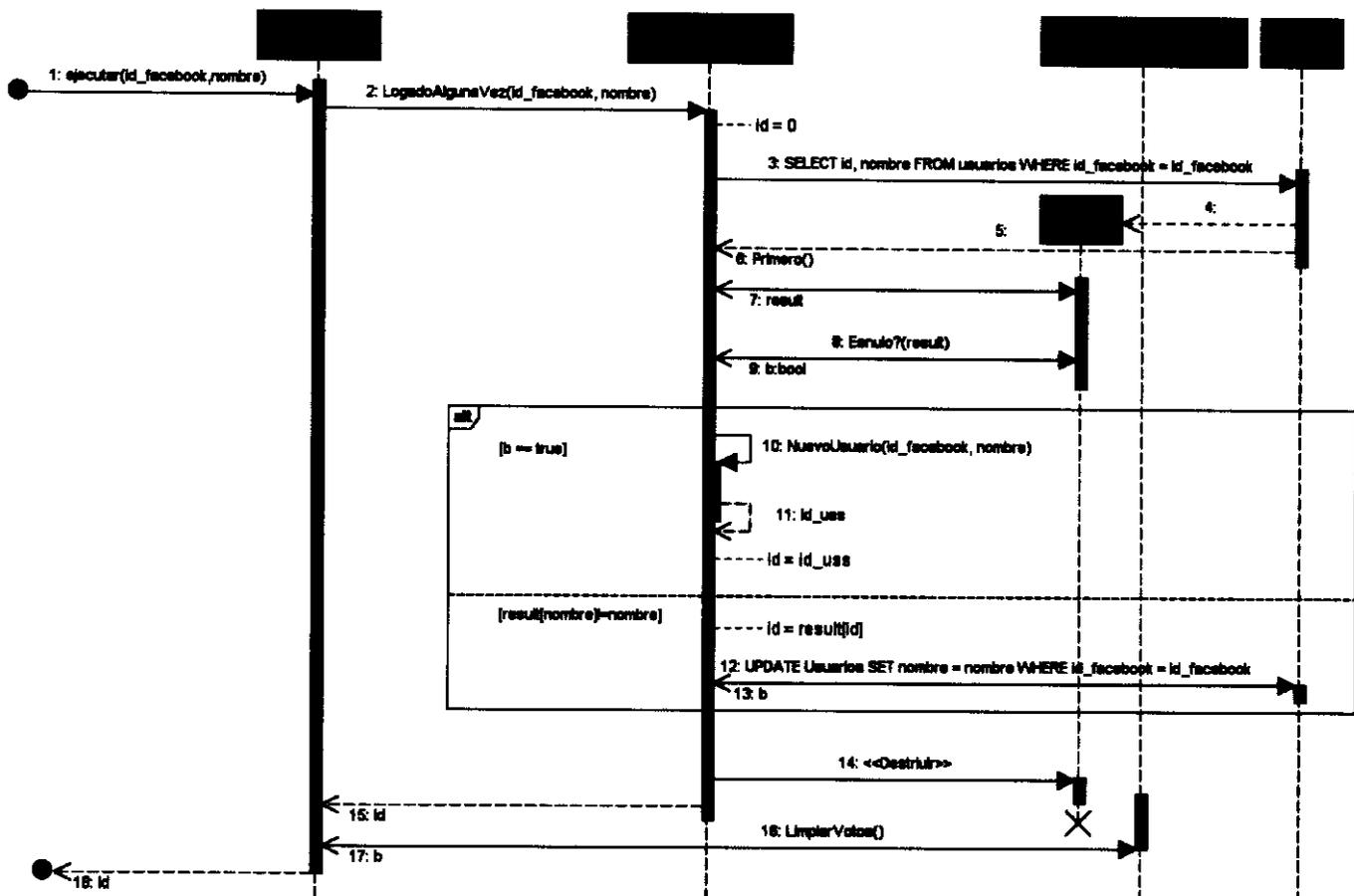
Fig. 14 : Ejemplo de traducción de una sentencia SQL a lenguaje natural.

Como puede comprobar en la traducción anterior, a excepción de algunos argumentos, la traducción entre SQL y el lenguaje natural es bastante intuitiva. Una vez se domina el lenguaje SQL, es más eficiente que el lenguaje natural.

En las siguientes páginas encontrara los diagramas de secuencia de todas las operaciones del sistema.

NOTA: Se recuerda que por motivos de practicidad y de espacio, previo acuerdo con el cliente, no se van a incluir todos los diagramas de secuencia, sino únicamente los más representativos. Los demás son prácticamente iguales y por tanto sería demasiado repetitivo para la lectura.

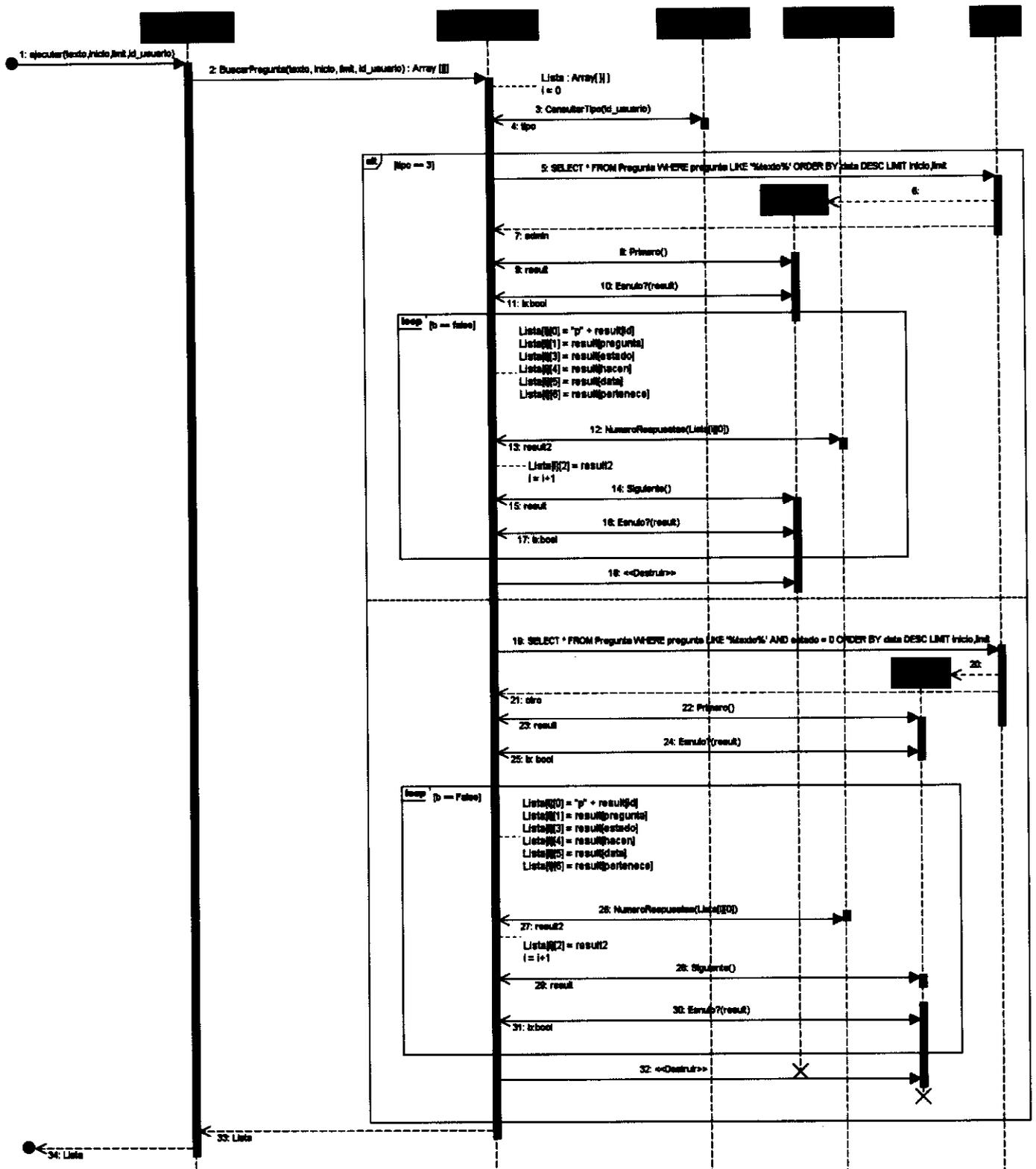
Operación Iniciar Sesión



Nota:

- La Operación NuevoUsuario está explicada detalladamente en la página 99
- La Operación LimpiarVotos está explicada detalladamente en la página 100

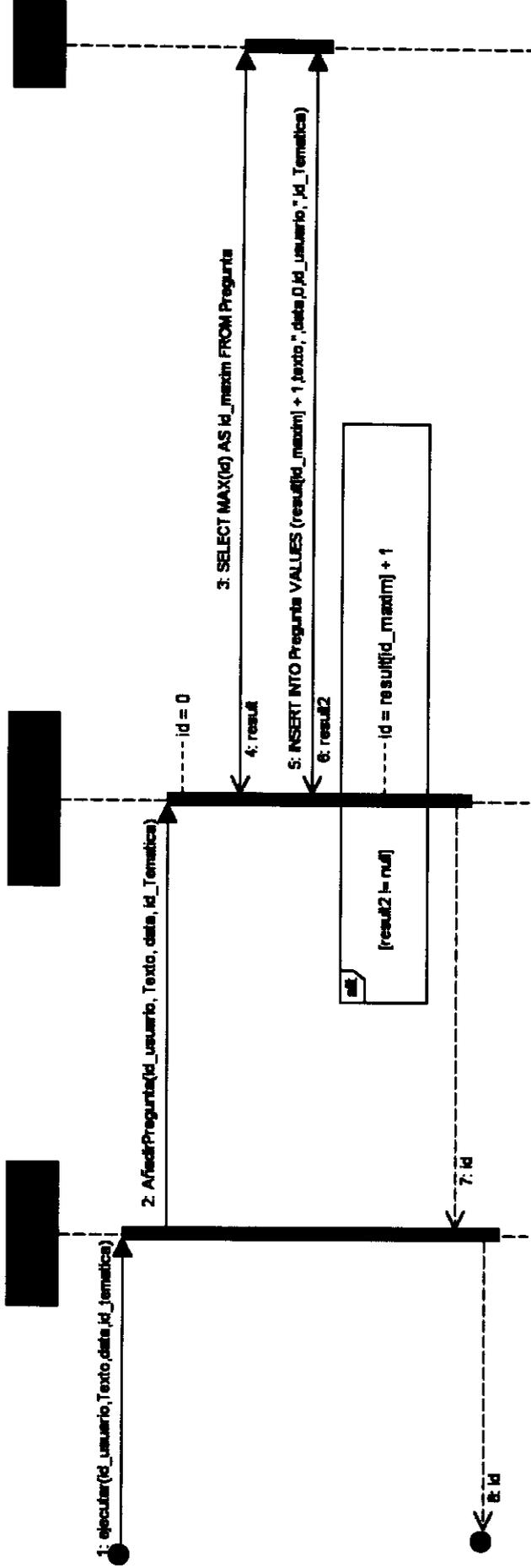
Operación Buscar Pregunta



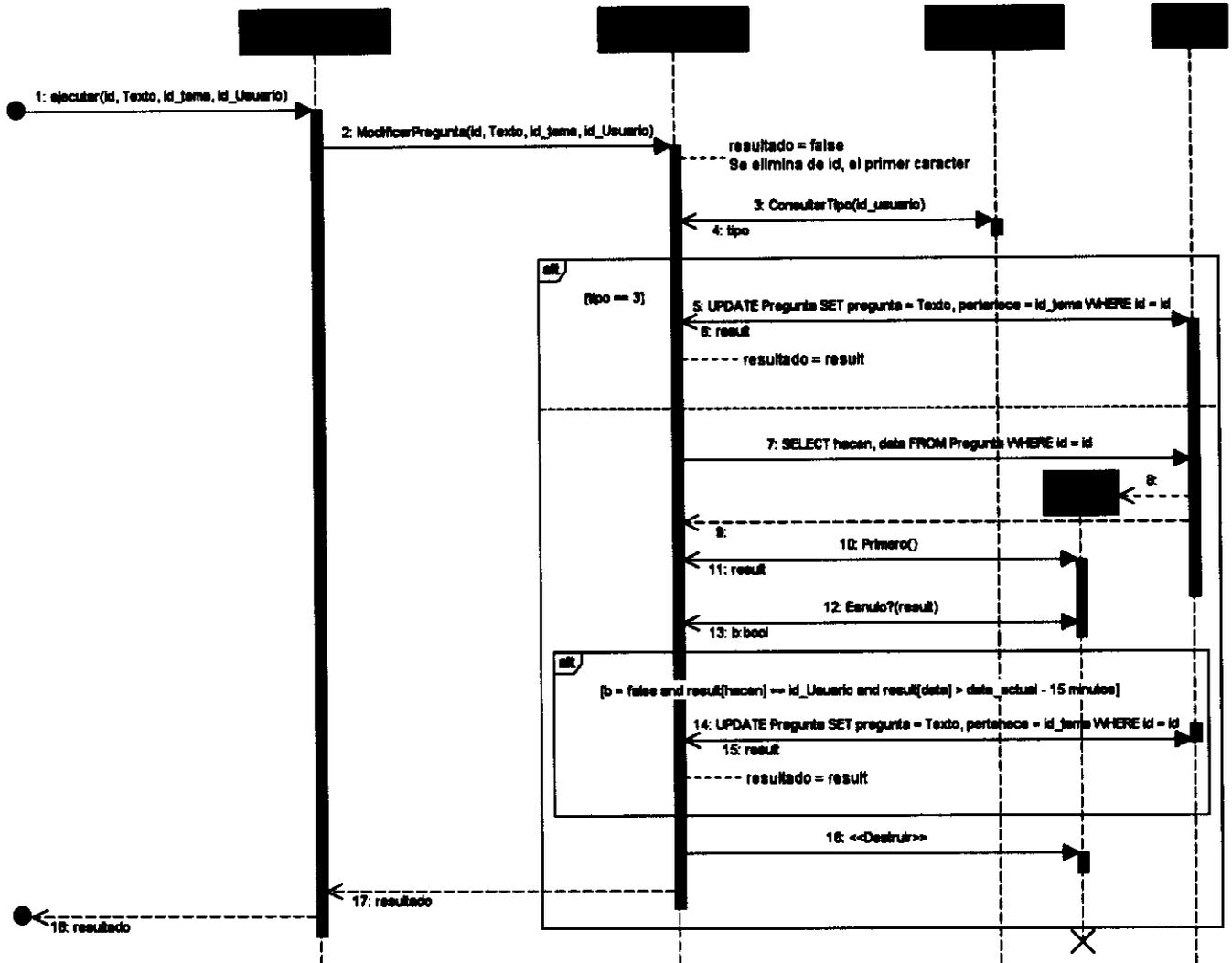
Nota:

- La Operación ConsultarTipo está explicada detalladamente en la página 101
- La Operación NumeroRespuestas está explicada detalladamente en la página 101

Operación Crear Pregunta



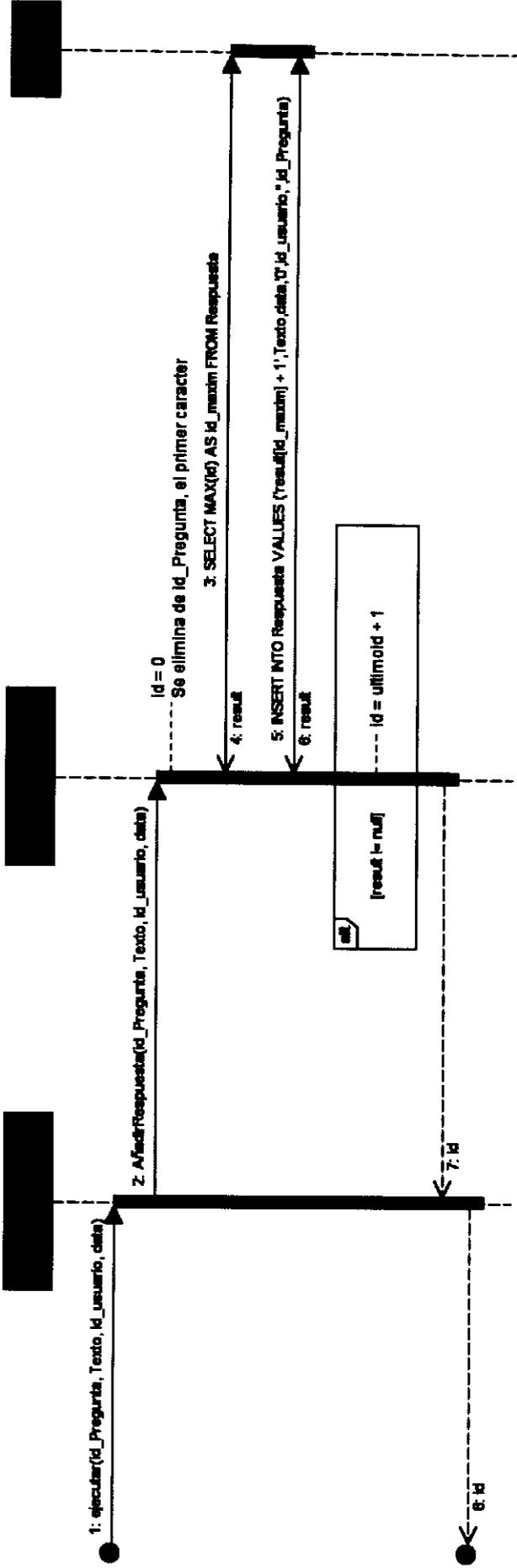
Operación Modificar Pregunta



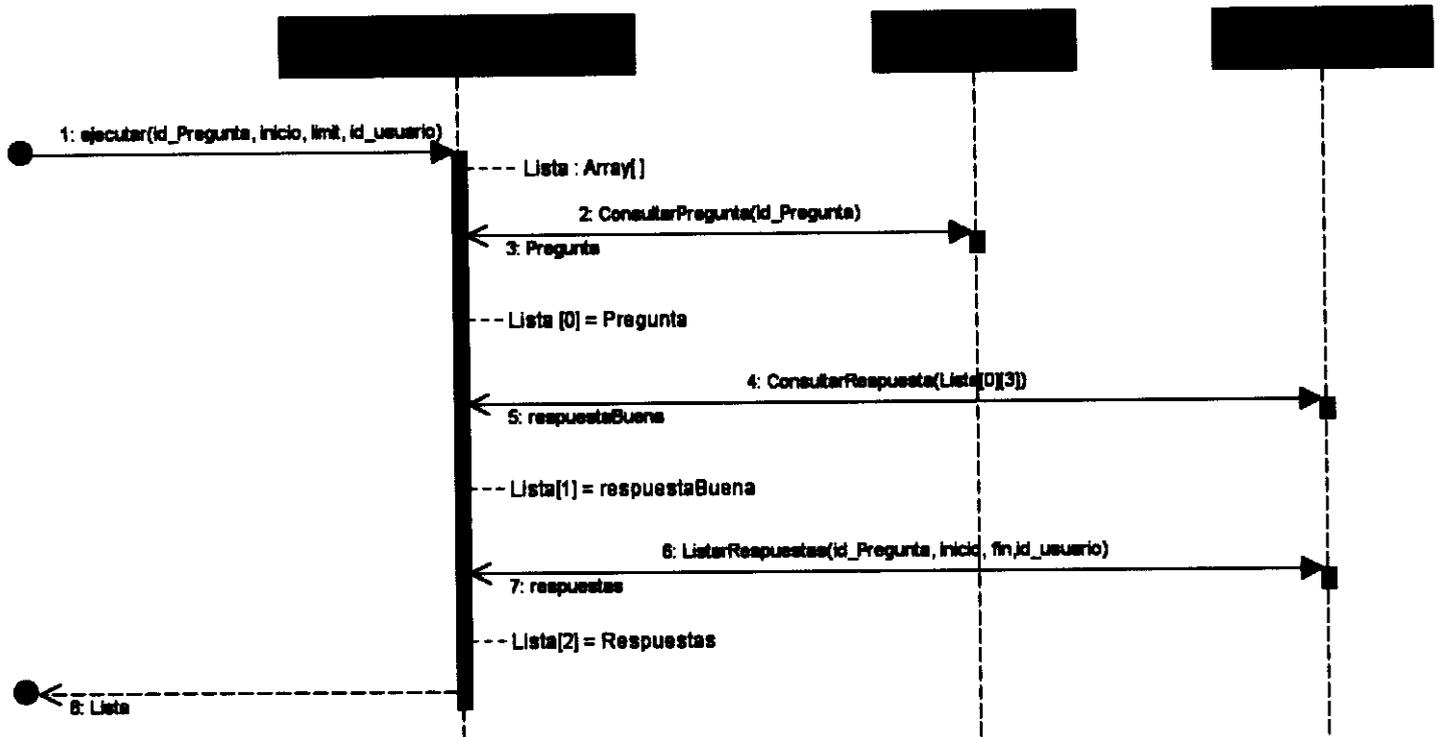
Nota:

- La Operación ConsultarTipo está explicada detalladamente en la página 101

Operación Responder a Pregunta



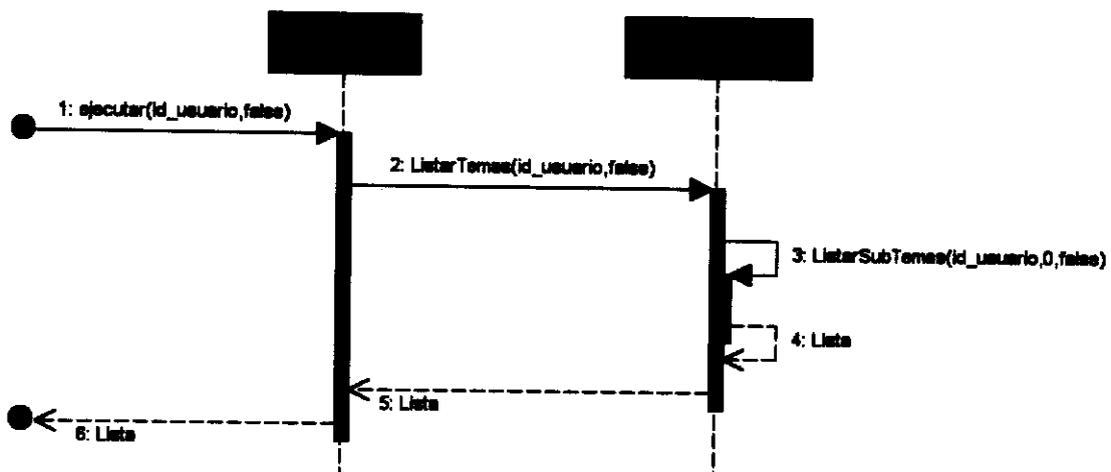
Operación Consultar Respuestas de una Pregunta



Nota:

- La Operación ConsultarRespuesta está explicada detalladamente en la página 98
- La Operación ListarRespuestas está explicada detalladamente en la página 102

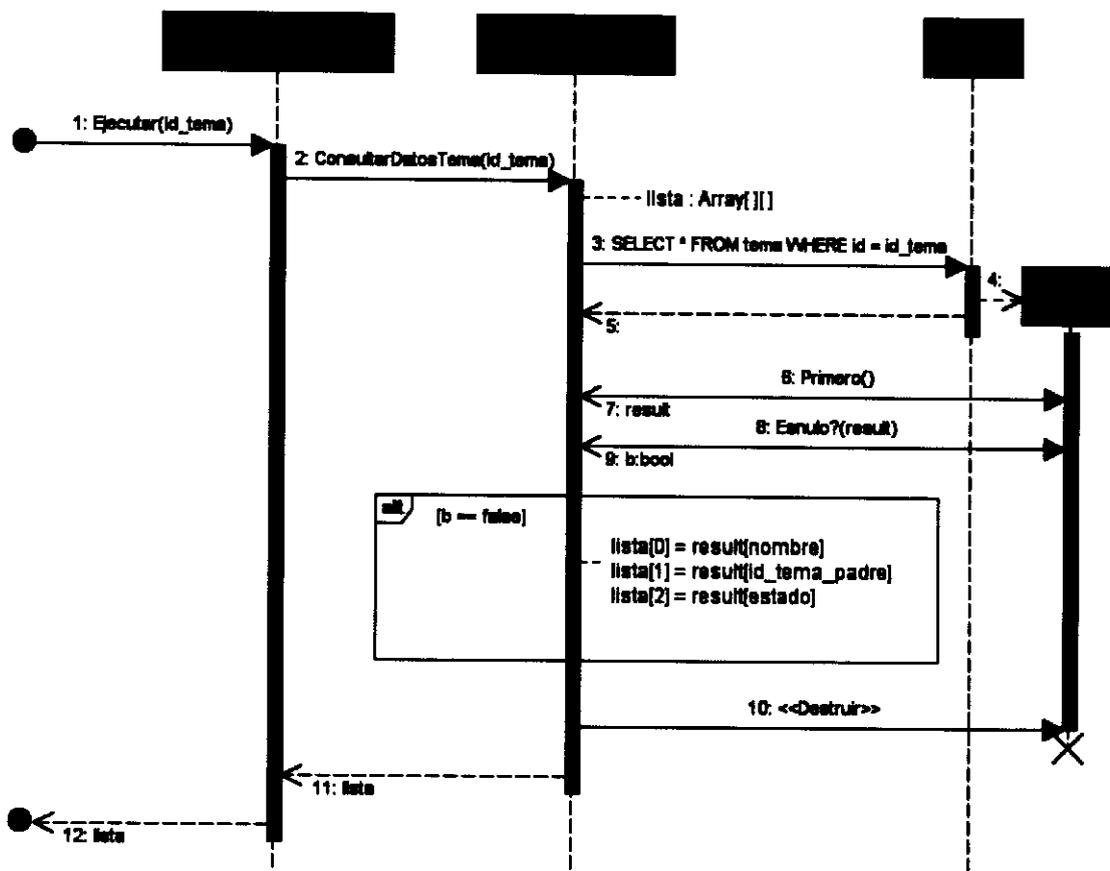
Operación Menú Temas



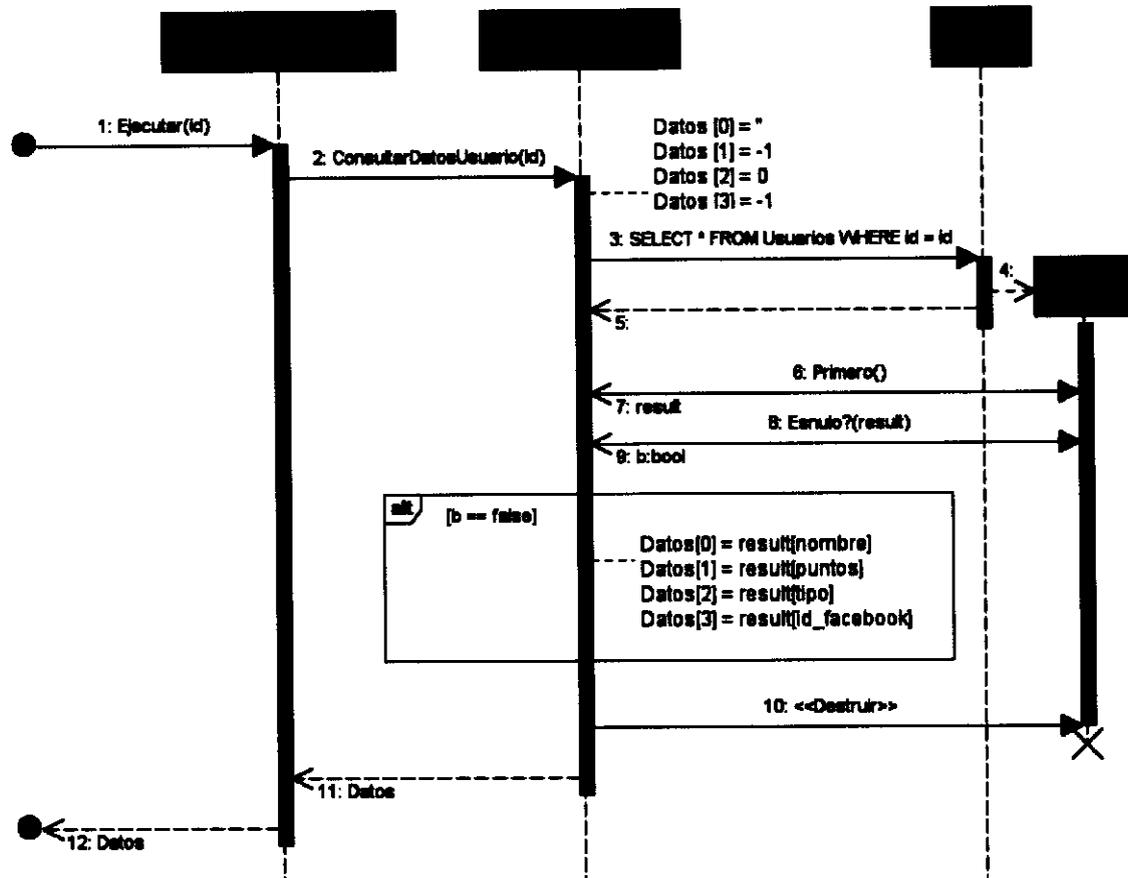
Nota:

- La Operación ListarSubTemas está explicada detalladamente en la página 103

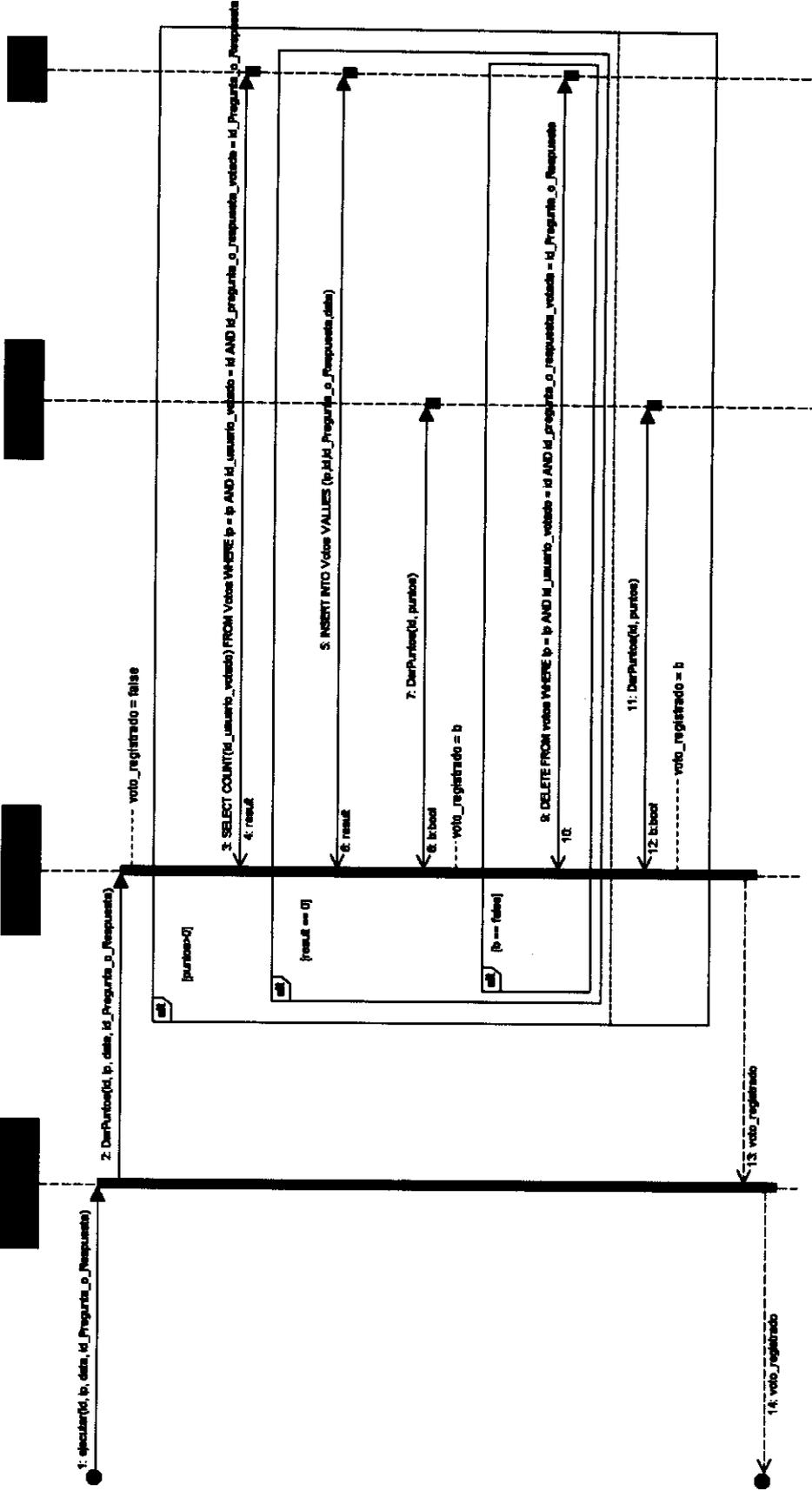
Operación Consultar Datos Tema



Operación Consultar Datos Usuario



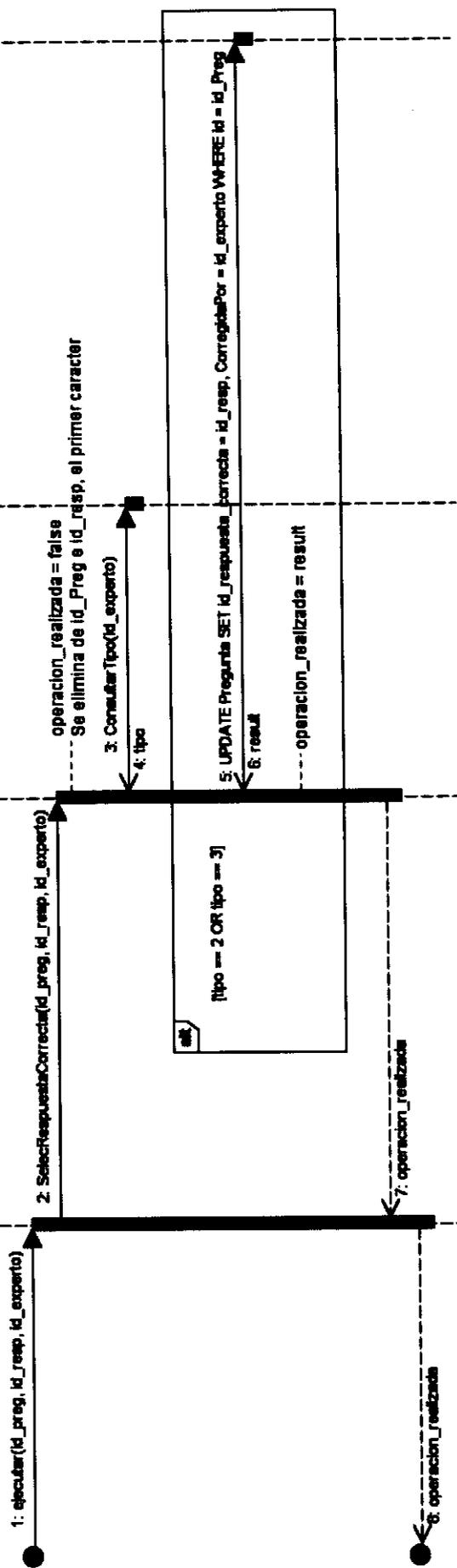
Operación Dar Puntos A Usuario



Nota:

- La Operación DarPuntos está explicada detalladamente en la página 104

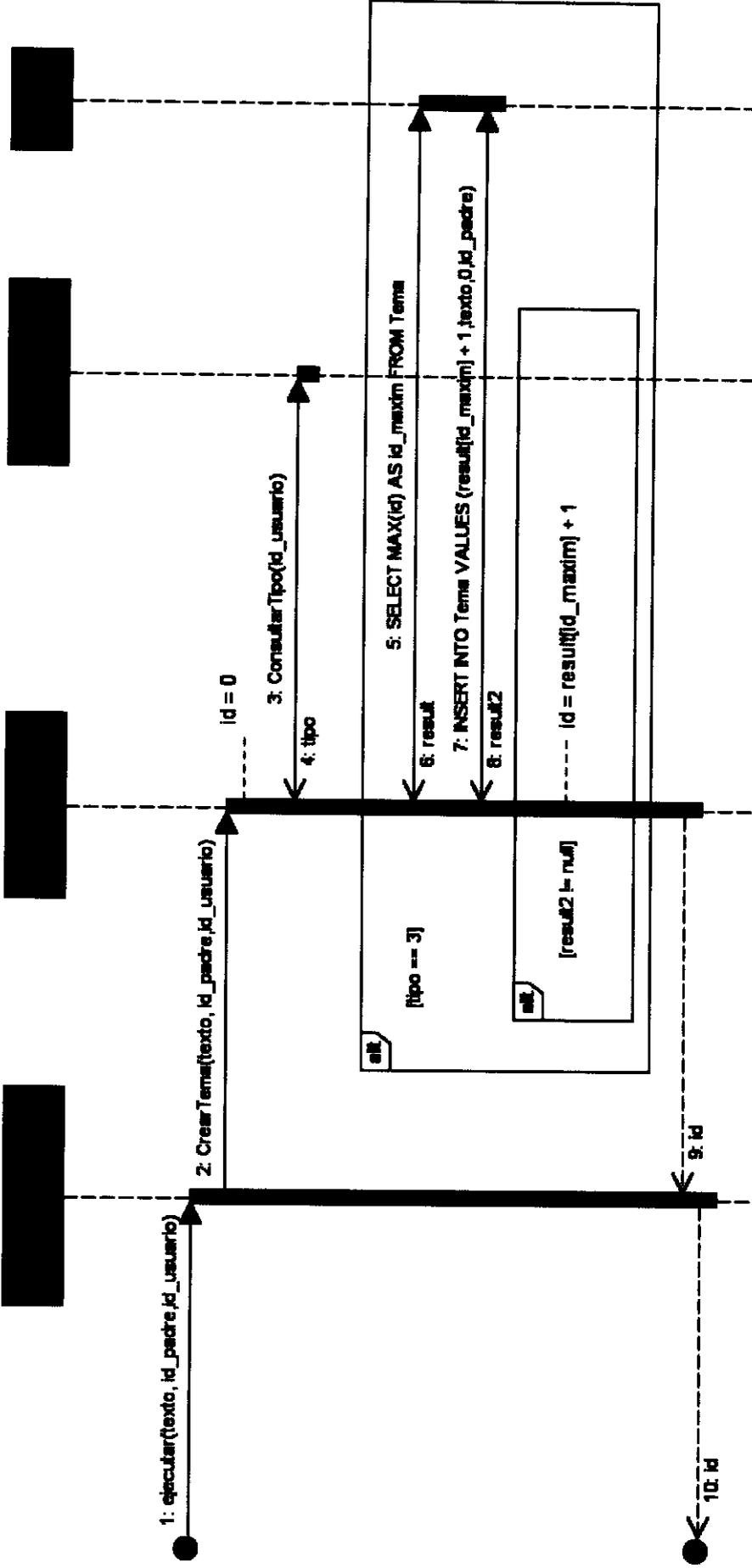
Operación Seleccionar Respuesta Correcta



Nota:

- La Operación ConsultarTipo está explicada detalladamente en la página 101

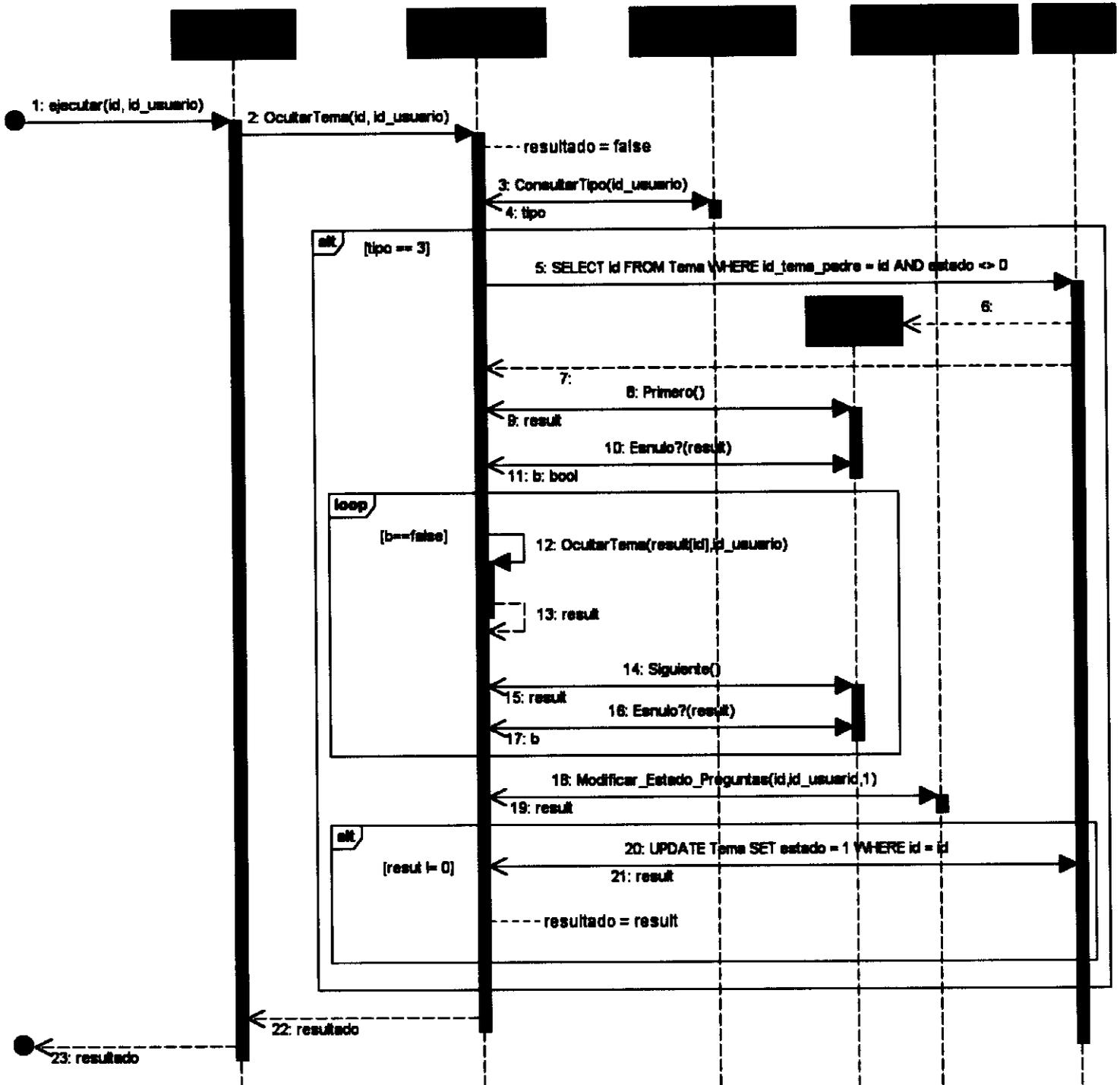
Operación Crear Tema/Subtema



Nota:

- La Operación ConsultarTipo está explicada detalladamente en la página 101

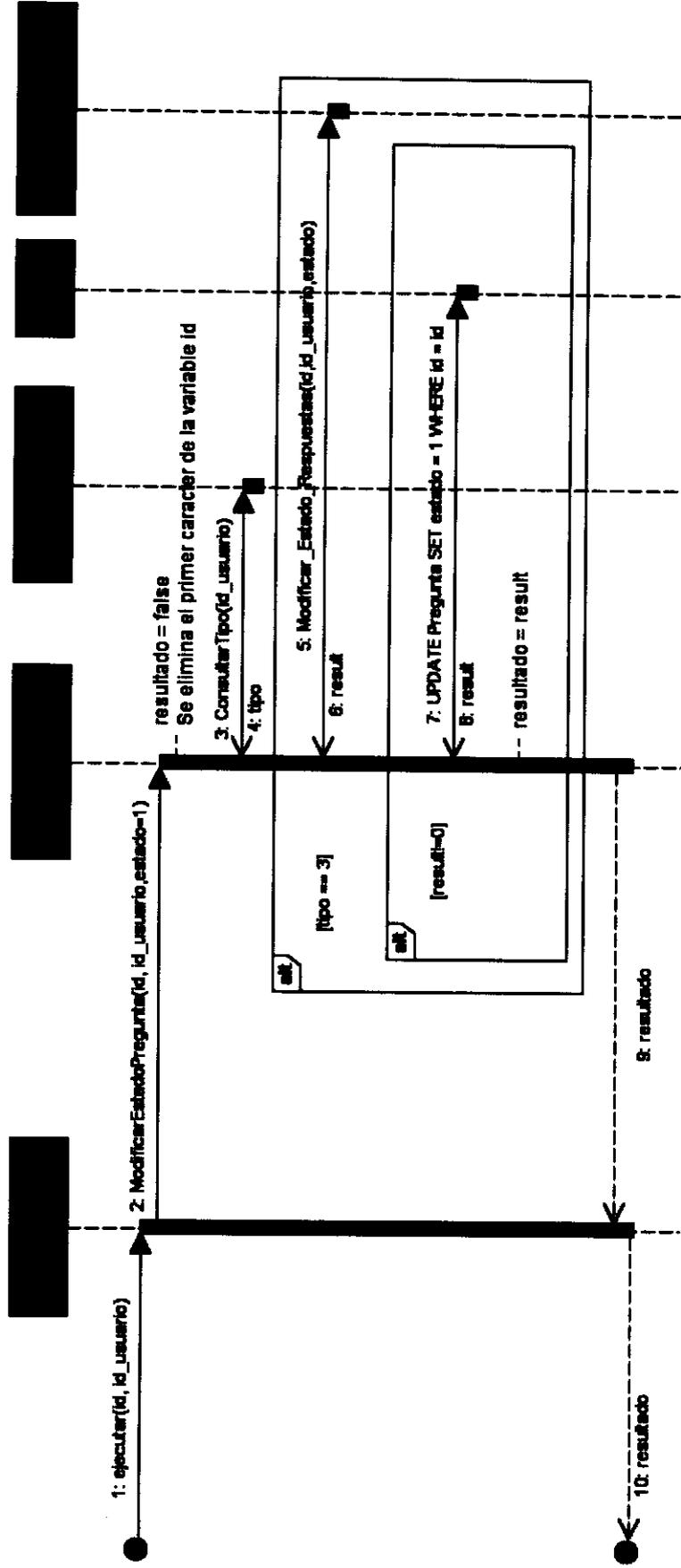
Operación Ocultar Tema



Nota:

- La Operación ConsultarTipo está explicada detalladamente en la página 101
- La Operación Modificar Estado Preguntas está explicada detalladamente en la página 106

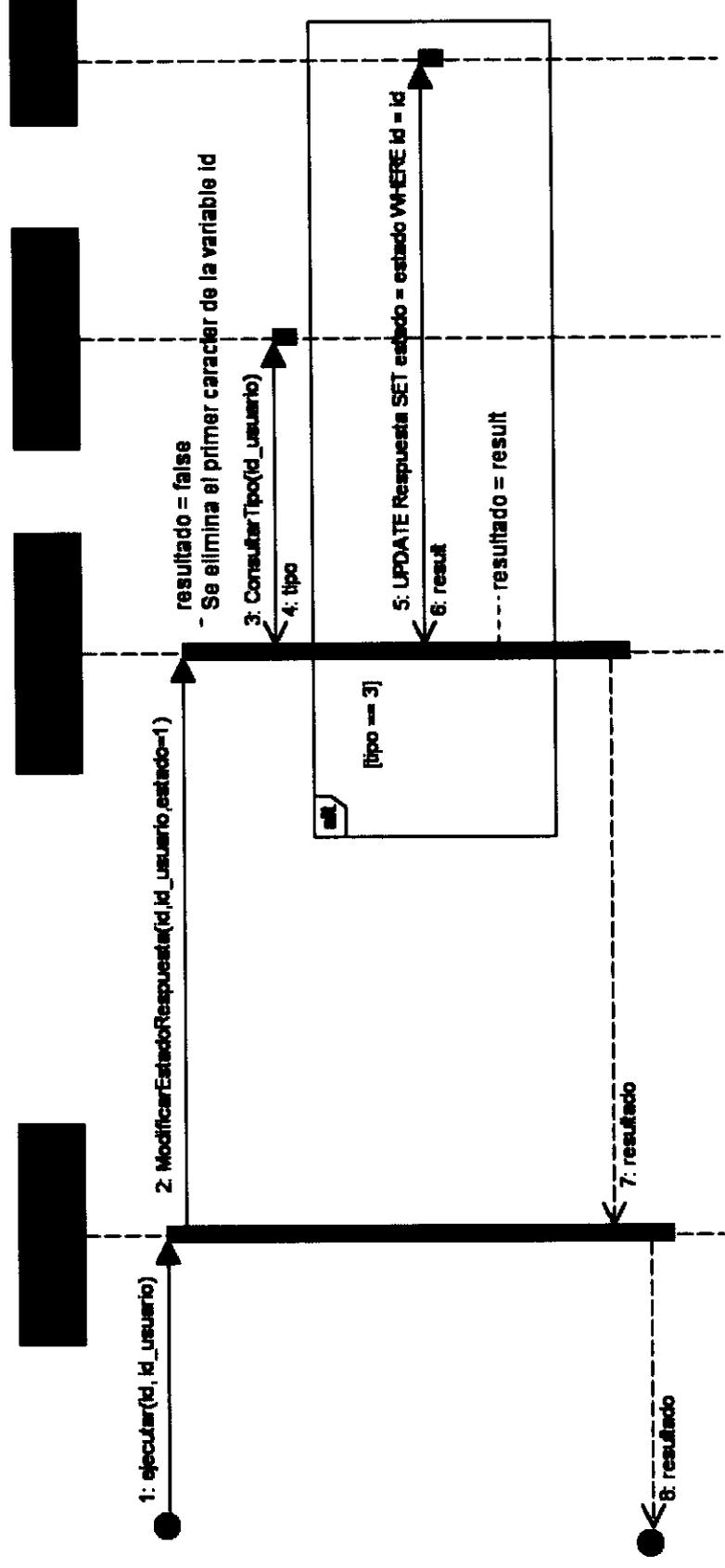
Operación Ocultar Pregunta



Nota:

- La Operación ConsultarTipo está explicada detalladamente en la página 101

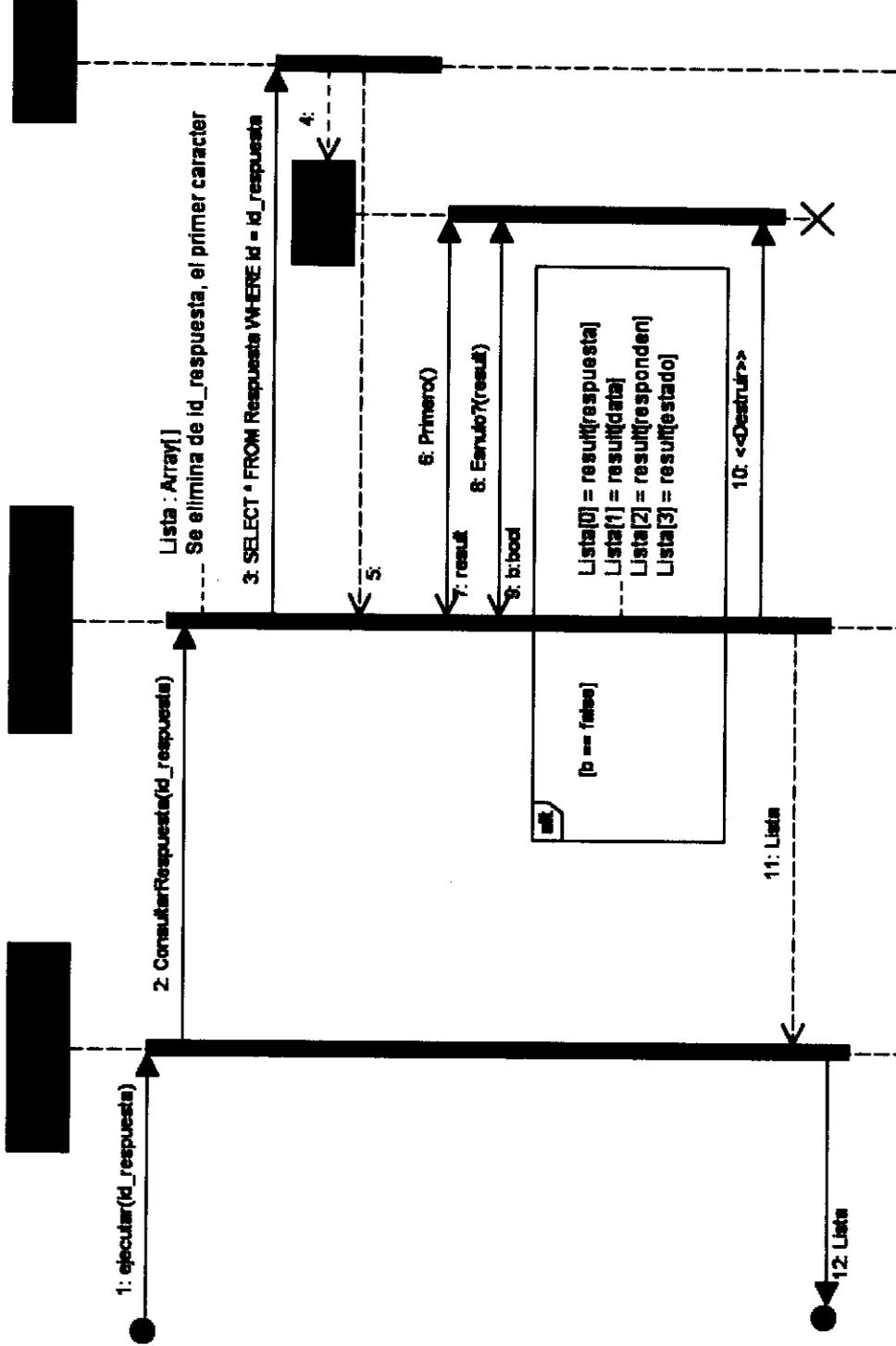
Operación Ocultar Respuesta



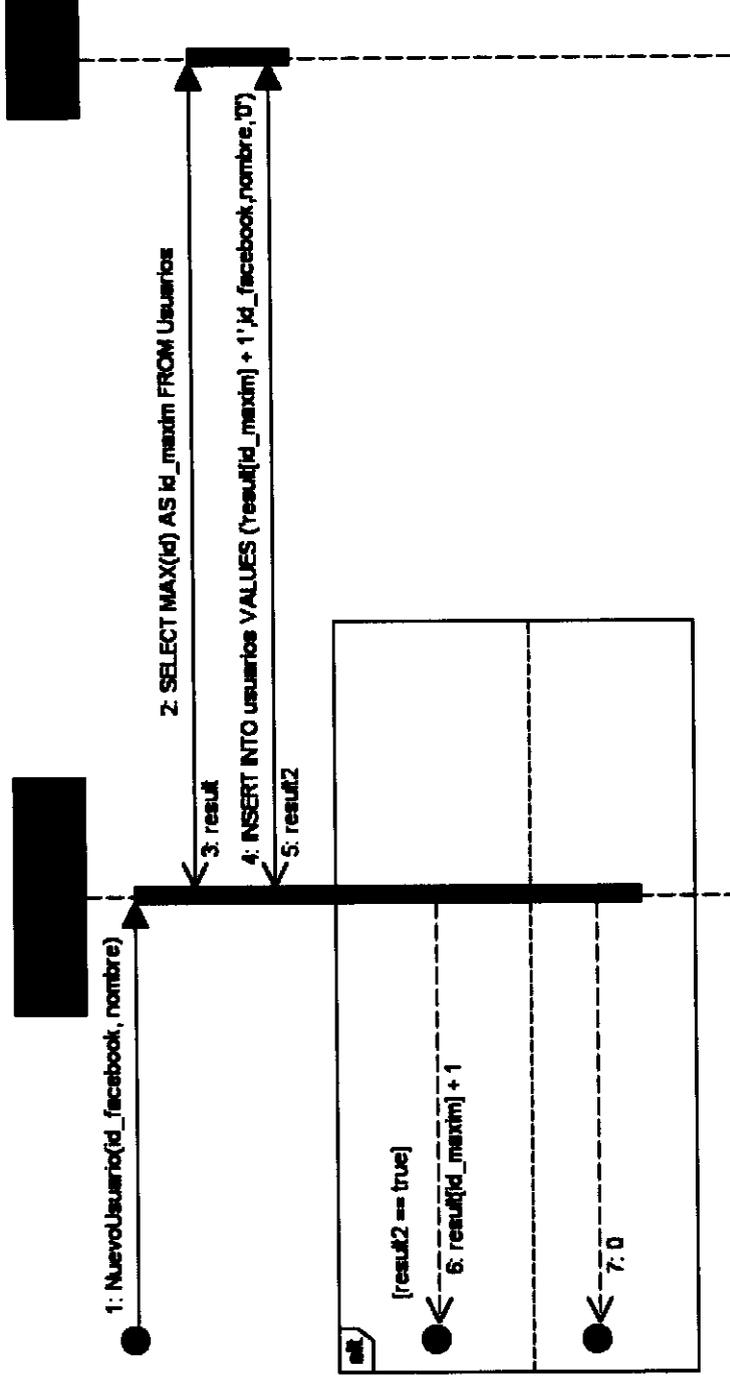
Nota:

- La Operación ConsultarTipo está explicada detalladamente en la página 101

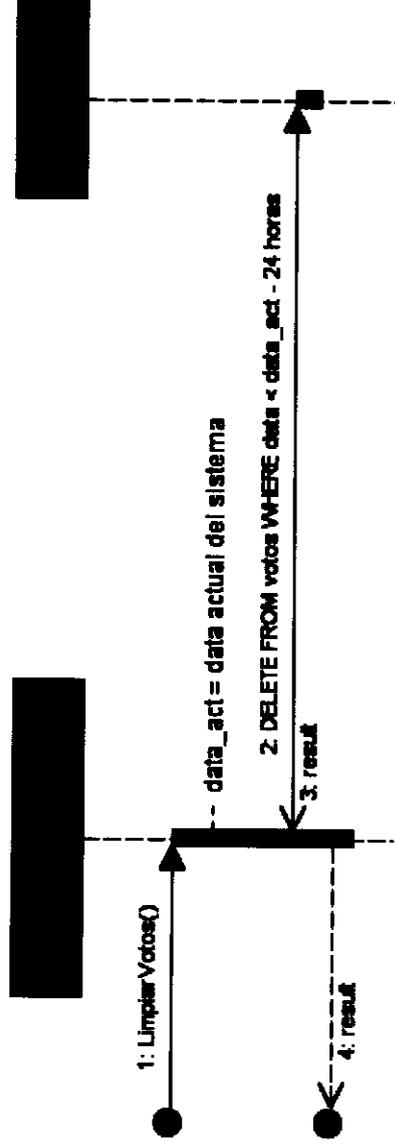
Operación ConsultarRespuesta



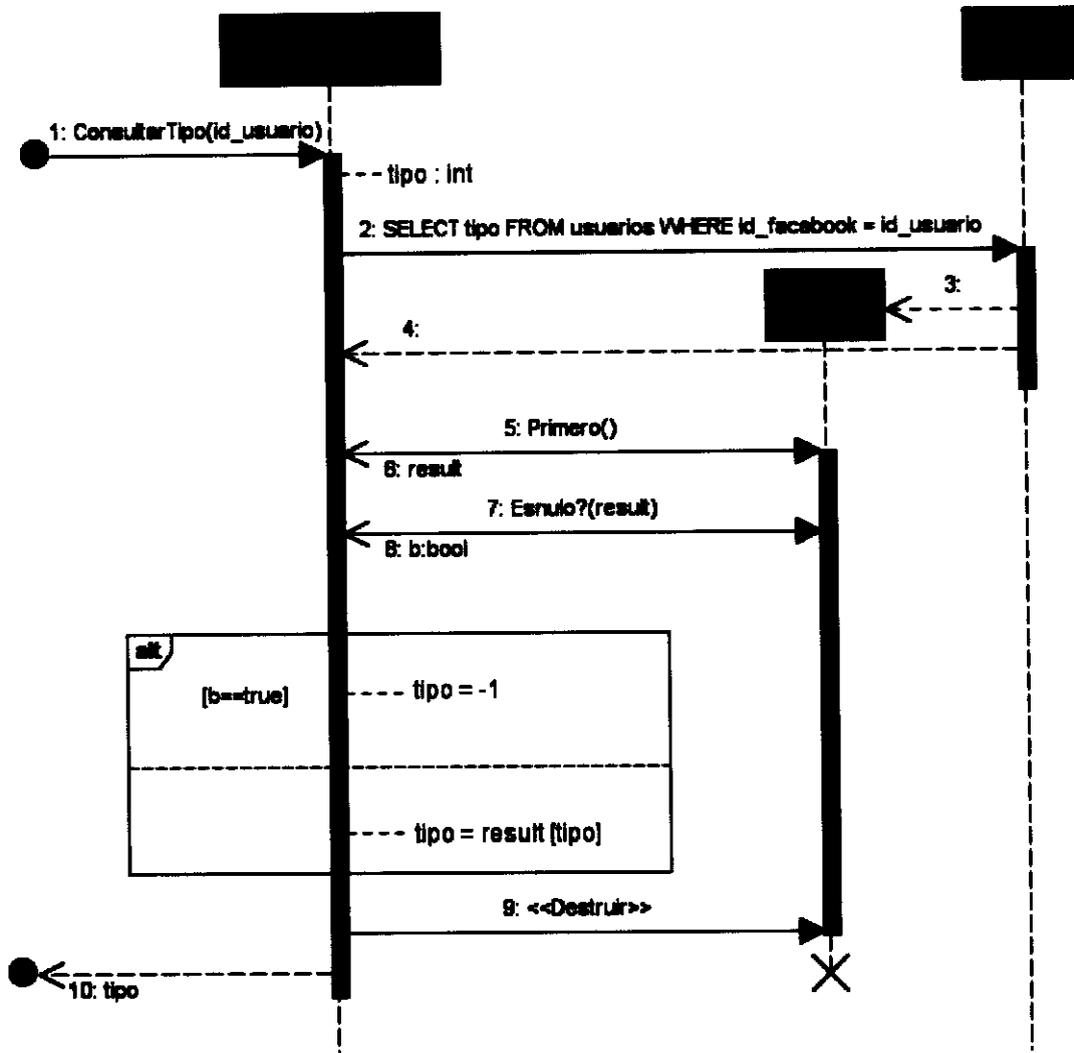
Operación NuevoUsuario



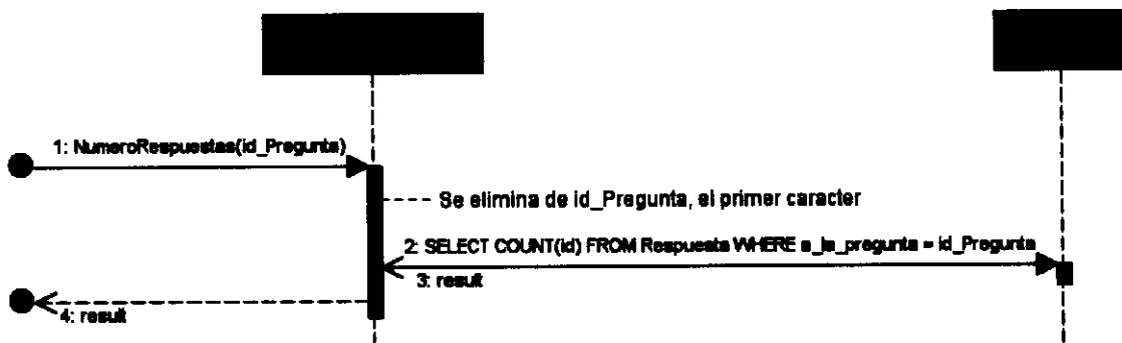
Operación LimpiarVotos



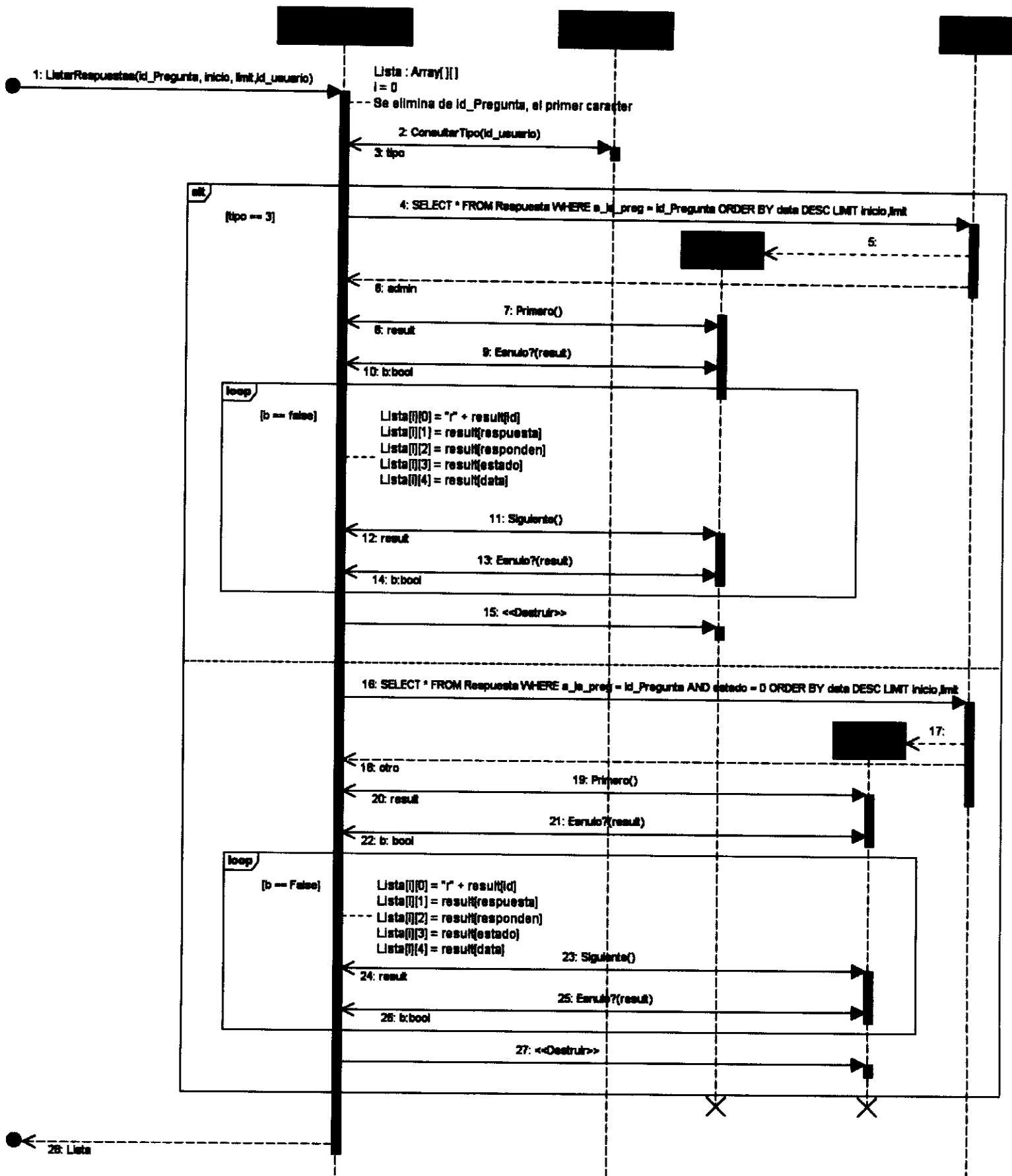
Operación ConsultarTipo



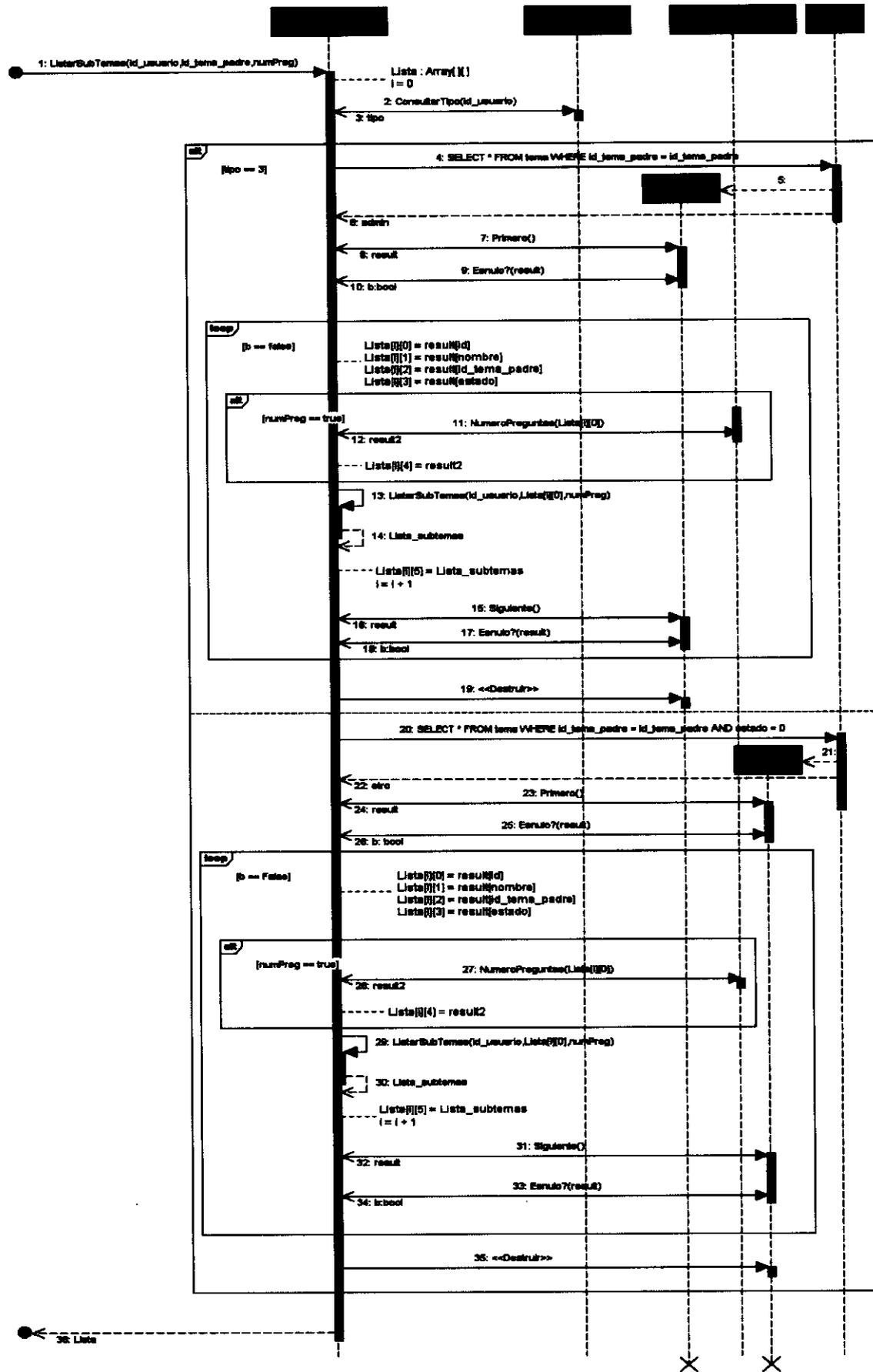
Operación NumeroRespuestas



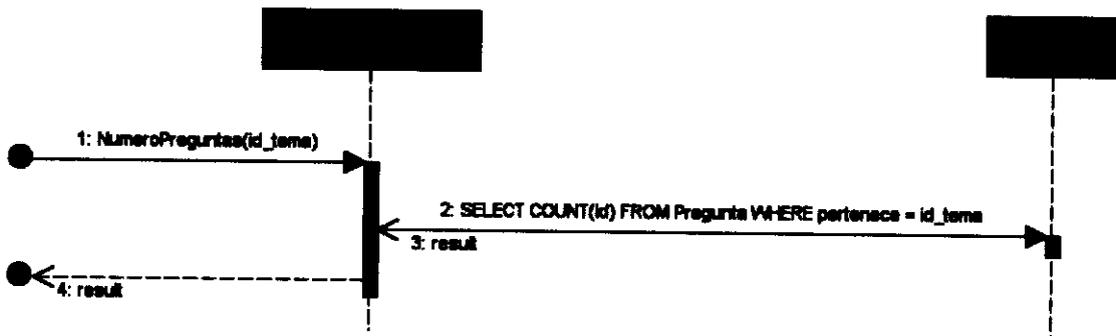
Operación ListarRespuestas



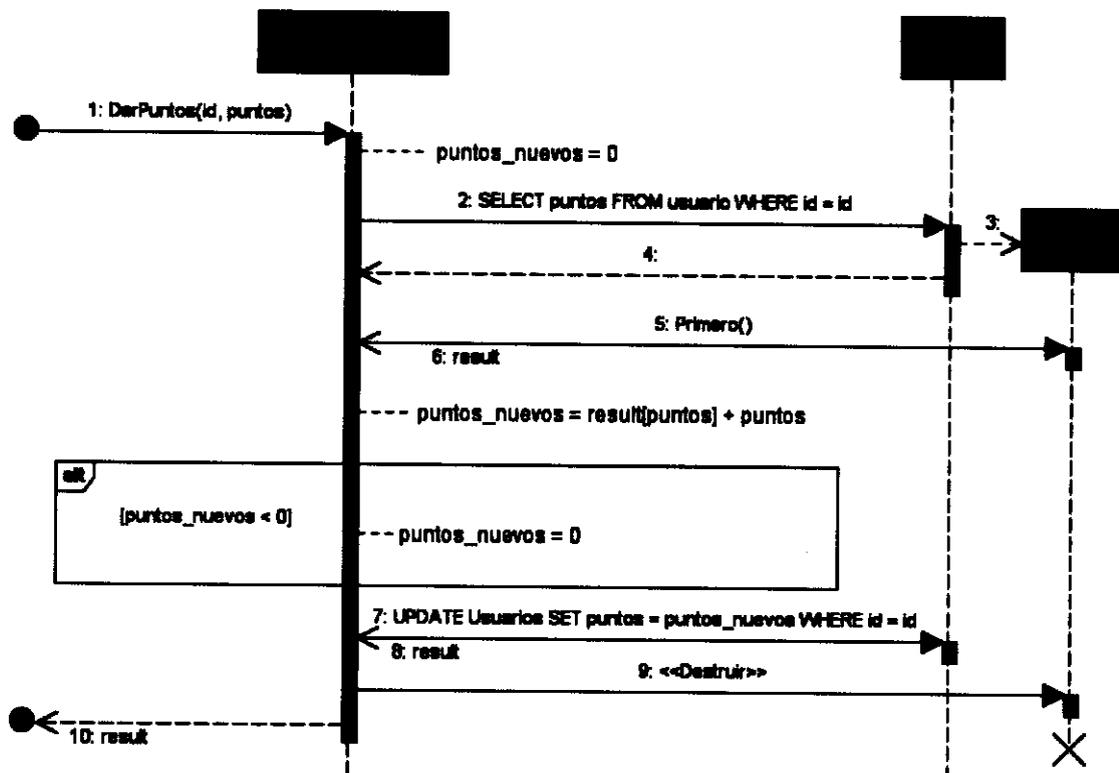
Operación ListarSubTemas



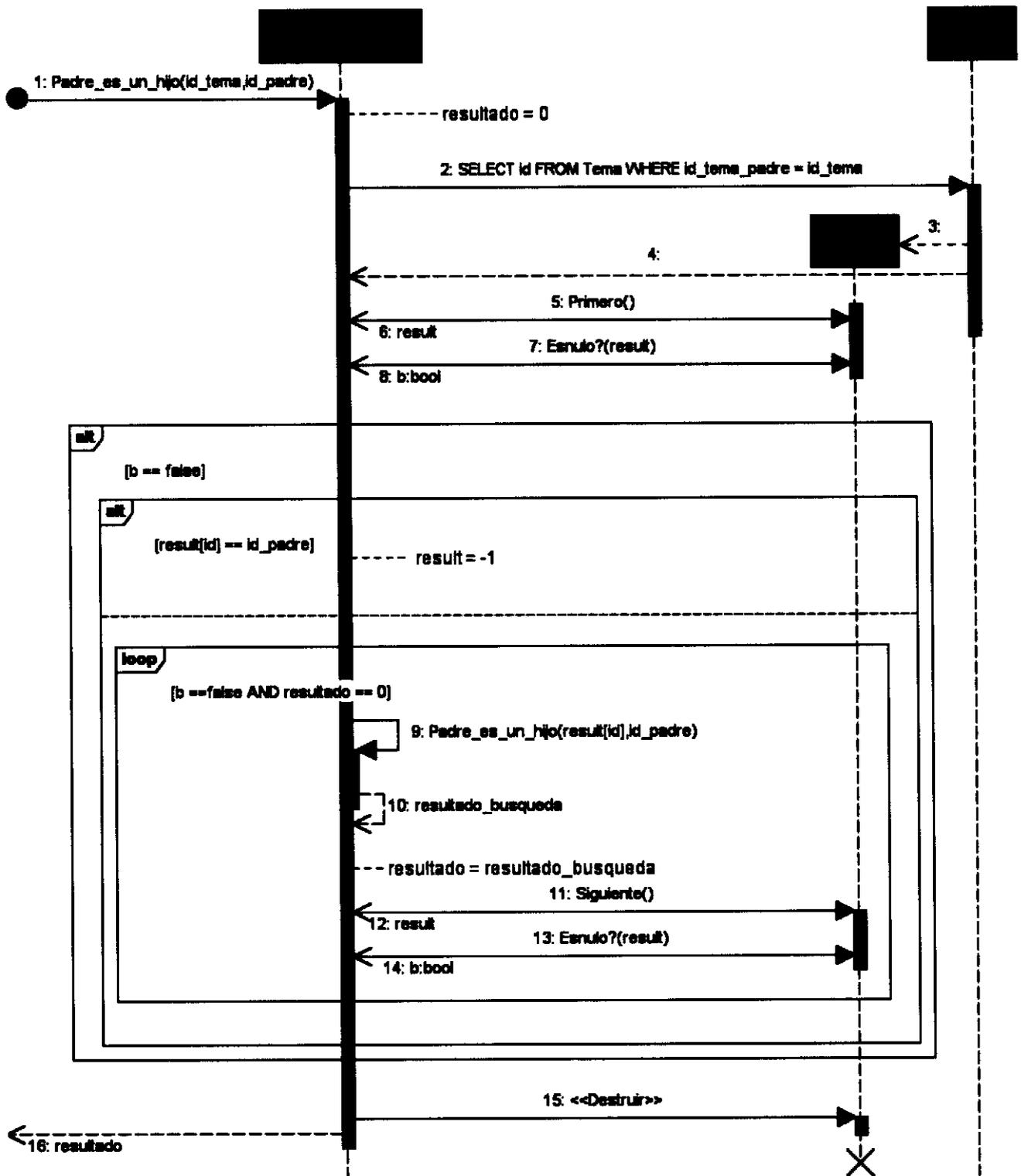
Operación NumeroPreguntas



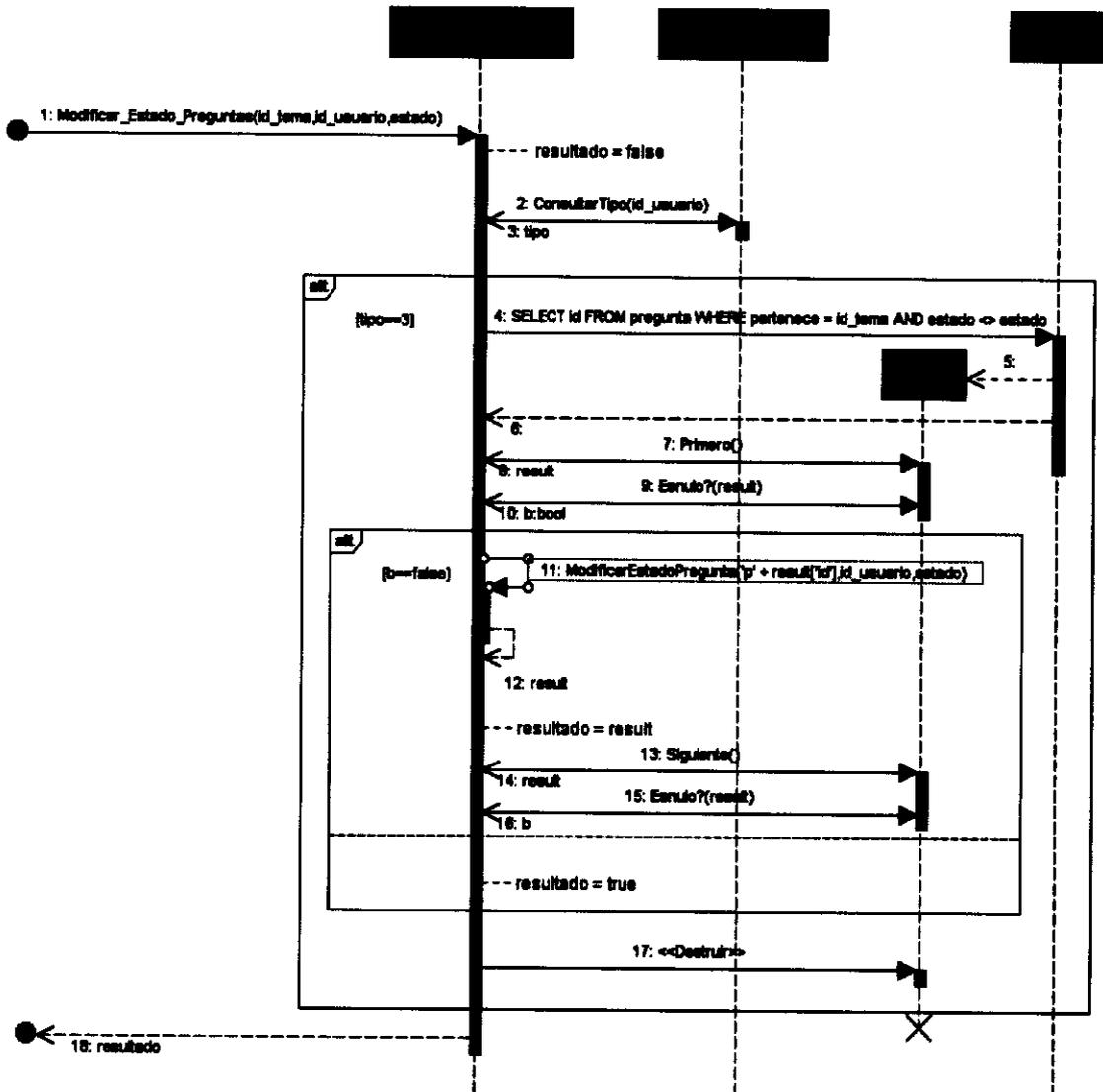
Operación DarPuntos



Operación Padre_es_hijo



Operación Modificar Estado Preguntas



Nota:

- La Operación ConsultarTipo está explicada detalladamente en la página 101
- La Operación Modificar Estado Pregunta está explicada detalladamente en la página 96

5.2 Diseño de la interfaz

El diseño de la interfaz es un punto de gran importancia dentro del desarrollo de una aplicación, puesto que es aquello con lo que el usuario está en contacto continuamente. Una buena o mala interfaz para un mismo programa (código) puede variar el uso de un usuario considerablemente (cuando más apropiada sea la interfaz mayor atractivo tendrá para el usuario y más veces usara la aplicación).

Los factores principales que pueden hacer mejor o peor una aplicación son:

- ✓ Colores empleados para mostrarla.
- ✓ Disposición de los elementos de comunicación.
- ✓ Mensajes de error poco claros.
- ✓ Cantidad de información mostrada simultáneamente.

Se podría matizar mucho sobre las características de la aplicación, pero para sintetizar se han extraído una serie de características que se han intentado lograr. Son las siguientes:

- ✓ Facilidad de navegación.
- ✓ Diseño grafico agradable
- ✓ Homogeneidad y consistencia entre las diferentes páginas que forman la aplicación (todas las pantallas dentro de la web siguen un estilo de comportamiento común y una misma estética).

5.2.1 Diseño de la interfaz del portal de El preguntometro

En este apartado se mostrarán las imágenes de las diferentes pantallas.

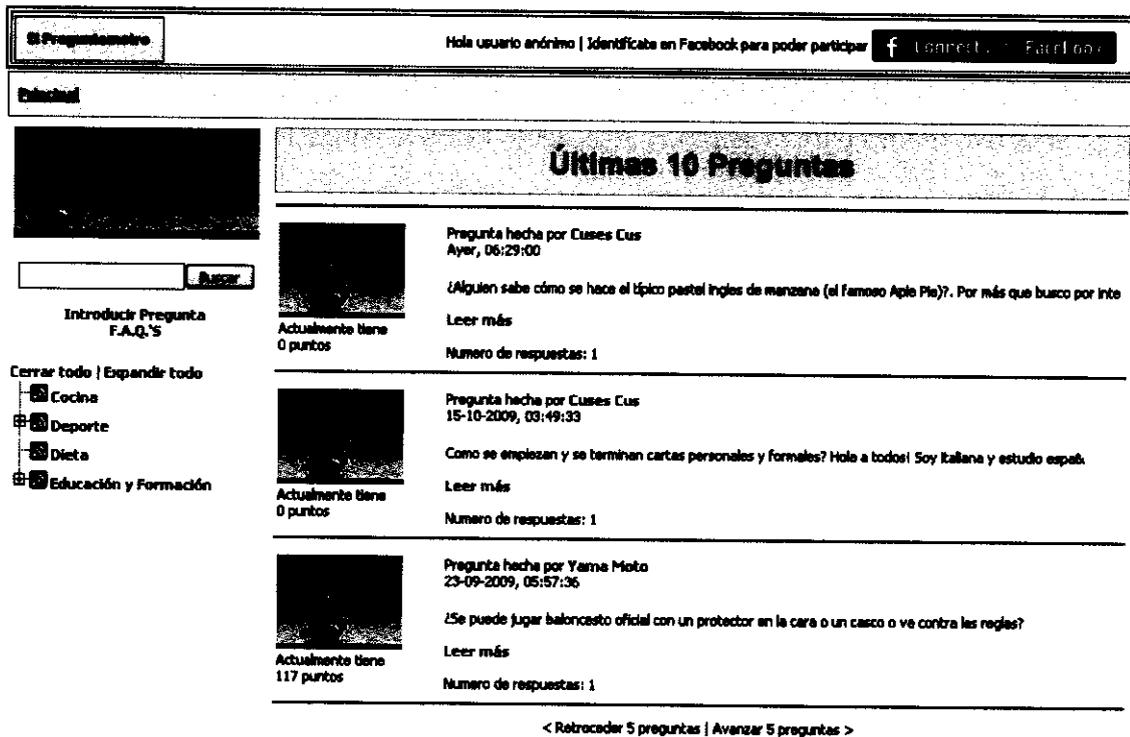


Fig. 15: Pagina principal de la aplicación

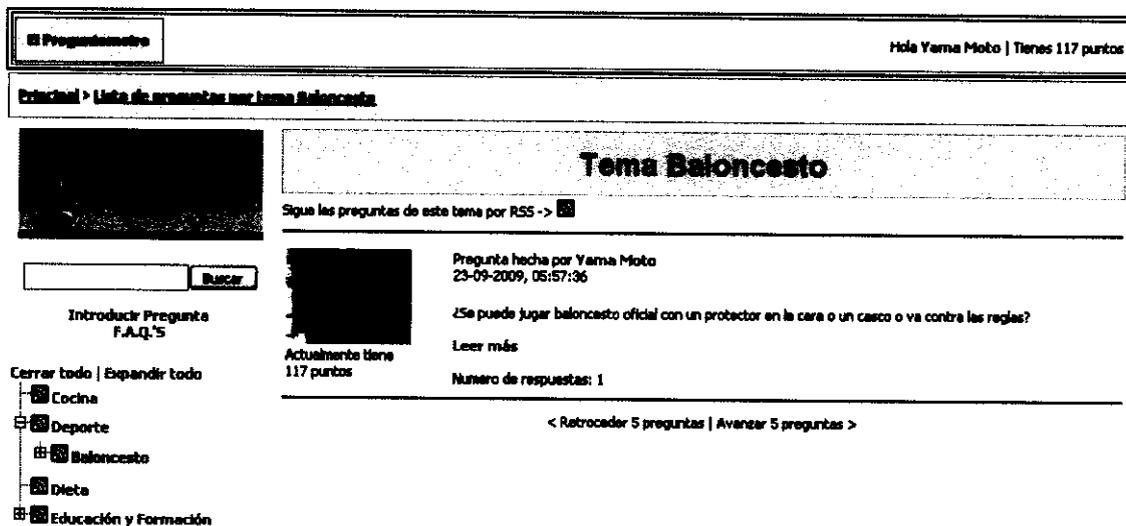


Fig. 16: Listado de preguntas de un tema.

El Preguntamestre Hola Yama Moto | Tienes 117 puntos

Principal > Lista de preguntas por tema Baloncesto > Pregunta hecha por usuario Yama Moto

Sigue las respuestas de esta pregunta por RSS ->



Introducir Pregunta
F.A.Q.'S

Cerrar todo | Expandir todo

- Cocina
- Deporte
- Dieta
- Educación y Formación

Pregunta hecha por Yama Moto
23-09-2009, 05:57:36

¿Se puede jugar baloncesto oficial con un protector en la cara o un casco o va contra las reglas?

Vota al usuario
Yama Moto

☆☆☆☆☆
Actualmente tiene
117 puntos

Es que ya me fracturaron la nariz y siempre tengo la mala suerte de que me rompan los labios y me paran golpeando la mandíbula.

Respuesta Correcta

Respuesta hecha por Yama Moto
23-09-2009, 06:00:22

Se usan los protectores para la cara, no los cascos, hay muchos jugadores que usan los protectores especialmente aquellos que han tenido algún ...

Actualmente tiene
117 puntos

Pregunta validada por Yama Moto

Leer más

Lista de Respuestas



Actualmente tiene
117 puntos

Respuesta hecha por Yama Moto
23-09-2009, 06:00:22

Se usan los protectores para la cara, no los cascos, hay muchos jugadores que usan los protectores especialmente aquellos que han tenido algún ...

Leer más

< Retroceder 5 respuestas | Avanzar 5 respuestas >

Fig. 17: Visor de una pregunta y listado de respuestas de una pregunta.

El Preguntamestre Hola Yama Moto | Tienes 117 puntos

Principal > Lista de preguntas por tema Baloncesto > Pregunta hecha por usuario Yama Moto > Respuesta hecha por usuario Yama Moto



Introducir Pregunta
F.A.Q.'S

Cerrar todo | Expandir todo

- Cocina
- Deporte
- Dieta
- Educación y Formación

Respuesta hecha por Yama Moto
23-09-2009, 06:00:22

Se usan los protectores para la cara, no los cascos, hay muchos jugadores que usan los protectores especialmente aquellos que han tenido algún golpe en la nariz (algunos que se han roto el tabique nasal). Un ejemplo es Richard Hamilton el jugador de Detroit Pistons que la empezó a usar por un golpe que tuvo y después se la dejó para siempre.



Actualmente tiene
117 puntos



Fig. 18: Visor de Respuestas

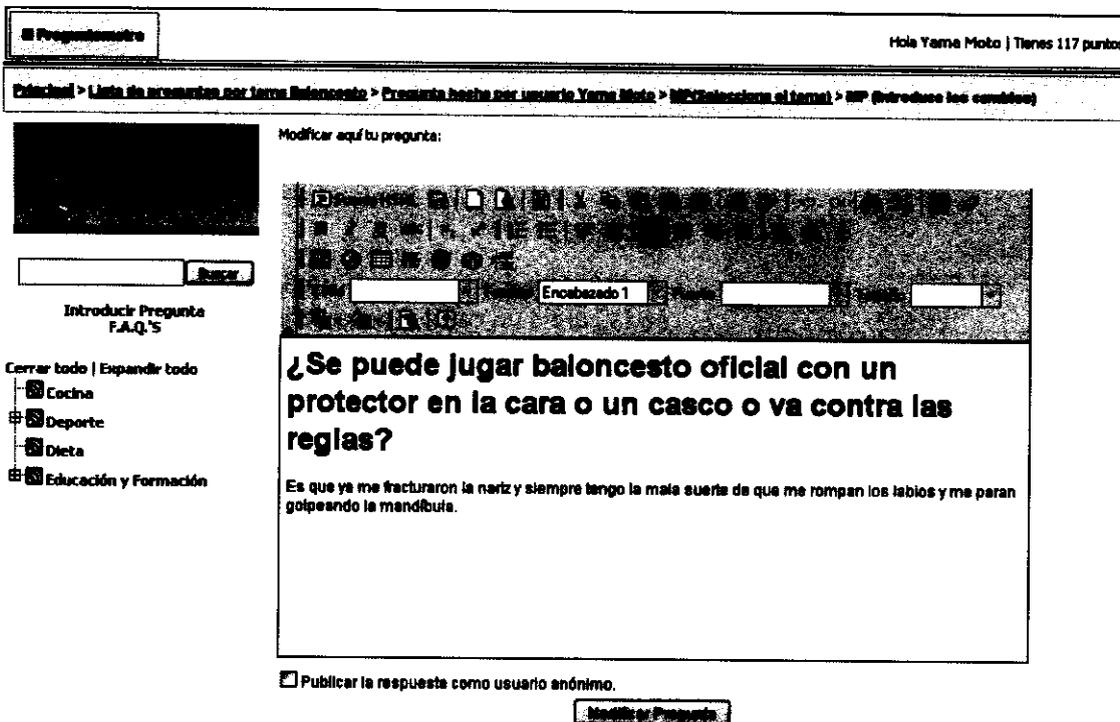


Fig. 19: Editor de preguntas y respuestas

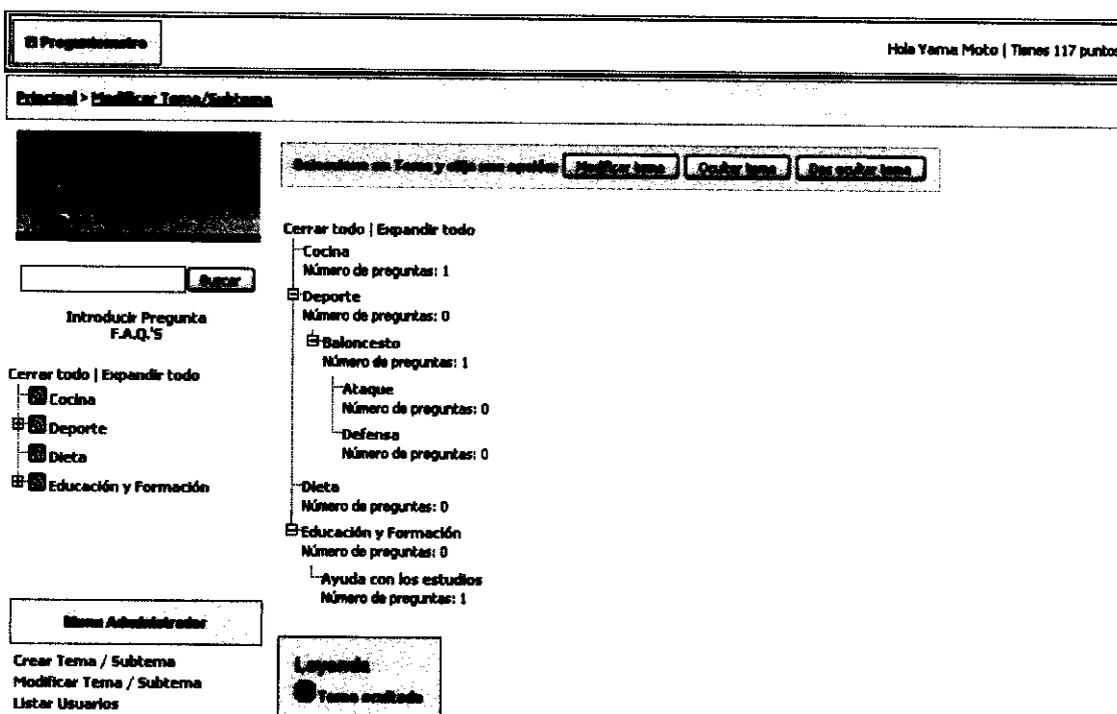


Fig. 20: Gestión de temas

El Preguntamestre
Hola Yama Moto | Tienes 117 puntos

[Inicio](#) > [Listar Usuarios](#)



Introducir Pregunta
F.A.Q.'S

Cerrar todo | Expandir todo

- Cocina
- Deporte
- Dieta
- Educación y Formación

Usuarios Registrados:

Ordenar por: Nombre | Puntos | Id Facebook

Usuarios ordenados por Nombre

Admin	Exp	Normal	Nombre	Puntos		
			Carlos Villavieja	0		
			Cuses Cus	0		
			Javier Vasallo González	5		
			Patricia Jiménez Mora	0		

< Retroceder 5 usuarios | Avanzar 5 usuarios >

Menus Administrador

[Crear Tema / Subtema](#)
[Modificar Tema / Subtema](#)
[Listar Usuarios](#)

Fig. 21: Gestión de usuarios

6. Implementación

6.1 Introducción

En este capítulo se describe todo lo referente a la etapa de implementación y a las tecnologías empleadas para la realización de este proyecto, justificando en todo momento las decisiones tomadas y mostrando ejemplos claros (también de código).

A continuación se muestra un pequeño resumen de las tecnologías lenguajes utilizados:

Navegadores(Browsers)	Internet Explorer v.7 y 8, Firefox v.3.5, Chrome v.3.0
SGBD(Sistema Gestor de Bases de Datos)	MySQL v.5.1

Tabla 12: Tecnologías empleadas

6.2 Tecnologías empleadas

A continuación se van a describir con detalle cada una de las tecnologías y conceptos más importantes que se han empleado en la implementación de este proyecto, así como los motivos de su elección.

6.2.1 La tecnología web

El principio de Internet se remonta hacia el año 1969 cuando comienza a formarse la llamada ARPANET, promovida por el DOD (Departament of Defense) de los EEUU. Se trataba de un conjunto de redes experimentales de conmutación de datos. Su objetivo era poder acceder a Bases de Datos militares desde cualquier punto del territorio en caso de enfrentamientos militares que imposibilitasen el transporte de personas y/o datos por los medios convencionales.

No es hasta 1993 cuando se comienza a hablar de la WWW (World Wide Web) como tal, aunque en esos momentos la red básicamente se usaba en entornos académicos y científicos. Poco a poco se fueron introduciendo elementos multimedia que la fueron haciendo más agradable para otros entornos no tan específicos. En los años siguientes fue erigiendo como una gran herramienta para la comunidad empresarial. A día de hoy, es difícil cuantificar en número de webs online que existen, (según estimaciones hechas en el 2006, el numero superaba los 600.000 millones de webs online, y su crecimiento era exponencial).

6.2.2 Cliente / Servidor

Existe un tipo específico de arquitectura orientada a las redes TCP / IP. Este tipo de programación es llamado de **Cliente / Servidor**.

El concepto de este tipo de arquitectura es muy sencillo y consiste en que el programa desarrollado se divide en dos: por un lado en el **servidor** y por otro en la aplicación de la máquina del usuario que realiza la petición al servidor (llamada aquí **cliente**).

Entre las principales **características de la arquitectura cliente / servidor** se pueden destacar la siguientes.

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

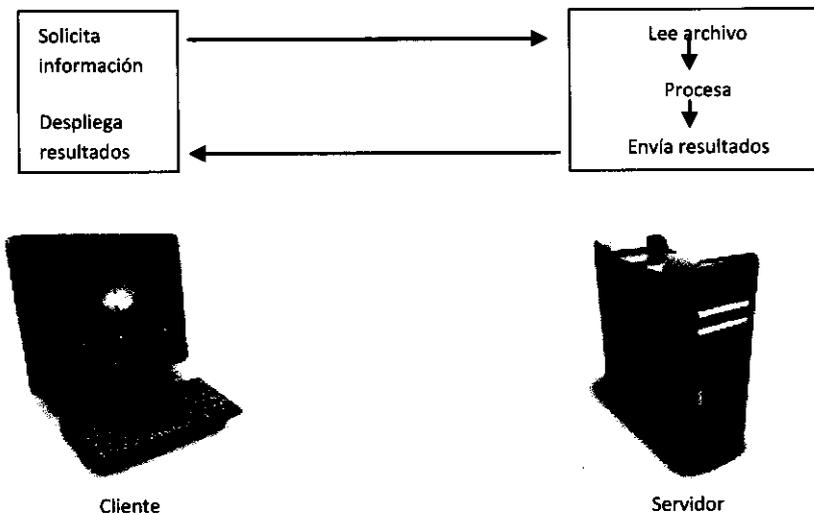


Fig. 22: Funcionamiento de la arquitectura Cliente / Servidor

Por último vamos a concretar cuáles son las herramientas utilizadas en el desarrollo de cada parte de estas aplicaciones concretamente en el entorno web:

- **Cliente:** la parte del cliente se desarrolla con código HTML, y con la opción de algunos lenguajes de programación un tanto limitados y específicos como JavaScript, Applets de Java y Flash.
- **Servidor:** La parte del servidor está formada por scripts o códigos de programa realizados en lenguajes más potentes como PHP, ASP, JSP, etc... Estos scripts son ejecutados en el servidor y la salida que producen, es enviada a la máquina del cliente.

6.2.3 El servidor web

El servidor web a elegir puede depender de factores externos. Debido a que las librerías de Facebook están hechas en php, esto nos delimitaba a dos tipos de servidor: **Apache** y el **Internet Information Server IIS**. Debido a que principal idea era la de publicar la web en un servidor web gratuito, (la mayoría de servidores web gratuitos que admiten la programación php, tienen instalado Apache por defecto), la decisión final fue usar Apache y descartar el IIS.

6.2.4 Navegadores (browsers)

Como ya se ha comentado anteriormente, el navegador será el encargado de ejecutar la aplicación del lado del cliente. Hay muchos y muy diversos navegadores en el mercado (para todos los sistemas operativos) y todos contemplan pequeñas diferencias entre sí, lo cual supone un problema a la hora de mostrar las aplicaciones en la parte del cliente, puesto que la interfaz no será exactamente igual si un navegante usa uno u otro browser. O simplemente versiones diferentes del mismo.

Llegados a este punto se debe elegir un navegador para el cual optimizar la aplicación. Con esto no estamos descartando a los usuarios que utilicen otro navegador, sino que estamos tomando uno como referencia principal, siempre sin dejar de pensar en el resto de personas, intentando cubrir el mayor número de navegantes de una manera satisfactoria.

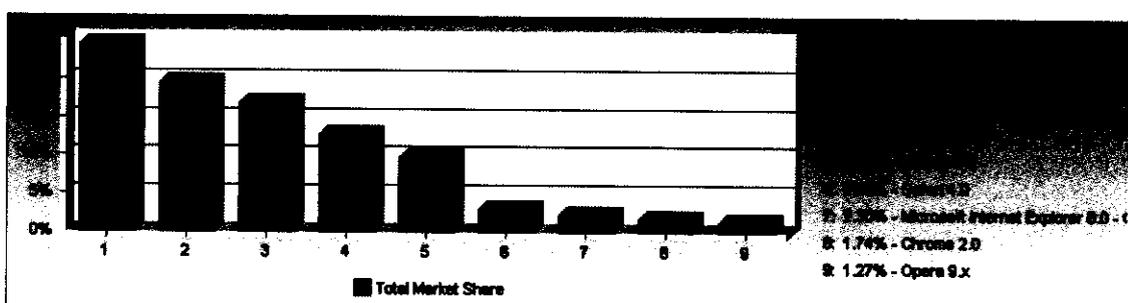


Fig. 23: Uso de navegadores (browsers) en Septiembre 2006

Si observamos las estadísticas de Market Share⁴, podemos comprobar que el navegador más utilizado es el Microsoft Internet Explorer 6.0, pero si analizamos con más profundidad sus inmediatos perseguidores, podemos comprobar que, desde su presentación a principios del 2009, Internet Explorer 8.0 ha tenido un crecimiento exponencial, robándosele de sus hermanos menores. Es de creer que esta tendencia continuara en los siguientes meses por lo que será esta versión la elegida como referencia. Aunque se han probado diversas versiones de Firefox, Safari y Chrome (los inmediatos perseguidores del gigante de Microsoft) obteniendo una visualización más

⁴ web que se dedica, entre otros, al estudio de usabilidad de navegadores:

Fuente: <http://marketshare.hitslink.com>

que satisfactoria, con lo que nos aseguramos llegar a un mayor número de usuarios de Internet.

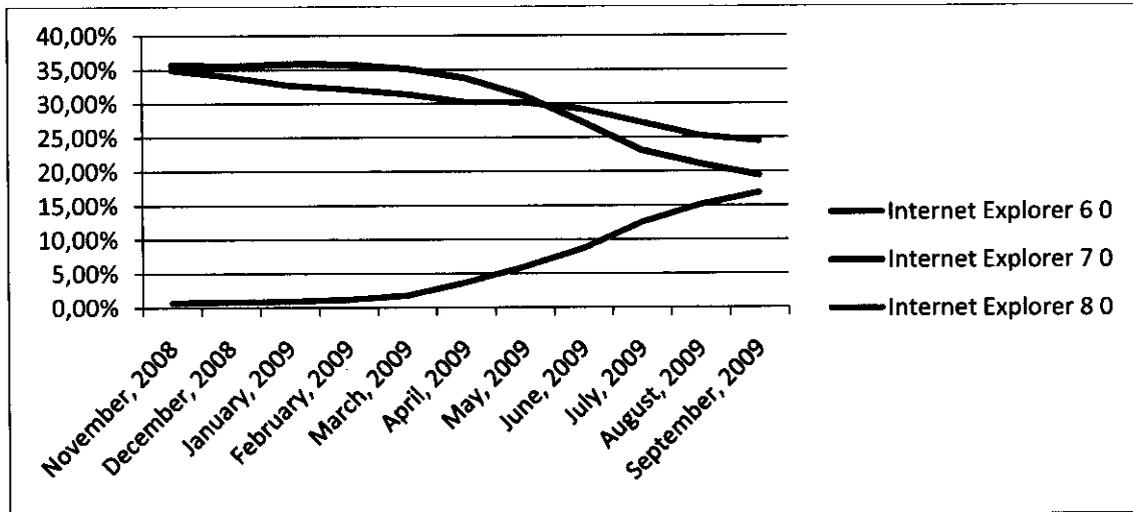


Fig. 24: Histórico de uso del navegador Internet Explorer

Por otro lado también se tendrá que tener en cuenta la resolución de las pantallas de los navegantes. El estándar actual es de 1024x768 píxels, por tanto deberemos procurar que dicha resolución de las páginas se visualicen correctamente, aunque también teniendo en cuenta cada vez más que los Internautas poseen equipos con más resolución y la página debe resultar también atractiva para estos últimos.

6.2.5 El SGBD: Sistema Gestor de Bases de Datos

Al igual que en el caso del servidor web, la decisión de usar un Sistema Gestor de Bases de Datos u otro se vio delimitado por la ubicación de la web. La mayoría de servidores webs gratuitos ofrecen como SGBD MySQL.

También hay que decir que existen estudios en los que se demuestra que MySQL es uno de los SGBD gratuitos más eficientes⁵ a la hora de realizar consultas o modificaciones en las Bases de datos. También es conocida sobre todo por los escasos recursos que consumen y por su facilidad de instalación y configuración.

6.3 Lenguajes utilizados. Explicación y ejemplos.

En este apartado se van a exponer los lenguajes utilizados para la implementación de la aplicación junto con ejemplos ilustrativos de su uso.

⁵ En estos mismos estudios, se ha demostrado que, solo en concretas operaciones, el SGBD PostGres mejora en rendimiento de MySQL. En internet se puede encontrar muchas comparativas al respecto. Fuente: http://www.wikivs.com/wiki/MySQL_vs_PostgreSQL

6.3.1 HTML

HTML son las siglas de HyperText Markup Language y, como ya hemos comentado con anterioridad, no es un lenguaje de programación propiamente dicho, sino que se trata de un marcador usado para dar formato a los textos definiendo su estructura y presentación, aunque el cómo se muestre una página hecha en HTML depende en parte del navegador empleado para su visualización.

HTML se define mediante lo que se ha dado en llamar tags o etiquetas que se encargan de los diferentes aspectos de la presentación final. El contenido debe ir introducido entre los tags que se consideren oportunos.

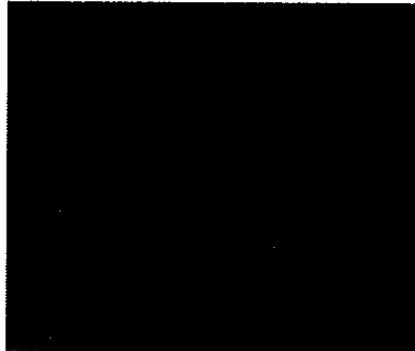


Fig. 25: Ejemplo sencillo de código HTML

Los Tags o etiquetas que controlan el comportamiento y la apariencia del documento son fragmentos de texto que van entre los símbolos <>. Además, algunas de estas etiquetas ofrecen la posibilidad de introducir atributos o parámetros específicos. Como podemos ver en la figura anterior, ejemplos de tags serían <html>, <body>, etc.

Para la edición de HTML se requiere únicamente de un editor corriente de texto ASCII, como el "Bloc de notas" de Windows, aunque con el tiempo han ido surgiendo editores especializados como Front Page o Dreamweaver que incorporan herramientas para facilitar la creación de páginas web. La única limitación que se imponen a los documentos creados es que tengan la extensión .htm o .html que los navegadores los puedan reconocer.

Una de las principales herramientas del HTML es la posibilidad de la inclusión de formularios dentro de la página. Estos formularios sirven para permitir la recogida de datos en el cliente para su posterior envío hacia el servidor, lo cual es indispensable para aplicaciones que requieren de entradas.

HTML es un "lenguaje" sencillo y muy limitado; es por ello que usualmente se usa acompañado de otras herramientas que le dan dinamismo y funcionalidad a las páginas como JavaScript que veremos en el siguiente apartado.

6.3.2 JavaScript

JavaScript es un lenguaje interpretado (desarrollado por la compañía Netscape Communications Corporation, aunque existe una implementación de Microsoft llamada JScript) pensado para ser incluido dentro de documentos HTML que nos permite añadir un poco de interactividad a las páginas web. Esta interactividad se hace especialmente necesaria en el tratamiento de formularios, por ejemplo en la validación de éstos. Si validamos los formularios mediante JavaScript, la validación estará siendo realizada en el cliente, y por tanto estaremos ahorrando trabajo al servidor gracias a lo cual irá más desahogado (tanto él como la red) y atenderá otras peticiones con más facilidad.

Características principales:

- No tiene comprobación de tipos de ninguna clase.
- Tiene expresiones, funciones y sentencias de control del flujo de ejecución.
- Tiene soporte para eventos.
- Es interpretado por el navegador cuando se lee la página. Esto implica que el navegador debe tener soporte para JavaScript para poderlo utilizar (Actualmente todos tienen).
- Es orientado a objetos, pero no tiene los conceptos de clase ni de herencia. Lo único que hay es unos objetos predefinidos de los cuales podemos obtener información y modificar propiedades.

A continuación, mostramos un ejemplo sencillo de código de JavaScript junto con código HTML (El código JavaScript se muestra en color azul.)

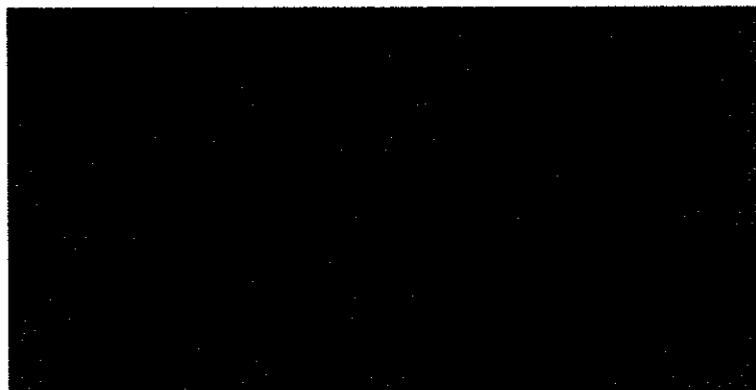


Fig. 26: Ejemplo sencillo de código JavaScript

6.3.3 PHP

PHP es el acrónimo de Hypertext Preprocessor. Es un lenguaje de alto nivel embebido en código HTML, que antes de ser enviado al cliente por el servidor web es sustituido por la salida que resulta de ser interpretado. Gracias a esta explicación podemos observar que PHP no es un lenguaje que se compile, sino que se interpreta.

Sus características principales son las siguientes:

- Gran similitud (sobretudo sintáctica) a otros lenguajes de programación de alto nivel como C, Perl, Java...
- Multiplataforma, pues hay versiones de PHP para los sistemas operativos más importantes.
- Soporta múltiples SGBD's (Sistemas Gestores de Bases de Datos).
- Soporta múltiples protocolos como IMAP, POP3, SNMP, TNP, FTP, HTML entre muchos otros.
- Consume pocos recursos, es decir, se puede utilizar prácticamente en cualquier PC.
- Está muy extendido y, por tanto, es sencillo encontrar infinidad de información sobre instalación, configuración, desarrollo y prácticamente resolver cualquier tipo de duda.

Una de las principales ventajas del PHP es que es embebido. Ello significa que se puede insertar código PHP en un documento HTML (con extensión .htm o .html), que será interpretado en el momento en que el cliente hace la petición de visualización de una página al servidor. Análogamente también se puede insertar HTML en un documento .php, y este HTML será enviado cada vez que se haga una petición del fichero .php donde se aloja.

Tendríamos así una parte del HTML a la que podríamos llamar "fija" y otra que se genera a partir de la interpretación del script php.

Por otro lado también se pueden tener documentos íntegros en PHP (con extensión .php); la idea sería que las funciones implementadas generarían y devolverían todo el HTML que posteriormente se envía al cliente.

A continuación, se muestra un ejemplo de código php embebido con HTML (el código php se muestra en azul)



Fig. 27: Ejemplo sencillo de código PHP

La elección de la utilización de este lenguaje se dio a que, las librerías facilitadas por Facebook para el uso de sus funcionalidades, están hechas en php.

6.3.4 SQL

Son las siglas de “Structured Query Language” y es un lenguaje de consultas a Bases de Datos que fue creado como estándar para asegurar que las consultas fueran independientes a la Base de Datos con la que se trabajase. Es decir, en el caso de este proyecto se trabaja con MySQL, pero las mismas consultas se podrían usar si se cambiase algún día el SGBD a Oracle, Informix o cualquier otro.

El código SQL se ha insertado en el código PHP, mediante instrucciones proporcionadas por el propio lenguaje (PHP).

En nuestro caso, hemos creado un controlador de datos que se encarga de consultar los datos que se soliciten a la BD y retorna el resultado.

Como ejemplo vamos a mostrar la función en PHP que consulta los datos de una respuesta en concreto, junto con la función del controlador de datos, que consulta los datos a la Base de Datos:

Esta función, ejecuta la consulta a la Base de datos y retorna el resultado

Vamos recorriendo fila por fila, todos los resultados devueltos por la Base de datos

Realizamos la consulta a la Base de datos y retornamos el resultado

Indicamos al SGBD que seleccione la Base de datos "elpreguntometro" para realizar las consultas.

Abrimos una conexión con el SGBD (Systema Gestor de Base de datos) para realizar la consulta a la Base de Datos.

6.3.5 Plataforma Facebook

Después de la creación de Facebook en el 2004, la empresa creó una plataforma para que terceras personas, pudiesen desarrollar aplicaciones que pudiesen utilizar la extensa información que contiene la Base de Datos de Facebook.

Actualmente la plataforma de Facebook ofrece 5 tipos de lenguajes propios para poder interactuar con Facebook y su Base de Datos. Todos ellos, están explicados en la wiki que ofrece Facebook a todas aquellas personas que quieran desarrollar aplicaciones con su plataforma. El acceso a esta wiki es de manera totalmente gratuita (<http://wiki.developers.facebook.com>).

Aunque si es cierto que todos los lenguajes que ofrece Facebook están explicados en su wiki, si es cierto que, en mi opinión, no están totalmente detallados, por lo que durante el desarrollo de la aplicación, en muchas ocasiones se ha acabado recurriendo a fuentes externas para acabar de comprender como funciona una funcionalidad en concreto. Es por eso, que a continuación se detallaran los primeros pasos básicos para poder conectar una aplicación web con la plataforma Facebook.

El primer paso que hay que realizar es registrar la aplicación en la Base de datos de Facebook, por lo que hay que tener una cuenta de Facebook.

Para crear una cuenta en Facebook, ha de entrar en la web de Facebook (www.facebook.es) y rellenar el formulario de inscripción. Una vez creada la cuenta, para registrar la aplicación, ha de visualizar su perfil, ir al final de la página y hacer clic encima de Desarrollador (El link se encuentra al fondo de la web en la parte derecha en letras pequeñas).

A continuación haga clic en "Start Building for your site".

facebook

Build and Grow with Facebook

Bring identity and connections to your site or application.

[Start Building for your site](#)

What would you like to do?

- ▶ Get simple widgets for your site or blog
- ▶ Integrate Facebook into your site using Connect
- ▶ Build Facebook-enabled iPhone apps
- ▶ Create an application that runs inside Facebook

How can we help you grow?

- ▶ Promote your app on Facebook
- ▶ Get your app verified

Where is the latest information about Platform?

- ▶ Developer Roadmap
- ▶ Platform Live Status
- ▶ Principles and Policies

What's New

October 22

- ▶ Simplifying the Stream

October 13

- ▶ Making it Easier to Build on Facebook

September 28

- ▶ Facebook Connect Wizard

September 29

- ▶ Translations for Facebook Connect

Blog

Postcards from October Developer Events

Ayer a las 21:55
Publicado por Julia Lam

We hosted six Facebook Developer Garages this month around the globe. In Istanbul, Turkey, over 250 developers were joined by Facebook's Justin Osofsky, who gave an overview of the benefits of... [Read More](#) ▶

Events

FACEBOOK DEVELOPER GARAGE LONDON

Facebook Developer Garage London
November 11th, 2009
6:00 PM - 9:00 PM

San Microsystems
45 King William Street, EC4R 9AN
London, United Kingdom

Facebook © 2009 [About](#) [Terms of Service](#) [Privacy Policy](#)

Fig. 28: Como dar de alta una aplicación en Facebook, paso 0.

A continuación ha de introducir los datos de la aplicación a crear. Ha de poner un nombre de aplicación y la url donde estará alojada la web (Facebook no ofrece el servicio de hosting, solo la conexión con sus Bases de datos, por lo que tendrá que tener alojada la aplicación en un servidor externo a Facebook)

facebook

Paso 1 **Paso 2: Subir archivo** **Paso 3: Añadir marca social**

Información básica sobre su sitio

Nombre de la página web:

Dirección web de la página:

Idioma:

Facebook © 2009 [About](#) [Terms of Service](#) [Privacy Policy](#)

Fig. 29: Como dar de alta una aplicación en Facebook, paso 1.

El siguiente paso será descargar un archivo en formato html necesario para que la aplicación se conecte con Facebook. Es importante que este archivo se guarde en el mismo directorio en donde se encuentre la pagina en la que aparecerá el botón de conectar a Facebook.

Este archivo lo facilita Facebook. También puede copiar el código que se muestra a continuación.

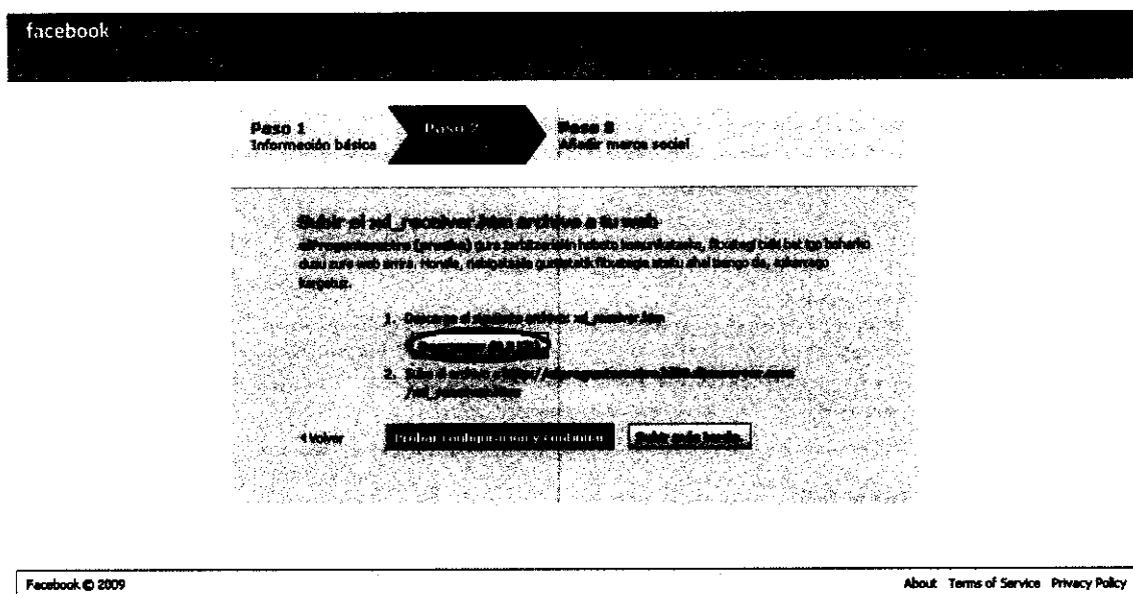


Fig. 30: Como dar de alta una aplicación en Facebook, paso 2.



Tabla 13: Código del archivo xd_receiver.htm



Tabla 14: Código a añadir en tu web para conectar tu aplicación con Facebook

Además del código anteriormente mostrado, se ha de añadir el siguiente código XFBML⁶ para que muestre en la web un botón con el texto “Conectar con Facebook”.



Tabla 15: Código que muestra el botón de Conectar con Facebook

También tendremos de modificar la cabecera de la web con el siguiente código para que html sepa resolver el código XFBML:



Tabla 16: Código que se ha de añadir al comienzo del archivo de tu web

Y con estas modificaciones su aplicación ya podrá conectarse a Facebook.

Si desea que, después de que un usuario se halla identificado en Facebook, la aplicación realice alguna acción (como por ejemplo refrescar la ventana) puede modificar la línea del código XFBML de la siguiente manera:



Tabla 17: Modificación para que una vez logado el usuario, refresque la web.

A continuación, mostramos un ejemplo simple de una web en la que se han realizado las modificaciones anteriores (que son las que se encuentran resaltadas en rojo):



⁶ XFBML es un lenguaje creado por Facebook que se usa como extensión del propio lenguaje HTML ampliando sus posibilidades. Una de sus grandes cualidades es que, una vez identificado el usuario en Facebook, este lenguaje permite mostrar información del usuario conectado con extrema facilidad (con una simple orden, como si fueran etiquetas HTML). Puede obtener más información sobre este lenguaje en la wiki de desarrolladores (<http://wiki.developers.facebook.com/index.php/XFBML>)

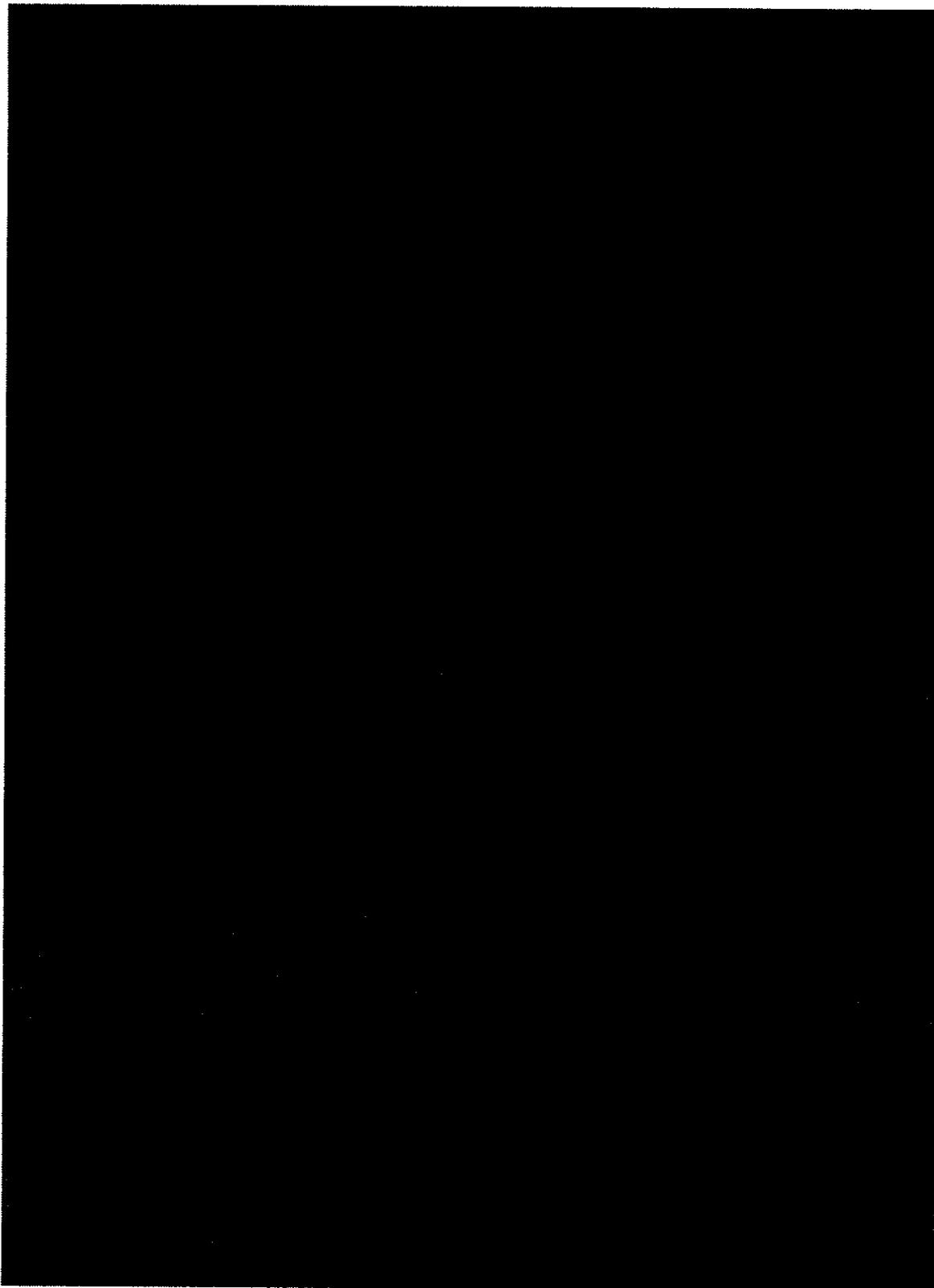


Tabla 18: Ejemplo completo para conectar tu aplicación a Facebook.

Una vez conectado, podemos utilizar cualquiera de los lenguajes que ofrece Facebook para interactuar con su Base de Datos. Personalmente recomiendo usar el lenguaje API si deseamos guardarnos alguna información del usuario que se ha conectado a nuestra

aplicación ya que, según mi opinión, es el que mejor está detallado en la wiki de Facebook (este lenguaje funciona conjuntamente con php).

Por el contrario, si lo que deseamos es únicamente mostrar información por pantalla, del usuario que se ha conectado a nuestra aplicación, XFBML es el más simple y potente de utilizar y no requiere del uso de php (con HTML y JavaScript es suficiente).

Anteriormente ya hemos mostrado un ejemplo de cómo utilizar el código XFBML. A continuación mostraremos un ejemplo práctico de cómo usar el lenguaje API. Antes de continuar, es imprescindible aclarar que API no se puede considerar un lenguaje en sí mismo sino un conjunto de librerías escritas en código php, por lo que, para utilizarlo, primero es necesario descargar las librerías que ofrece Facebook y que podrá encontrar en la wiki en el apartado "*Client Libraries*" o en la url <http://svn.facebook.com/svnroot/platform/clients/packages/facebook-platform.tar.gz>

Una vez descargado, descomprima el directorio php del archivo tar descargado y cópielo en el directorio donde tiene hospedado su web. Después solo es necesario incluirlo en el archivo en el que se desea ejecutar el código API.

A continuación se muestra un código de ejemplo para obtener el nombre del usuario actual (Se ha de substituir la cadena "API_KEY" por el identificador que le haya facilitado anteriormente por Facebook al registrar la aplicación y "SECRET" por la contraseña también facilitada anteriormente por Facebook):



Tabla 19: Ejemplo de función que obtiene el nombre de usuario del usuario logado actualmente.

Del mismo modo que, en el ejemplo anterior se obtiene el nombre del usuario logado, **se puede obtener cualquier dato del usuario identificado** (a excepción de la dirección de correo electrónico) **siempre y cuando el mismo usuario no haya indicado a Facebook que el dato que solicitamos en concreto no sea compartido.**

6.4 Pruebas

Durante el desarrollo de la aplicación web, se realizó una aplicación web de pruebas en la que se pudo experimentar con las funcionalidades que ofrece Facebook en un entorno aislado y seguro.

Con esta aplicación, pudimos probar muchas de las funcionalidades que ofrece Facebook como la posibilidad de poder escribir un mensaje en el wall (muro) del usuario, o el poder utilizar los estilos de Facebook para la propia web.



Aplicación de testeo

Inicie sesión para empezar a utilizar la aplicación



Si no tiene una cuenta, puede darse de alta aquí.

Fig. 32: Imagen de la aplicación web de testeo creada para poder probar muchas de las funcionalidades que ofrece Facebook.

6.5 Diseño Gráfico

Si bien es verdad que desde un principio se abordó el tema con el cliente desde la perspectiva de que un informático no es equivalente a un diseñador gráfico, y que para desempeñar este trabajo existen empresas especializadas, el diseño gráfico de la aplicación se ha basado en los requerimientos establecidos en un inicio acerca de la facilidad y sencillez que debían tener los menús de navegación.

La estructura de la web fue lo primero que se abordó en una reunión concertada en los primeros meses, dado que es fundamental saber cómo se quiere distribuir la información. Una vez se esbozó una estructura de la web, se procedió a realizar algunos bocetos de lo que serían las vistas de cada funcionalidad. Finalmente se decidió presentar un menú de navegación en la parte superior y un menú lateral en el que se muestran las opciones de la aplicación. Estos dos menús siempre estarían visibles para facilitar la navegación por la web. A partir de aquí se siguió trabajando en función a esta estructura y una vez concluida la implementación de la aplicación se concertaron varias reuniones para “afinar” conceptos relacionados con los colores y formas de los textos y botones, así como para pensar en mejoras y aplicaciones de la web.

Aun y así, sería necesario realizar un estudio de diseño y usabilidad para poder finalizar el proyecto.

7. Acerca de Facebook

La plataforma Facebook, como plataforma de desarrollo y red social, permite a usuarios y empresas almacenar datos y obtener información de otros usuarios. Uno de los objetivos de este proyecto final de carrera, además de desarrollar una aplicación, era estudiar el impacto de una aplicación para una empresa, estudiar la forma de obtener información disponible de un usuario desde una aplicación y ver posibles usos de esta información para obtener beneficios económicos directos o indirectos.

7.1 ¿Exactamente qué datos podemos obtener de los usuarios de Facebook?

Antes de detallar exactamente qué datos podemos obtener de un usuario de Facebook, hemos de diferenciar el estado que tendrán al estar utilizando la aplicación. Este estado puede ser:

- Un **usuario** de Facebook **no identificado** en la aplicación: A este estado nos referimos a un usuario que tiene una cuenta de Facebook pero que está navegando por la aplicación sin haberse identificado en ella con un usuario de Facebook, o que simplemente no está conectado a Facebook o incluso a internet pero tiene creada una cuenta en Facebook.
- Un **usuario** de Facebook **identificado** en la aplicación: A este estado nos referimos a un usuario que ha clicado encima del botón Conectar con Facebook y ha introducido su usuario y contraseña de Facebook para acceder a la aplicación.

Actualmente, de un usuario no identificado solo podremos obtener la información disponible en las búsquedas públicas (tu nombre, redes, foto de perfil y lista de amigos).

Sin embargo, de un usuario identificado, además de la información mencionada anteriormente, podemos obtener los siguientes datos:

- ✓ Información básica (fecha de nacimiento, sexo, ciudad, ideología política y cuando actualizaste tu perfil por última vez).
- ✓ Información personal (actividades, intereses, etc.)
- ✓ Ciudad actual.
- ✓ Historial académico.
- ✓ Historial laboral.
- ✓ Estado del perfil.
- ✓ El muro del usuario.
- ✓ Las notas del usuario.
- ✓ Los grupos a los que pertenezco.
- ✓ Eventos a los que ha sido invitado el usuario.
- ✓ Fotos hechas por el usuario.
- ✓ Situación sentimental.
- ✓ El tipo de relación que está buscando el usuario.
- ✓ Que sexos le interesan al usuario.
- ✓ Con quien mantiene una relación.
- ✓ Creencias religiosas
- ✓ El sitio web que pueda tener reflejado el usuario.

7.2 ¿Por qué puedo acceder a los datos privados de un usuario de Facebook?

Una de las peculiaridades por las que se conoce Facebook es por tener una licencia de uso un tanto peculiar. Si le echamos un vistazo detenidamente podremos leer en las condiciones de uso el siguiente párrafo (versión del 28 de Agosto del 2009)



Fig. 33: Parte de la licencia de condiciones de uso de Facebook

Resumiendo el texto anterior, entendemos que estamos autorizando a Facebook a utilizar nuestros datos del modo que a ellos les convenga, siempre y cuando no vulneren la licencia de la propiedad intelectual.

Es de costumbre, que la mayoría de los usuarios no se lean las condiciones de uso de una aplicación, por ello, la gran mayoría de los usuarios que utilizan Facebook, desconocen que ciertas aplicaciones, pueden acceder a sus datos personales.

Todo y con ello, cada vez que un usuario intenta utilizar una aplicación de Facebook, esta le advierte de que esta aplicación podría acceder a sus datos personales.

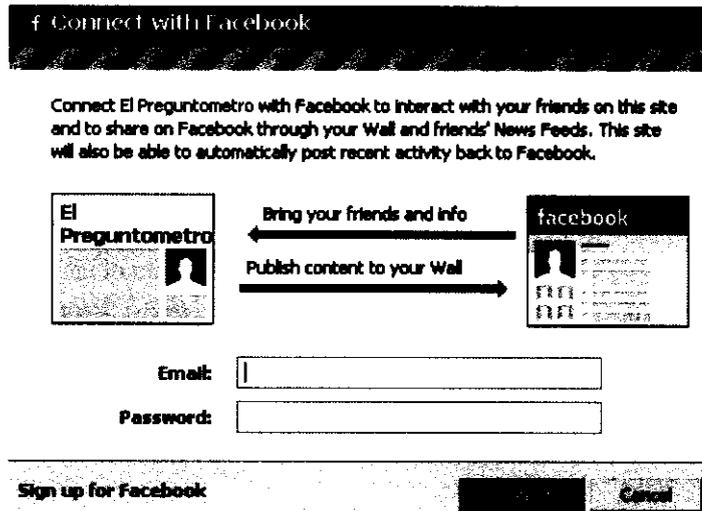


Fig. 34: Aviso de Facebook de que la aplicación que va a utilizar, podría acceder a sus datos personales.

7.3 ¿Qué puedo hacer si quiero utilizar una aplicación de Facebook pero no deseo que esta pueda acceder a ciertos datos privados?

Actualmente, Facebook permite que el usuario restrinja la información que desea que ciertas aplicaciones pueda acceder a ella. Se puede restringir el acceso por aplicaciones (que sólo ciertas aplicaciones puedan acceder a cierta información) o se puede hacer una restricción total (para que ninguna aplicación pueda acceder a cierta información). Ambas opciones se pueden realizar desde la configuración de la privacidad de la cuenta, en el apartado Aplicaciones. Su configuración es muy fácil e intuitiva.

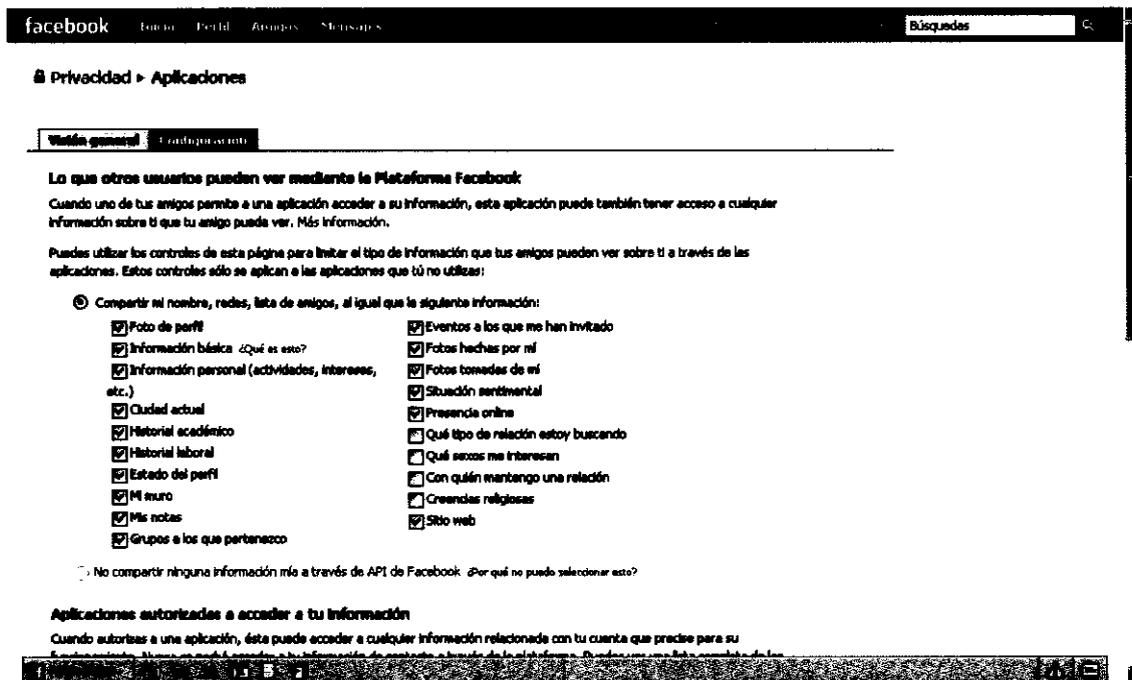


Fig. 35: Imagen de la pantalla de configuración de la privacidad de una cuenta de Facebook.

8. Desarrollo en Facebook ¿es un negocio rentable?

Uno de los objetivos de este proyecto es evaluar las posibilidades de negocio que la plataforma Facebook puede ofrecer, y es lo detallaremos en este capítulo.

Es lógico que un canal de comunicación que llega a más de 300 millones de usuarios registrados e incluye posibilidades infinitas de segmentación y personalización, sea un blanco interesante y seductor para todo tipo de empresa.

Ya hace tiempo que muchas personas han empezado a utilizar Facebook como una útil herramienta para darse a conocer profesionalmente o para formar una red de contactos que pueda generar nuevas oportunidades de negocio por ejemplo.

Facebook ha sido muy valioso en este sentido ya que a partir de esta red social uno puede contactar otras personas para enviar un CV, difundir un blog personal o simplemente ubicar potenciales clientes segmentando por el perfil de cada usuario.

Pero más allá de como cada individuo explote la herramienta de forma personal, ya se ha vuelto una tendencia creciente la aparición de miles de empresas en Facebook.

El camino más sencillo y directo que encontraron las empresas para tener presencia en Facebook fue contar con un perfil o página propia desde donde invitar a gente para que se haga Fan de la empresa o marca. Así podemos ver como Coca-Cola tiene más de 4 millones fans, Adidas Originals más de 2 millones, y Apple Students supera los 1.400.000 fans.⁷

Pero muchas han optado además, por adaptar sus webs para que interactúen con Facebook. Este es el caso de Radio y Televisión Española, que permite compartir noticias que te interesan añadiéndolas en tu perfil de Facebook para que tus amigos puedan verlas.

⁷ Numero de fans comprobado a fecha 4 de enero del 2010.

8.1 Reglas a seguir para realizar una aplicación con Facebook

Facebook se apoya en los siguientes principios básicos:



Tabla 20: Traducción al castellano de parte de la Declaración de Derechos y Responsabilidades de Facebook (del 28 de agosto del 2009). El artículo completo lo puede encontrar en la web <http://www.facebook.com/terms.php?ref=pf>

Esta es la última versión publicada por Facebook. Muchos han sido los cambios que se han producido en esta declaración de derechos y responsabilidades, debido a los muchos ataques que ha tenido esta plataforma desde su nacimiento (robos de información para vender a SPAMERS, virus, amenazas a usuarios,...). Debido a ello, es posible que muchas empresas no encuentren utilidad a esta plataforma. Nada más lejos de la realidad.

8.2 ¿Realmente es útil la presencia en Facebook para una empresa?

La principal ventaja de estar dentro de Facebook es la posibilidad de generar conversación entre las empresas y los usuarios. O sea, el proceso comunicacional deja de ser un monólogo (como en TV y Gráfica por ejemplo) y la empresa sabe que todo lo que diga o proponga a través de Facebook puede ser contestado, aprobado o descalificado por el usuario de manera rápida y sencilla.

Es por eso que la empresa se asegura que el mensaje que se logre va a tener mayor credibilidad que en cualquier otro medio, ya que la empresa se sitúa a la misma altura que el usuario, e invita a participar y proponer, creando así un vínculo importante cliente - empresa.

Además, Facebook ofrece las siguientes ventajas:

- ✓ **Audiencia:** Facebook anuncia que ya tiene cerca de **400.000 usuarios españoles** entre los actuales 300 millones activos en todo el mundo. Pero esta cifra va a crecer mucho más porque Facebook **ha conseguido** esos 400.000 usuarios **sin disponer de una versión en castellano**. Con la recién traducida versión, se espera que la cifra de usuarios aumente y también que el tiempo de uso de los usuarios crezca.
- ✓ **Demografía:** Parece que las redes sociales siguen siendo usadas por los más jóvenes, pero lo cierto es que se está produciendo un “envejecimiento” demográfico con crecimientos superiores en el rango de 25-34 y por encima de los 34 años. El nivel de ingresos es medio-alto en general y sin entrar en más detalles parece un gran segmento interesante para muchos productos y servicios.
- ✓ **Conversación:** Las marcas, las empresas que participan en Facebook están obligadas a conversar, a utilizar como instrumento la conversación y la relación se produce con usuarios. En este entorno cerrado las reglas las marcan los usuarios, las personas, que no están dispuestos a escuchar el canto tradicional publicitario. Frescura y relaciones honestas.
- ✓ **Mensaje y utilidad:** Facebook provee de las herramientas necesarias para que las empresas puedan crear fácilmente un entorno comunicativo donde interactuar con sus usuarios y medir sus resultados. No existe el freno operativo de tener que construir un foro web, por ejemplo.

8.3 ¿Pero realmente es seguro?

Muchas han sido las empresas que han adaptado sus webs a Facebook para conseguir más publicidad, pero muchos usuarios aun se preguntan si es seguro utilizar dichas aplicaciones.

Para solucionar este problema, Facebook ofrece la posibilidad de certificar que una aplicación es segura. Solo es necesario registrarse en la aplicación de verificación de Facebook y pagar la correspondiente cuota anual de 375 dólares o de 175 si quien realiza la aplicación es un estudiante o una organización sin ánimo de lucro.

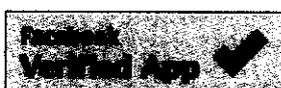


Fig. 36: Logo que aparece en una aplicación que ha sido verificada por Facebook

Muchas han sido las quejas recibidas por los desarrolladores de este sistema de validación de aplicaciones, pero estos también declaran que es lo suficientemente bueno como para que lo consideren un dinero bien invertido porque aumenta considerablemente la visibilidad y utilización de las aplicaciones certificadas. Así, por ejemplo, algunos desarrolladores, como el de App Broker, aseguran que, tras validar su aplicación con Facebook Application Verification, han visto incrementar sus usuarios hasta en un 1.200 por cien.

8.4 Además de la web, otros beneficios de desarrollar en Facebook

Además de la plataforma de desarrollo web descrita en este proyecto, Facebook ofrece la posibilidad de desarrollar aplicaciones en otros entornos como pueden ser aplicaciones para ordenadores personales, sistemas de entretenimiento, o dispositivos móviles.

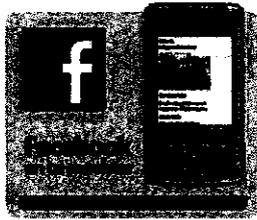


Fig. 37: Movistar ofrece en casi todos sus terminales, la posibilidad de conectar con Facebook y utilizar sus aplicaciones.

Muchas han sido las empresas de hardware que ha querido actualizar sus productos para que funcionen con Facebook. Un ejemplo claro está en los miles de terminales móviles que son compatibles con Facebook. Así permiten al usuario utilizar Facebook y sus aplicaciones en cualquier lugar. También Sony y Microsoft han aportado su grano de arena haciendo que sus consolas (Play Station 3 y Xbox 360) permitan ejecutar aplicaciones Facebook.

Con ello conseguimos un perfecto estado WIN-WIN, en donde empresas de servicios y producción de hardware ganan haciendo que unos puedan ver las aplicaciones que se crean en Facebook en cualquier lugar y otros puedan vender más porque su producto es compatible con la red social que más clientes potenciales tiene.



Fig. 38: Xbox ofrece la posibilidad de visitar varias redes sociales como Twitter o Facebook.

9. Conclusiones

9.1 Objetivos alcanzados

Una vez el proyecto ha concluido es el momento de comparar los objetivos iniciales con los alcanzados.

- Realización de un estudio de las funcionalidades que ofrece la plataforma de desarrollo de Facebook.
 - ✓ Si bien es cierto que actualmente Facebook ofrece multitud de funcionalidades (y a cada poco añaden nuevas funcionalidades), la gran parte se encuentran resumidas en el apartado 7.1 de esta memoria.

- Evaluar las posibilidades de negocio que esta plataforma puede ofrecer.
 - ✓ Después de buscar por internet opiniones al respecto, he podido realizar un resumen de los beneficios que aporta desarrollar en esta plataforma, y que podrá encontrar en la sección 8 de esta memoria. Cada vez se escucha mas por los medios de comunicación (en la publicidad) la frase *“sin una cuenta de Facebook no eres nadie”*. Las empresas empiezan a ser conscientes del gran impacto que tienen las redes sociales en la sociedad y están empezando a apostar fuerte en estos medios de comunicación, en especial en Facebook.

- Desarrollar una aplicación de demostración que utilice algunas de las funcionalidades de Facebook.
 - ✓ El preguntometro es un pequeño y sencillo ejemplo del potencial que puede tener esta plataforma. La aplicación web esta online y funcionando desde principios de noviembre y ya tiene a unos cuantos fans (amigos y familiares).
 - ✓ Una de las cosas que más nos ha sorprendido ha sido la facilidad con la que se puede obtener visitas, gracias a las propias recomendaciones que los propios usuarios pueden hacer de una aplicación de Facebook. Gracias a ello, conseguimos en un mes 107 visitas (que no está nada mal para una pequeña web que empieza de cero), entre ellas, la visita de un usuario desde Bélgica. (Estas mediciones se han hecho con la aplicación de google Analytics).

➤ En lo personal:

- El hecho de tener la confianza y el respaldo de mi director de proyecto para la realización de este proyecto, unido a la presión de la evaluación del mismo como PFC que es, me ha hecho, pienso, más responsable y cuidadoso en los pequeños detalles, cosa que antes era algo que echaba en falta y pese a saberlo e intentar cambiarlo nunca lo logré conseguir.
- Realmente creo que ha supuesto un cambio fundamental en mi manera de afrontar los problemas, las responsabilidades y los quehaceres cotidianos, por mi voluntad de realizar un muy buen trabajo, me han hecho madurar para bien como persona. Sólo espero que esto no cambie o si lo hace, sea todavía para mejor
- Ha sido una experiencia muy enriquecedora y positiva, sobretodo viéndola ahora que ha concluido.

9.2 Posibles ampliaciones y mejoras (el trabajo futuro)

Este punto hace referencia a las diferentes posibilidades que se pueden plantear en el futuro para mejorar y/o ampliar el preguntometro.

Durante la realización del proyecto, el cliente fue sugiriendo ampliaciones, de las cuales algunas se pudieron implementar antes de la finalización del proyecto pero que, por motivos de tiempo, no se pudieron documentar.

Posibles modificaciones:

- Que un usuario pueda introducir una pregunta/respuesta anónimamente. Solo los usuarios de tipo administrador podrán saber quien fue el usuario que introdujo la pregunta/respuesta en la aplicación (**Hecho**).
- Al introducir una pregunta/respuesta, el usuario pueda publicar en el wall de su cuenta un resumen de esta (**Hecho**).
- Poder suscribirse por RSS a un tema o a una pregunta para así no tener que estar constantemente visitando la web para ver si alguien ha realizado alguna modificación (**Hecho**).
- Poder realizar búsquedas más complejas.

Bibliografía

Material de asignaturas de la facultad:

1. "Enginyeria del software I: Especificación de sistemas orientados a objetos con notación UML". Ernest Teniente.
2. "Enginyeria del software II: Diseño de sistemas orientados a objetos con notación UML". José Antonio Raya
3. "Gestio de Sistemes Informatics: Apuntes y transparencias de Gestión funcional de S.I. y estimación de costes de un proyecto Informático". Marc Alier.
4. "Bases de dades: Apuntes". Carme Martin
5. "Seguretat en Sistemes Informàtics: Apuntes y transparencias sobre la Ley orgánica de protección de datos" Manuel Medina.

Material de Internet⁸

1. <http://php.net/index.php>
Web oficial de soporte al lenguaje php.
2. <http://www.javascriptkit.com/>
Completa página sobre JavaScripts, CSS y DHTML que consta de tutoriales y ejemplos de código abiertos.
3. <http://www.forosdelweb.com/>
Foro de amantes de la programación en donde se pueden exponer y encontrar miles de soluciones a problemas de programación.
4. <http://www.w3schools.com/>
Igual que la anterior pero más centrado en la programación web. De esta página también se pueden obtener estadísticas del uso de los navegadores más utilizados.
5. http://wiki.developers.facebook.com/index.php/Main_Page
Wiki oficial de Facebook donde podrás encontrar toda la información oficial de todas las funcionalidades que ofrece la plataforma Facebook.

⁸ Enlaces comprobados a fecha 4 de enero de 2010.

6. <http://www.faceblog.es/>
Bolg no oficial de Facebook donde se puede encontrar información y ejemplos de algunas de las funcionalidades más usadas en Facebook.
7. <http://www.ruhanirabin.com/integrate-social-networking-wordpress-using-facebook-connect-api-wordpress-plugin-tutoria/>
Web en la que resumen de una manera sencilla como adaptar tu web para que se conecte a Facebook.
8. <http://vimeo.com/2752138>
Podcasts oficiales de Facebook en donde se muestran paso a paso como utilizar algunas de las funcionalidades más usadas en la plataforma Facebook.
9. <http://es.wikipedia.org/wiki/LOPD>
Artículo de la wikipedia en donde resumen los derechos y obligaciones de la Ley Orgánica de Protección de Datos.
10. http://www.boe.es/aeboe/consultas/bases_datos/doc.php?coleccion=iberlex&id=1999/23750
Articulo del Boletin Oficial del Estado en donde se encuentra redactada los derechos y obligaciones que tiene la Ley de protección de datos.
11. [http://www.youtube.com/watch?v=BnZlct8 -MI&feature=player_embedded](http://www.youtube.com/watch?v=BnZlct8-MI&feature=player_embedded)
Podcast que explica que es un RSS y que beneficios tiene el utilizarlo. Podcast publicado por commoncraft y traducido por Ludo (Marc Alier).
12. <http://www.webnova.com.ar/articulo.php?recurso=364>
Web en la que explica de una manera sencilla como realizar un RSS dinamico.
13. <http://marketshare.hitslink.com/>
Web que se dedica, entre otros, al estudio de usabilidad de navegadores.
14. <http://www.rae.es>
Página de la Real Academia Española donde he podido solucionar algunas dudas ortográficas que me han surgido durante la realización de esta memoria.
15. <http://www.el-mundo.es>
Portal del periódico El Mundo. Ha sido de mucha utilidad algunos artículos relacionados con el uso de las redes sociales y Facebook.

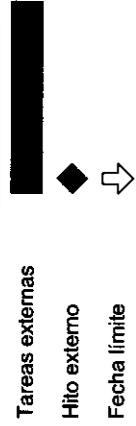
16. <http://www.google.com>

Y el último pero seguramente el más importante, GOOGLE. El que para muchos es el mejor y más eficiente buscador de toda la red. Gracias a él he podido encontrar en las anteriores webs y portales toda la información necesaria para la consecución de este proyecto.

ANEXO A: Diagrama de GANTT del proyecto realizado en Microsoft Project

En la página siguiente, se muestra el diagrama de Gantt realizado en el programa Microsoft Project. El camino crítico se muestra en color rojo.

Id	Nombre de tarea	Duración	Comienzo	15 jun '09							22 jun '09							29 jun '09							
				D	L	M	M	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J	V	S	D
1	Inicio del proyecto	0 días	lun 15/06/09																						
3	Reunión con el cliente para especificar el proyecto	1 día?	lun 15/06/09																						
4	Análisis previo y de requerimientos	9 días?	mar 16/06/09																						
5	Revisar las Condiciones de uso de Facebook	1 día	mar 16/06/09																						
6	Hacer un estudio del funcionamiento de las casas de Facebook	1 día	mié 17/06/09																						
7	Hacer un estudio del LOPD	1 día?	jue 18/06/09																						
8	Buscar un servidor donde hacer pruebas	1 día?	lun 22/06/09																						
9	Hacer un Gantt	2 días?	lun 22/06/09																						
10	Redactar el Análisis previo y los análisis de requerimientos en la memoria	5 días?	lun 22/06/09																						
11	Margen de maniobra	2 días	lun 29/06/09																						
12	Inscribir el proyecto	1 día?	mié 01/07/09																						
13	Margen de maniobra	1 día?	jue 02/07/09																						
14	Realización de la especificación (en la Memoria)	94 días?	vie 03/07/09																						
15	Realización del Modelo de Entidad Relación	2 días	vie 03/07/09																						
16	Margen de maniobra	1 día?	mar 07/07/09																						
17	Reunión con el cliente para aprobación del modelo anterior	1 día?	mié 08/07/09																						
18	Aplicar posibles cambios al modelo acordados en la reunión anterior	1 día?	jue 09/07/09																						
19	Realización del Diagrama de la capa dominio + datos	2 días	vie 10/07/09																						
20	Margen de maniobra	1 día?	mar 14/07/09																						
21	Reunión con el cliente para aprobación del modelo anterior	1 día?	mié 15/07/09																						
22	Aplicar posibles cambios al modelo acordados en la reunión anterior	1 día?	jue 16/07/09																						
23	Realización de un boceto de las vistas	2 días	vie 17/07/09																						
24	Margen de maniobra	1 día?	mar 21/07/09																						
25	Reunión con el cliente para aprobación de las vistas anteriores	1 día?	mié 22/07/09																						
26	Aplicar posibles cambios a las vistas acordadas en la reunión anterior	1 día?	jue 23/07/09																						
27	Realización del Diagrama de la capa de vistas	2 días	vie 24/07/09																						
28	Margen de maniobra	1 día?	mar 28/07/09																						
29	Reunión con el cliente para aprobación del modelo anterior	1 día?	mié 29/07/09																						
30	Aplicar posibles cambios al modelo acordados en la reunión anterior	1 día?	jue 30/07/09																						
31	Realización del Diagrama de Casos de Uso	1 día?	mar 01/09/09																						

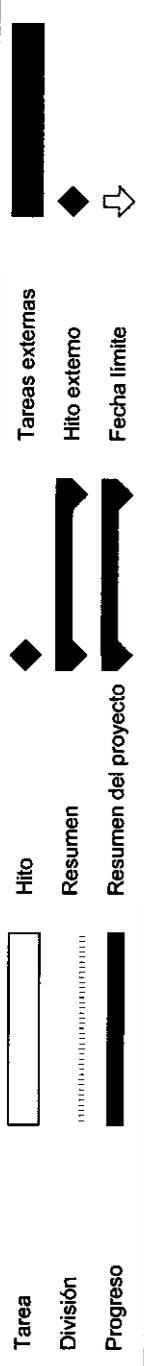
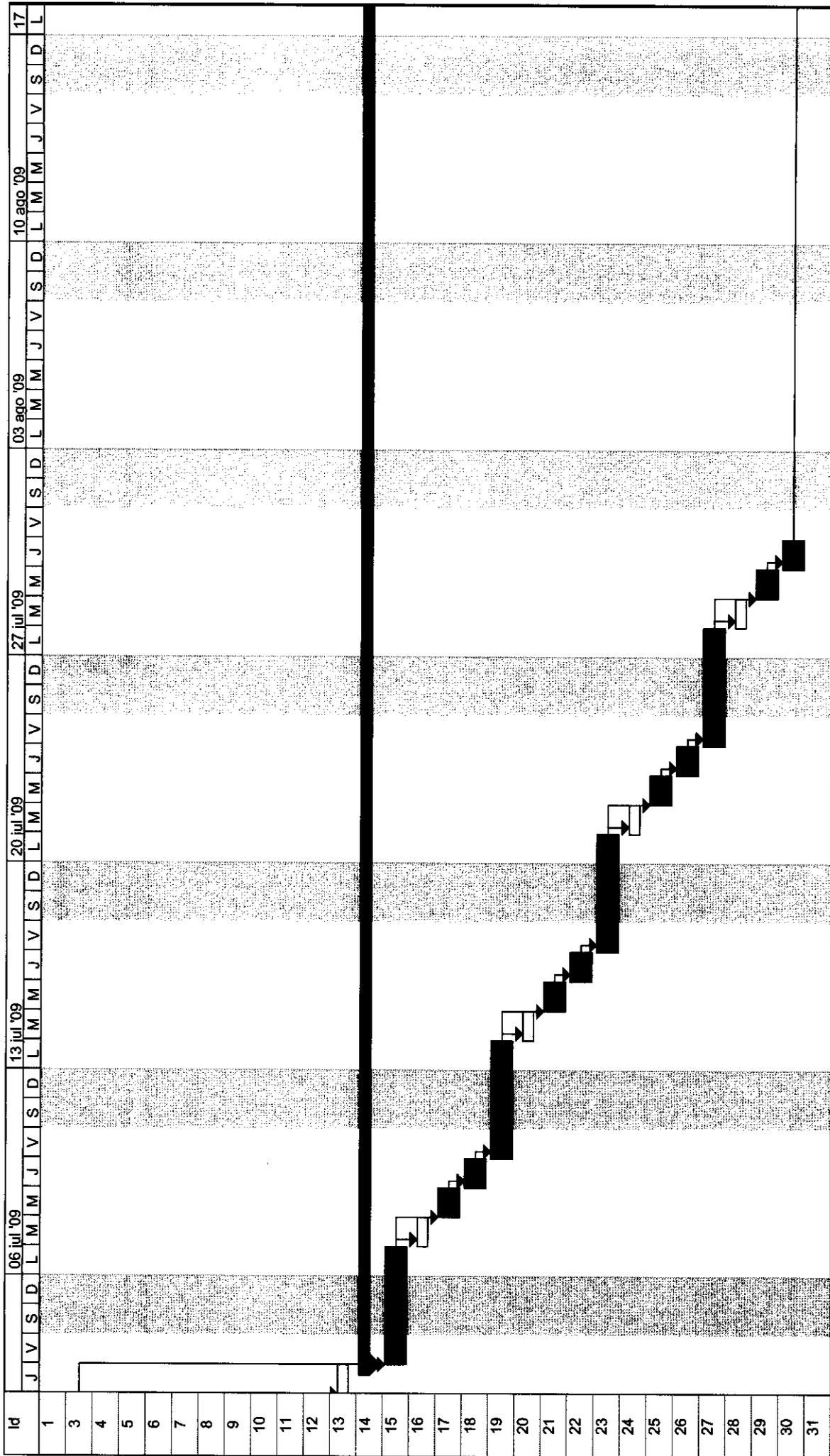


◆ Tareas externas
◆ Hitos externos
⇩ Fecha límite

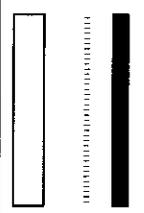
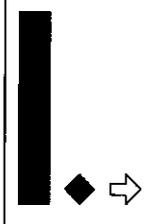
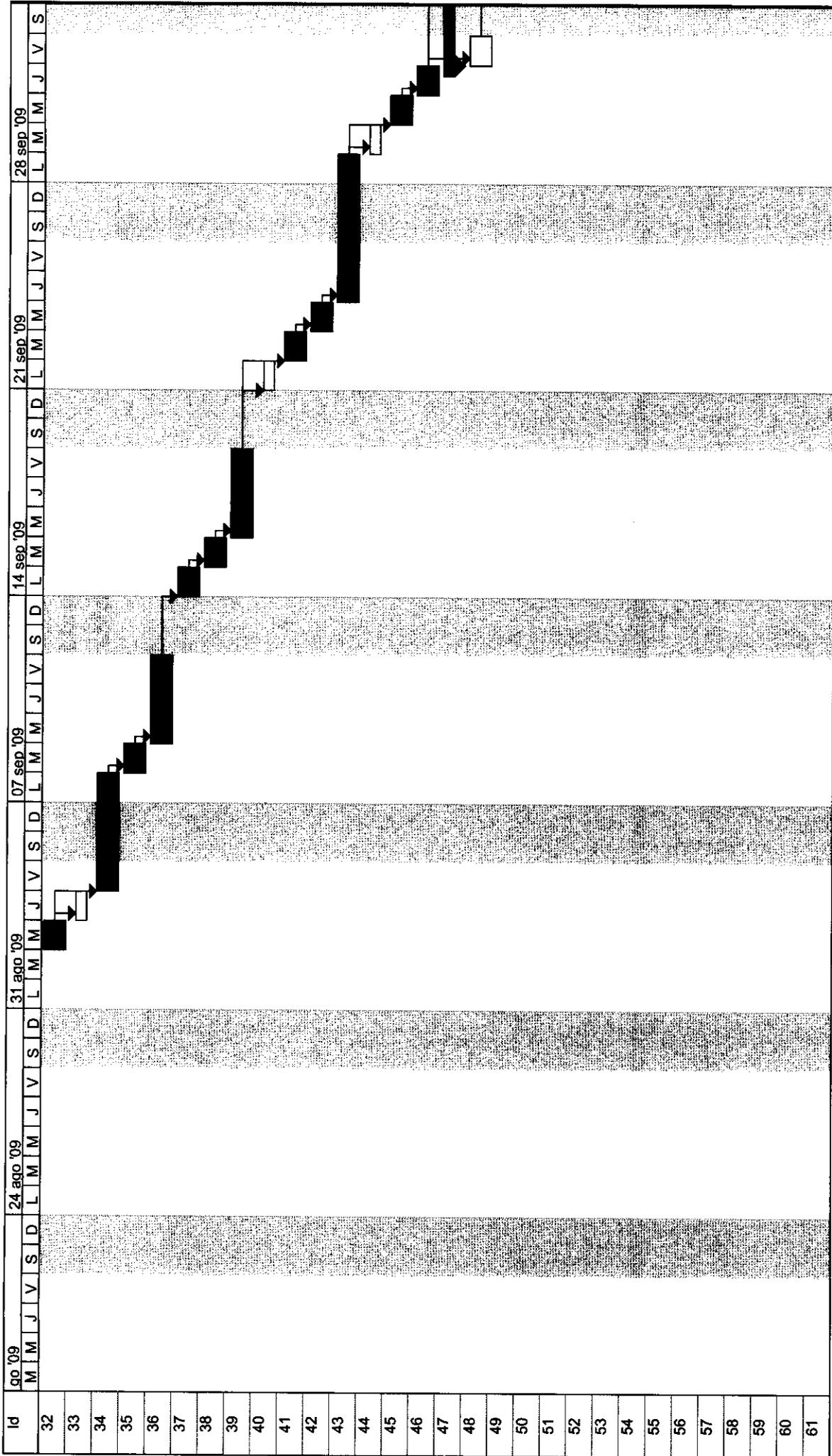
◆ Hitos
⇐ Resúmenes
⇐ Resúmenes del proyecto

▭ Tarea
⋯ División
▬ Progreso

Proyecto: gam2
 Fecha: vie 27/1/09

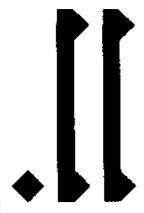
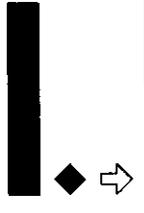


Proyecto: gani2
 Fecha: vie 27/1/09

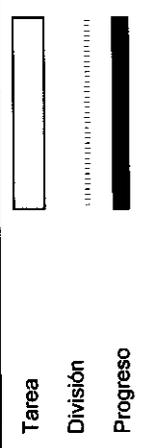
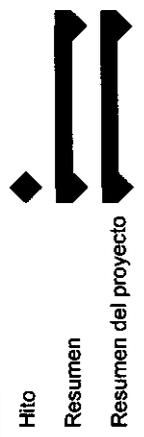
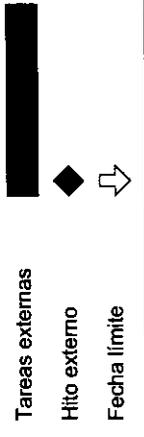
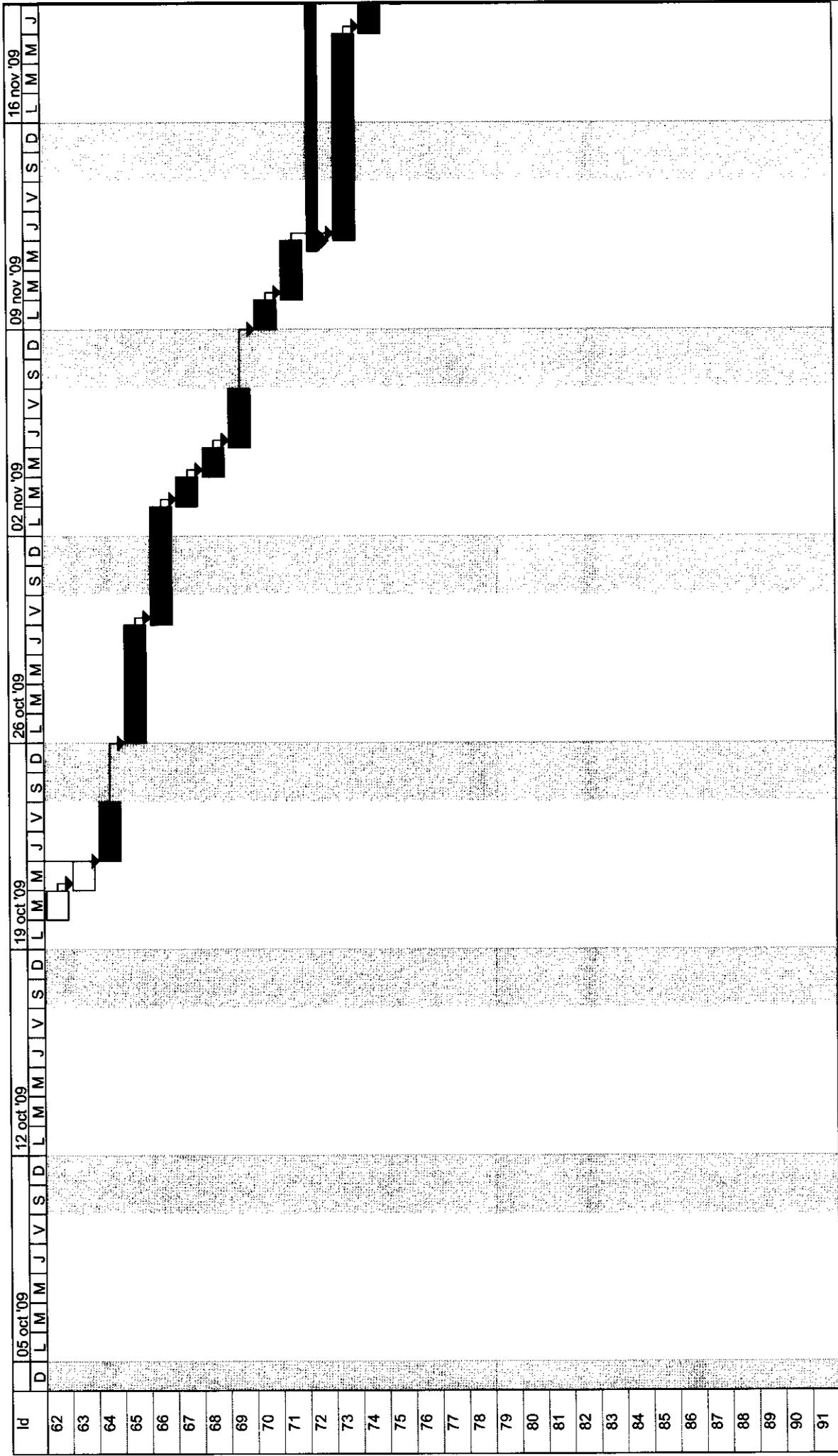


Proyecto: gan12
 Fecha: vie 27/1/09

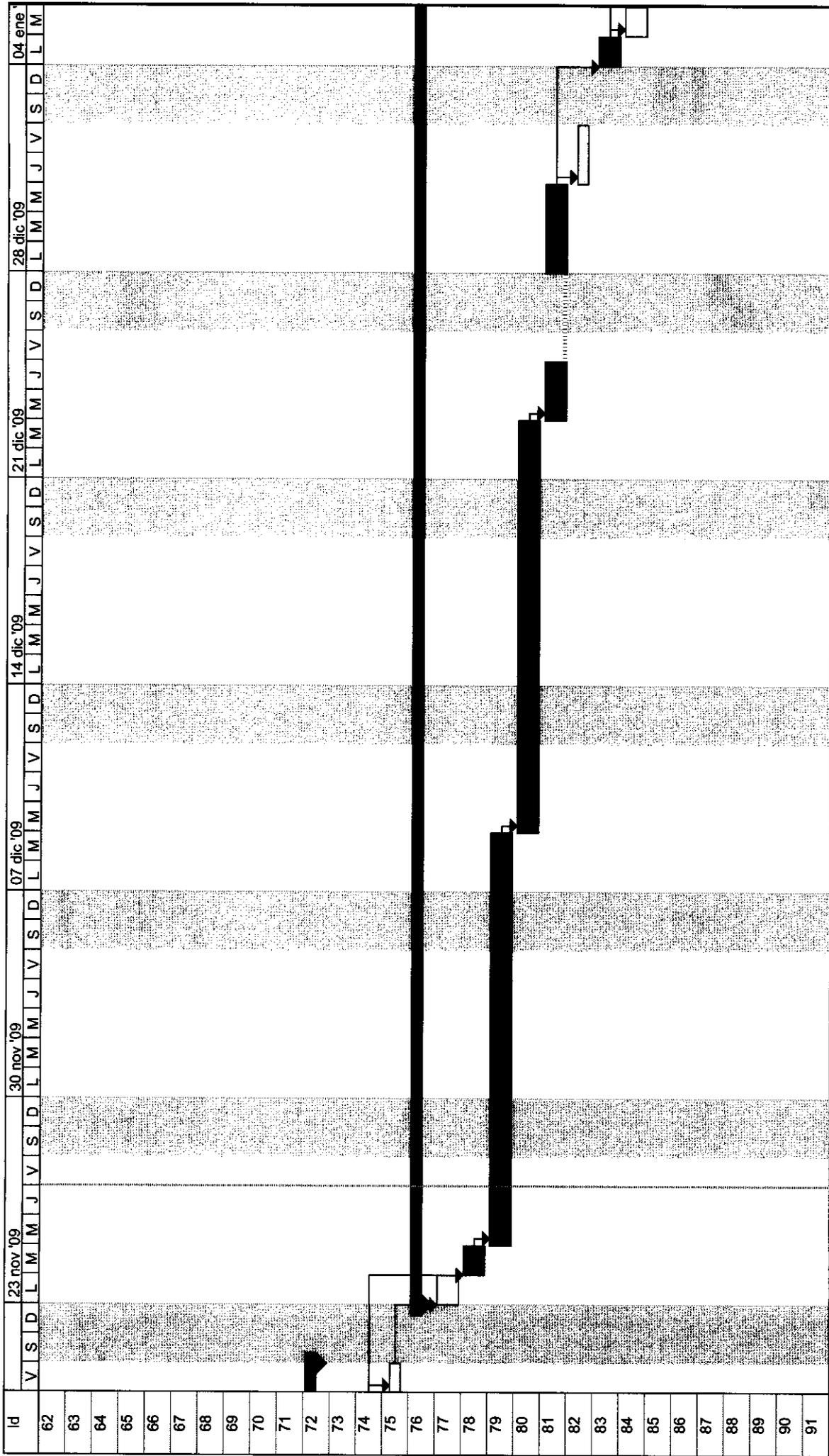
Id	05 oct '09			12 oct '09			19 oct '09			26 oct '09			02 nov '09			09 nov '09			16 nov '09							
	D	L	M	J	V	S	D	L	M	J	V	S	D	L	M	J	V	S	D	L	M	J	V	S	D	L
1																										
3																										
4																										
5																										
6																										
7																										
8																										
9																										
10																										
11																										
12																										
13																										
14																										
15																										
16																										
17																										
18																										
19																										
20																										
21																										
22																										
23																										
24																										
25																										
26																										
27																										
28																										
29																										
30																										
31																										



Proyecto: gant2
Fecha: vie 27/11/09



Proyecto: gant2
 Fecha: vie 27/11/09



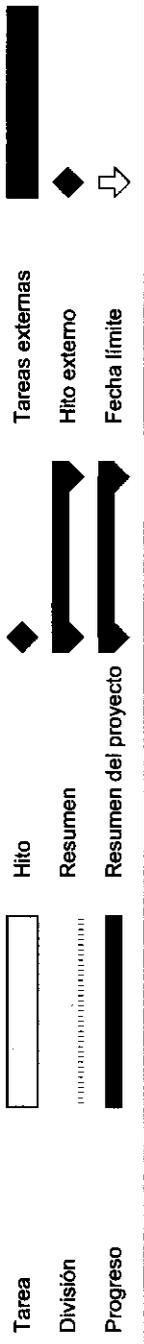
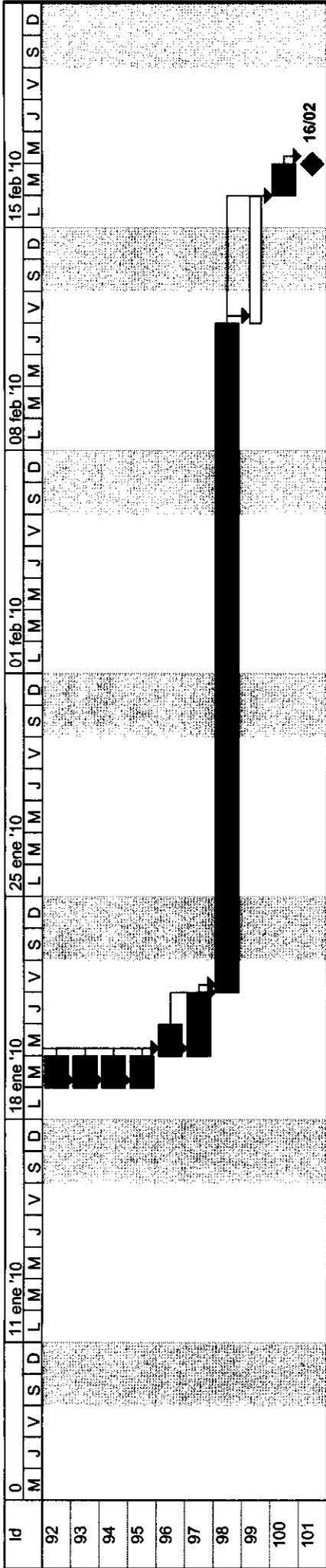
Proyecto: gami2
 Fecha: vie 27/11/09

Tarea		Hito		Tareas externas	
División		Resumen		Hito externo	
Progreso		Resumen del proyecto		Fecha límite	



	Tarea		Hito		Tareas externas
	División		Resumen		Hito externo
	Progreso		Resumen del proyecto		Fecha limite

Proyecto: gant2
 Fecha: vie 27/11/09



Proyecto: gant2
 Fecha: vie 27/11/09