

MTIG 11

Proyecto final

Universidad Aut3noma de Barcelona.
Departamento de Geograf3a.
Auding-Intraesa S.A.

Autor:

Roy A. Justo Torres

Tutores:

Miquel 3ngel Vargas (LIGIT)
Fernando L. Rodr3guez (Auding-Intraesa)

DISEÑO, DESARROLLO E IMPLEMENTACI3N DE UNA APLICACI3N WEB-MAP PARA LA VISUALIZACI3N Y CONSULTA DE LOS PROYECTOS EJECUTADOS POR AUDING-INTRAES A

Agradecimientos

Esta tesis de maestría, si bien ha requerido de la perseverancia y el esfuerzo dedicado por el autor, no hubiese alcanzado sus objetivos sin la colaboración y el entusiasmo mostrado por muchas personas. En este sentido me permito dedicar este espacio para mencionar a todos los que de alguna forma se han visto involucrado en este proyecto y que sirva para mostrar mi agradecimiento por el interés expresado y el tiempo dedicado para que todo saliera con la mayor calidad. Estas personas son:

- ❖ Miquel Ángel Vargas, tutor de este trabajo por la parte del LIGIT.
- ❖ Fernando L. Rodríguez, tutor por la parte de Auding-Intraesa S.A.
- ❖ Todos los profesores y estudiantes de la oncenava edición del máster en tecnologías de la información geográfica.
- ❖ El departamento de Gestión de la Información Territorial de Auding-Intraesa S.A., principalmente a Thaís Lima Nemer.
- ❖ Y muy especialmente a Ignacio Ferrero Beato por su invaluable ayuda, su interés y por todas las horas dedicadas a este proyecto.

A modo personal quisiera expresar mi infinito agradecimiento a todos los familiares y amigos que han apoyado incondicionalmente mis proyectos, y sin los cuales, no hubiese sido posible llegar a este punto del camino. Muchas gracias a todos....!!!

Resumen

Este trabajo consiste en la creación de una aplicación WEB-MAP para la consulta de información referente a los proyectos ejecutados por la empresa Auding-Intraesa S.A. Se trata de mostrar en un mapa la localización de cada proyecto y la información asociada a estos.

Para desarrollar esta aplicación se realizó un análisis de los requerimientos básicos, las funcionalidades deseadas, la información previa y los intereses corporativos.

A partir de esto efectuó un proceso de selección tecnológica con el fin de identificar la más adecuada a los requerimientos de la empresa. La aplicación proyectada ha requerido un sistema de gestión de bases de datos, un servidor web y un entorno de desarrollo de la aplicación. En este orden se seleccionaron, MySQL, HTTPApache y AdobeFlex. Estas tecnologías, además de su carácter gratuito y abierto, están avaladas por su extensa difusión en el mercado y por la gran cantidad de usuarios que tiene.

Una vez alcanzado este nivel se planificó el desarrollo del trabajo en tres grandes fases. La primera que involucraba la instalación y configuración de la tecnología, así como el desarrollo de la base de datos. Una segunda fase donde se diseña y se desarrolla la interfaz de usuarios, y finalmente una tercera donde se dota de funcionalidad a la aplicación para su publicación final.

Se diseñó el modelo de base de datos que recoge la información con carácter comercial orientada a la divulgación. Como parte del tratamiento de los datos fue necesario añadir la componente espacial a cada registro, incluyendo en cada caso las coordenadas geográficas (latitud y longitud) para su visualización en Google-Maps. También se recopilieron las fichas técnicas elaboradas por la empresa para cada proyecto donde se describen sus características principales, con el objetivo de que estas pudieran ser consultadas a través de la aplicación desde un directorio público.

Posteriormente se diseñó la interfaz de usuario partiendo de los siguientes criterios básicos: mantener la imagen corporativa, que fuera intuitiva, amigable, eficiente y atractiva. Se desarrolló una herramienta de búsqueda en un primer nivel que guía al usuario a obtener una lista de proyectos ejecutados dentro de un área de trabajo, un tema específico de trabajo y finalmente una tipología determinada. Con este resultado se puede realizar un nuevo filtro para obtener nuevas listas en función del año a partir del cual se quiere consultar los proyectos visualizados. Al mismo tiempo aparecen en un mapa la localización asociada a cada proyecto. Tanto de esta lista, como de su símbolo correspondiente en el mapa, se puede acceder a la información específica de cada uno de los proyectos.

Se utilizó el API de Google Maps para Flash. Tanto la tecnología de servidor y de consulta de mapas e imágenes que ofrece Google Maps, así como las posibilidades de las librerías

desarrolladas por Flex, permitieron una gran calidad y eficiencia en el proceso de consulta y visualización. Cubriendo así, uno de los más importantes requerimientos establecidos.

Para dotar de la funcionalidad requerida a la aplicación, se desarrollaron un conjunto de rutinas en ActionScript 3. Estas funcionalidades incluyen la consulta y actualización en tiempo de ejecución de la información mostrada en la interfaz, la consulta y descarga de ficheros externos en formato PDF, la fácil navegación por el mapa con capacidades de zoom, pan, cambio de cartografías y el rápido acceso a la información deseada.

La aplicación dotó a la empresa de un modelo de base de datos unificada y homogénea orientada a la divulgación de la información con carácter comercial de los proyectos ejecutados. Mejoró la capacidad de divulgar y distribuir de forma atractiva eficiente y masiva el trabajo desempeñado desde su fundación. Optimizó la comunicación con los clientes. Y, permitió la implantación de nuevas tecnologías para la administración y divulgación de su información de índole comercial.

Finalmente se proponen un conjunto de recomendaciones a modo de mejoras, que no pudieron ser desarrolladas en el tiempo previsto para este ejercicio académico.

Palabras claves: Google Maps, Adobe Flex, AS3, MySQL, PHP, Visor WebMap, Sistemas de Información Geográfica, RIA, Aplicaciones web enriquecidas.

Abstract

This Project is based on develop a WEB-MAP application so as users would be able to consult information related to projects executed by Auding-Intraesa S.A. Mainly consist on showing in a map every project located on its geographic context, and a tool for access to the information related to each of them.

To make this done, it was necessary to analyze corporation demands to get fulfill during the development process. Technological platforms selected to work with were, MySQL for server technology, HTTP Apache for Web server technology and Adobe Flex Builder 3 for IDE (stand for Integrated Development Environment). All of them are very widespread technologies, widely used and documented in the World Wide Web.

At this level, the project was planned to be carried out in three phases. The first one involves the setting up and configuration process and, designing database structure. Next dealt with designing user interface, and the last one, had to do with providing of behavior and functionality the Application.

It was collected attractive information from the marketing point of view, PDF files created by the enterprise for describing projects, geographical coordinates for every project among others.

User interface was designed according to enterprise image concept. Interactive components in the web application pretended to be intuitive, short time-consuming and easy to use. Communication flow between user and application is summed up in three steps. First one, guide the user to select a framework area where projects belong to, next a second level of filter guide user to select a specific theme of work among area selected before, and finally a typology filters that are provided a list of projects titles because of the previous consulting criteria. This list can also be filtered by the year from which desire projects, were assigned to the corporation. At the same time that mentioned list is filled, the map show spatial locations related to each project shown.

This Web Application allows users to consult information related to services provided by the enterprise in an attractive way. Also, allow user download projects profiles, and interact with the map. This work provides Auding-Intraesa with a solid central project database for marketing. Also, provide a way to show its influence on the market and to implement new technologies to distribute its information.

Some recommendations are provided at the end of this paper work, in order to improve what couldn't be done during this project.

Índice

INTRODUCCIÓN.....	1
CAPÍTULO I. MARCO TEÓRICO METEDOLÓGICO	4
CAPÍTULO II. PLATAFORMA TECNOLÓGICA	7
II.1 Análisis de la información de partida.	7
II.2 Fundamentación y requerimientos tecnológicos.	9
II.3 Desarrollo de la estructura de la base de datos.	15
CAPÍTULO III. DISEÑO DE LA INTERFAZ DE USUARIO.....	17
III.1 Diseño del cliente web con el API de Google Maps para Flash.	17
III.2 Conexión entre el cliente web y la base de datos.	19
III.3 Estructura y personalización del cliente web.	21
CAPÍTULO IV. DESARROLLO E IMPLEMENTACIÓN DE LA APLICACIÓN	27
IV.1 Estructura y funcionamiento de la aplicación.	27
IV.2 Modelo lógico de desarrollo	31
IV.3 Características de puesta en marcha y rendimiento de la aplicación	36
CONCLUSIONES	38
RECOMENDACIONES	39
BIBLIOGRAFIA	40
ANEXOS	41

Índice de figuras y tablas

FIGURAS:

FIGURA 1: Plan de trabajo	5
FIGURA 2: Fases del proyecto	6
FIGURA 3: Esquema de funcionamiento de la tecnología empleada	11
FIGURA 4: Panel buscador	22
FIGURA 5: Panel proyectos	22
FIGURA 6: Ventana de información.....	22
FIGURA 7: Estructura de la interfaz de usuario	23
FIGURA 8: Estructura del diseño del panel Buscador.....	24
FIGURA 9: Estructura del diseño del panel Proyectos.....	24
FIGURA 10: Estructura del diseño de InfoWindow.mxml.....	25
FIGURA 11: Diseño del encabezamiento de la página web	25
FIGURA 12: Diseño de la interfaz de usuario de la aplicación.....	26
FIGURA 13: Estructura para la creación de la aplicación AudingSpatial según la lógica de desarrollo de Flex	28
FIGURA 14: Modelo lógico de desarrollo de la aplicación	31
FIGURA 15: Esquema de funcionamiento de Infowindow.mxml	34

TABLAS:

TABLA 1: Clasificación de los proyectos ejecutados por Auding-Intraesa S.A. en función del marco de trabajo donde se realizan	7
TABLA 2: Tecnologías adoptadas para el desarrollo de la aplicación	11
TABLA 3: Estructura de la tabla Projauding, de la base de datos Auding_progis	15
TABLA 4: Ficheros generados por la aplicación de conexión a base de datos.....	19
TABLA 5: Componentes de la aplicación audinig_spatial.mxml.....	29
TABLA 6: Métodos creados dentro del fichero Projauding.php, para la consulta personalizada a la base de datos	32

Introducción

El uso de la web para la diseminación de cualquier tipo de información susceptible a ser representada espacialmente, además de representar un salto significativo en el desarrollo de la cartografía, ha abierto innumerables oportunidades para muchas empresas. La posibilidad de compartir y actualizar información en un contexto geográfico, personalizar el contenido de los mapas e identificar patrones de distribución espacial en el comportamiento de los negocios, son solo algunas de las ventajas.

En los últimos años la industria geoespacial ha desarrollado cuantiosos productos en función de las crecientes necesidades del mercado. Hoy en día asistimos a la emergencia de una nueva serie de herramientas cartográficas online de bajo coste como Google Earth y Google Maps, NASA World Wind, Yahoo Maps entre otros. La ventaja comercial está relacionada con la gran masa de usuarios que consultan estos productos y con las posibilidades de desarrollo y personalización que brindan muchos de estos proveedores de servicios geoespaciales. El desarrollo de APIs, permite a los usuarios personalizar sus mapas, desarrollar entornos de consultas, publicar actividades y servicios, e incrementar la comunicación con los clientes.

Desde su fundación, en 1979, Auding proporciona a sus clientes una extensa gama de servicios profesionales de ingeniería y consultoría, relacionados con el estudio, definición y ejecución de las infraestructuras y los servicios públicos. En todo este periodo esta empresa ha acumulado una gran experiencia así como mayor ampliación de su influencia en el mercado reflejada en la gran cantidad de proyectos realizados en diferentes ámbitos geográficos. Su reciente fusión con Intraesa S.A. consolida aun más su posición dentro del ámbito empresarial de la Ingeniería Civil en España. En el proceso de diversificación de servicios que ha experimentado la empresa, se creó el departamento de Gestión de la Información Territorial, que se ocupa de la problemática territorial y está orientado a la ejecución de proyectos SIG en función de las necesidades específicas de cada cliente. En este marco realiza una gestión global que va desde la consultoría a la integración de sistemas, pasando por el desarrollo de aplicaciones personalizadas y la formación y mantenimiento de los productos implementados. Desde este departamento surge la iniciativa de crear una aplicación orientada a divulgar de forma interactiva y actualizada, el campo de influencia de la empresa. Se basa fundamentalmente y generar una aplicación Web que muestre la localización de cada proyecto ejecutado por Auding-Intraesa S.A. y a su vez, ofrezca información sobre las peculiaridades de cada uno de ellos.

La idea principal es que los usuarios, ya sean clientes reales o potenciales, o personas e instituciones interesadas en el tema, puedan acceder de manera eficiente y sencilla a un conjunto de informaciones relacionadas con cada proyecto desarrollado por esta empresa. Los usuarios podrían tener acceso a información sobre el título de cada proyecto, en que marco de trabajo se encuentra adjudicado, información relacionada con las fechas de inicio y

finalización, importe etc. Tendrán además la posibilidad de descargar una ficha técnica de cada proyecto si así lo desean. La búsqueda en función de la localización, y de la especialidad en que se encuentra enmarcado cada proyecto, permitirá orientar rápidamente a los usuarios en función de sus intereses. Esta aplicación aportará una base de datos coherente y centralizada de los proyectos realizados por Auding-Intraesa desde su fundación, creará las bases tecnológicas para la creación de una herramienta corporativa para la gestión de la información sobre los proyectos, y ofrecerá una visión sobre la distribución espacial del campo de acción de la empresa.

Es en este marco que envuelve al departamento de Gestión de la Información Territorial de Auding-Intraesa S.A, en cooperación con el Laboratorio de Información Geográfica y Teledetección (LIGIT) del departamento de Geografía de la Universidad Autónoma de Barcelona (UAB), donde se realiza este proyecto.

Este informe constituye el trabajo final para optar por título de máster en tecnologías de la información geográfica, y describe las particularidades del proceso de diseño y desarrollo de la aplicación, quedando estructurado en tres capítulos que se describen a continuación.

Capítulo I. Marco teórico metodológico.

En este capítulo se describe los fundamentos teóricos metodológicos del desarrollo de la aplicación. Se define el objetivo principal, las tareas ejecutadas para alcanzar este objetivo, los requerimientos necesarios, así como particularidades del proceso de planificación y organización en la ejecución de todo el trabajo.

Capítulo II. Plataforma Tecnológica

En este capítulo se hace un análisis de la información de partida. Se analiza también las diferentes alternativas tecnológicas en función de los intereses de la empresa y a partir de esto se comenta las particularidades de la plataforma tecnológica seleccionada, y finalmente se explica cómo fue estructurada la base de datos de la aplicación.

Capítulo III. Diseño de la interfaz de usuario.

Este capítulo está enfocado en el desarrollo del cliente web, o sea, las características de la interface de usuario. Se comenta las particularidades del desarrollo de este tipo de clientes usando al API de Google-Maps para flash, como se conecta la aplicación cliente con la base de datos alojada en el servidor de datos, y finalmente como se diseñó y se estructuró el cliente WEB.

Capítulo IV. Desarrollo e Implementación de la aplicación.

En este capítulo se hace énfasis en la lógica de programación empleada y en la estructura lógica de funcionamiento de la aplicación. Se describe esquemáticamente como funciona cada

uno de los elementos que componen el proyecto en general. Y finalmente una valoración de las características de su implementación y rendimiento.

El informe también contiene unas conclusiones generales y unas recomendaciones donde se proponen mejoras a la aplicación que en el periodo de tiempo establecido no se pudieron elaborar.

Capítulo I. Marco teórico metodológico.

Este trabajo ha sido realizado en el departamento de Gestión de la Información Territorial de Auding-Intraesa S.A. como parte de un convenio de colaboración con la Universidad Autónoma de Barcelona. Este convenio permite acoger la realización del proyecto final del Máster en Tecnologías de la Información Geográfica, 11ª edición, organizado por el Departamento de Geografía, y llevado a cabo por el Laboratorio de Información Geográfica y Teledetección (LIGIT).

El objetivo fundamental es: *Diseñar, Desarrollar e Implementar una aplicación Web Map para la visualización y consulta de los proyectos ejecutados por Auding-Intraesa S.A.*

Basado en este objetivo fundamental, la aplicación estará dotada de las siguientes funcionalidades.

1. Reunir en una base de datos centralizada toda la información de carácter divulgativo, referente a los proyectos ejecutados por Auding-Intraesa S.A.
2. Mostrar los proyectos ejecutados por la empresa en un marco territorial, así como la información asociada a cada uno de ellos.
3. Consulta de la base de datos desde el cliente siguiendo diferentes criterios.

Estas funcionalidades deseadas implican la ejecución de diferentes tareas tanto de carácter organizativo, así como de diseño y desarrollo. Se pueden resumir en las siguientes:

- Análisis de la información inicial.
- Estudio y propuesta de la plataforma tecnológica
- Instalación y configuración de un servidor web y un servidor de bases de datos.
- Desarrollo de la estructura de la base de datos.
- Diseño del cliente web y la interfaz de usuario.
- Conexión de la base de datos con el cliente web.
- Desarrollo y personalización de las funcionalidades de la aplicación.
- Publicación de la aplicación.

A fin de concluir en el plazo establecido se llevó a cabo un plan de trabajo enmarcando cada tarea en un periodo determinado para su culminación. (Figura 1)

Figura 1. Plan de trabajo

Roy A. Justo Torres

Calendario y Plan de Trabajo.

Tesis de Maestría: Diseño desarrollo e implementación de una aplicación WebMap para la definición, visualización y consulta de los proyectos ejecutados por Auding-Intraesa S.A.

mes	septiembre		octubre				noviembre				diciembre			
semana	14 al 18	21 al 25	28 al 02	05 al 09	12 al 16	19 al 23	26 al 30	02 al 06	09 al 13	16 al 20	23 al 27	30 al 04	07 al 11	14 al 18
tareas														
Análisis de la plataforma tecnológica														
Implementación del servidor														
Desarrollo de la estructura de la base de datos.														
Desarrollo del cliente web														
Conectar la base de datos con el cliente.														
Publicación de entidades geográficas														
Añadir funcionalidades al cliente web														
Personalización del cliente.														
Alojar la aplicación en el servidor de Auding.														

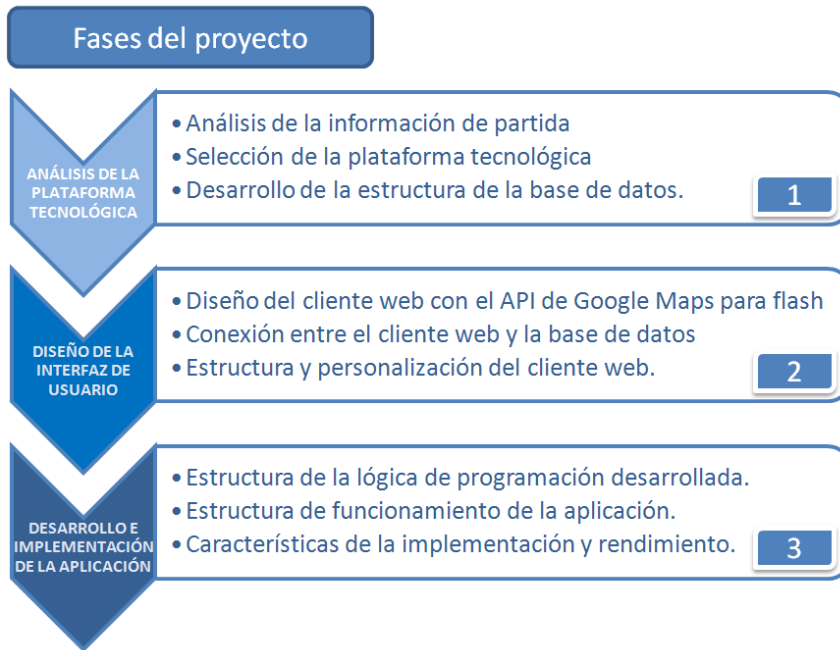
Fuente: elaborado por el autor

Los beneficios que resultarían de esta aplicación serían:

- Una base de datos unificada y homogénea orientada a la divulgación de información con carácter comercial de los proyectos ejecutados por la empresa.
- Localización espacial de cada uno de estos proyectos.
- Divulgación y distribución masiva, eficiente y atractiva de la información relacionada con los proyectos en su marco territorial.
- Visión corporativa de la información relacionada con sus proyectos.
- Mejora de la comunicación con los clientes.
- Implantación de nuevas tecnologías y mejoras del proceso de administración y divulgación de la información.
- Mayor Promoción el trabajo desarrollado por Auding-Intraesa S.A.

A modo organizativo, se puede resumir en tres grandes fases toda la labor realizada para alcanzar el objetivo principal de este trabajo como se muestra en la figura 2. Cada una de estas fases describe los procedimientos y las características principales de las tareas realizadas.

Figura 2. Fases del proyecto.



Fuente: elaborado por el autor

CAP II. PLATAFORMA TECNOLÓGICA

Contenido del capítulo:

- *Análisis de la información de partida.*
- *Fundamentación y requerimientos tecnológicos.*
- *Desarrollo de la estructura de la base de datos*

I.1 Análisis de la información de partida.

Cada uno de los proyectos ejecutados por Auding-Intraesa S.A. tiene asociado un conjunto de información que es de interés comercial y corporativo. Actualmente la información de los proyectos se gestiona desde una aplicación desarrollada sobre Visual Basic y Acces.

En este sentido la empresa ha adoptado la siguiente clasificación basada en seis grandes áreas de trabajo y varios temas de trabajo asociados a cada una de estas grandes áreas. El objetivo es normalizar la información de carácter divulgativo de los proyectos siguiendo esta clasificación.

Tabla 1. Clasificación de los proyectos ejecutados por Auding-Intraesa S.A. en función del marco de trabajo donde se realizan.

Área de Trabajo	Tema de trabajo
Infraestructuras del Transporte	Carreteras Autopistas Ferrocarriles Metro Tranvía Aeropuertos
Agua y Medio Ambiente	Planificación hidrológica Potabilización y desalación Depuración y regeneración Hidrología e hidráulica fluvial Ingeniería agroforestal Impacto ambiental Energías renovables

Área de Trabajo	Tema de trabajo
Tecnología e Instalaciones	Energía. Generación, transformación y transporte Sistemas electromecánicos Señalización y control Gestión de incidencias y emergencias Sistemas de Transporte Inteligente
Urbanismo y Edificación	Planeamiento Infraestructuras urbanas Parques Polígonos industriales Edificación residencial y pública
Estructuras e Ingeniería del Terreno	Puentes Obras singulares Cimentación de estructuras Túneles y obras subterráneas Tratamientos de mejora del terreno Estabilidad de taludes
Consultoría	Economía de servicios Gestión de activos Participación público-privada Gestión de información territorial Control de explotación Auditorías

Fuente: elaborado por el autor

A su vez, cada uno de estos proyectos tiene una TIPOLOGÍA asociada. Clasificándose así, según este criterio, en proyectos, anteproyectos, estudios, dirección de obras etc. Esta jerarquía de la información permite establecer filtros o una guía para su consulta.

El resto de la información está relacionada con características particulares de cada proyecto, o sea, las fechas de inicio y fin, el presupuesto adjudicado, el título, el contratista o cliente, el contexto geográfico (comunidad autónoma, municipio, provincia, país), el o los autores entre otros, y unas fichas técnicas con fotos y detalles sobre dichos proyectos.

I.2 Fundamentación y requerimientos tecnológicos

Los requerimientos tecnológicos para el desarrollo de esta aplicación se pueden resumir en los siguientes:

1. Un servidor de bases de datos, así como una herramienta para su gestión.
2. Un servidor Web
3. Un entorno de desarrollo de la aplicación. IDE (Integrated development environment)

En el mercado existe una gran diversidad de productos y tecnologías que pueden ser usadas para lograr las funcionalidades que se requieren para esta aplicación. Los sistemas de gestión de bases de datos (SGBD) son software muy específicos desarrollados principalmente para servir de interfaz entre las bases de datos y los usuarios y las aplicaciones. Su propósito es manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización. Los SGBD existentes en el mercado se pueden dividir en tres grupos fundamentales en función de su coste. Existen los SGBD libre, los no libres y los no libres - gratuitos. El SGBD escogido para este trabajo se encuentra en el primer grupo. Más adelante se explica sus particularidades, las ventajas de su uso y como se complementa con el resto de las tecnologías escogidas. Si es importante aclarar que una premisa muy importante en la selección de la tecnología era el carácter liberado y gratuito de su uso.

Un servidor web es un programa que está diseñado para transferir hipertextos, páginas web o páginas HTML (HyperText Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música. El programa implementa el protocolo HTTP (HyperText Transfer Protocol) que pertenece a la capa de aplicación del modelo OSI. El término también se emplea para referirse al ordenador que ejecuta el programa. El Servidor web se ejecuta continuamente en un ordenador, manteniéndose a la espera de peticiones por parte de un cliente (un navegador web) y que responde a estas peticiones adecuadamente, mediante una página web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error. Además de la transferencia de código HTML, los Servidores web pueden entregar aplicaciones web. Éstas son porciones de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre aplicaciones en el lado del cliente y en el lado del servidor. En este caso se usó la del lado del cliente web. Este es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones donde el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones. Comúnmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje JavaScript y Java, aunque

pueden añadirse más lenguajes mediante el uso de PLUGINS¹. Es frecuente requerir de estos complementos que amplían las funciones de las páginas web para ver contenidos interactivos. Un ejemplo conocido es FlashPlayer de Adobe. Un complemento que carga animaciones multimedia y aplicaciones interactivas desde el cliente web.

Instalar un servidor web en nuestro PC permite, entre otras cosas, poder montar nuestra propia página web sin necesidad de contratar hosting, probar nuestros desarrollos vía local y acceder a los archivos de nuestro equipo desde un PC remoto.

En el caso del entorno de desarrollo integrado para desarrollo de aplicaciones web existen también muchas en el mercado. Sin embargo en los últimos años se han venido desarrollando una filosofía de desarrollo web conocido como RIA (Rich Internet Applications) o Aplicaciones de internet enriquecidas.

Son aplicaciones web que tienen la mayoría de las características de las aplicaciones tradicionales. Estas aplicaciones utilizan un “navegador web” estandarizado para ejecutarse y por medio de “plugins” o independientemente una “virtual machine” o “sandbox”, se agregan las características adicionales. Surge como una combinación de las ventajas que ofrecen las aplicaciones Web y las aplicaciones tradicionales. Buscan mejorar la experiencia del usuario

Normalmente en las aplicaciones Web, hay una recarga continua de páginas cada vez que el usuario pulsa sobre un enlace. De esta forma se produce un tráfico muy alto entre el cliente y el servidor, llegando muchas veces, a recargar la misma página con un mínimo cambio.

En los entornos RIA, en cambio, no se producen recargas de página, ya que desde el principio se carga toda la aplicación, y sólo se produce comunicación con el servidor cuando se necesitan datos externos como datos de una Base de Datos o de otros ficheros externos.

Hay muchas herramientas para la creación de entornos RIA. Entre las más usadas se pueden mencionar las plataformas Adobe Flash, Adobe Flex, de Adobe, Silverlight de Microsoft, la trilogía HTML/JavaScript/Ajax, y OpenLaszlo entre otros.

En la siguiente tabla se muestra la tecnología adoptada para cada caso. Para escoger cada una se tuvo en cuenta los objetivos propuestos, las características de cada una de estas tecnologías y como se complementan cada una de ellas entre sí, con el fin de optimizar el rendimiento y las prestaciones de la aplicación final.

¹ PLUGIN: Un **complemento** es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API. También se lo conoce como **plugin** (del inglés "enchufable"), **add-on**(agregado), **complemento**, **conector** o **extensión**.

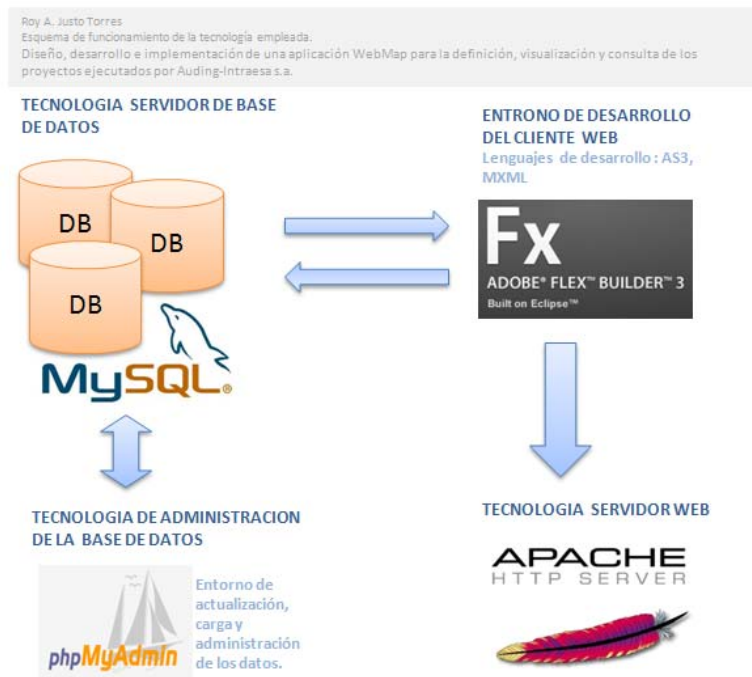
Tabla 2. Tecnologías adoptadas para el desarrollo de la aplicación.

TECNOLOGIA	SOFTWARE
Servidor WEB	HTTP Apache
Sistema de gestión de bases de datos	MySQL
Administrador Base de Datos	PHPMyAdmin
Desarrollo cliente Web	Adobe Flex API Google Maps para Flash
LENGUAJES DE DESARROLLO	
✓ Cliente: MXML, AS3.	
✓ Servidor: PHP, SQL.	

Fuente: elaborado por el autor

El esquema de funcionamiento e interacción entre estas tecnologías se muestra en la siguiente figura.

Figura 3. Esquema de funcionamiento de la tecnología empleada.



Fuente: Elaborada por el autor.

A continuación se describe las características principales de cada una de las tecnologías utilizadas.

Servidor WEB HTTP Apache

El servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociación de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración. Tiene amplia aceptación en la red. Desde 1996 es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo.

Apache es usado primariamente para enviar páginas web estáticas y dinámicas en la World Wide Web. Es el componente de servidor web en la popular plataforma de aplicaciones LAMP, junto a MySQL y los lenguajes de programación PHP, Perl, Python entre otros.

La mayor parte de la configuración se realiza en el fichero *apache2.conf* o *httpd.conf* según el sistema donde se esté ejecutando. Cualquier cambio en éste archivo requiere reiniciar el servidor, o forzar la lectura de los archivos de configuración nuevamente.

La licencia Apache (Apache License o Apache Software License para versiones anteriores a 2.0) es una licencia de software libre creada por la Apache Software Foundation (ASF). La licencia Apache (con versiones 1.0, 1.1 y 2.0) requiere la conservación del aviso de copyright y el disclaimer, pero no es una licencia copyleft, ya que no requiere la redistribución del código fuente cuando se distribuyen versiones modificadas.

Como cualquier otra de las licencias de software libre, la Licencia Apache permite al usuario del software la libertad de usarlo para cualquier propósito, distribuirlo, modificarlo, y distribuir versiones modificadas de ese software. La Licencia Apache sólo exige que se mantenga una noticia que informe a los receptores que en la distribución se ha usado código con la Licencia Apache. (Bib. c. h.)

Sistema de Gestión de Bases de Datos MySQL

MySQL es un sistema de gestión de base de datos relacional y multiusuario con más de seis millones de instalaciones. Es desarrollado como software libre en un esquema de licenciamiento dual por MySQL AB, una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y el copyright del código está en poder del autor individual, MySQL es propietario y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

MySQL es muy utilizado en aplicaciones web, como Drupal o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

Lenguaje de programación PHP

Este es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica.

PHP es un acrónimo recursivo que significa PHP *Hypertext Preprocessor* (inicialmente PHP Tools, o, *Personal Home Page Tools*). Fue creado originalmente por Rasmus Lerdorf en 1994 y publicado bajo la PHP License. La Free Software Foundation considera esta licencia como software libre.

Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, aunque el número de sitios en PHP ha compartido algo de su preponderante sitio con otros nuevos lenguajes no tan poderosos desde agosto de 2005. Es también el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web.

Entorno de desarrollo del cliente WEB. Adobe Flex

Es un término que agrupa una serie de tecnologías publicadas desde marzo de 2004 por Macromedia para dar soporte al despliegue y desarrollo de Aplicaciones Enriquecidas de Internet, basadas en su plataforma propietaria Flash.

Los programadores tradicionales de aplicaciones ven como un desafío adaptar la metáfora de la animación sobre la plataforma con la cual fue originalmente construido Flash. Flex minimiza

elegantemente este problema proveyendo un flujo de trabajo y un modelo de programación que es familiar a los desarrolladores de aplicaciones.

Flex fue inicialmente liberado como una aplicación de la J2EE o biblioteca de etiquetas JSP que compilara el lenguaje de marcas Flex (MXML) y ejecutara mediante ActionScript aplicaciones Flash (archivos SWF binarios). Versiones posteriores de Flex soportan la creación de archivos estáticos que son compilados, y que pueden ser distribuidos en línea sin la necesidad de tener una licencia de servidor.

El objetivo de Flex es permitir a los desarrolladores de aplicaciones web construir rápida y fácilmente Aplicaciones de Internet Ricas, también llamadas RIAs. En un modelo multi-capas, las aplicaciones Flex son el nivel de presentación.

Flex pone en relieve el desarrollo de Interfaces gráficas de usuario usando un lenguaje XML llamado MXML. Flex tiene varios componentes y características que aportan funcionalidades tales como Servicios Web, objetos remotos, arrastrar y soltar, columnas ordenables, gráficas, efectos de animación y otras interacciones simples. El cliente solo carga la aplicación una vez, mejorando así el flujo de datos frente a aplicaciones basadas en HTML (eg. PHP, ASP, JSP, CFMX), las cuales requieren de ejecutar plantillas en el servidor para cada acción. El lenguaje y la estructura de archivos de Flex buscan el desacoplamiento de la lógica y el diseño.

El servidor Flex también actúa como una puerta de enlace, permitiendo al cliente comunicarse con servicios web XML y objetos remotos, tales como: Coldfusion CFCs, clases Java, y cualquiera que soporte el formato de mensajes de acciones.

API Google Maps para Flash

Esta API permite a los desarrolladores Flex insertar Google Maps en aplicaciones Flash. De forma similar a la versión Java Script, esta API de ActionScript proporciona varias utilidades para manipular y añadir contenido a mapas a través de distintos servicios, lo que te permite insertar aplicaciones de mapas interactivas y complejas en un sitio web. El equipo de Google Maps publica constantemente nuevos datos de mapas y mejora la cobertura internacional. No existe ningún límite relativo al número de visitas de la página de mapas del sitio web. El API de Google Maps está disponible para todos los sitios de acceso gratuito. Se puede utilizar en sitios web comerciales y sin ánimo de lucro, siempre que dichos sitios cumplan las condiciones del servicio. Se puede utilizar siempre que el sitio sea accesible desde el punto de vista general a los clientes de forma gratuita.

I.3 Desarrollo de la estructura de la base de datos.

La información relacionada con cada uno de los proyectos ejecutados por la empresa no ha sido fundamentalmente gestionada a nivel corporativo. El carácter diverso de los servicios ofertados hace que en cada caso existan particularidades con respecto a la información manejada para cada proyecto, y por tanto una gran variedad de datos e informaciones asociadas. De esta forma la primera tarea consistió en elaborar una selección de la información que tuviera un carácter de interés general y que fuera atractiva desde el punto de vista comercial, para ser mostrada y compartida en la aplicación.

Una vez realizado esto se creó la base de datos denominada `auding_progis`, y dentro de esta una tabla denominada `Projauding`. La simplicidad de los datos utilizados para la difusión comercial, no requirió del diseño de una base de datos relacional. Sin embargo las condiciones técnicas están creadas para un futuro desarrollo en caso de que sea interés aumentar la complejidad de la base de datos.

Tabla 3. Estructura de la tabla `Projauding`, de la base de datos `Auding_progis`.

Base de datos: AUDING_PROGIS Tabla : PROJAUDING	
CAMPOS	TIPO
PROYECTO_ID	Integer (6)
AREA_TRABAJO	Varchar (200)
TEMA_TRABAJO	Varchar (200)
TIPOLOGIA	Varchar (200)
TITULO	Varchar (300)
CLIENTE	Varchar (300)
IMPORTE	Double
MONEDA	Varchar (3)
MUNICIPIO	Varchar (100)
CC_AA	Varchar (100)
PAIS	Varchar (50)
FECHA_INI	Varchar (5)
FECHA_FIN	Varchar (5)
CUSTODIA	Varchar (50)

CAMPOS	TIPO
LOCALIZACION*	Varchar (100)
FICHAS*	Varchar (300)
CERTIFICADOS*	Varchar (300)
PROYECTE	Varchar (300)
AUTOR	Varchar (50)
LAT*	Double (10.6)
LNG*	Double (10.6)

Fuente: elaborado por el autor

Los campos señalados con (*), han sido añadidos por el autor de este trabajo y su contenido en casi todos los casos son URL que apuntan a ficheros externos que contienen información asociada a los proyectos. En el caso de la LOCALIZACION se refiere a ficheros KML² que están alojados en la carpeta pública del servidor apache (*htdocs*), y contiene la localización geográfica de cada uno de los proyectos (Ver anexo 1). En el caso de las FICHAS y los CERTIFICADOS, la URL apunta a otra carpeta que guarda en formato PDF las fichas técnicas de cada proyecto y los certificados de los proyectos que los han obtenido, con el fin de que puedan ser descargados desde la aplicación. Sobre estos aspectos se hablara con más detalle en los capítulos siguientes. Los campos de LAT y LNG son campos que contienen las coordenadas geográficas latitud y longitud respectivamente, de cada proyecto.

² **KML** (del acrónimo en inglés **Keyhole Markup Language**) es un lenguaje de marcado basado en XML para representar datos geográficos en tres dimensiones. Fue desarrollado para ser manejado con Keyhole LT, precursor de Google Earth (Google adquirió *Keyhole LT* en Octubre de 2004 tras lanzar su versión LT 2).

CAP III. DISEÑO DE LA INTERFAZ DE USUARIO

Contenido del capítulo:

- *Diseño del cliente Web con el API de Google Maps para Flash.*
- *Conexión entre el cliente Web y la base de datos.*
- *Estructura y personalización del cliente Web.*

III.1 Diseño del cliente web con el API de Google Maps para Flash.

El desarrollo de contenido Flash que integre Google Maps requiere la inclusión de la biblioteca de interfaces del API de Google Maps para Flash en el código de la aplicación creada. Esta biblioteca consta de un archivo (**.swc*) situado en el directorio "lib" del kit de desarrollo de software (SDK) del API de Google Maps para Flash disponible en la siguiente URL:

<http://maps.googleapis.com/maps/flash/release/sdk.zip>

El SDK incluye dos archivos SWC: una versión Flex para utilizarla con FlexBuilder (o con la aplicación gratuita Flex SDK) y una versión no Flex para utilizarla con Flash CS3. La versión Flex del archivo (**.swc*) aparece marcada con el sufijo "_flex" en el nombre de archivo.

Estos archivos SWC contienen interfaces para todas las clases públicas del entorno de desarrollo del API de Google Maps para Flash. La compilación de una aplicación con esta biblioteca garantiza que se pueda utilizar y comunicar con todas las funciones públicas de la biblioteca del tiempo de ejecución del API de Google Maps para Flash, que se recupera desde los servidores de Google cada vez que un cliente carga la aplicación.

La mayor parte del código utilizado para la ejecución de la aplicación de Google Maps para Flash permanecerá en la biblioteca del tiempo de ejecución del API de Google Maps para Flash independiente. Esto permite al equipo de Google realizar mejoras, solucionar errores y hacer modificaciones en las funciones de la biblioteca principal sin que los usuarios tengan que volver a compilar la aplicación.

El API de Google Maps para Flash utilizado en la aplicación, al igual que el API de Google Maps para JavaScript, requiere el uso de una clave de desarrollador disponible de forma gratuita. La clave del API se compila en el archivo SWF y debe corresponder al dominio en el que se aloje el archivo SWF, que no tiene por qué ser necesariamente la ubicación del archivo HTML alojado.

Las aplicaciones Flex se definen en las declaraciones `MXML`. Por lo general, se debe proporcionar estas definiciones de aplicación `MXML` en la raíz del directorio de código fuente. De forma predeterminada, FlexBuilder crea un directorio `src` en el proyecto para colocar el código fuente y los archivos `MXML` deben residir en la raíz de ese directorio.

La declaración `MXML` define elementos de la interfaz de usuario en una aplicación Flex, mientras que el código `ActionScript` integrado en la etiqueta `<mx:Script>` define acciones sobre dichos elementos de la interfaz de usuario. En el caso más sencillo, sólo se tiene que declarar un objeto `com.google.maps.Map` en `MXML` e inicializar el mapa con código `ActionScript`.

Además del código `ActionScript`, las aplicaciones de Google Maps en Flash requieren un framework de interfaz de usuario para visualizar el mapa y cualquier elemento asociado de la interfaz de usuario en una página web. En el framework de Flex, estos componentes de la interfaz de usuario se especifican en una declaración `MXML`. Una declaración `MXML` se compone de un archivo de configuración con el sufijo `.mxml`. Este archivo `MXML` reside normalmente en la raíz del directorio de desarrollo de `ActionScript`.

Para visualizar el mapa de Flash en una página web, se necesita al menos una declaración `MXML`. Todas las declaraciones `MXML` requieren un componente `<mx:Application>` raíz. Además, el componente `<mx:Application>` también registra el prefijo `mx` para hacer referencia a los componentes estándar de Flex.

El objeto `<mx:Script>` es un componente de Flex que contiene una referencia al código `ActionScript`. Dado que `MXML` es una variante de `XML`, se tiene que informar al analizador `MXML` de que ignore el código `ActionScript` en este archivo mediante el uso de los delimitadores `'<![CDATA[][' y ']]>'`.

III.2 Conexión entre el cliente web y la base de datos.

Flex permite generar automáticamente servicios o aplicaciones que se conectan a bases de datos. En este caso se utilizó una aplicación basada en PHP. Cuando se crea esta aplicación, se generan un conjunto de ficheros que controlan todo el proceso de comunicación entre la aplicación principal (*AudingSpatial.mxml*) y la base de datos. Los ficheros generados y sus funciones se muestran en la siguiente tabla. El directorio donde se generan estos ficheros siempre se localiza según este criterio [`web_server_root\nombre del proyecto_debug`]. En este directorio se crean además un subdirectorio PEAR, que es un framework y un sistema de distribución para la codificación de componentes en PHP, donde se guarda un fichero PEAR XML que se ocupa de las funcionalidades relacionadas con los ficheros XML. Y un subdirectorio HISTORY, donde se guarda el historial del manejo de la aplicación. En el caso de esta aplicación los archivos se generaron en el directorio: `...htdocs\AudingSpatial\bin-debug\`.

Tabla 4: Ficheros generados por la aplicación de conexión a base de datos.

Ficheros	Funciones
Projauding.php	Contiene todas las sentencias de consulta manejo y actualización a la base de datos. Es el medio principal de comunicación entre la base de datos y la aplicación.
AudingSpatialconn.php	Crea la conexión a la base de datos: nombre de la base de datos, usuarios, contraseñas etc.
Functionsinc.php	Optimiza el tratamiento de los valores de la base de datos.
XmlSerializer.class.php	Contiene las funciones de lectura y generación de documentos XML.
Projauding.swf	Aplicación de gestión de las tablas de la base de datos que puede ser ejecutada en FlashPlayer y visualizada en los navegadores.
Projauding.html	Documento web donde se visualiza y consulta las tablas de la base de datos.
AC_OETags.js	Genera una plantilla HTML con funcionalidades de chequeo de versiones y alojamiento de los ficheros SWF. Para lograr esto las plantillas HTML llaman a un método de JavaScript llamado <code>AC_FL_RunContent()</code> , que se encarga de alojar las aplicaciones de Flex en el documento HTML y de chequear la versión de FlashPlayer que el navegador está usando en ese momento para verificar su compatibilidad con el SWF generado.
SRC\Projauding.mxml	Ámbito de desarrollo de estructuración de la aplicación que maneja las tablas de la base de datos.

SRC\ ProjaudingConfig.as	Clases que controlan las funcionalidades de la aplicación Projauding.mxml
SRC\ ProjaudingScript.as	Clases que controlan las funcionalidades de la aplicación Projauding.mxml

Fuente: elaborado por el autor

De esta forma queda configurada la conexión con la base de datos. A partir de aquí cualquier consulta o petición que se quiera ejecutar, solo se debe modificar el fichero projauding.php para generar métodos de consultas personalizadas.

III.3 Estructura y personalización del Cliente Web.

Esta aplicación está dirigida a la difusión comercial de los proyectos que ejecuta la empresa y pueden ser diferenciados en función de dos tipos usuarios fundamentalmente. En primer lugar a personas familiarizadas con el mundo de la ingeniería, ya sean administraciones públicas, ayuntamientos y empresas que se desenvuelven profesionalmente dentro de este mundo, y que demandan este tipo de servicio. Y en segundo lugar usuarios particulares que no están familiarizados con los servicios que ofrecen las empresas de ingenierías, pero que pueden convertirse en demandantes de estos productos.

En este sentido, la filosofía de cara al usuario en general, consiste en mostrar una herramienta eficiente, intuitiva y amigable, de consulta y búsqueda de los proyectos en los que esté interesado. Organizar el proceso de búsqueda en función de la estructura organizativa de la empresa contribuye a proyectar una imagen de profesionalidad diversidad y experiencia en el sector.

A estos requerimientos se les añade la capacidad que debe tener la aplicación que permita al usuario interactuar con el mapa, o sea, funciones de zoom, pan, etc. En el caso de esta aplicación estas capacidades las ofrece el API de Google Maps, lo que constituye una ventaja puesto que poseen una de las tecnologías de servicio de mapas e imágenes más rápidas y eficientes que existen.

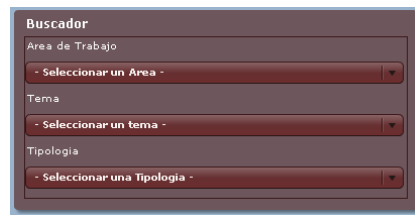
Partiendo de estas premisas se diseñó una herramienta de búsqueda consistente en un panel con tres objetos ComboBox que permiten al usuario navegar refinando su búsqueda en tres niveles hasta llegar a una lista de los proyectos que este acorde con sus intereses. Una primera búsqueda se refiere al Área de Trabajo, luego al Tema de Trabajo dentro de esta Área, y finalmente la Tipología de los proyectos que le interesaría consultar.

Una vez alcanzado este punto, otro panel aparecería ilustrando una lista con los títulos de los proyectos asociados al criterio de búsqueda establecido en el paso anterior y sobre esta lista ya se puede entonces acceder a información más detallada de cada uno.

Cada uno de estos proyectos que aparecen en la lista desplegada tiene una localización geográfica asociada, la cual se representa en el mapa en el mismo momento en que aparece el panel mencionado. Ambos objetos, el panel y los marcadores que indican la posición espacial de cada proyecto, están relacionados. Tanto desde cada ítem de la lista, como del marcador representado en el mapa, se puede acceder a la información específica del proyecto seleccionado, así como la posibilidad de descargar una ficha técnica referente al mismo.

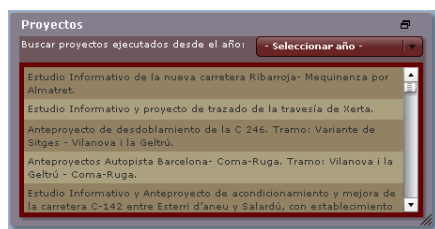
El funcionamiento general quedaría resumido en los siguientes pasos.

1. Inicialización de la aplicación. (Accediendo a la URL pública de la web desde un navegador)
2. Búsqueda mediante el panel Buscador. *(Figura 4. Panel buscador)*
 - a) Primer nivel de búsqueda: selección del área de trabajo el tema seleccionado en los dos pasos anteriores.
 - b) Segundo nivel de búsqueda: selección del tema de trabajo asociado al área seleccionada anteriormente.
 - c) Tercer nivel de búsqueda: tipología de los proyectos relacionados con el área y



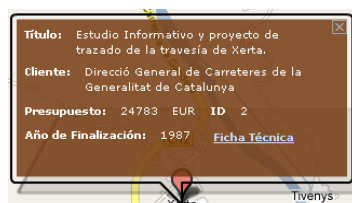
3. Se despliega el panel con la lista de los proyectos que responden el criterio de búsqueda anterior, así como los marcadores que indican su posición espacial. Aparece otra opción de búsqueda de proyectos a partir de una fecha. Esta opción hace un filtro mostrando solo los proyectos ejecutados a partir del año seleccionado hasta la actualidad, dentro de los criterios seleccionados en los pasos anteriores.

Figura 5. Panel proyectos (elaborado por el autor)



4. Click sobre un ítem de la lista, o sobre un marcador: se despliega una ventana sobre el marcador con la información referente a ese proyecto seleccionado.

Figura 6. Ventana de información (elaborado por el autor)



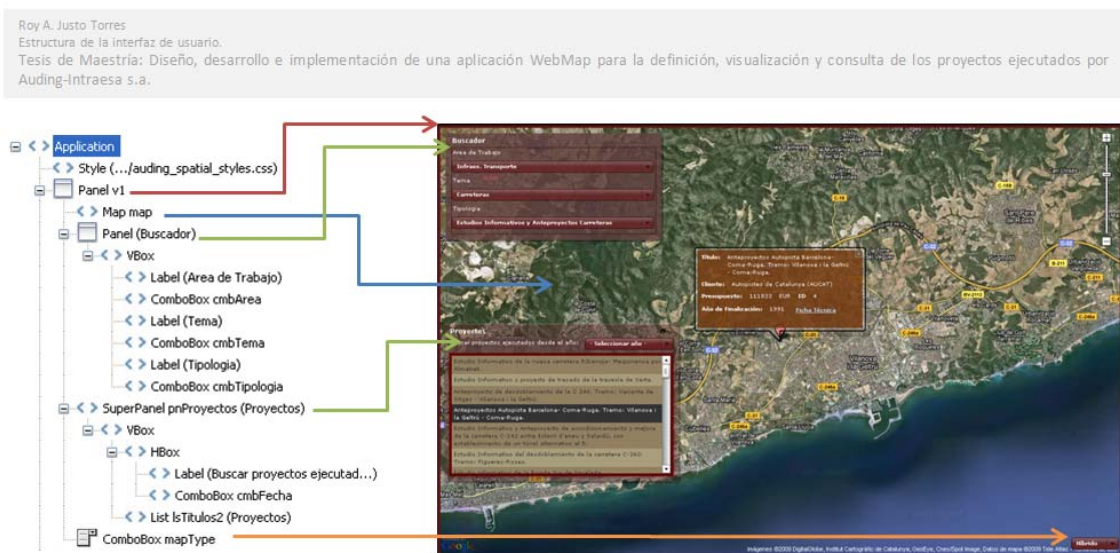
El diseño de las interfaz desarrolladas en Flex se basa fundamentalmente en un conjunto de objetos añadidos a modo de etiquetas en un documento con extensión (*.mxml). MXML es un lenguaje que describe interfaz de usuarios, crea modelos de datos y tiene acceso a los recursos del servidor, del tipo RIA (Rich Internet Application). Tiene una estructura similar a HTML, pero con una sintaxis menos ambigua que proporciona una gran variedad e inclusive permite extender etiquetas y crear sus propios componentes.

Los objetos añadidos en Flex deben seguir o respetar una jerarquía. Existen objetos contenedores y objetos controladores o controles. Los primeros funcionan como contenedores de otros objetos y su función fundamental es organizar la forma en que se muestran los datos que contienen los objetos controladores, o sea, las imágenes, tablas, textos, gráficos, botones etc.

El proyecto AudingSpatial contiene dos documentos mxml, uno es la aplicación principal y el segundo es una ventana de información que se muestra cuando es llamada una función dentro del programa.

Como se observa en la figura la aplicación principal contiene un gran panel que a su vez contiene el resto de los objetos, fundamentalmente tres. Uno está relacionado con la herramienta de búsqueda contenida en un objeto de tipo panel. Otro está relacionado con un contenedor panel donde se muestra una lista de los proyectos que responden a un criterio de búsqueda determinado, y finalmente un objeto ComboBox que contiene los diferentes tipos de mapas que pueden ser mostrados.

Figura 7: Estructura de la interfaz de usuario.



Fuente: elaborado por el autor.

El panel Buscador está formado por tres objetos etiquetas (*mx:Label*), y tres objetos *mx:ComboBox*, que están dentro de un Objeto *mx:VBox* encargado de organizar verticalmente los objetos que contiene.

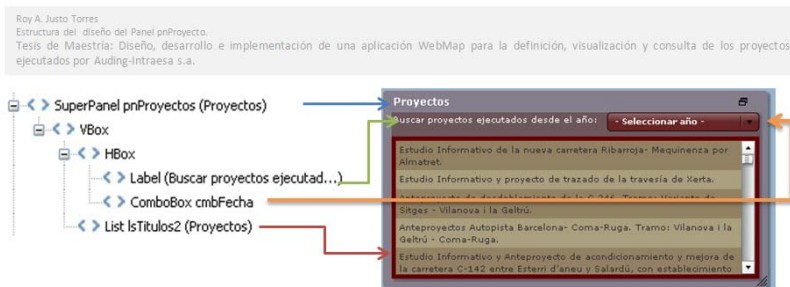
Figura 8: Estructura del diseño del panel Buscador.



Fuente: elaborado por el autor.

El SuperPanel pnProyectos es una extensión del objeto panel, al cual se le añade las propiedades de ser desplazable, expansible y que se pueda colapsar. Este además contiene un objeto de tipo *mx:List* donde se muestra la lista con los títulos de los proyectos que responden a un criterio de búsqueda determinado. Y un objeto *mx:ComboBox* que contiene una lista de los años a partir de los cuales se puede filtrar para visualizar en la lista los proyectos ejecutados después del año seleccionado.

Figura 9: Estructura del diseño del panel Proyectos.

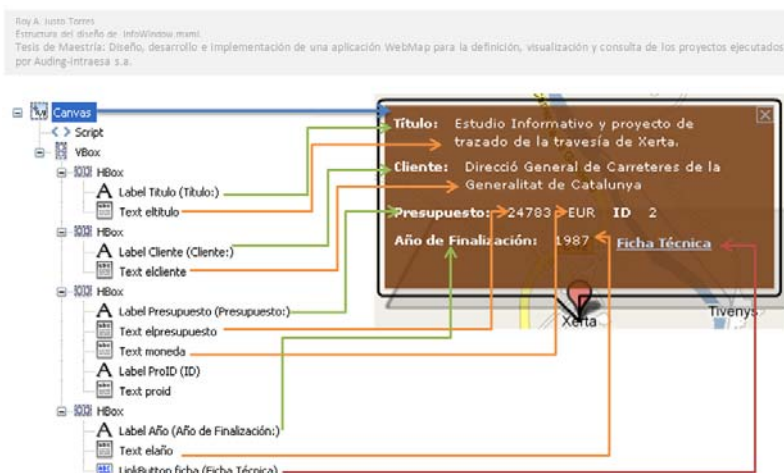


Fuente: elaborado por el autor.

La ventana de información se diseñó en función del contenido que se quería mostrar. Esencialmente contiene etiquetas o *mx:Label* que nombran la información que se muestra interactivamente en los objetos de tipo *mx:Text*. Estos objetos se llenan desde la base de datos, de forma interactiva en función de las consultas que realice el usuario. También hay un objeto de tipo *mx:LinkButton* que permite descargar ficheros PDF que se encuentran en un directorio externo. Todos estos componentes u objetos están contenidos en componentes *mx:Hbox* o *mx:Vbox* en función del sentido de orientación en que se quiera estructurar la información desde el punto de vista visual, o sea, de forma vertical o de forma horizontal. Y el contenedor general es un objeto de tipo *mx:Canvas*. Canvas es un tipo de contenedor que define un área rectangular donde se pueden alojar otros contenedores o controladores. El

proceso mediante el cual este objeto es invocado y visualizado mediante el API de Google-Maps para flash será explicado en el Capítulo IV.

Figura 10: Estructura del diseño de InfoWindow.mxml.

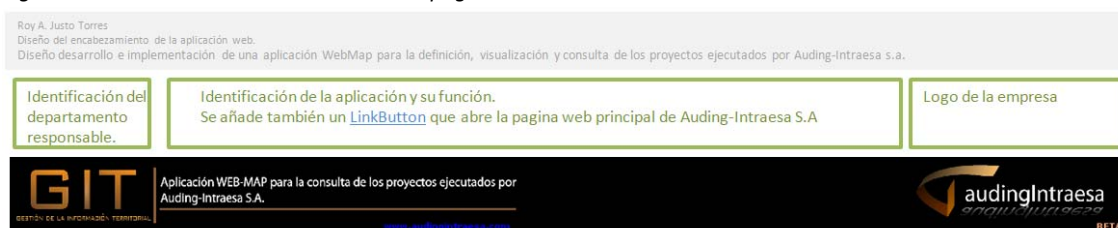


Fuente: elaborado por el autor.

Cada objeto insertado en la aplicación tiene asociado un estilo. La definición de cada uno de estos estilos se encuentra en un documento con extensión CSS denominado css_styles.css.

La selección de colores y estilos en general se basa fundamentalmente en la imagen corporativa de la empresa. En la web de Auding, se puede acceder a folletos de promoción de la empresa y de cada uno de los departamentos. Estos folletos en conjunto con otros materiales de promoción y marketing que está desarrollando la empresa contienen los conceptos básicos de la imagen corporativa que la empresa promueve. En este sentido fueron seleccionados los diferentes colores y estilos de la aplicación, así como el diseño del encabezamiento.

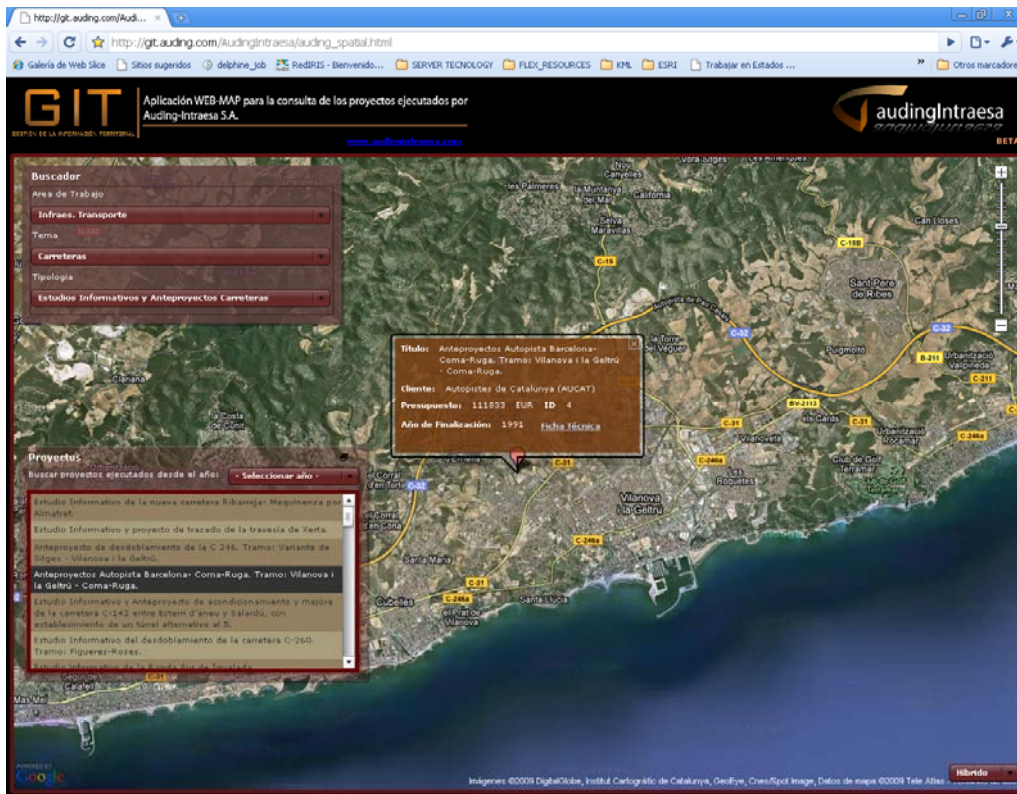
Figura 11: Diseño del encabezamiento de la página web.



Fuente: elaborado por el autor.

De esta forma quedo diseñada la interfaz de usuario. Como se muestra en la figura existe una homogeneidad en el diseño. Se conserva la imagen corporativa aunque se puede identificar la identidad propia de este producto. Desde el punto de vista funcional es intuitiva, amigable, y permite un intercambio eficaz con el usuario.

Figura 12: Diseño de la interfaz de usuario de la aplicación.



Fuente: elaborado por el autor.

CAP IV. DESARROLLO E IMPLEMENTACION

DE LA APLICACIÓN

C Contenido del capítulo:

- *Estructura y funcionamiento de la aplicación.*
- *Modelo lógico de desarrollo*
- *Características de puesta en marcha y rendimiento de la aplicación.*

IV.1 Estructura y funcionamiento de la aplicación.

Los proyectos realizados en Flex son a grandes rasgos, un compendio de objetos o componentes estructurados (MXML) que configuran la interface y un conjunto de rutinas escritas en ActionScript que crean y gestionan el comportamiento de la aplicación. Cada proyecto está compuesto, tradicionalmente, por cuatro carpetas.

- **BIN-DEBUG**: que es la abreviatura de binary debugging, y que contiene el código de compilación (el fichero SWF), y para las aplicaciones web contiene también un fichero contenedor HTML con sus ficheros suplementarios.
- **HTML-TEMPLATE**: contiene plantillas HTML que generan los ficheros HTML siempre y cuando se trate de una aplicación WEB.
- **LIBS**: es la abreviatura de Libraries, o bibliotecas, que contiene el conjunto de código compilado que se usa en la aplicación.
- **SRC**: o source, que contiene todo el código fuente, ya sean ficheros *.mxml*, ficheros *.as* o ficheros descriptores de aplicaciones para aplicaciones AIR.

Además de estas carpetas y ficheros que crea por defecto, se le pueden añadir más clases, componentes, y ficheros en función de la complejidad de cada proyecto que se desee realizar.

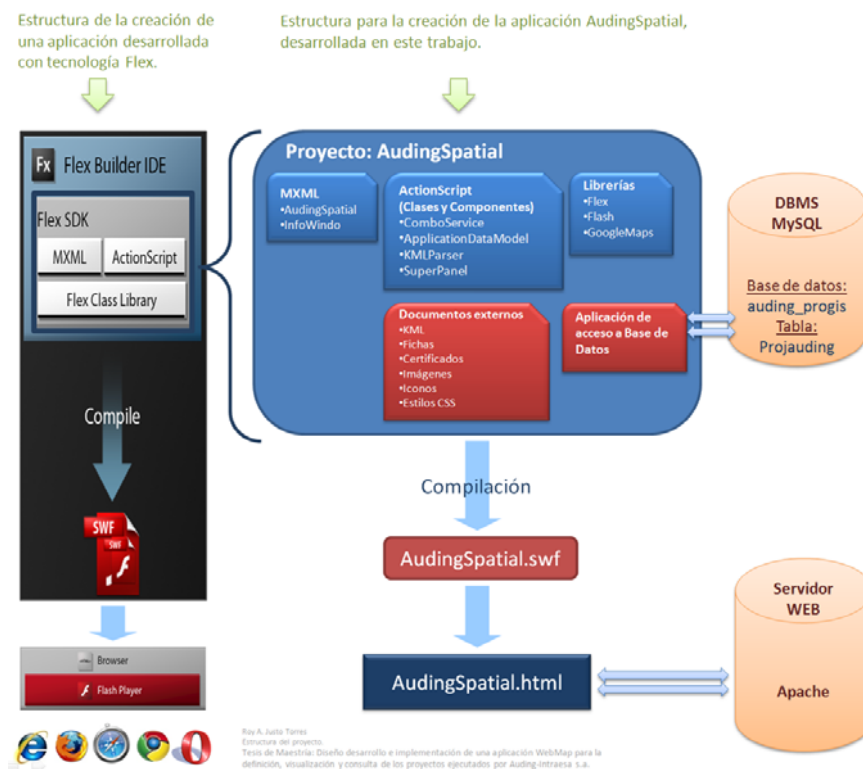
Una vez acabados estos proyectos son compilados para crear un fichero con extensión SWF que puede ser alojado en un documento HTML para ser publicado en la WEB. Estos ficheros

SWF son ejecutados por FlashPlayer, una aplicación que es compatible y está instalada en los navegadores más usados del mercado.

En proyecto *AudingSpatial* está constituido, además de las carpetas y ficheros que crean por defecto, por un conjunto de elementos que fue necesario añadir para obtener las funcionalidades deseadas (Figura 13). Para desarrollar esta aplicación había que partir de un conjunto de premisas básicas:

- Necesidad de conectar la aplicación con una base de datos externa.
- Localizar espacialmente los proyectos y relacionar su ubicación con la información asociada a cada uno de ellos.
- Hacer que ficheros externos a la aplicación y a la base de datos pudieran ser consultados y adquiridos desde la aplicación en tiempo de ejecución.

Figura 13. Estructura para la creación de la aplicación *AudingSpatial* según la lógica de desarrollo de Flex.



Fuente: elaborado por el autor.

Una vez creada y estructurada la base de datos, desarrollada la aplicación que se conecta y gestiona el intercambio de información entre la base de datos y la aplicación WebMap (Capítulo III), y diseñada la interface de usuario; se procedió a dotar de funcionamiento a la

aplicación. El proyecto fue estructurado como se observa en la *figura 13* y está formado por los siguientes elementos:

Tabla 5. Componentes de la aplicación *audinig_spatial.mxml*

Ficheros	Descripción
AudingSpatial.mxml	Es la aplicación principal. En este fichero está definida toda la interface de usuario, así como las funciones de los objetos y componentes que contiene.
InfoWindow.mxml	Es una Ventana de información asociada a los Marcadores o puntos geográficos donde están localizados los proyectos realizados por la empresa. Desde esta ventana se accede a a información relacionada con cada proyecto y se puede descargar tanto las fichas técnicas de los mismos, así como los certificados de estos en caso de que lo tengan.
ComboService.as	Rutina que se encarga de hacer las peticiones de información a la base de datos con las que se llenan los combos del panel de búsqueda.
ApplicationDataModel.as	Rutina que le da formato y hace accesible a cualquier parte de la aplicación, a los datos recibidos desde el servidor.
SuperPanel.as (Version 1.0)	Es una extensión del componente mx:panel de flex, desarrollada con el objetivo de dotarlo de mas funcionalidades y propiedades que las que tiene por defecto. Esta hace que un panel sea minimizable, maximizable, extensible y desplazable. Fue creada por Wiestse Veenstra el 22 de febrero del 2007 y está disponible en su WEB http://www.wietseveenstra.nl . Ha sido modificada por el autor de este trabajo en función de los intereses particulares de esta aplicación.
FICHAS	Esta es la carpeta donde se alojan todos los ficheros PDF de las fichas técnicas de cada uno de los proyectos
CERTIFICADOS	Esta es la carpeta donde se alojan todos los ficheros PDF de los certificados de cada uno de los proyectos en caso de que hayan tengan alguno asignado.
IMAGENES	Carpeta donde se guardan las imágenes que se usan en la aplicación.
Iconos	Carpeta donde se guardan los iconos que se usan en la aplicación.
Styles	Carpeta donde se guarda el fichero CSS de estilos de cada componente de la aplicación.

Fuente: elaborado por el autor.

Estos componentes están relacionados entre sí y tienen asociado funciones específicas que responden a eventos que ocurren cuando se ejecuta alguna acción sobre los objetos de la interface de usuario. Estos eventos pueden ser iniciados por el usuario, o desencadenados por otros eventos.

Cuando el usuario accede a esta aplicación para hacer una búsqueda o consulta, la herramienta que encuentra es un panel de búsqueda que tiene tres combos a partir de los cuales se establecen los criterios de la búsqueda. Cuando se inicializa la aplicación el primer

combo se llena con los valores únicos del campo que guarda las diferentes áreas de trabajo de la empresa. Al seleccionar uno de estos ítems, la aplicación pide a la base de datos los temas de trabajo asociados al área de trabajo seleccionada. Con estos valores se llena el segundo combo de temas de trabajo. Por último al seleccionar un tema de trabajo de este combo se realiza la misma operación pero con otros parámetros que devuelve las diferentes tipologías asociadas al tema de trabajo anteriormente seleccionado. Con estos se llena este último combo. Una vez establecido estos criterios de búsqueda solo se necesita seleccionar una de las tipologías desplegadas en el último combo, y entonces pasan dos cosas: aparece un nuevo panel con una lista con los títulos de todos los proyectos que responden a ese criterio de búsqueda y aparecen en el mapa las localizaciones asociadas a cada uno de estos proyectos.

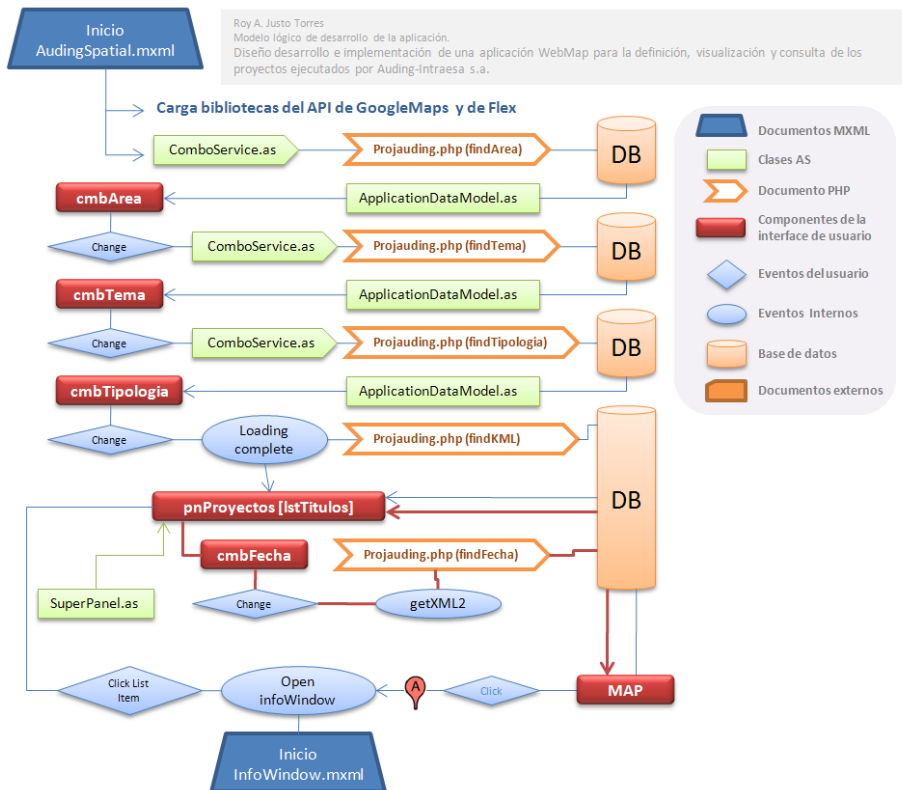
La información espacial está localizada en la tabla de la base de datos. En esta parecen dos columnas (LAT y LNG) donde se guardan los valores de latitud y longitud geográfica, asociadas a cada uno de los proyectos.

Desde este nuevo panel donde se visualiza la lista de los títulos de cada proyecto y aparece un nuevo combo, con una lista de los años en que se iniciaron los proyectos visualizados, para filtrar los proyectos desarrollados a partir del año que se seleccione es el combo mencionado. Desde la lista se puede invocar una ventana que contiene información específica del proyecto seleccionado referente al título, cliente, presupuesto, y año de finalización del proyecto consultado. También esta ventana tiene un botón que permite descargar la ficha técnica asociada al proyecto seleccionado. Esta ventana puede ser invocada también desde los marcadores que representan la localización de los proyectos.

IV.2 Modelo lógico de desarrollo

El modelo de programación seguido es bastante lineal. Se basa principalmente en rutinas que consultan la base de datos filtrando información hasta llegar a resultados deseados (Figura 14). Este comportamiento implica programación tanto en el cliente como en el servidor.

Figura 14. Modelo lógico de desarrollo de la aplicación



Fuente: elaborado por el autor

La programación desde el servidor se realiza en el fichero Projauding.php. En este fichero, por defecto genera cuatro métodos generales de consulta a la base de datos: FindAll (buscar), Insert (insertar), Update (actualizar), Delete (borrar) y Count (contar). Para personalizar las consultas que se necesitaban se crearon cinco métodos y se modificó el método FindAll para una consulta en concreto. En la siguiente tabla se señalan cada uno de estos métodos y su descripción.

Tabla 6. Métodos creados dentro del fichero Projauding.php, para la consulta personalizada a la base de datos.

Métodos	Descripción
FindArea	Este método pide a la base de datos cada uno de los valores únicos que existen en el campo de AREA_TRABAJO de la base de datos.
FindTema	Este método pide a la base de datos cada uno de los valores únicos que existen en el campo de AREA_TRABAJO filtrados por el campo AREA_TRABAJO.
FindTipologia	Este método pide a la base de datos cada uno de los valores únicos que existen en el campo de TIPOLOGIA filtrados por el campo TEMA_TRABAJO.
FindKML	Este método devuelve de la base de datos un XML que contiene los valores los campos LAT, LNG y TITULO, filtrados por el campo TEMA_TRABAJO y TIPOLOGIA
FindAll (\$campo)	Este método recupera todos los registros de la base de datos filtrado por cualquier campo. Para esto se define una variable \$campo, dentro del método, y así puede tomar cualquier valor. Este fue llamado filtrando por el campo PROYECTO_ID.
FindFecha	Este método devuelve de la base de datos un XML que contiene los valores los campos LAT, LNG, TITULO y FECHA_INI filtrados por el campo TEMA_TRABAJO y TIPOLOGIA

Fuente: elaborado por el autor

Cuando desde la aplicación se llama a estos métodos, ellos hacen un pedido específico a la base de datos y esta les devuelve un fichero de transporte de datos XML con la información solicitada.

Estos métodos son invocados desde la aplicación mediante un componente específico de Flex diseñado para esto, mediante un *HTTPService*. Este componente se puede usar con cualquier tipo de tecnología de servidor. En este caso se usa con *PHP*. Con este componente se llama a la página *PHP* mediante el método *GET* si se quiere consultar datos de la base de datos, o con el método *POST* si se quiere enviar datos a esta. Como se trata solo de una aplicación de consulta, donde el usuario no puede actualizar o modificar la base de datos, solo se realizan peticiones usando el método *GET*. Casi todas las peticiones realizadas a la base de datos están desarrolladas en la clase *ComboService.as*. Esta clase recibe los ficheros XML que envía el servidor y los reformatea en E4X.

La especificación de ECMAScript para XML define un conjunto de clases y funcionalidad para trabajar con datos XML. Este conjunto de clases y funcionalidades se denomina *E4X*. ActionScript 3.0 incluye las siguientes clases de E4X: XML, XMLList, QName y Namespace.

Los métodos, propiedades y operadores de las clases de E4X se han diseñado con los siguientes objetivos:

- Simplicidad: siempre que sea posible, E4X facilita la escritura y comprensión del código para trabajar con datos XML.
- Coherencia: los métodos y la lógica que subyacen a E4X son coherentes internamente y con otros componentes de ActionScript.
- Familiaridad: los datos XML se manipulan con operadores conocidos, como el operador punto (.).

Posteriormente los convierte en colecciones de datos más fáciles de representar dentro de los componentes de la aplicación. Las colecciones que se usan son de tipo `ListCollection` y `ArrayCollection`.

Cuando los datos son recibidos son asignados a un modelo de datos (*ApplicationDataModel.as*). La función de este modelo es instanciar los datos de forma tal que se puedan usar desde cualquier parte de la aplicación manteniendo su integridad y actualizándose continuamente en la medida que se actualiza la información recibida desde el servidor. Mantiene los datos actualizados para toda la aplicación. Esto permite llenar continuamente los `ComboBox` de área de trabajo, tema de trabajo y de tipología, y la lista de títulos de proyectos, en la medida en que se van actualizando los datos como resultado de las consultas que realice el usuario.

Cuando se está inicializando la aplicación ya esta realiza un primer pedido a la base de datos usando el método *getAreas* definido en *ComboService.as* que llena el combo de *cmbAreas*. Así cuando la aplicación carga completamente ya e usuario puede acceder a este primer nivel de búsqueda. Cuando este selecciona un ítem de la lista que se despliega en este combo, ocurre el evento *Change*, o sea, un cambio de ítem. Este evento llama al método *getTemas* definido en la clase *ComboService.as* que hace el nuevo pedido a la base de datos y llena el combo *cmbTemas* con el resultado de este pedido, filtrando según el área de trabajo (ítem) que se seleccionó en el paso anterior. Así ocurre también en el caso del combo *cmbTipologia*, usando el método que le corresponde (*getTipologia*).

Llegado a este último nivel de búsqueda donde se selecciona un ítem de los desplegados en el combo *cmbTipologia* se invocan un evento que a su vez invoca dos procesos.

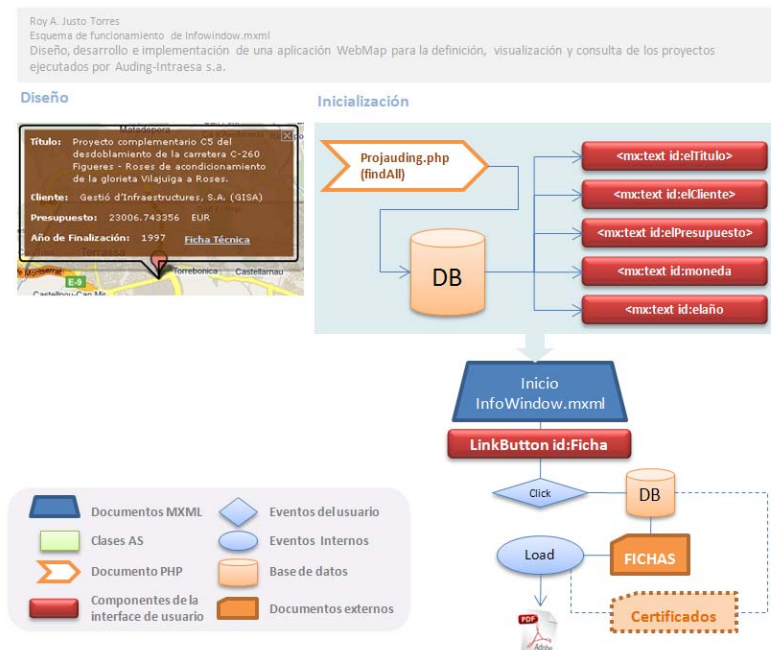
- 1- Se hace visible el componente `<myCom:SuperPanel>`
- 2- Se realiza un nuevo pedido a la base de datos mediante el método *findKML* definido en *projauding.php* con el cual se llena la lista que contiene el componente `<myCom:SuperPanel>`, y el combo *cmbFecha* que aparece también en este panel, con los resultados filtrados en función del ítem seleccionado.

Si el usuario decide implementar un nuevo filtro en la lista desplegada para obtener los títulos de los proyectos ejecutados a partir de un año determinado, este debe escoger en el combo *cmbFecha* el ítem que le interese. Este proceso invocaría un evento llamado *get XML2* que repetiría el proceso anterior para llenar la lista de proyectos pero esta vez mediando un método llamado *findFecha* dentro de *projauding.php*.

Llegado a este punto el usuario tiene acceso a una lista con los títulos de los proyectos y un conjunto de marcadores representados en el mapa. Para acceder a información sobre un proyecto en concreto puede hacerlo de dos formas: ya sea por interés geográfico accediendo desde el marcador, o por interés respecto a algún título de proyecto en específico que aparezca en la lista. En ambos casos cuando el usuario hace click, ya sea sobre el marcador o sobre algún ítem de la lista, se invoca un mismo evento dentro de la aplicación que realiza varias funciones.

Cuando se invoca desde la lista, este evento centra en el mapa y hace un zoom hacia el marcador que corresponde con el ítem seleccionado de la lista. También este evento ejecuta una función del API de GoogleMaps denominada *OpenInfoWindow*, el cual abre una ventana de información sobre el marcador seleccionado. El contenido personalizado de esta ventana es el fichero *InfoWindow.mxml*, o sea que una vez ejecutado *openinfowindow*, se invoca la inicialización del fichero MXML mencionado.

Figura 15. Esquema de funcionamiento de *Infowindow.mxml*.



Fuente: elaborado por el autor.

Como se muestra en la figura 15 esta aplicación en cuando se inicializa automáticamente hace un pedido a la base de datos a través del método *findAll* desarrollado en *Projauding.php*, que devuelve la información correspondiente al ítem o marcador seleccionado. Esta información se muestra en los componentes *mx:Text* que les corresponde. En la estructura de esta ventana de información también existe un componente de tipo *mx:LinkButton* que accede a la base de datos en busca del *URL* que guarda el campo *FICHAS* de la base de datos y que apunta a un directorio externo donde se guardan los ficheros en formato *PDF* de las fichas técnicas de cada proyecto. La función que invoca este *LinkButton* permite a los usuarios descargar estos ficheros.

IV.3 Características de la puesta en marcha y rendimiento de la aplicación.

Como se observa en la primera figura de este capítulo, una vez que todo el proceso de desarrollo esta completado y comprobado que todo funciona correctamente, se pasa a la compilación de la aplicación. Cuando esto sucede se genera un fichero son extensión SWF. Este es un formato de archivo de gráficos vectoriales. Suelen ser suficientemente pequeños como para ser publicados en la WEB en forma de animaciones o applets con diversas funciones y grados de interactividad.

Los archivos SWF son compilados y comprimidos a partir de archivos editables con los que se ha trabajado en Flex. El objetivo principal del formato SWF es crear archivos pequeños pero que permitan la interactividad y que funcionen en cualquier plataforma, aún sobre un ancho de banda reducido (cómo un navegador web conectado a través de un módem). El plugin que permite reproducir ficheros SWF está disponible en Adobe Systems para diferentes navegadores y diferentes sistemas operativos, incluido Microsoft Windows, Apple Macintosh y Linux. Este plugin está instalado aproximadamente en un 98% de los ordenadores de los internautas.

Una vez compilada la aplicación se alojó automáticamente en un documento o plantilla HTML, el cual fue colocado en el directorio publico de nuestro servidor web Apache. El acceso público a esta aplicación se hace mediante un dominio adquirido por Auding-Intraesa para este propósito. La aplicación puede ser consultada en este directorio http://git.auding.com/AudingIntraesa/auding_spatial.html.

En sentido general el rendimiento es bueno, aunque mejorable. El servicio está siendo ejecutado, en esta fase inicial, desde una maquina virtual alojada en el servidor de la empresa. Sin embargo es interés de la empresa alojar definitivamente esta aplicación, una vez terminada, en el servidor principal para optimizar su rendimiento.

En el apartado de recomendaciones de este informe se mencionan las limitantes actuales y las soluciones propuestas por el autor para optimizar el funcionamiento.

La tecnología empleada permite un ejercicio de mantenimiento muy ventajoso. La aplicación pude ser modificada cuantas veces se desee sin que afecte el servicio web. Cuando las modificaciones estén listas solo se necesita compilar nuevamente la aplicación, alojándola en el directorio publico reescribiendo el documento HTML existente. Este proceso solo tarda unos segundos. Este sería el tiempo que se necesitaría para actualizar el servicio.

El mantenimiento de las bases de datos es independiente de la aplicación. De momento se usa la herramienta de gestión del MySQL mencionada en el Capítulo II: phpMyAdmin. Esta interfaz permite manipular los datos de las tablas. En este sentido se señala a modo de recomendaciones la creación de una aplicación de autogestión de la base de datos,

sincronizada con la aplicación de consulta de forma tal que los encargados de alimentar esta base de datos tengan una herramienta fácil de gestión y administración de la misma.

Conclusiones

Este trabajo perseguía objetivos específicos referentes a tres niveles principalmente: un nivel relacionado con la selección de la plataforma tecnológica y el diseño de la estructura de la base de datos, un segundo nivel relacionado con el diseño funcional y visual de la interfaz de usuario, y finalmente en el desarrollo de una estructura lógica de programación que fuera capaz de cubrir con las exigencias de funcionamiento señaladas por la empresa. En este sentido se puede concluir lo siguiente:

- En cuanto a la selección tecnológica y la implementación de la base de datos se puede concluir que existe una sincronía entre las plataformas utilizadas optimizando el flujo de información entre el servidor y el cliente, con una alta calidad en su representación de cara al usuario final.
- La estructura de la base de datos cubre las exigencias tanto de la aplicación como de los intereses corporativos de difusión de su información.
- La interfaz de usuario es sencilla, intuitiva, amigable y eficiente desde el punto de vista funcional. Desde el punto de vista visual su diseño es atractivo y transmite la identidad corporativa.
- La lógica de programación empleada es lineal, limpia, modular y sólida. Cubre las exigencias corporativas y permite posteriores mejoras y modificaciones tanto en el diseño como en el funcionamiento, empleando un mínimo de cambios en la estructura del código escrito.
- El mantenimiento de la aplicación es ajeno al funcionamiento de la misma. La tecnología empleada permite modificar la aplicación cada vez que se estime conveniente en un entorno de prueba independiente. Una vez validados los cambios puede ser publicada la actualización requiriendo muy poco tiempo.

Recomendaciones

Las recomendaciones que se plantean en este trabajo responden a dos directrices fundamentales. Una está relacionada con las necesidades actuales de la empresa que no se pudieron cubrir en el tiempo destinado para el desarrollo de este proyecto, y otras con respecto al mejoramiento del funcionamiento de la aplicación en general. En este sentido las recomendaciones que se sugieren son las siguientes.

- Elaborar una base de datos única y centralizada de gestión de la información relacionada con los proyectos ejecutados por Auding-Intraesa S.A.
- Elaborar una herramienta amigable de gestión de esta base de datos a un nivel corporativo, donde se gestione toda la información relacionada con los proyectos (pliegos de licitación, facturas, documentos de los proyectos, hojas de cierre, certificados etc.), estableciendo restricciones tanto del personal autorizado a la gestión de la mismas, así como en cuanto a la información susceptible a ser publicada o no.
- Añadir más capacidades de consulta de la base de datos desde la aplicación, como consultas de proyectos que estén en un rango de años seleccionado por el usuario, visualizar solo los proyectos de un municipio, país o comunidad autónoma determinada. Elaborar herramientas de visualización de gráficos estadísticos que representen información de proyectos elaborados por años, décadas, tendencias, etc.
- Usar la información espacial generada a partir de este proyecto para elaborar patrones espaciales influencia en el mercado, o estrategias de expansión empresarial.
- Personalizar los marcadores.

Bibliografía

LIBROS y ARTÍCULOS

1. Adobe Systems Incorporated (2008). Programming ActionScript.
2. Alaric Cole. (2008). Learning Flex 3. Getting up to speed with Rich Internet Applications. O'Reilly Media Inc.
3. Goodchild Michael F. (2009) Assertion and Authority: The Science Of User Generated Geographic Content.
4. Jack Herrington & Emily Kim. (2008) Getting Started with Flex 3. O'Reilly Media Inc.
5. Joshua Noble & Todd Anderson. (2008) Flex 3 CookBook. O'Reilly Media Inc.

WEB:

- a. Adobe Flex 3 Help Resource Center. <http://www.adobe.com/support/documentation/>
- b. Adobe Flex 3 Language Reference. <http://livedocs.adobe.com/flex/3/langref/package-summary.html>
- c. Apache Software Foundation Homepage. <http://www.apache.org/>
- d. API Google Maps para flash. Google Code. <http://code.google.com/intl/es-ES/apis/maps/documentation/flash/>
- e. MySQL 5.0 Reference Manual. <http://dev.mysql.com/doc/refman/5.0/es/index.html>
- f. PHPMyAdmin Documentation. http://www.phpmyadmin.net/home_page/
- g. Wietse Veenstra Blog. Factum Vicesimus Quartus. Exploring web x.0, innovation and RIA techniques. <http://www.wietseveenstra.nl/blog/>
- h. Wikipedia. La enciclopedia libre. <http://es.wikipedia.org/wiki/Wikipedia:Portada>

Anexos

ANEXO 1: ESTRUCTURA DE LA APLICACIÓN CON UNA REPRESENTACION DE GEOMETRIA COMPLEJA A PATIR DE FICHEROS KML.

En una primera fase del proyecto, uno de los requerimientos de la empresa consistía en representar además de puntos, también líneas y polígonos en el mapa de Google. Para esto se desarrolló una lógica de funcionamiento basada en la generación de la información espacial en formato KML. Conservándose así estos ficheros en un directorio externo al cual se accedía desde una URL guardada en la base de datos en la columna LOCALIZACION. En este campo se guardaron las URL que apuntaban a cada uno de los ficheros de localización de los proyectos. Este campo permitía vincular la información alfanumérica asociada a cada proyecto con su ubicación espacial.

Para poder mostrar en la interfaz de usuario la localización de los proyectos seleccionados según un criterio de búsqueda, había que realizar un pedido a la base de datos mediante un método llamado *getKML*. Este pedido devolvía una lista de cadenas de texto con esta estructura: '*http://dominio/KML/proyecto_id.kml*'. Cada uno de estos ficheros de localización estaba nombrado por el identificador de proyecto que le corresponde del campo PROYECTO_ID de la tabla. De esta forma se asociaba la información espacial con la alfanumérica. Una vez obtenida la lista de URL, se ejecutaba otra función (*loadFile*) que se encarga de descargar estos ficheros.

KML es un formato de archivo que se utiliza para mostrar información geográfica en navegadores terrestres como Google Earth, Google Maps y Google Maps para móviles. KML utiliza una estructura basada en etiquetas con atributos y elementos anidados y está basado en el estándar XML. Todas las etiquetas distinguen entre mayúsculas y minúsculas y deben aparecer exactamente como aparecen en la Referencia de KML establecida por Google. En esta referencia se indica qué etiquetas son opcionales. Dentro de un elemento determinado, las etiquetas deben aparecer en el mismo orden en el que aparecen en la referencia.

Los ficheros KML se pueden encontrar estructurados de diferentes formas. Para su representación en GoogleMaps deben estar estructurados al menos como se muestra en la siguiente figura, o sea, con estas etiquetas básicas que señala la referencia de Google para estos ficheros:

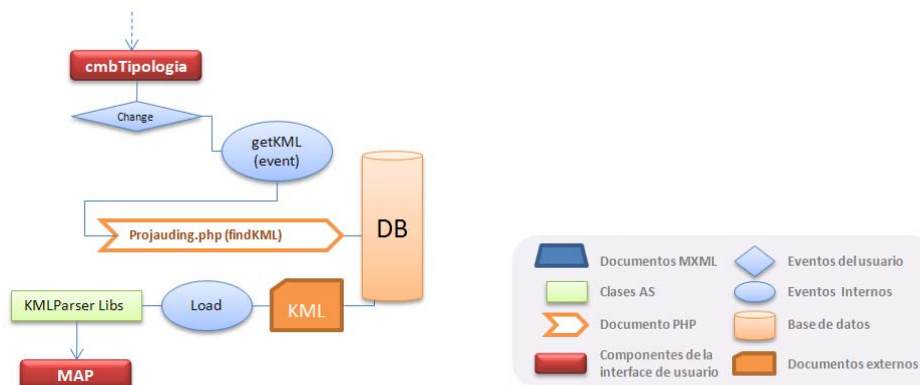
Ejemplo fichero KML que representa un punto en el espacio geográfico.


```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <kml xmlns="http://earth.google.com/kml/2.2">
3   <Folder>
4     <name>Placemarks</name>
5     <Placemark>
6       <name>Simple placemark</name>
7       <description>Attached to the ground</description>
8       <Point>
9         <coordinates>1.337300452237236,42.33734492723217,0</coordinates>
10      </Point>
11     </Placemark>
12   </Folder>
13 </kml>
    
```

Para que esta estructura sea interpretada por las librerías de FLEX, y puedan ser representadas en el mapa, el equipo del API para flash desarrolló unas librerías con un conjunto de clases que se llama *KMLParser*. Esta librería se encarga de identificar en los KML que tipo de geometría estos representan. Asociando y representando en el mapa todo tipo de geometría simple (puntos, líneas y polígonos. Funciona leyendo la información referente a las coordenadas y convertirlas en marcadores o *markers* (objetos del API de GoogleMaps para flash) que representan los puntos en el mapa, polylines (líneas), o polígonos. El esquema de funcionamiento quedaría representado como se muestra en la siguiente figura.

Esquema de funcionamiento de la aplicación usando ficheros KML para la representación espacial de los proyectos.



Fuente: elaborado por el autor.