

ESCOLA UNIVERSITÀRIA D'ENGINYERIA
TÈCNICA DE TELECOMUNICACIÓ LA SALLE

TREBALL FI DE CARRERA

ENGINYERIA TÈCNICA EN MULTIMÈDIA

MAPA ACÚSTICO

**asociado al tráfico marítimo
costa peninsular**

ALUMNE

PROFESSOR PONENT

Marcos de la Torre de la Torre

Daniel Arguedas Rodríguez

ACTA DE L'EXAMEN DEL TREBALL FI DE CARRERA

Reunit el Tribunal qualificador en el dia de la data, l'alumne

D. Marcos de la Torre de la Torre

va exposar el seu Treball de Fi de Carrera, el qual va tractar sobre el tema següent:

MAPA ACÚSTICO

**asociado al tráfico marítimo
costa peninsular**

Acabada l'exposició i contestades per part de l'alumne les objeccions formulades pels Srs. membres del tribunal, aquest valorà l'esmentat Treball amb la qualificació de

Barcelona,

VOCAL DEL TRIBUNAL

VOCAL DEL TRIBUNAL

PRESIDENT DEL TRIBUNAL

ABSTRACT

Este proyecto nace de la necesidad de crear un mapa acústico submarino de la costa peninsular, donde el usuario tiene varias herramientas para visualizar y escuchar el ruido asociado a las embarcaciones marinas, así como información acerca de la solapación de ondas entre embarcaciones y cetáceos.

El mapa acústico nos permite explorar las distintas áreas estudiadas y, mediante un sistema audiovisual, poder interpretar el nivel de contaminación acústica que sufren las costas de la península.

Esta herramienta es la que permite visualizar los estudios elaborados por el equipo de científicos del Laboratori d'Aplicacions Bioacústiques de la Universitat Politècnica de Catalunya. Han sido financiados por una Obra Social de "la Caixa".

RESUMEN

El proyecto consiste en el desarrollo de una herramienta multimedia, con la cual se genera un mapa acústico submarino, mostrando el nivel de contaminación del litoral peninsular.

También tiene otras vías de información, tanto con texto como una simulación de falso 3D en la que se ilustra la solapación de ondas entre cetáceos y embarcaciones. Y por último, también se dispone de las especificaciones del nivel de ruido de embarcaciones y cetáceos.

Este proyecto es, a su misma vez, parte de otro proyecto más grande, de una Obra Social de “la Caixa”, llamado “La Caixa a favor del mar” *Ref.1.*

El proyecto global se desarrolló por un equipo de científicos del LAB (Laboratorio de Aplicaciones Bioacústicas) de la UPC (Universitat Politècnica de Catalunya). Dicho laboratorio necesitaba el desarrollo de una aplicación con la que plasmar todos los datos recogidos en sus estudios, y aquí es donde entra en juego el desarrollo de la aplicación multimedia. Por lo tanto, podemos decir que este proyecto es la capa de presentación del estudio científico financiado por la citada Obra Social.

El objetivo del proyecto global es mentalizar al ciudadano de la realidad de nuestras costas, acercándonos a él mediante esta aplicación entre otros medios de comunicación, así como ruedas de prensa que se llevaron a cabo durante la campaña del proyecto, y en las que la aplicación sirvió de canal de presentación para ilustrar los estudios.

ÍNDICE

1. Introducción	7
1.1. Marco	7
1.2. Estado del arte	9
1.2.1. Aplicaciones actuales	9
1.2.2. Tecnologías del momento en el que se comenzó a desarrollar la aplicación	14
1.2.3. Tecnologías del presente	16
1.3. Descripción del problema	19
1.4. Solución propuesta	20
1.4.1. Solución al mapa acústico	21
1.4.2. Solución al método en que mostrar la solapación de ondas entre cetáceos y embarcaciones	22
1.4.3. Características de la aplicación a desarrollar	22
1.5. Perspectiva general	23
1.5.1. Fundamentos teóricos	23
1.5.2. Parte práctica	23
1.5.3. Resultados	24
1.5.4. Conclusiones y líneas de futuro	24
2. Fundamentos teóricos	25
2.1. Contaminación acústica	25
2.2. Solapación de ondas entre cetáceos y embarcaciones	27
2.2.1. Acerca de los cetáceos	28
2.2.2. Acerca de las embarcaciones	28
2.3. Tecnologías web	29
2.3.1. HTML	29
2.3.2. XML	30
2.3.3. Javascript	31
2.3.4. CSS	33
2.3.5. Web 2.0	34
2.4. Lenguajes de representación visual	35
2.4.1. FLASH	35
2.4.2. Representación de curvas Bézier cuadráticas	37
2.5. Desarrollo gráfico del mapa acústico	38
2.5.1. Adaptación de mapas con Photoshop	38
2.5.2. Adaptación de itinerarios	40
3. Parte práctica	42
3.1. Análisis de requisitos	42
3.2. Análisis funcional	43
3.2.1. Mapa peninsular	44
3.2.2. Mapas de áreas estudiadas	45
3.2.3. Mapa acústico	46
3.3. Diseño de la arquitectura	47
3.4. Toma de decisiones	49
3.4.1. Tecnología a emplear	49
3.4.2. Tecnologías desestimadas	50

3.4.3.	Tratamiento de imágenes	51
3.4.4.	Preparación de datos: De excel a xml	51
3.4.5.	Simulación: Funcionalidad interactiva para la representación del solapamiento de frecuencias entre embarcaciones y cetáceos.....	52
3.4.6.	Automatización de la carga de barcos	52
3.5.	Estructura/módulos de la aplicación	53
3.5.1.	Diagrama de clases	53
3.5.2.	Estructura de archivos XML.....	54
3.5.3.	Clases y métodos destacados	56
3.6.	Fase de desarrollo del proyecto	65
3.6.1.	Fase 1	65
3.6.2.	Fase 2	68
3.6.3.	Fase 3	73
3.6.4.	Fase 4	76
3.7.	Metodología de desarrollo	77
3.7.1.	Fase 1	77
3.7.2.	Fase 2	79
3.7.3.	Fase 3	81
3.7.4.	Fase 4	81
4.	Resultados	82
4.1.	Guión de la aplicación	82
4.2.	Rendimiento de la aplicación.....	86
4.3.	Coste en horas del proyecto	87
4.4.	Diagrama de las partes colaboradoras del proyecto	88
4.4.1.	Personal que colaboró en el proyecto	89
5.	Conclusiones y líneas de futuro	90
5.1.	Conclusiones	90
5.2.	Líneas de futuro	92
5.2.1.	Mejoras mediante marketing on-line	92
5.2.2.	Mejoras en la aplicación a nivel de desarrollo	93
5.2.3.	Posible método de desarrollo alternativo	95
6.	Bibliografía	96

1. INTRODUCCIÓN

1.1. Marco

Actualmente, y cada vez más, se hace uso de herramientas multimedia como canal de comunicación para llegar al usuario.

Incluso a nivel de enseñanza, el uso de aplicaciones multimedia es cada vez más frecuente, ya que ayudan a interactuar de una forma más dinámica con los alumnos. Hay muchas herramientas didácticas que emplean funcionalidades totalmente interactivas como medio de aprendizaje. Además pueden desarrollarse bajo diversos medios de difusión, como CD's, aplicaciones de escritorio y a través de páginas web.

Cuando hablamos de estudios científicos que lo que obtienen son únicamente datos y coordenadas, es apropiado utilizar un entorno interactivo, y en nuestro caso audiovisual, para poder plasmar todos los datos, de forma que el usuario pueda absorber dicha información de una forma sencilla, y a poder ser atractiva.

El proyecto desarrollado podemos catalogarlo dentro del ámbito multimedia, ya que se trata de una aplicación interactiva, en la que el usuario puede ver y oír el nivel de contaminación acústica que sufre la costa peninsular, a causa de la ruidos generados por embarcaciones, tanto de ocio como mercantiles.

En cuanto a la temática del proyecto, hablamos de un punto de unión entre tecnología y preservación del medioambiente. Donde los estudios recopilados y luego la aplicación multimedia, forman un canal con el que poder informar al usuario sobre la contaminación acústica de la costa peninsular.

El impacto medioambiental que está teniendo el tráfico marítimo fue el que motivó a crear esta Obra Social de "la Caixa", la cual ha financiado el proyecto. Eso permitió realizar los estudios pertinentes y recogida de datos para poder crear el mapa acústico. También ha servido para financiar indirectamente el desarrollo de esta aplicación. Aunque antes de finalizar los estudios, la Obra

Social dejó de financiar el proyecto, con lo cual, se quedó sin fondos para completar el mapa acústico de la costa sureste peninsular, concretamente fueron las comunidades autónomas de Andalucía (desde Málaga), Murcia y Valencia, que no se pudieron estudiar, y por lo tanto, de las que no se obtuvieron datos.

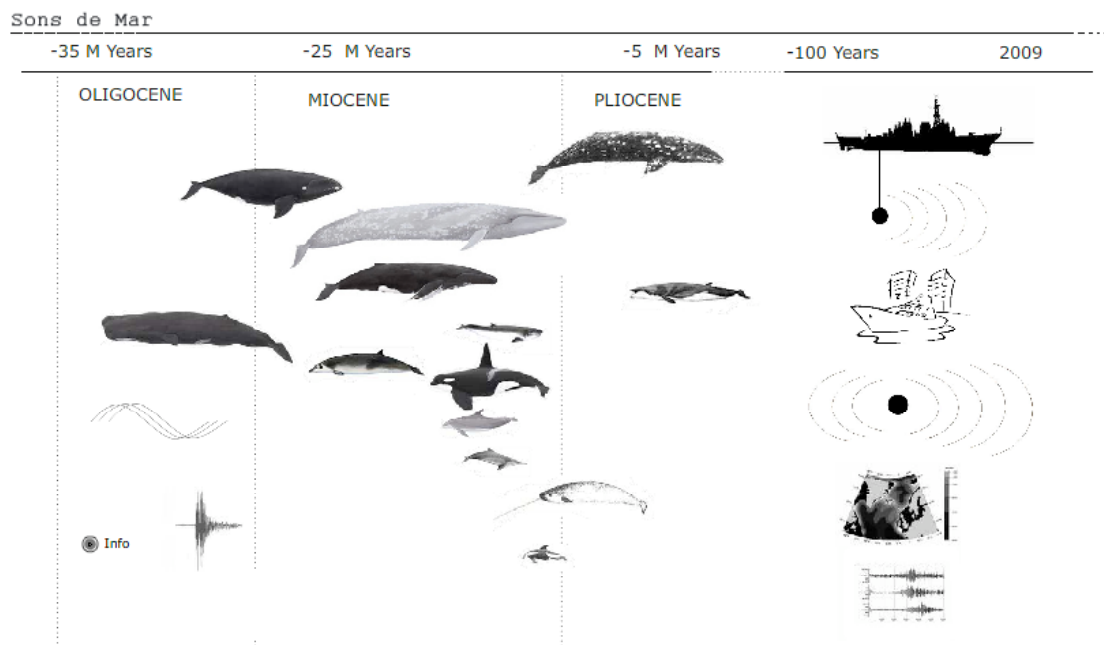
A través del mapa acústico, esta obra social tiene los siguientes objetivos:

- Concienciar al ciudadano sobre la realidad acústica del litoral y de sus efectos sobre ecosistemas y organismos marinos, en particular los cetáceos.
- Constituir una herramienta de gestión que permite proyectar y controlar el ruido asociado a la creación de nuevas rutas marítimas, como la finalidad de implementar autopistas marítimas para descongestionar la emanación de gases producidos por el tránsito marítimo.
- Desarrollar medidas de reducción en áreas donde se encuentra el foco de mayor afluencia acústica y la implementación de soluciones tecnológicas que combinan los intereses relativos a la actividad humana y la conservación de los ecosistemas marinos. La insonorización de motores de las embarcaciones para reducir el nivel de ruido producido, como la creación de rutas marítimas adecuadas, son claros ejemplos para menguar el problema de la contaminación en nuestras aguas.

También cabe destacar la vital importancia del impacto sobre la sociedad de este tipo de investigaciones. Ya que es como se consigue avanzar en una materia que no es nada sencilla, la del desarrollo de sistemas con éxito comercialmente hablando y, al mismo tiempo, minimizando el impacto medioambiental.

Desde el Laboratorio de Aplicaciones Bioacústicas LAB, se está colaborando con el Ministerio de Medio Ambiente para elaborar una normativa que regule el nivel de contaminación acústica submarina.

- **Sons de mar** *Ref.14*: Es una aplicación que muestra el rango temporal, desde los primeros cetáceos que conocemos hasta la fecha. Por medio de explicaciones y escucha de los sonidos que emiten, se quiere hacer comprender cómo los océanos y cetáceos, tienen que hacer frente a nuevas fuentes sonoras que afectan a las capacidades orientativas de los cetáceos, llegando incluso a provocarles la muerte, si las fuentes de ruido son muy cercanas y potentes. Estas fuentes pueden ser provocadas por embarcaciones, sónares y radares militares, experimentos científicos, etc, como se muestra en la imagen:



Herramienta on-line "Sons de Mar". Ref. 14

1.2.2. Tecnologías del momento en el que se comenzó a desarrollar la aplicación

Años atrás hubiese sido bastante más complejo desarrollar una aplicación de este tipo, debido a las pocas herramientas específicas que habían para crear aplicaciones interactivas. Actualmente se dispone de herramientas y librerías suficientes como para hacer viable, con relativa facilidad, una aplicación de estas características.

Tras las primeras reuniones con el cliente, se hizo un listado de todas las especificaciones que requería la aplicación, haciendo al mismo tiempo un pequeño estudio de las herramientas de las que se podía disponer para desarrollar la aplicación.

- **Macromedia Director:** Herramienta para hacer aplicaciones multimedia. Aunque, más que orientado a web, está enfocado a aplicaciones para soportes físicos como CD's o DVD's. Por otro lado, comparte ciertas similitudes con Flash, pero no se estaba extendiendo a nivel comercial a la velocidad que lo estaba haciendo Flash. Lo que me limitaría bastante a la hora de encontrar ejemplos, tutoriales, librerías y demás recursos para facilitar el desarrollo de la aplicación.
- **Matlab:** Obviamente hablamos de una herramienta muy potente para cualquier tipo de cálculo matriciales e interpretación de señales, y para nuestro caso, una posible representación de gráficas. Pero no es un programa con el que se puedan crear interfaces, o al menos no está pensado para ello. Con lo cual únicamente podría funcionar como herramienta para el desarrollo de gráficas, cálculos de movimientos de barcos, etc. Esto quiere decir que necesitaría de otro programa con el que crear la interfaz y poder ejecutar las instrucciones de Matlab. Opción compleja y poco versátil pensando en los medios en que se quiere ejecutar la aplicación, sobretodo vía web.
- **Google Maps:** cuando se inició el proyecto, la *API de geolocalización de Google Maps* Ref.5 aún no se había liberado, y los pocos servicios que se ofrecían para desarrolladores como myMaps, aún estaban en fase beta, con lo cual, no se contempló su uso para esta aplicación, ya que hablamos de un proyecto comercial.

- **Flash:** La herramienta que cumplía todos los requisitos para crear la aplicación multimedia, sin duda era Flash. Por la facilidad de manejar objetos a través de la propia interfaz del programa, por que tenía soporte para ejecutar la aplicación vía web, a través del plugin Flash Player, y porque a través de la red, se podía disponer de suficiente información como para llevar a cabo el proyecto.

Solamente hubo un problema durante el desarrollo de la aplicación. Debido a la gran cantidad de barcos que se tenían que representar en el mapa acústico, se quiso hacer un sistema de carga dinámico de los barcos. Pero debido a las herramientas de las que se disponía por aquel entonces dentro de Flash para hacer este tipo de animaciones (hablamos de Flash CS2), tuvimos que recurrir al desarrollo de un sistema propio para controlar dinámicamente la carga de barcos. Pero el sistema desarrollado tenía un coste computacional demasiado alto cuando se mostraban muchos barcos en pantalla. Y al tener plazos de entrega constantes, no hubo tiempo para seguir investigando, y se tubo que recurrir a la animación manual de cada barco. Con lo cual, el sistema de carga dinámico de barcos se hizo como mejora en la fase 4 del proyecto.

- **Actionscript2:** En el momento de inciar el proyecto, ya había posibilidad de emplear el nuevo lenguaje de programación de Flash actionscript3, pero al no haberlo utilizado anteriormente y no haber demasiados ejemplos en internet, no se asumió el riesgo de utilizarlo, por el mismo motivo citado anteriormente, los plazos de entrega eran muy seguidos y el tiempo de investigación se tenía que suplir por el de desarrollo.

1.2.3. Tecnologías del presente

- **Integración en redes sociales > Paso a web 2.0:** Sin duda, uno de los medios más efectivos para difundir una aplicación multimedia a través de internet, son las redes sociales. Por lo tanto la integración de las aplicaciones multimedia con redes sociales se convierte en algo, sino indispensable, muy aconsejable.

- **API de Google Mapa para Flash** *Ref. 5:* A día de hoy, Google Maps dispone de una API para desarrolladores, incluso se puede utilizar bajo Flash. Esta API de ActionScript proporciona varias utilidades para manipular y añadir contenido a mapas a través de distintos servicios, lo que permitiría combinar mapas interactivos con el resto de funcionalidades del proyecto.

Cabe destacar que la geolocalización está motivando a creativos y desarrolladores a aprovechar esta funcionalidad y crear así nuevas herramientas de búsqueda en diversos ámbitos. Además sirven para simplificar el desarrollo de funcionalidades que dependen de mapas para mostrar información.

- **HTML 5** *Ref. 6:* Es la nueva generación del HTML, concretamente su quinta revisión importante. Dentro de esta nueva versión, encontramos una serie de mejoras que, en muchos casos, ayudan a evitar el uso de programas adicionales que necesitan de plugins en los navegadores, como Adobe Flash.

La combinación de *HTML + CSS + API's de Javascript*, dan lugar a una nueva forma de crear entornos web multimedia.

Por lo tanto, actualmente, podría ser otra vía con la que desarrollar una aplicación multimedia, ya que tiene infinidad de nuevas prestaciones, entre ellas la incorporación de canvas para dibujar, reproductor de audio y vídeo y efectos múltiples para animaciones simples y modificación visual de objetos creados directamente desde HTML. Pero estas no son prestaciones suficientes como para poder crear una aplicación como la desarrollada. Ya que no se dispone de una interfaz gráfica en la que controlar los objetos en espacio y tiempo, no se permite la interacción audiovisual con cualquier tipo de objeto, máscaras complejas, etc.

- **Smokescreen:** Un proyecto muy interesante que aún se encuentra en fase de desarrollo es *smokescreen* Ref.11. Consiste en una herramienta opensource que transforma un archivo Flash en HTML5 + Javascript, de modo que no se necesita el plugin de Flash para visualizar el proyecto. Aunque no funciona en ciertos navegadores como IE8 e inferiores. Además consume más recursos de sistema que el player de Flash, así que para aplicaciones con alto contenido gráfico y movimiento, puede ser algo limitado. Puede que se acabe orientando a banners o similares.
- **Flash** Ref. 3: Actualmente Flash están en la versión CS5, la cual tiene una serie de mejoras respecto a la que se utilizó desde el inicio del desarrollo. Ahora se puede trabajar en entorno 3D sin tener que recurrir a engines externos, aunque las posibilidades siguen siendo limitadas, sobretodo a nivel de programación. En cuanto a manipulación de cajas de texto ofrece algunas mejoras respecto a la versión utilizada en la aplicación. Pero el cambio más grande que se puede observar entre la versión y otras más antiguas, es el lenguaje de programación: *actionscript3*.
- **Actionscript3:** Este es el lenguaje que se emplea o debería emplearse actualmente para crear web's o demás aplicaciones en Flash. Por fin se puede hablar de un lenguaje de programación robusto y totalmente orientado a objetos. Esto permite estructurar el código debidamente. Además los programadores que vengan de lenguajes como C++ o Java, verán una clara similitud entre los distintos lenguajes.
- **Librerías para Flash:** Desde versiones de Flash anteriores, se puede recurrir al uso de librerías específicas para Flash y conseguir efectos visuales avanzados a través de código, evitando así las animaciones manuales, que siempre serán estáticas. Actualmente, pensando en las características que puede tener un *movieClip*, como interpolaciones de cualquier tipo (alpha, posición, tamaño, efectos de desenfoque, etc.) y control de tiempo, uno de los frameworks más extendidos y potentes, es *Greensock* Ref.2. De hecho es la librería que se ha utilizado para hacer la carga dinámica de barcos en el mapa acústico.

Otras librerías similares pero menos eficientes y completas son: *Tweener* ^{Ref.8}, Twease, ZigoEngine, Fuse, Tween de Adode, etc.

Existe una comparativa ^{Ref.9} en la que se demuestra que el framework de Greensock es el que menos recursos consume.

- **Engines 3D para Flash:** Cada vez más, se ven aplicaciones que emplean las 3 dimensiones para ganar espectacularidad, o simplemente, por necesidad de la propia naturaleza del proyecto. En nuestro caso, podría emplearse para mejorar la simulación que ahora es con falso 3D.

Hay unos cuantos engines que funcionan en Flash. Papervision3D es el más extendido, aunque hay otros similares como: Sandy3D, Five3D y Swift3D.

- **Microsoft Silverlight** ^{Ref. 24}: Es el competidor directo de Adobe Flash. Se puede trabajar sobre un entorno gráfico llamado Expression Blend ^{Ref. 25}. A grandes rasgos, una de las diferencias respecto Flash, es que utiliza el modelo de animación WPF. Este modelo se basa en el tiempo en vez de frames, y sólo necesita saber cuanto tiempo durará la escena y las condiciones, él la recrea.

El sistema de archivos ocupa más que en Flash al no estar comprimidos.

A nivel de lenguaje de programación, Silverlight utiliza tecnología propia creada por Microsoft. Por lo que podremos programar con lenguajes tales como Visual C#.Net o Visual Basic.Net, facilitando la tarea a los programadores.

Uno de los motivos más importantes por los que no se ha empleado Silverlight como herramienta de desarrollo para la aplicación, es por que no está tan extendida como Flash. Además actualmente, en sistemas operativos que no sean Windows, no dispone de plugin para navegadores como Google Chrome e Internet Explorer en ninguna de sus versiones. El plugin es necesario para su reproducción, del mismo modo que Flash emplea el Flash Player.

- **Adobe Flex** ^{Ref. 26}: Emplea el lenguaje MXML para el desarrollo de interfaces gráficas de usuario, está basado en XML. Actualmente van por la versión 4, y se centran en el desarrollo de aplicaciones RIA, facilitando el flujo de trabajo entre diseñadores y desarrolladores de estas aplicaciones.

Podría servir como herramienta para parte del desarrollo de la aplicación, pero al centrarse únicamente en interfaz para mostrar datos, se tendría que complementar de todos modos con Flash.

1.3. Descripción del problema

El equipo de científicos que llevó a cabo la investigación, fue recolectando información de las distintas áreas estudiadas. Pero no disponía de una herramienta con la que mostrar esta información.

Por un lado, se tenía que mostrar mediante gráficas convencionales las señales de las frecuencias emitidas por embarcaciones y cetáceos, y poder hablar de la solapación entre unas ondas y otras. También era necesario representar una animación audiovisual para poder representar esta solapación de ondas, ya que de este modo, el usuario podría interactuar con los elementos y comprender mejor el problema ocasionado por las embarcaciones. Y por último, el objetivo principal, que era crear el mapa acústico de todo el litoral, con una representación tanto visual como auditiva de la contaminación acústica submarina que generan las embarcaciones.

En este punto se vieron con la necesidad de crear una herramienta multimedia personalizada, con la que manejar los datos recopilados. Un sistema con el que poder mostrar más información que la que se podía mostrar mediante tablas numéricas, gráficas o mapas estáticos.

El equipo de científicos tenía los medios y conocimientos para llevar a cabo el estudio, pero carecía de personal para poder desarrollar la herramienta con la que poder mostrar todas las funcionalidades.

Gracias a esta necesidad, se pusieron en contacto conmigo, y a través de una primera reunión, poder exponer la situación del proyecto y tratar de encontrar una forma viable de desarrollar esta herramienta.

El siguiente paso fue estudiar con detenimiento las distintas funcionalidades, cómo mostrarlas y qué software de desarrollo utilizar.

1.4. Solución propuesta

El laboratorio de bioacústica consiguió fondos de la Obra Social para financiar el estudio, y así poder difundirlo con más fuerza y medios.

La aplicación que necesitaba el cliente, servía de presentación para exponer todos los estudios en las ruedas de prensa realizadas, tanto en las comunidades donde se habían ido realizando los estudios, como en ciertas capitales estratégicas. Siempre con el objetivo de informar sobre la realidad de nuestras costas, y llegar a los organismos que pudiesen tomar cartas en el asunto.

Por otro lado, la Obra Social, tenía la finalidad de hacer llegar al ciudadano toda esta información. Así que el desarrollar una herramienta capacitada para utilizarla en presentaciones o exposiciones tenía sentido.

Años atrás, era más difícil crear una herramienta con la que mostrar contenidos complejos, y más si hablamos de una herramienta enfocada, no sólo a un sector específico, sino al público en general.

Por las necesidades del cliente, lo que convenía era crear una aplicación multimedia, capaz de interactuar con el usuario para hacerle comprender mejor, y poder estudiar con detalle, el problema de la contaminación acústica mediante el mapa interactivo.

Internet se ha convertido en uno de los canales más utilizados para difundir contenidos multimedia. Así que, se dio un paso más y se aprovechó el canal de internet, para poder difundir el estudio.

1.4.1. Solución al mapa acústico

Era necesario crear un sistema en el que proyectar la contaminación acústica mediante mapas. Tras los estudios, se disponía de información de todo el tráfico de embarcaciones de los principales puertos marítimos durante el tiempo que duró el estudio.

Pero habían dos problemas, uno de cómo representar a nivel temporal este tráfico y otro cómo acotar las zonas estudiadas.

Principalmente se pensó en crear un único mapa que fuese representando mediante niveles cromáticos, las variaciones de nivel de contaminación acústica. Pero de este modo, no se podía dar información de la contaminación producida en cada puerto estudiado de forma detallada, dado que cada puerto tenía un conjunto de itinerarios utilizado por distintas embarcaciones.

De modo que se propuso acotar toda esta información en distintos mapas que representasen cada comunidad estudiada, y a su misma vez, cada comunidad en distintas áreas según la cantidad de puertos e itinerarios que albergase.

En cuanto al problema de cómo representar los estudios obtenidos durante un año entero, era evidente que hacer un histórico de todo el año podía convertirse en una tarea innecesaria. Así que se propuso escoger únicamente cuatro días de estudio que fuesen representativos de todo un año. Por lo tanto se eligió un día de cada estación.

Finalmente, para la representación del mapa acústico se dispondría de un sistema por el que poder desplazarse tanto de forma temporal como espacial.

1.4.2. Solución al método en que mostrar la solapación de ondas entre cetáceos y embarcaciones

Se propusieron dos formas para poder representar la solapación de ondas: Una más técnica y la otra más ilustrativa:

1. Creación de una gráfica frecuencia-ruido, en la que se pudiese elegir el tipo de embarcación y/o de cetáceo para representarse la solapación.
2. Se propuso crear una animación interactiva en falso 3D, para poder representar la solapación de ondas entre cetáceos y embarcaciones.

1.4.3. Características de la aplicación a desarrollar

Finalmente, podemos destacar las siguientes características que debía cumplir la aplicación final:

1. Crear un sistema de mapas para navegar entre las distintas áreas estudiadas, y poder acceder así a los distintos mapas acústicos. Los cuales debían representar mediante animaciones de barcos la contaminación acústica de las zonas estudiadas, así como la distribución de cetáceos.
2. Mostrar mediante gráficas la solapación de ondas entre cetáceos y barcos.
3. Crear una simulación interactiva de la animación de un barco y un cetáceo para representar de forma audiovisual el nivel de contaminación y solapación de ondas frecuenciales.
4. Completar mediante texto la información que se obtiene a través de las distintas herramientas de la aplicación.

1.5. Perspectiva general

En los siguientes apartados de la memoria, se desarrollarán los puntos descritos a continuación:

1.5.1. Fundamentos teóricos

- Se hablará del concepto de la contaminación acústica y del repercusión que está teniendo el impacto humano en el medio marino.
- Teoría sobre la solapación de ondas frecuenciales entre cetáceos y embarcaciones.
- Información sobre las tecnologías web del momento, tanto las empleadas en la aplicación como las que están relacionadas.
- En cuanto a lenguajes de representación visual, se explicará la herramienta empleada, así como el sistema con el que se representan las animaciones dinámicas de las embarcaciones dentro del mapa acústico.
- Desarrollo gráfico para la elaboración de mapas.

1.5.2. Parte práctica

- Se analizan todos los requisitos de la aplicación
- Se crea un diagrama del análisis funcional, en el que se muestra la estructura de módulos de la que se compone la aplicación, explicando también el funcionamiento de cada uno de estos módulos.
- Representación de un diagrama en el que se ilustra la estructura de archivos a nivel de sistema. Nos ayuda a visualizar su funcionamiento, explicando de forma detallada cada uno de los archivos.
- Explicación de las decisiones que fueron tomándose a lo largo del proyecto, como las tecnologías empleadas y desestimadas, cambios en los mapas utilizados para mejorar la calidad visual, tratamiento de los datos cedidos por el cliente para adaptarlos a la aplicación, creación de una simulación para representar de una forma más visual la solapación de frecuencias entre cetáceos y embarcaciones, y el desarrollo de la animación dinámica de barcos en el mapa acústico.

- Elaboración de un diagrama de clases de la aplicación, de este modo puede visualizarse cómo se estructura el código. Además se muestra otro diagrama para visualizar la estructura a nivel de archivos xml, y ver así cómo afecta la jerarquía de mapas a nivel de archivos de contenido. Posteriormente se explica en qué consiste la gran mayoría de métodos y librerías desarrollados en la aplicación.
- Se explica cómo se ha desarrollado el proyecto separándolo por fases reales. Además se pueden ver pantallas de la aplicación que ilustran las mejoras a nivel gráfico de una fase a otra, así como las nuevas funcionalidades que se fueron desarrollando.
- Explicación de la metodología empleada para el desarrollo. Ya que cada fase se desarrolló en unas circunstancias distintas, se tuvo que adaptar la metodología a emplear en cada una de ellas.

1.5.3. Resultados

- Se habla sobre el rendimiento de la aplicación
- El coste invertido en horas
- Un diagrama en el que se observa la relación entre las partes colaboradoras del proyecto, y en global, de la obra social.

1.5.4. Conclusiones y líneas de futuro

- Se habla de aspectos generales para concluir la memoria: si se le ha sacado provecho al proyecto, el grado de satisfacción por parte del cliente, si sería necesario readaptar la aplicación, si es comercializable y si se tendría que haber orientado de otra forma.
- Se habla de las posibles mejoras que podría tener la aplicación, tanto a nivel de difusión como a nivel de desarrollo. También se explica una forma alternativa en la que se podría desarrollar el proyecto.

2. FUNDAMENTOS TEÓRICOS

2.1. Contaminación acústica

Aunque sea un tema que quede un poco al margen del desarrollo de esta herramienta multimedia, el concepto de contaminación acústica está totalmente ligado al objetivo de esta aplicación, ya que principalmente, se desarrolló esta herramienta con el fin de crear un mapa acústico.

Cuando se habla de contaminación acústica, suele hablarse de la contaminación producida por el tráfico de vehículos, industrias, locales de ocio, aviones, etc, al fin y al cabo es el ruido que se genera en nuestro entorno y afecta a nosotros directamente. Pero no se ha de olvidar que también podemos hablar de contaminación acústica en otro medio, como el submarino. Pues su impacto tanto en el hábitat animal como vegetal es cada vez mayor debido en parte al gran nivel de tráfico marítimo, ocasionado tanto por embarcaciones mercantiles como de ocio.

El ruido submarino producido por actividades humanas aumenta cada día e incluye el tráfico marítimo, la explotación y producción de gas y petróleo, el sonar industrial y militar, las fuentes sonoras de experimentación industrial, los explosivos submarinos, etc. De hecho, no existe ningún rincón del mundo que no esté afectado por la contaminación acústica.

Dentro del conjunto de los factores de riesgo que pesan sobre el hábitat marino, esta contaminación constituye una de las mayores amenazas a corto plazo y escala mundial para el equilibrio de los océanos. Dado que los cetáceos dependen del sonido en todos los aspectos de sus vidas, no cabe duda de que son especialmente vulnerables a las fuentes de ruido artificial.

Los cetáceos son altamente dependientes de su aparato auditivo para su supervivencia. Muchas especies utilizan sonidos para localizar presas, navegar y comunicarse, alcanzando distancias considerables en el caso de numerosas

especies de grandes ballenas. Los cetólogos están cada vez más preocupados por la contaminación acústica de los océanos derivada del intenso tráfico marítimo, los muestreos sísmicos, la extracción del petróleo o los dispositivos de sónar, elementos que pueden estar teniendo importantes impactos negativos en muchas especies.

A la luz de recientes mortalidades en aguas españolas, parece claro que fuentes de ruido, a diferentes niveles de intensidad, pueden afectar de forma negativa a poblaciones de cetáceos. El impacto de estas fuentes puede variar de forma significativa, desde causar molestias y desplazamiento de poblaciones hasta lesiones de distinta gravedad en el sistema auditivo: de ligeras, y posiblemente reversibles, a graves, produciendo sorderas permanentes e incluso la muerte inmediata del animal.

La modernización y aumento del tráfico marítimo, junto con prácticas de exploración geológica, prospecciones petrolíferas y el empleo militar de sonares activos en España, vuelven especialmente sensibles a las poblaciones de cetáceos existentes en aguas como las del estrecho de Gibraltar y Baleares. *Ref.10.*

Hasta que no se comenzaron a recoger los datos estudiados en el 2008 por el equipo de científicos e ingenieros del Laboratorio de Aplicaciones Bioacústicas *Ref. 16,* no se han podido obtener datos realistas del nivel de contaminación acústica submarina en otras zonas de la costa peninsular.

2.2.1. Acerca de los cetáceos

El sistema auditivo de los cetáceos está caracterizado por una serie de adaptaciones morfológicas únicas: una de las más interesantes es la capacidad de discriminación fina de imágenes acústicas a través de los canales auditivos que actúan como filtros de frecuencias.

En un organismo sano, esta selectividad de frecuencias del oído (y por lo tanto de las señales acústicas que producen) está evolutiva y directamente en relación con el uso específico de su hábitat, y caracteriza por lo tanto, cada especie de cetáceos.

Por otro lado, dentro de esta selectividad de frecuencias, la sensibilidad del oído a algunas frecuencias, permite medir el estado fisiológico y/o patológico del sistema auditivo de un determinado individuo y estimar su capacidad acústica para utilizar su hábitat.

2.2.2. Acerca de las embarcaciones

Las fuentes sonoras asociadas al transporte marítimo se introducen en el espacio acústico y físico de los organismos marinos y no existen actualmente niveles de referencia que permitan prever las consecuencias negativas de estas interacciones a corto, medio o largo plazo sobre el equilibrio natural de los mares y océanos.

Muchos factores pueden, potencialmente, estar involucrados en estos procesos: el nivel de fuente del sonido, su transmisión a través del agua, la posición del animal en la columna de agua, su comportamiento y estado fisiológico, así como efectos sinérgicos, incluida cualquier lesión física crónica.

Todos pueden jugar un papel, pero se desconocen los parámetros básicos del mecanismo de impacto que permitirían controlar los efectos negativos de esta contaminación acústica, y posteriormente, legislar sobre la introducción de fuentes sonoras artificiales en el medio marino.

2.3. Tecnologías web

A continuación se muestra información general sobre los distintos lenguajes de programación que se han empleado en la aplicación. Hablamos de lenguajes que se centran en la capa de presentación.

2.3.1. HTML

Es el lenguaje base con el que poder crear páginas web. Se usa para describir la estructura y el contenido en forma de texto. Dentro de él podemos encontrar objetos que referencian a otros contenidos, como por ejemplo, enlaces a hojas de estilo (lenguaje CSS), a scripts tipo javascript, a objetos Flash, etc.

Actualmente, existe ya su quinta revisión importante, dándole el nombre de HTML 5 *Ref. 6*, que mejoran el lenguaje al añadir nuevas etiquetas, quizás las más relevantes sean las que permiten difundir audio y vídeo. Evitando así el uso de plugins para ejecutar archivos que el navegador no sabría interpretar, como los plugins de Flash, Quicktime, Java, etc.

Hay que tener en cuenta que esta serie de mejoras, únicamente pueden ser interpretadas por navegadores modernos. Es decir, cada nueva versión de HTML añade o corrige una serie de elementos y para ello los navegadores han de ser capaces de saber interpretarlo. Y esto da pie al problema de estandarización dentro del lenguaje web.

Si nos basamos en estadísticas, podemos comprobar que hay más usuarios con navegadores antiguos con plugins como Adobe Flash, que no navegadores modernos, capaces de interpretar correctamente las nuevas especificaciones del HTML. Esto hace, por desgracia, que tales avances, se tengan que tener muy en cuenta a la hora de ser utilizados, o no, al desarrollar páginas web.

2.3.2. XML

El XML es un metalenguaje extensible de etiquetas desarrollado por la W3C *Ref. 19*.

El lenguaje XML es un sistema muy extendido y que sirve de base para muchos otros lenguajes de programación, de ahí que se defina como metalenguaje.

Podemos decir que es un lenguaje de etiquetas, y que por su propia estructura, está marcando la jerarquía de los contenidos. Por lo tanto, es muy importante que los XML empleen una buena semántica para definir las etiquetas y sus atributos.

Una forma de introducir contenidos sin ser analizados, es mediante las secciones CDATA, de este modo, podemos introducir texto sin que este sea parseado como XML.

Para realizar una serie de reglas sintácticas a la hora de definir las etiquetas, podemos crear un archivo DTD, aunque su formato en sí, no sigue la estructura de los XML, por este motivo, se utilizan los XML schema, que sirven para lo mismo y mantienen una estructura de tipo XML, aunque algo más complicada de especificar. En cualquier caso, ambos lenguajes sirven para validar los XML y comprobar la integridad de sus datos.

2.3.3. Javascript

2.3.3.1. HISTORIA

El javascript es un lenguaje de scripting, del cual se desarrollaron las primeras versiones a principios de los 90. Según iba creciendo el nivel de complejidad de las páginas web, y dependiendo de unas conexiones a internet muy lentas, nació la necesidad de poder ejecutarse un lenguaje de programación directamente en el navegador del usuario, de este modo se evitaba el depender siempre de las respuestas de servidor, minimizando así el tiempo de respuesta de las páginas web.

Tras su éxito al funcionar bajo el Navegador Netscape, el lenguaje Javascript se estandarizó, evitando así problemas legales, tras querer competir Microsoft con el mismo lenguaje pero con otro nombre: Jscript. Y posteriormente, la W3C (World Wide Web Consortium) *Ref.19*, estandarizó las distintas versiones que había de Javascript para compatibilizarlo con los navegadores del momento.

Una vez se estableció Javascript como lenguaje de scripting estándar para ejecutarse en el navegador, comenzaron a desarrollarse distintas librerías o *frameworks* con el objetivo de facilitar la tarea de los desarrolladores web.

2.3.3.2. FRAMEWORKS PARA JAVASCRIPT

Son librerías que proveen de un amplio conjunto de componentes. Los más destacados:

- Compatibilidad con distintos motores intérpretes del javascript.
- Comunicación asíncrona (Ajax): Aumenta la interactividad del usuario con la web al no tener que recargar la página con tanta frecuencia.
- Validación de formularios.
- Efectos visuales.
- Manejo de eventos.
- Almacenamiento Client-Side, permite a las web's guardar datos de usuario de forma segura.

Respecto a los frameworks más extendidos:

- Dojo toolkit
- ExtJS
- JQuery
- Mootools
- Prototype + Scriptaculous

La continua mejora de estos frameworks para trabajar con componentes, está creando cada vez más competencia respecto al uso de aplicaciones de Adobe como Flex y Flash.

2.3.3.3. UTILIDADES DE JAVASCRIPT

- Generar cambios dinámicos en las hojas de estilo (CSS), consiguiendo así efectos visuales interesantes.
- Comprobación de la inserción correcta en formularios.
- Comprobación de versiones de plug-in's externos al navegador, versión y tipo de navegador entre otras configuraciones del equipo, como la resolución de pantalla, etc.
- Carga de datos desde servidor de forma asíncrona mediante Ajax.

2.3.3.4. AJAX

Ajax es una técnica de programación, que no lenguaje ni sublenguaje. Y aunque años atrás ya existían los métodos que hacían posible la carga asíncrona de datos, fue años después que se comenzaron a establecer como una técnica de programación, consiguiendo crear un nuevo concepto de desarrollo web.

Esta técnica evita el recargar la página cada vez que se necesita acceder a servidor para transmitir datos. Por lo que puede llegar a dinamizar muchísimo las funcionalidades de una web.

2.3.4. CSS

Las hojas de estilo en cascada o CSS, definen el formato visual de los contenidos de un documento HTML o XML. Las especificaciones son estandarizadas por la W3C *Ref.19*.

El objetivo de las hojas de estilo es separar la estructura de un documento de su presentación, obteniendo así más versatilidad para mostrar una misma página web en distintos dispositivos o en distintos medios, como versión para imprimir, versión para lectores de pantalla, etc.

Las hojas de estilos pueden ser de tres tipos:

- Externas: Las hojas de estilo se almacenan en un archivo diferente. Es un tipo de documento de texto plano y no enriquecido, igual que el HTML o XML, pero con extensión CSS.
- Internas: La hoja de estilo está incrustada dentro del documento HTML.
- En línea: Se aplican los estilos dentro de las etiquetas HTML.

2.3.5. Web 2.0

Podemos decir que la web 2.0, no es una tecnología, sino una filosofía a tener en cuenta cuando se desarrolla una página web.

Un conjunto de conceptos son los que acaban definiendo como tal una web 2.0, entre los cuales, se pueden destacar los siguientes:

- Web con semántica rica y correcta para mejorar la indexación de contenidos
- Diseños de interfaz sencillos y funcionales
- Separación de contenidos y aspecto visual (HTML+CSS)
- Integración de la web con redes sociales con el fin de hacerlas colaborativas, manteniendo los contenidos actualizados de forma automática

El objetivo acaba siendo ceñirse a unas pautas para garantizar el éxito comercial de la página web.

Otro aspecto importante de la web 2.0 es evitar la descarga de contenidos, es decir, contenidos en la nube.

Gracias a herramientas como los blogs, las wikis, google Docs, correo como Gmail, Yahoo, Hotmail, las redes sociales de música como mySpace, last.fm e incluso Spotify como aplicación de escritorio, flickr, y un sinnúmero de herramientas más, se consigue disponer de la información que necesitamos sin necesidad de descargarla.

El tener la información en la nube, inicialmente podía parecer que no tuviese demasiado sentido, quizás por la tendencia a tenerlo todo almacenado y archivado a nivel local, como se hace tradicionalmente con las cosas físicas. Pero el tener esta información en internet tiene una serie de ventajas respecto al almacenamiento local, como la libertad de disponer de esta información sin la necesidad de hacerlo siempre desde el mismo terminal, esto nos da libertad de movimiento. Otra ventaja es la de dar la oportunidad a que contenidos que no nos importa que sean publicados, se enriquezcan gracias a la participación de otras personas, o que otras personas puedan beneficiarse de estos contenidos. Un claro ejemplo es la wikipedia.

2.4. Lenguajes de representación visual

2.4.1. FLASH

Es una herramienta de desarrollo para crear aplicaciones interactivas. Utiliza Flash Player como máquina virtual para interpretar los archivos generados por Flash, generando por regla general la extensión de archivo SWF.

Para poder ejecutarse los SWF a través de un navegador web, se necesita un plug-in del player de Flash. Del mismo modo, se pueden ejecutar localmente si se tiene instalado el reproductor de Flash. Esto permite crear tanto páginas web *full Flash* como aplicaciones de escritorio, aplicaciones para dispositivos móviles o presentaciones. En todos los casos dando la capacidad de crear contenido multimedia e incluso comunicarse con lenguajes de programación en servidor como PHP, ASP, etc.

El entorno de la herramienta Flash permite trabajar con objetos vectoriales y tipo mapa de bits sobre una línea de tiempo para poder animarlos. Por otro lado, también permite trabajar con vídeos, audio y documentos que almacenen contenido como XML, documentos de texto, etc. Incluso permite sincronizarse en tiempo de ejecución con periféricos externos, como puede ser una webcam. Todas estas características dan una gran versatilidad a la herramienta.

Como lenguaje de programación, las versiones actuales de Flash permiten trabajar con actionscript 2 y 3. Además se pueden crear o hacer uso de librerías externas con extensión AS. Actionscript 3 ha apostado por la programación 100% orientada a objetos a través de clases.

Por otro lado son capaces de generar y cargar componentes para facilitar el desarrollo de aplicaciones.

Además, actualmente hay una gran cantidad de frameworks compatibles o incluso desarrollados específicamente para Flash, que enriquecen a la herramienta. Por ejemplo frameworks para trabajar en un entorno 3D, motores de físicas 2D y 3D, etc.

Flash ha tenido etapas en las que quizás, ha sido utilizado como herramienta para crear páginas web sin una causa justificada. Haciendo un mal uso o un uso excesivo de los efectos visuales que son creados con suma facilidad, perdiendo el enfoque de la web.

Sin embargo, Adobe Flash es una herramienta que permite crear aplicaciones y web's complejas totalmente interactivas, como es el caso del actual proyecto.

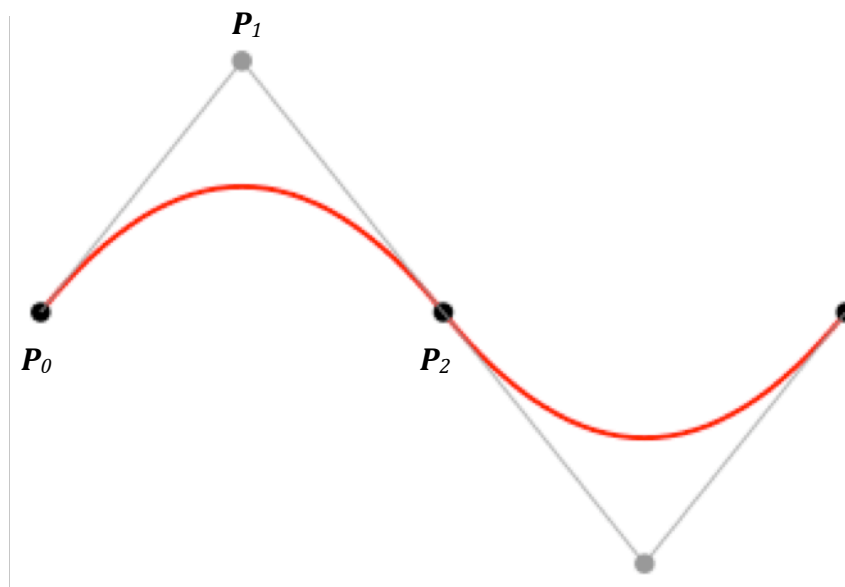
Gracias a todas las cualidades descritas anteriormente, Flash se utilizó como herramienta de desarrollo para realizar este proyecto.

2.4.2. Representación de curvas Bézier cuadráticas

Las curvas bézier cuadráticas se emplean para múltiples funcionalidades, como la creación de las letras tipo True Type.

En nuestro caso, ha sido útil para la generación de los itinerarios por los que desplazarse los barcos dentro del mapa acústico.

Estas curvas tienen un punto de control inicial, otro final, y otros intermedios que darán la curvatura que se necesita. El punto intermedio entre dos puntos de control siempre será tangente y por donde pase la curva:



Curva Bézier cuadrática Ref. 22

En términos matemáticos, podemos definir una curva bézier cuadrática como el camino trazado por la función $B(t)$, dados los puntos P_0 , P_1 y P_2 . Ref. 21.

Su fórmula sería:

$$B(t) = (1-t)^2P_0 + 2t(1-t)P_1 + t^2P_2, t \in [0,1]$$

2.5.2. Adaptación de itinerarios

En la primera área estudiada, por parte de cliente se obtuvieron tablas tipo Excel con el listado de los puertos de origen y destino de cada embarcación. Posteriormente se buscaban mediante la herramienta de Google Maps y se representaban en un mapa las trayectorias necesarias.

En las siguientes áreas estudiadas, el cliente cedía directamente los mapas a modo de boceto para luego incluir estos itinerarios en Flash.



Mapas con itinerarios cedidos por el cliente

Una vez se disponía de los mapas con itinerarios, se generaban los mapas retocados con Photoshop y se incluían dentro de los archivos Flash.

Los mapas de las zonas de interés (mapas de itinerarios), representaban una estética algo distinta a la de los mapas de navegación (mapa peninsular y mapas de áreas estudiadas). Pues en este caso, se empleaba un color azul oscuro constante para la zona de mar. De este modo, se podía representar mejor el nivel de contaminación mediante color rojo con degradado de alpha.

3. PARTE PRÁCTICA

3.1. Análisis de requisitos

A partir de una primera entrevista con el cliente, en la que se habló del proyecto en general, personal que trabajaba y posterior presentación con mi trayectoria profesional, se establecieron varias reuniones, en las que se analizaron todas las necesidades para crear la aplicación multimedia.

Después de exponer el cliente sus objetivos con la aplicación y datos recogidos en los estudios, se habló sobre el funcionamiento de la aplicación, analizando uno a uno los requisitos:

3.1.1.1. MAPA DE CONTAMINACIÓN ACÚSTICA SUBMARINA

Era necesario crear un mapa acústico que, mediante un sistema interactivo y a poder ser de forma audiovisual, pudiese representarse el nivel de contaminación de embarcaciones en un período de tiempo determinado.

3.1.1.2. MUESTRA DE ESPECIFICACIONES SOBRE FRECUENCIAS DE CETÁCEOS Y EMBARCACIONES

Mediante el uso de gráficas debía darse información del rango de frecuencias utilizados por los cetáceos y embarcaciones. Pudiendo ver de este modo la solapación de ondas frecuenciales que hay entre unos y otros.

3.1.1.3. ANIMACIÓN DE UN CETÁCEO Y UN BARCO PARA VER LA SOLAPACIÓN DE ONDAS.

A través de una animación, quería representarse de un modo más visual e interactivo la solapación de ondas frecuenciales entre cetáceos y embarcaciones al acercarse uno a otro.

3.1.1.4. SISTEMA DE INFORMACIÓN GENERAL DEL ESTUDIO.

También era necesario poder ampliar la información que se mostraba en la aplicación mediante textos complementarios.

que tenía el cliente por las distintas comunidades autónomas (áreas estudiadas), donde se presentaba la aplicación como soporte audiovisual.

De modo que para acceder a un mapa acústico el proceso sería el siguiente:

Mapa peninsular > Área de interés > Zona de interés > Mapa acústico.

3.2.1. Mapa peninsular

Desde el mapa peninsular se podría acceder a los mapas de cada área estudiada clicando en la región que nos interesase del propio mapa, o acceder a la ayuda inicial e información de bienvenida mediante el panel inferior derecho.

3.2.1.1. AYUDA

Se debía mostrar algunos consejos básicos para facilitar la navegación por la aplicación.

3.2.1.2. INFORMACIÓN DE BIENVENIDA

La herramienta que se quería desarrollar, iba a ser totalmente interactiva, muy visual e incluso con sistemas de escucha. Pero era necesario ampliar la información que se mostraba a través del mapa acústico y demás utilidades con texto complementario.

Por tanto, se propuso añadir un sistema de información en cada una de las funcionalidades de la aplicación. Tanto a modo introductorio con explicación general de la herramienta, como información detallada de embarcaciones y cetáceos, solapación de ondas y demás conceptos que se dan a conocer dentro de la aplicación.

3.2.2. Mapas de áreas estudiadas

Cada área de interés tendría un mapa que funcionase del mismo modo que el peninsular. En este caso serviría para acceder a las distintas zonas de interés del área seleccionada (mapas acústicos). Además, mediante el panel inferior derecho, se podría acceder nuevamente a la ayuda, las gráficas de acústica, la simulación y a la información general.

3.2.2.1. AYUDA

Para cada área estudiada, no estaba de más, poder recordar la ayuda inicial del mapa peninsular, puesto que servía para la aplicación en general.

3.2.2.2. ACÚSTICA (MÓDULO DE ESPECIFICACIONES)

Se requería de un módulo en el que poder observar mediante gráficas los distintos rangos frecuenciales de los principales tipos de embarcaciones y cetáceos, todo ello para cada una de las áreas estudiadas.

También se utilizaría este módulo para detallar las especificaciones técnicas de cada tipo de embarcación.

Esta herramienta tenía como finalidad, poder ilustrar el solapamiento de ondas frecuenciales entre embarcaciones y cetáceos. Dando por obvio que tal solapamiento repercute seriamente sobre la comunicación y orientación de los cetáceos.

Todas estas especificaciones se complementarían con un sistema de información y ayuda interactiva para acabar de transmitir el objetivo.

3.2.2.3. SIMULACIÓN

Para poder transmitir de una forma más visual este emmascaramiento de frecuencias entre cetáceos y embarcaciones, era necesario desarrollar una herramienta audiovisual e interactiva.

Por lo tanto, mediante un sistema de falso 3D, y a través de una herramienta de zoom y una línea temporal, se podría ver la animación de un barco y un cetáceo.

3.2.2.4. INFORMACIÓN DE BIENVENIDA

Del mismo modo que con la ayuda, debería mantenerse la información de bienvenida.

3.2.3. Mapa acústico

Se requería desarrollar un mapa en el que poder generar, inicialmente por vía visual, el nivel de contaminación acústico creado por el tráfico marítimo de las costas del litoral. Este se representaría mediante la animación de barcos y representación de contaminación en zonas costeras. También debería poder accederse a ciertas funcionalidades:

3.2.3.1. AYUDA INTERACTIVA

Interesaba mostrar información al usuario acerca de los distintos elementos de la interfaz, para ello era necesario crear un sistema de ayuda.

3.2.3.2. SISTEMA ACÚSTICO

Se pensó en crear un sistema acústico para escuchar el nivel de contaminación de los barcos y de las zonas costeras, dependiendo del nivel de contaminación y tipos de embarcaciones, el sonido variaría.

3.2.3.3. GRUPOS DE CETÁCEOS

También era necesario representar visualmente la distribución de los cetáceos a lo largo de las costas en un tiempo determinado.

3.2.3.4. INFORMACIÓN DE PUERTOS

Dado que había mucho tráfico entrante y saliente de embarcaciones en los puertos principales, sería necesario poder mostrar mediante una tabla toda esta información de cada puerto.

3.2.3.5. INFORMACIÓN GENERAL

Debería facilitarse información complementaria sobre el impacto de la contaminación acústica en la vida de los cetáceos.

4. **area<numArea>.swf:** Se muestra el mapa del área estudiada a la que hemos accedido, así como ayuda, información y los módulos de acústica y simulación. También se da acceso a las zonas de interés del área estudiada.
5. **area<numArea>zi.swf:** Se carga el mapa acústico, además de ayuda e información.

El mapa acústico tiene las siguientes funcionalidades:

- Barra de tiempo desplazable
 - Cambio de día estudiado
 - Sistema acústico interactivo
 - Visualización de áreas frecuentadas por cetáceos
 - Información del tráfico marítimo en los puertos principales
 - Información sobre el origen/destino de cada itinerario
 - Información detallada de cada barco
6. **generadorItinerarios.swf:** Es una utilidad independiente a la aplicación que se utiliza para generar, a partir de puntos de control, las coordenadas de las curvas Bézier que representan cada uno de los itinerarios. Los datos que genera, son posteriormente pasados a archivos de tipo XML.
Esta utilidad es necesaria para la carga dinámica de barcos.
 7. **Datos tipo excel:** Archivos que contienen toda la información de los estudios. Mediante un proceso manual, estos datos son pasados a formato XML, ya que la información cedida por el cliente no podía parsearse automáticamente debido a la naturaleza de los contenidos en sí.
 8. **Datos tipo XML:** Contienen toda la información a nivel de texto que necesita el mapa peninsular, así como rutas de otros archivos XML con contenidos de las distintas áreas de interés.
 9. **Datos tipo XML:** Contienen toda la información a nivel de texto que necesitan las áreas de interés y sus zonas estudiadas, así como rutas de otros archivos XML con contenidos del mapa acústico (itinerarios y manchas) y los módulos de información, ayuda, simulación y acústica.
 10. **Librerías actionsript:** Todas las librerías que se emplean se centran en la programación del mapa acústico (información de barco, puerto, colisiones, manchas y librería externa Greensock).

3.4. Toma de decisiones

3.4.1. Tecnología a emplear

Dado que hablamos de una aplicación que, por su naturaleza debía ser de tipo audiovisual e interactiva, ya se pensó en trabajar con una plataforma que facilitase su desarrollo en estos aspectos. Y que además ofreciese un avanzado sistema de renderización de gráficos para poder transmitir lo más fiel posible, todos los datos recogidos.

Es por estos motivos anteriores, y debido a la familiarización que ya se tenía con esta tecnología en proyectos anteriores, que se decidió utilizar *Adobe Flash* Ref. 3 como herramienta sobre la que desarrollar toda la aplicación.

Inicialmente, la aplicación estaba pensada únicamente como soporte para las ruedas de prensa que se realizaron a lo largo de la costa peninsular y ciudades importantes del país. Pero dado que Flash es una tecnología, en gran medida orientada a la plataforma web, fue motivo suficiente para ampliar los canales de difusión de esta aplicación, dejando de ser una herramienta específica para las ruedas de prensa.

En cuanto al lenguaje de programación dentro de Flash, se optó por *actionscript2*. Ya que hablamos de un proyecto iniciado en julio del año 2007, el uso de *actionscript3* no estaba tan extendido como ahora. Además que a nivel de trabajo, aún no había desarrollado ningún proyecto en esta nueva versión de *actionscript*. Lo cual se sumaba al problema de plazos de entrega, que de por sí, eran demasiado justos como para tener que trabajar sobre un lenguaje no utilizado anteriormente.

3.4.2. Tecnologías desestimadas

En cuanto a tecnologías contempladas como vías de desarrollo pero que no procedieron:

- Empleo de la **API de Google Maps dentro del entorno Flash** para el uso de mapas. Esto complicaría el tratamiento dinámico de determinados objetos del mapa acústico, como el mar, la tierra, el ruido representado en rojo de barcos y contaminación costera causada por pequeñas embarcaciones. Es por este principal motivo que se descartó el uso de la API de Google Maps.
- **Java:** Se desestimó al no ofrecer una interfaz para el tratamiento visual de los elementos como dispone Flash.
- **Silverlight:** Como herramienta multimedia, se pueden conseguir resultados muy similares a los de Adobe Flash. Pero en el inicio del desarrollo, estaba en una fase demasiado inicial, de modo que no se disponía de gran información sobre la herramienta ni de ejemplos y tutoriales.
- **Director:** Este software también es similar a Flash, aunque está orientado a soportes multimedia como CD's, DVD's, etc, más que orientado a web como Flash. Así que, pensando en las presentaciones sí que hubiese sido una buena herramienta de desarrollo. Pero en el momento en que se pensó en la web como medio igual de importante con el que difundir la aplicación, empezó a dejar de tener sentido el utilizar director como herramienta de desarrollo de la misma. Y más aún cuando la tendencia era dar cada vez menos soporte a esta herramienta, tanto a nivel de desarrollo de nuevas versiones, como por falta de interés y seguimiento por parte de los programadores.
- **Flex:** Esta herramienta es muy útil cuando pensamos en tratamiento dinámico de datos y presentación de los mismos mediante componentes. Pero, puesto que hablamos de una interfaz gráfica totalmente específica para esta aplicación, dejaba de tener sentido el tratar los contenidos con componentes ya predefinidos. Sólo hubiese sido útil en algunos módulos puntuales de la aplicación, que hubiesen podido sincronizarse bien con Flash, ya que utiliza el mismo plugin para ejecutarse. Pero para mostrar animaciones, el mapa

acústico y demás utilidades de la aplicación, no era necesario. Con lo cual se desestimó también.

3.4.3. Tratamiento de imágenes

Puesto que hablamos de una aplicación, básicamente centrada en los mapas acústicos, se decidió adaptar las primeras propuestas de diseños esquemáticos, a mapas reales retocados, empleando la herramienta de edición de imagen *Photoshop* Ref.4.

3.4.4. Preparación de datos: De excel a xml

Los estudios recogidos por el cliente, los tenían guardados en archivos de tipo Excel.

Dado que Flash no permite parsear datos directamente de ficheros Excel, se hizo una conversión de estos archivos a un formato que trabaja bien con Flash, el XML. Dado que los archivos XML ya fuerzan una estructura de sus contenidos, ayudó a poder interpretar rápidamente todos los contenidos visualmente desde XML a la hora de hacer testeos de los contenidos parseados e interpretados desde Flash.

También se aprovechó para adaptar los contenidos de excel a XML, con tal de facilitar su tratamiento posterior en Flash.

Otros contenidos de tipo texto, fueron transformados a XML para unificar el sistema de entrada de datos.

3.4.5. Simulación: Funcionalidad interactiva para la representación del solapamiento de frecuencias entre embarcaciones y cetáceos

Inicialmente, el mapa acústico era el único sistema para poder observar el nivel de contaminación acústica producido por las embarcaciones. Pero con este no se daba información sobre el solapamiento de ondas frecuenciales entre las embarcaciones y cetáceos.

Por este motivo se tomó la decisión de crear una animación con capacidad de interactuar entre la embarcación y el cetáceo elegidos.

Se decidió generar un entorno de falso 3D en la animación para dar más sensación de realismo a los elementos que se muestran.

3.4.6. Automatización de la carga de barcos

Debido al gran trabajo que supone animar manualmente cada uno de los barcos que acaban generando el mapa acústico de las zonas estudiadas de toda la costa peninsular, se optó por emplear un framework para Flash llamado *Greensock* Ref.2.

Este proceso de automatización se llevó a cabo como mejora del proyecto real entregado al cliente, adaptándolo así para el proyecto de fin de carrera (Fase 4).

Se empleó básicamente para dos aspectos:

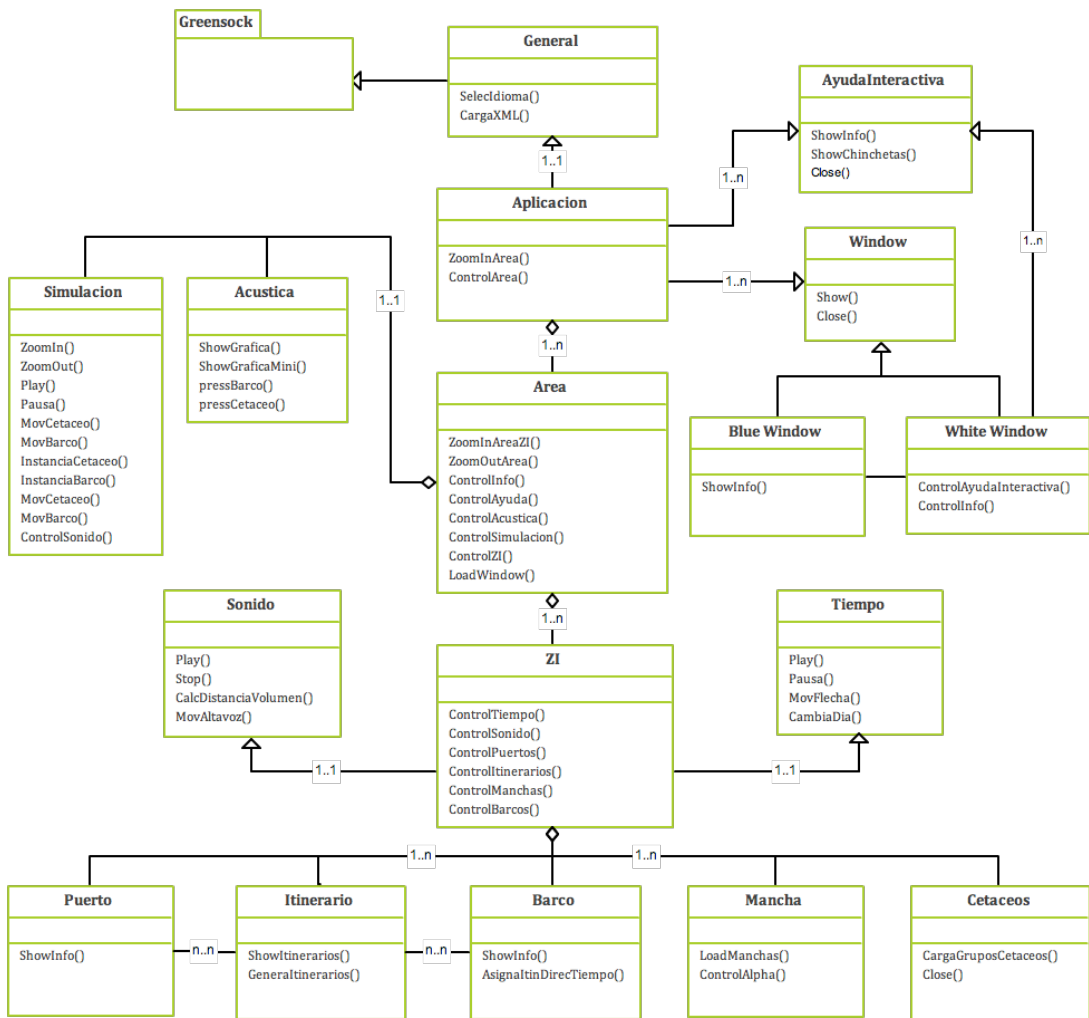
1. Crear una línea temporal, sincronizada con la línea de tiempo del mapa acústico, con la que controlar el tiempo de carga y descarga de barcos en pantalla.
2. Crear las interpolaciones de movimiento de los barcos por sus correspondientes itinerarios (curvas Bézier). Esto implicó suplir las líneas guía para animaciones de Flash por curvas Bézier. Y para ello, se creó la herramienta *generador de itinerarios*.

Como cada itinerario pasó a ser un conjunto de coordenadas que formaban las curvas Bézier, se crearon archivos XML para almacenar las coordenadas de las curvas.

3.5. Estructura/módulos de la aplicación

3.5.1. Diagrama de clases

Al hablar de *actionsript 2*, no existe una estructura de clases que englobe toda la programación utilizada, puesto que no es una programación 100% orientada a objetos, sino orientada en parte también a eventos. Así que a continuación se muestra un diagrama con la estructura de clases que podría tener la aplicación 100% orientada a objetos.



[online diagramming & design] creately.com

Diagrama de clases Ref.20.

3.5.2. Estructura de archivos XML

El siguiente diagrama muestra la estructuración de archivos XML que se ha empleado para la aplicación.

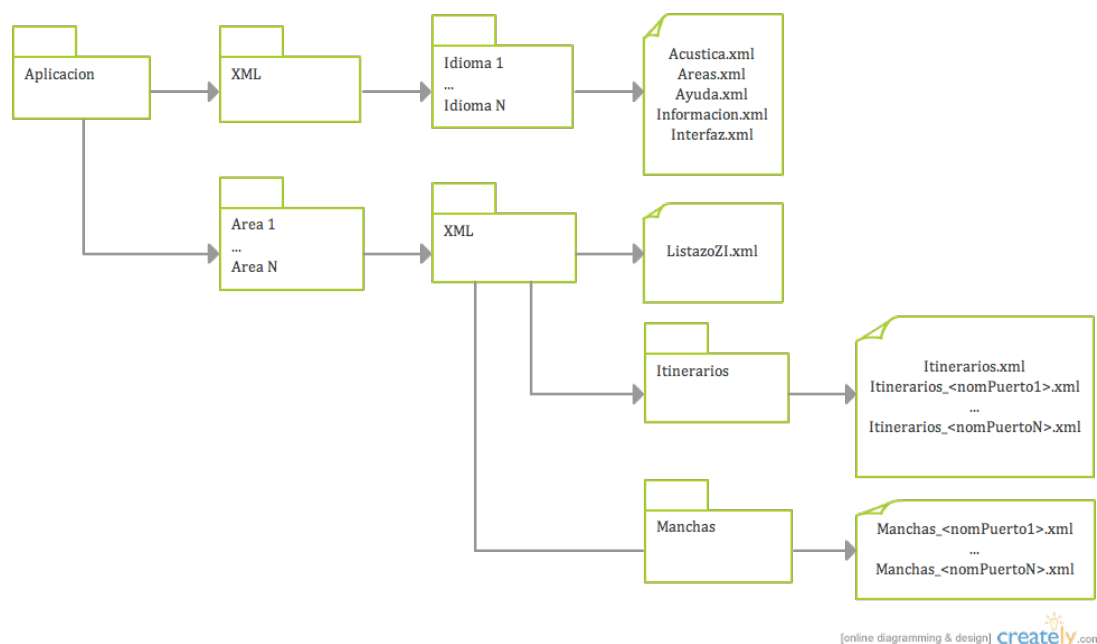


Diagrama de archivos XML Ref.20.

Toda la información que varía según el área y zona estudiada, es decir, la que se muestra en el mapa acústico, excepto la información y ayuda interactiva, se encuentra en el directorio XML dentro del área estudiada.

Por lo tanto, cada área estudiada tiene los siguientes archivos:

- *listado_zona_interes.xml*: por cada zona estudiada, hay un nodo zona. Este tiene dos nodos hijos:
 - Nodo de tipo itinerario: Es un conjunto de nodos por cada puerto de la zona. Cada nodo puerto, especifica la URL del XML que contiene los itinerarios relativos a cada puerto.
 - Nodo de tipo mancha: Al hablar de mancha nos referimos al nivel de contaminación global que se genera en un puerto, representado gráficamente como una mancha roja. Es un conjunto de nodos por cada puerto de la zona. Pero en este caso, no de puertos que contienen itinerarios, sino de todos los puertos que contienen manchas.

- *Itinerarios > Itinerarios.xml*: Contiene la información generada con la miniaplicación *GeneralItinerarios fla*.

Hay un nodo por cada puerto de la zona estudiada. Cada puerto tiene un conjunto de nodos itinerario que contienen las coordenadas de los puntos de control que forman las curvas bézier.

- *Itinerarios > Itinerarios<nombrePuerto>*: Contienen la información de todos los barcos que pasan por el puerto para cada día estudiado.

Hay un nodo por cada día estudiado, compuesto por nodos barco. Y cada nodo barco contiene la información necesaria para ser animado (fecha y hora de entrada y salida del puerto, y tipo de barco, que determinará su velocidad). También alberga información a mostrar cuando se clica sobre un barco.

- *Manchas > Manchas<nombrePuerto>*: Mantiene la misma estructura que el XML anterior. En este caso, en vez de dar información por cada barco, se da información de la contaminación que sufre cada puerto en el tiempo estudiado, indicando la intensidad de contaminación. Lo que se transforma en más o menos opacidad de la mancha roja.

El resto de archivos XML, están en el directorio XML principal, que contiene los archivos relativos a acústica, ayuda, información, áreas e interfaz.

3.5.3. Clases y métodos destacados

3.5.3.1. APLICACIÓN.FLA

En este archivo se han encapsulado funciones que no corresponden como tal a este nivel de la aplicación, sino que corresponden a las áreas y zonas de interés. Se ha hecho con el fin de no replicar código por cada área y zona de interés.

Como tenemos dos archivos Flash por cada área estudiada (area y areaZI), estos están libres de código, como se ha dicho anteriormente, con el fin de no replicarlo, puesto que el funcionamiento es idéntico entre una áreas y otras.

Cada punto siguiente hace referencia a una capa con código en el primer o segundo frame del archivo Flash. Cada capa de código hace referencia a cada funcionalidad de la aplicación, como indican sus nombres, pudiendo tener varias funciones.

- **Información**

- *controlInformacion()*: Gestionamos los eventos de ratón cuando se clica a la sección de información, sea desde acústica, simulación o pantalla inicial.
- *controlCargaInfo()*: En función de qué sección de información sea, se carga o descarga el panel de contenido de información.
- *cargaInfo()*: Se accede al contenido del XML que corresponde a la información que queremos mostrar, almacenándose en el objeto que se pasa por parámetros.

- **Ayuda**

- *controlAyuda()*: Se llama a *controlCargaAyuda()*, pasando el objeto según desde la sección que se haya accedido.
- *controlCargaAyuda()*: Se cargan las chinchetas de la ayuda interactiva, cargando un contenido distinto según la sección desde la que se acceda: simulación, acústica o zona de interés.

- **Acústica**

- *controlAcustica()*: Para cada tipo de barco y de cetáceo seleccionado, se carga su gráfica correspondiente. Las gráficas de cetáceos son directamente imágenes, y las de los barcos, se generan dinámicamente según los valores entrados para cada tipo de embarcación.

- **Simulación**

- *controlSimulacion()*: Se inicializan los datos y objetos de la simulación. Una vez seleccionamos un barco, se llama a *controlTiempoSimulacion()* y a *calculaDistancia()*.

Cuando se selecciona un cetáceo se llama a *simulacionMouseCetaceos()* y a *simulacionSonidoCetaceos()*.

- *calculaDistancia()*: Se recalcula la distancia entre barco y cetáceo tras hacer zoomIn y zoomOut, devolviendo el nivel de decibelios.
- *controlTiempoSimulacion()*: Comienza la animación del barco clicado, sincronizado con la línea de tiempo. Se gestiona el tiempo y el control de zoomIn y zoomOut de la animación. Se llama a *simulacionSonidoBarcos()*.
- *simulacionSonidoBarcos()*: Cargamos el audio del barco seleccionado. Al pasar con el puntero por encima de la onda del barco, y según la distancia calculada al centro del mismo, se varía el volumen del sonido del barco.
- *simulacionSonidoCetaceos()*: Cargamos el audio del cetáceo seleccionado.

- **Interfaz**

- *playBlurInterfazPrincipal()* y *stopBlurInterfazPrincipal()*: Hacen que el fondo se desenfoque o enfoque según si se carga o descarga una sección.

- **Zona interés**

- *controlCargaZonaInteres()*: Para cada área o zona de interés, se cargan sus nombres correspondientes y se llama a la función *zoomIn()* al clicar sobre un área o zona.
- *zoomIn()*: Una vez se hace el zoom hacia el área o zona seleccionada, se llama a *cargaZonaInteres()*.
- *cargaZonaInteres()*: Según si venimos del mapa peninsular o de una área, cargamos el archivo SWF del área o de la zona de interés seleccionada.

- **Carga XML**
 - *cargaXMLAplicacion()*: Se llama a una función que se encarga de cargar los contenidos de los XML relativos a cada módulo: información, ayuda, áreas, interfaz y acústica.
- **Sonido**
 - Se cargan los sonidos de todos los cetáceos y embarcaciones.
- **Funciones**
 - Tenemos varias funciones que se encargan de inicializar la interfaz global de la aplicación y del control del mapa en general: botones de acceso a las distintas áreas y panel para acceder a la ayuda e información.
 - Además tenemos la función *scaleEaseTo()* que se encarga de hacer los zooms de los mapas.

3.5.3.2. AREAX.FLA

Contiene las funciones para generar la gráfica de la sección de acústica: *cargaEstadisticas()*, *pintaMatriz()*, *pintaBarra()*, *pintaNumeros()* y *creaMatriz()*.

3.5.3.3. AREAXZI.FLA

Se instancia la zona de interés que se haya seleccionado en el área estudiada.

Todo el control de tiempo, barcos, puertos, itinerarios y cetáceos está programado dentro de las librerías de actionscript.

La ayuda interactiva e información están programados en *aplicacion.fla*.

3.5.3.4. GENERADORITINERARIOS.FLA

Archivo Flash encargado de generar los itinerarios (puntos de control) que trazan la trayectoria de la animación dinámica de los barcos. Esta es una funcionalidad añadida para automatizar la carga de barcos. No realizada durante el desarrollo del proyecto entregado a cliente, sino específicamente creado para el proyecto de fin de carrera (Fase 4).

Se encarga de generar el código XML que define las coordenadas de los itinerarios. Para ello se emplean curvas Bézier generadas en tiempo de ejecución.

Se utilizan como plantillas las imágenes previamente dibujadas en Flash para tener una referencia visual de la trayectoria que ha de seguir cada itinerario.

Siempre se toma como referencia el puerto estudiado, siendo el primer punto de control. Posteriormente se añaden los puntos de control necesarios para trazar cada itinerario.

Las curvas Bézier implementadas son de tipo cuadrático.

A continuación se explican las funciones que pueden encontrarse en la capa de código que hay dentro del archivo Flash:

- *Init()*: Inicializamos todos los puntos de control que formarán las curvas y variables de texto que generan el formato de etiquetas XML.
- *setIti()*: Cada vez que se clicca al botón “Añadir itinerario” se llama a esta función, que se encarga de concatenar el texto que se ha generado para formar el nodo XML itinerario.
- *addControlPoint()*: Se llama al crear el primer punto de control sobre el puerto estudiado y cada vez que queremos crear un nuevo punto de control. Este se añade después del último punto de control seleccionado. Se añade un nuevo punto en el array *controlPoints*.
- *removeControlPoint()*: Se elimina el punto de control seleccionado y se elimina también del array *controlPoints*.

- *redraw()*: Se encarga de redibujar la curva bézier cada vez que añadimos o quitamos un punto de control, cada vez que añadimos un itinerario y cada vez que desplazamos un punto de control. Para ello llama a las siguientes funciones: *drawCurvePoints()*, *drawLinePoints()*. También se refresca el texto que muestra en el SWF las coordenadas de cada punto de control.
- *drawCurvePoints()*: Se añaden 10 círculos a lo largo de la curva entre cada punto de control a distancia equidistante uno de otro. Para ello se llama a la función *pennerPointOnCurve()*. Sirve para trazar luego las líneas que crearán la curva, y también se aprovecha para dar información de lo que tardará en recorrer el barco cada tramo del itinerario en función de la separación entre estos círculos.
- *pennerPointOnCurve()*: Calcula mediante la fórmula de curva bézier cuadrática las coordenadas de cada punto que formarán la curva.
- *drawLinePoints()*: Dibuja las líneas entre los 10 círculos que hay entre cada punto de control.

3.5.3.5. LIBRERÍAS ACTIONSCRIPT PROPIAS

- **zonalInteres.as**

- *zonalInteres()*: En la clase *zonalInteres* se trata todo el funcionamiento del mapa acústico. Básicamente el control de tiempo, barcos, itinerarios, manchas y cetáceos. Los apartados de ayuda e información ya se ha explicado anteriormente que se han desarrollado en *aplicación.fla* de forma global para todas las zonas de interés.

Se declara la línea de tiempo *TimelineMax()*, los arrays que contendrán la información relativa a las manchas, barcos y puertos.

Se carga la información de itinerarios, barcos y manchas mediante *loadItinerarios()*, *loadInfoBarcos()* y *loadManchas()*. Y una vez cargados sus contenidos XML, inicia todo el control del mapa acústico a través de *barcosZI()*, *puertosZI()*, *controlZI()*, *StartManchas()* y *CopyToPortMC()*.

- *loadItinerarios()*: Parsea el XML de itinerarios mediante la función *parseaCoords()*. Cada nodo XML tiene las coordenadas de los puntos de control que forman cada itinerario. Creamos el array *itis* de itinerarios por cada puerto. Y un array *nomItis* de nombre de itinerarios por cada puerto, que servirá para asignar los itinerarios a cada barco.
- *loadInfoBarcos()*: Se parsea el XML *listadoZonaInteres*, único para cada área estudiada. Y por cada nodo puerto se obtiene la ruta del XML de todos los puertos relativos a la zona de interés. Parseamos la información de los itinerarios por cada puerto / XML. Creamos una instancia de la clase *infoBarco* y por cada día estudiado se almacena toda la información de cada barco en el array *informacionPuertosBarcos*.
- *sortBarcos()*: Ordenamos el array de barcos *informacionPuertosBarcos* por orden de entrada y salida de puerto.
- *barcosZI()*: Nos recorremos el array de barcos *informacionPuertosBarcos* y por cada barco asignamos su itinerario y lo instanciamos.
- *asignaItinerario()*: Recorremos el array *nomItis* para encontrar el itinerario que se asigna a cada barco.
- *instanciaBarco()*: Instanciamos en el escenario cada uno de los barcos del día estudiado. Según su dirección invertimos el orden de los puntos de

control para que vaya en dirección opuesta. Mediante la clase *TweenMax.as*, animamos el barco indicando el array de coordenadas de puntos de control, es decir, indicando la curva bézier que ha de seguir, indicando en qué momento comienza y termina la animación según los datos parseados y calculados de cada barco.

- *loadManchas()*: Se cargan las manchas por cada uno de los puertos de la zona estudiada. Para ello se utiliza la función *AfegirTaca()* de *infoPuerto.as*.
- *StartManchas()*: Creamos un intervalo de tiempo para llamar a la función *UpdateManchas()*, que llama a la función *Update()* de la clase *manchaPuerto.as* para actualizar la mancha.
- *controlZI()*: Llamamos a la funciones *controlSonidoBarcos()* y *controlSonidoManchas()*, que se encargan de preparar los audios para cuando entremos en modo acústico. También se controlan los eventos del panel de secciones de información, ayuda, cetáceos, modo acústico y cerrar. Así como el control de tiempo: play/pausa y cambio de día.
- *puertosZI()*: Al clicar sobre un puerto se carga la información de todos los barcos que pasan por el puerto a través de un componente DataGrid.
- *cambiaMouse()*: Cada vez que se llama a esta función, se intercambia el puntero del mouse por un altavoz o la flecha normal. Cuando está en modo altavoz, se detecta si hay colisión con tierra o mar a través de la clase *colision.as*, con tal de reproducir un sonido u otro.
- *controlSonidoBarcos()*: Se activa el sonido para cada barco, si pasamos por encima de un barco se reproducirá el audio de mercante. Puesto que todos los barcos que van por itinerarios son mercantes.

El nivel de volumen según la intensidad de contaminación se calcula desde la movie barco dentro del archivo *areaXzi fla*.

- *controlSonidoManchas()*: Se activa el sonido para cada puerto/mancha, si pasamos por encima de una mancha se reproducirá un audio concreto, dependiendo del tipo de embarcaciones que hayan en el puerto.

El nivel de volumen según la intensidad de contaminación se calcula desde la movie mancha dentro del archivo *areaXzi fla*.

- **infoBarco.as**
 - *setBarco()*: En el momento que desde *zonaInteres.as* llamamos a *setBarco*, se almacena en el objeto *barco* los atributos parseados del XML propios de cada barco, y además se asignan unos valores u otros dependiendo de si hablamos de un puerto que llega a puerto o sale de puerto:
 - Tiempo: se asigna el tiempo parseado directamente de XML o el obtenido mediante la función *calculaTiempo()* dependiendo de si llega a puerto o sale de puerto.
 - Dirección: será 1 ó -1 dependiendo de si entra o sale. Nos sirve para asignar el itinerario a cada barco con los puntos tal cual están almacenados o en orden invertido para que el barco lo recorra al revés.
 - Itinerario: Nombre del itinerario que se asigna al barco, que será el campo origen o destino, dependiendo de si entra o sale de puerto.
 - *calculaTiempo()*: Devuelve la hora en que llega el barco a puerto. Para ello se resta la duración al tiempo de llegada, la duración depende de la velocidad, y por tanto del tipo de barco.
- **colision.as**
 - *colision()*: Calcula la colisión en base al nivel de opacidad del objeto con el que se colisiona. Siempre será una mancha o barco, por esto importa el nivel de opacidad, pues el volumen dependerá de estos valores.
- **infoPuerto.as**
 - *AfegirTaca()*: A partir de los datos pasados a la función y obtenidos del XML de manchas, por cada mancha se instancia la clase *mancha.as*.
- **mancha.as**
 - *mancha()*: En base a los datos propios de cada mancha, se calcula el tiempo de inicio, fin e intensidad de la mancha.
- **manchaPuerto.as**
 - *Update()*: Se encarga de actualizar el nivel de opacidad de la mancha según la intensidad que tenga en cada momento. Se llama según el intervalo marcado para actualizar las manchas en *zonaInteres.as*.

3.5.3.6. LIBRERÍAS ACTIONSCRIPT GREENSOCK REF.9

Es un conjunto de librerías estilo la clase *Tween* de Adobe. Se emplea para hacer interpolaciones y construir líneas de tiempo por código. Así como seguimiento de objetos por líneas Bézier.

Estos dos últimos puntos son los que han permitido hacer la carga dinámica de barcos en la fase 4 del proyecto.

3.6. Fase de desarrollo del proyecto

El proyecto se dividió en cuatro fases de desarrollo. En cada una de ellas se añadieron mejoras y fueron desarrollando las distintas zonas estudiadas.

El cliente realizó una serie de ruedas de prensa por todo el litoral presentando en cada comunidad autónoma cada una de las zonas estudiadas. Por este motivo, las entregas del proyecto iban directamente ligadas a las ruedas de prensa.

3.6.1. Fase 1

En la primera fase, se definió la base del proyecto a realizar, centrándose en los datos obtenidos en la comunidad autónoma de las Islas Baleares, concretamente se estudió la zona del canal entre Ibiza y Formentera.

Especificaciones del desarrollo realizado durante la primera fase:

- Conceptualización y desarrollo del diseño
- Estructuración visual de contenidos
- Parseo de contenidos dinámicos a XML (Información relativa a los barcos)
- Mapa acústico de la zona estudiada (4 días estudiados). Donde se muestra la información de cada barco y puerto.
- Simulación de las embarcaciones en falso 3D para ver y escuchar el nivel de contaminación acústica. Disponiendo de una barra temporal con cursor deslizable y una herramienta de zoom, para poder observar el radio de contaminación generado por la embarcación seleccionada.
- Especificaciones de las embarcaciones.

3.6.2. Fase 2

Las ruedas de prensa se comenzaron en la segunda fase, ya que ahora ya había una versión estable de la aplicación.

Nuevas zonas de estudio y mejoras que se desarrollaron:

- Paso de un idioma a dos. Implicó crear una series de archivos XML para externalizar los contenidos.
- Ayuda interactiva para ver información sobre todos los controles de la aplicación.
- Nueva área de interés estudiada, la comunidad autónoma de Cataluña, en la que se estudian tres zonas de interés.
- Información de cetáceos en especificaciones.
- Incorporación de cetáceos en la animación de falso 3D para comprobar la solapación de ondas con los mercantes.

La solución fue susbtituir el ratón por el cetáceo seleccionado, dando así libertad de movimiento al cetáceo, y pudiendo cambiar en cualquier momento el tipo de embarcación. De esta forma, se podía ver y escuchar este cruce de frecuencias entre cetáceos y embarcaciones.

- Creación de un sistema visual y auditivo de contaminación en las costas, causado por pequeñas embarcaciones. Sistema similar al de los barcos animados del mapa acústico.
- Animación de barcos en el mapa acústico, ahora también con sistema de audio para percibir el nivel de contaminación acústico a través de la escucha.
- Sistema de audio interactivo. Para ello se empleó el ratón como medio para explorar las zonas de mayor y menor nivel de contaminación.

3.7. Metodología de desarrollo

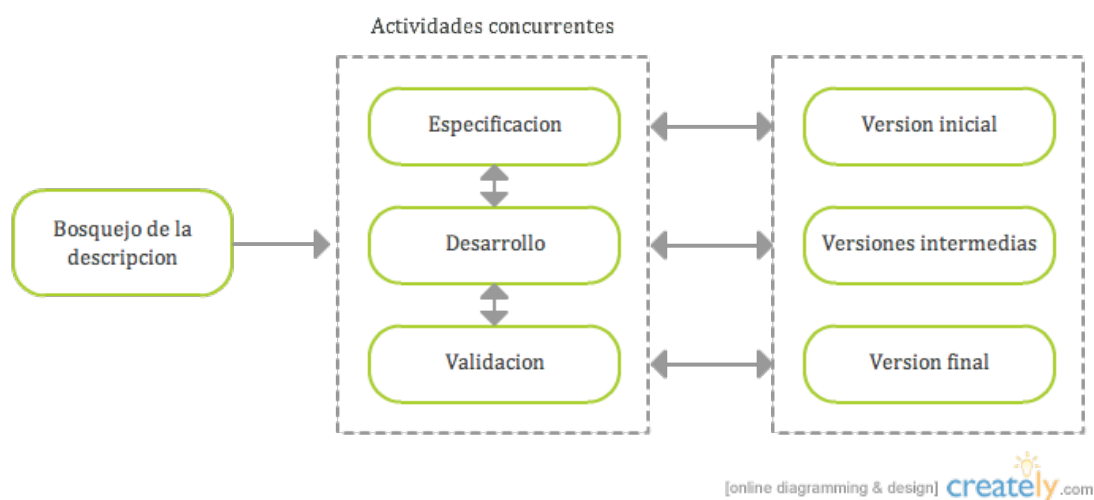
A lo largo del desarrollo de todo el proyecto, se llevaron a cabo distintas estrategias de trabajo. En cada fase, la situación era distinta, con lo cual, citaremos las diferentes metodologías aplicadas según la fase del proyecto.

3.7.1. Fase 1

En la fase inicial, se tenía como objetivo hacer poco más que una demo de lo que sería la aplicación final.

Por parte del cliente, no se disponía de unas especificaciones cerradas, ya que las funcionalidades que ofrecía la aplicación, se iban mejorando según los avances de la misma. Por este motivo, era necesario un feedback casi constante con el cliente. Esto implicaba ir generando versiones rápidas de desarrollar, de este modo se minimizaban los riesgos de desarrollo por malas especificaciones.

El conjunto de situaciones expuestas anteriormente apuntaron a emplear el **Modelo de Desarrollo Evolutivo**.



Modelo de desarrollo evolutivo.

Concretamente en el modelo de desarrollo evolutivo exploratorio se comienza a desarrollar las partes bien especificadas por el cliente, como era el caso de la principal funcionalidad del mapa acústico: La animación de barcos. De este modo el cliente veía resultados de forma rápida. Y la aplicación fue evolucionando conforme se añadían nuevas características propuestas por el cliente o mejoras de las funcionalidades ya existentes.

Explicando más en profundidad las distintas fases del modelo al crear nuevas funcionalidades:

- **Especificación:** Se concertaban reuniones con el cliente para exponer los problemas y mediante participación conjunta se proponían soluciones y nuevas ideas.
- **Desarrollo:** Se acababan de conceptualizar las ideas, estudiando la viabilidad a nivel tecnológico para llevarlas a cabo. Luego se comenzaba a trabajar en la preparación de bocetos, estilo de la interfaz, sistemas de navegación, y en general, desarrollo de la aplicación.

Por otro lado, se adaptaba el material a la aplicación. En el caso del mapa acústico:

- Excel con datos estudiados > XML
- Imágenes de rutas > Generación de itinerarios
- Regiones estudiadas > Generación de mapas
- **Evolución:** Mediante múltiples entregas se iba refinando el trabajo realizado, volviendo a repetir los procesos anteriores hasta dar con la versión definitiva.

3.7.2. Fase 2

En la segunda fase del proyecto habían nuevas funcionalidades que desarrollar y otras que, simplemente, tenían que ampliarse o mejorarse. Por este motivo, la metodología a aplicar debía adaptarse según fuesen nuevas herramientas o mejorar las ya existentes.

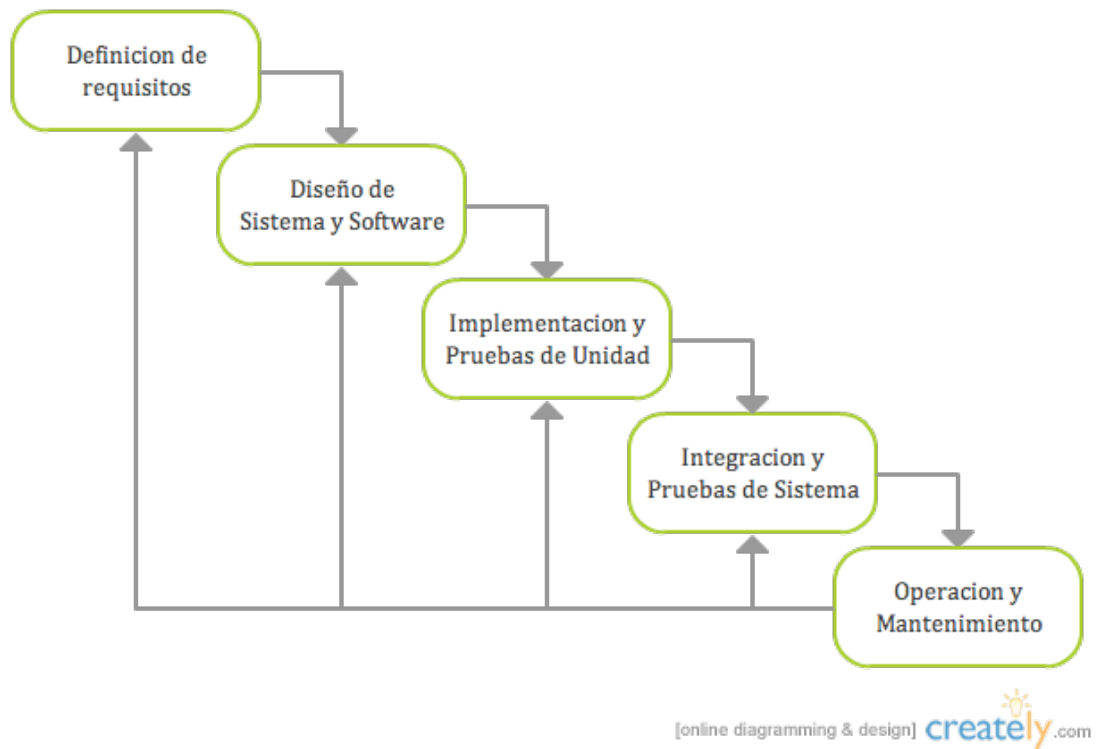
Para las nuevas funcionalidades se siguió utilizando el mismo modelo. Puesto que era necesario una gran comunicación con el cliente y preparar pequeñas entregas para hacer seguimiento de la evolución del trabajo. Las principales herramientas de la aplicación que debían desarrollarse eran la simulación acústica en falso 3D, el sistema de ayuda interactiva, las gráficas del sistema auditivo de cetáceos y barcos, y algunas nuevas funcionalidades en el mapa acústico.

La metodología utilizada hasta este punto, también provocó un aspecto negativo: la falta de documentación, debido a la necesidad constante de preparar entregas al cliente, quitando por tanto, tiempo para generarla.

Las ampliaciones, son relativas al mapa acústico. Esta parte del trabajo ya no requería de una presentación constante de nuevas versiones a cliente. Los requisitos ya estaban bien definidos. Con lo cual, el objetivo era desarrollar las nuevas zonas de interés siguiendo siempre el mismo proceso.

Cada nueva zona estudiada era presentada al cliente para comprobar su funcionamiento correcto, y si era necesario, hacer las adaptaciones pertinentes.

El **modelo de desarrollo en cascada** cumple con las características comentadas anteriormente. Por lo tanto, podemos decir que este fue el modelo que se empleó para ampliar el mapa acústico con nuevas zonas estudiadas.



Modelo de desarrollo en cascada.

3.7.3. Fase 3

En la tercera fase, y última entregada al cliente, se mantiene el mismo sistema de trabajo. Se emplean los dos modelos, puesto que hay una nueva funcionalidad a parte del desarrollo de nuevas áreas de estudio.

La nueva funcionalidad es la construcción del mapa peninsular como sistema de navegación entre las distintas áreas estudiadas. Durante este proceso, eran necesarias continuas revisiones por parte del cliente, así que el modelo de desarrollo evolutivo, es el que se utilizó.

El resto de trabajo para finalizar esta fase del proyecto, fue más repetitivo. Volvemos a hablar de especificaciones cerradas. Por lo tanto, el modelo de desarrollo en cascada era el más apropiado. No hicieron falta iteraciones en las distintas fases del modelo, puesto que el riesgo a tener que hacer correcciones era prácticamente nulo.

3.7.4. Fase 4

Para el trabajo de la automatización en la animación de barcos dentro del mapa acústico, se tuvo que rediseñar alguna estructura y métodos de la clase *zonaInteres.as*. Se aprovechó para eliminar código de los archivos Flash, y así controlarlo todo desde la propia clase. Este proceso requirió el tener que revisar gran parte del código antes ponerse con el desarrollo.

Aunque, como en esta fase, todo el desarrollo era interno y no afectaba al resultado de la aplicación más que en el rendimiento, ni tenía que hacerse un seguimiento externo, como en su momento tuvo que hacer el cliente, no fue necesario ceñirse a una metodología de trabajo específica.

4.2. Rendimiento de la aplicación

La aplicación muestra distintos comportamientos según la funcionalidad que se esté ejecutando. Sobretodo a la hora de acceder a un mapa acústico, el coste computacional es bastante más elevado que en otras funcionalidades de la aplicación. Además, requiere un tiempo previo a la animación de barcos para hacer los cálculos necesarios que requiere la animación de manchas. Pues los datos que se obtienen de XML necesitan de cálculos posteriores para poder hacer las animaciones pertinentes. Aunque estos cálculos no afectan luego al tiempo de ejecución.

El mapa acústico, anteriormente estaba desarrollado mediante un proceso manual de animación. Y cuando había un gran número de barcos desplazándose al mismo tiempo, afectaba al rendimiento de forma considerable. Esto se debe a la cantidad de cálculos que han de procesarse en cada iteración por cada barco.

Al desarrollar la cuarta fase del proyecto, en la que los barcos se animan de forma dinámica mediante el uso de la librería Greensock *Ref. 2*, se consigue disminuir considerablemente el coste computacional.

Se ha de tener en cuenta que el rendimiento de la aplicación también varía dependiendo de cómo se esté ejecutando. El player mediante plug-in de navegador suele tener un rendimiento inferior al del player local.

En las ruedas de prensa, para evitar posibles problemas de conexión y de rendimiento se ejecutaba una versión offline, directamente sobre el player local de Adobe Flash.

4.4.1. Personal que colaboró en el proyecto

En el proyecto habían dos equipos, uno formado por el laboratorio y otro por Sicalipsis.

Por parte del cliente, se encargaban recolectar los datos estudiados y de preparar el material cedido a Sicalipsis. Para ello utilizaron un barco con el cual hicieron mediciones a lo largo de todas las zonas estudiadas de la península:

- Michel André *Ref.23*: Director del laboratorio y líder del proyecto global. Ingeniero Superior en Biotecnología, Licenciado en Bioquímica y Fisiología Animal, Doctor en Biología.
- Alex Mas: Investigador en el laboratorio. Ingeniero en Biomédica.
- Maria Morell: Investigadora en el laboratorio. Ingeniera licenciada en Biología.
- Mike van der Schaar: Investigador en el laboratorio. Ingeniero licenciado y master en Matemáticas Aplicadas.

Por parte de Sicalipsis, las personas que colaboraron son:

- Marcos de la Torre: Jefe de proyecto. Diseñador y programador de la aplicación en las cuatro fases.
- Grisel Rodríguez: Colaboradora en la animación manual de barcos de ciertas zonas de estudio y en la mejora de la programación de la Fase 3.

5. CONCLUSIONES Y LÍNEAS DE FUTURO

5.1. Conclusiones

Comenzando por un aspecto negativo, podemos hablar de los problemas que acarrea el no disponer de un timing ajustado al tiempo que se ha de invertir en el planteamiento y desarrollo. Pues esto implicó fallos en el enfoque de la programación, impidiendo la escalabilidad a la hora de estudiar nuevas áreas.

Aún y así, hubo un trabajo en equipo bastante satisfactorio y bien sincronizado por parte de cliente y desarrolladores. Puesto que las propias ruedas de prensa forzaban el seguimiento y entregas periódicas según iban ampliándose las áreas estudiadas.

Con el resultado estoy contento, ya que tuvo un gran impacto, y como herramienta, cumplió con las expectativas del cliente.

La finalización del proyecto la provocó el no disponer de fondos, al dejar de financiar la Obra Social el proyecto global. Eso hizo que por parte del cliente, hubiese cierta decepción, pues no pudo completarse el mapa acústico de la costa peninsular por falta del estudio de la costa sureste. Pero en cuanto al resultado de la aplicación en sí, y del éxito mediático que consiguió, sí que quedó satisfecho.

Quizás el readaptar la aplicación, carezca de sentido, mientras no hayan datos de estudio con los que poder recrear posteriormente las animaciones de los barcos y completar el mapa acústico.

A nivel interno, tendría sentido si fuese una aplicación en la que tuviese que colaborar más gente a nivel de desarrollo. Pero en la fase que quedó, no sería necesario. Así que a nivel de readaptar la aplicación, se podría hablar más bien de añadir nuevas funcionalidades, como podría ser un proceso de socialización de la herramienta. De este modo se conseguiría difundir la aplicación a mucha más gente. Con lo cual, lo que quizás valdría la pena, es hacer campañas de marketing on-line para difundir la herramienta.

Desde el principio del desarrollo, no se tuvo demasiado en cuenta la escalabilidad del proyecto, pues comenzó con un mapa de una zona estudiada. Luego se pasó a tener un nuevo nivel de navegación en forma de mapa para elegir varias zonas estudiadas, no solamente una. Y posteriormente, un nuevo nivel de mapa para navegar por distintas áreas estudiadas, que a su vez tenían distintas zonas.

Esto dificultó el crear una estructura de archivos y programación con suficiente flexibilidad como para evitar la réplica de código y archivos al añadir nuevas áreas y zonas estudiadas, como acabó sucediendo. Aunque en parte sí que se readaptó el código para evitar al máximo tener código replicado en distintas áreas.

La aplicación se difundió en distintas ruedas de prensa. Así que podemos decir que la aplicación sirvió como herramienta comercial mientras duraron las campañas.

5.2. Líneas de futuro

Puesto que hablamos de una aplicación con múltiples herramientas y pionera en su ámbito, es fácil encontrar una serie de mejoras y ampliaciones en la aplicación que la enriquecerían aún más.

5.2.1. Mejoras mediante marketing on-line

- **Proceso de socialización de la aplicación:** Por razones ajenas al laboratorio que llevaba a cabo los estudios, como ya se ha comentado anteriormente, dejó de financiarse el proyecto. Por este motivo, no llegó a proponerse la creación de campañas a través de internet para difundir el uso de la aplicación, y así poder llegar al máximo de usuarios posibles.

Por lo tanto, a nivel comercial podría haberse hecho uso de las redes sociales para mejorar la difusión de la aplicación a través de internet. Se podrían haber hecho una serie de adaptaciones en la aplicación facilitando así el proceso de socialización:

1. Añadir enlaces a redes sociales y bloggers.
2. Creando páginas en las redes sociales destacadas como Facebook, Youtube, Blogger, etc.
3. Crear un blog en el que los usuarios pudiesen participar, dando sus opiniones sobre los temas relacionados con la campaña.

Creación de un viral: Manteniendo el objetivo de ganar protagonismo en la red, otro método para conseguirlo sería creando un buen viral. Es decir, encontrar una idea lo suficientemente creativa e innovadora para que diese de qué hablar en blogs y redes sociales, difundiéndose así la aplicación.

5.2.2. Mejoras en la aplicación a nivel de desarrollo

- **Posicionamiento web:** Cuanto mejor se indexasen los contenidos de la aplicación, mayor posicionamiento se conseguiría. Para ello, sería necesario readaptar parte de los contenidos para mejorar su indexación en buscadores, así como crear un mapa web de las distintas secciones / mapas acústicos. Otro sistema para mejorar el posicionamiento sería mediante enlaces a web's que pudiesen estar relacionadas de algún modo con el estudio, la obra social en sí, etc. Y también consiguiendo, del mismo modo, tener enlaces entrantes de web's con un alto posicionamiento. Por descontado que cuanto mayor participación en las redes sociales, también influiría en el número de visitas y por consiguiente en el posicionamiento.
- **Sistema de geolocalización mediante Google Maps:** Se podría añadir un sistema de geolocalización. El nivel de sincronización entre Google Maps y la aplicación sería el aspecto más delicado, pues los mapas creados no tienen ningún tipo de almacenamiento de coordenadas para poder adaptarlos a los mapas de Google Maps. Así que la comunicación entre aplicación y Google Maps en este aspecto sería bastante limitada.
- **Simulación en 3D real:** Actualmente, hay engines 3D que trabajan sobre Flash, como por ejemplo PaperVision3D. Este engine podría utilizarse para crear una animación mucho más realista, incluso añadiendo una mejor interacción entre embarcaciones / cetáceos – usuario. Ya que podríamos representar el nivel de contaminación de forma tridimensional. Con esto se pasaría de dar una información aproximada a una información prácticamente del todo realista.
- **No limitar a cuatro los días estudiados:** Como ideal, se deberían mostrar los datos de los días que el usuario quisiese escojer. Aunque los datos de los que se dispone pertenecen a fechas concretas del estudio.

- **Modificación del sistema de navegación entre zonas estudiadas:** En vez de jugar con el zoom in y zoom out para cambiar de zonas estudiadas y ver así los distintos mapas acústicos, podría hacerse un mapa acústico global por cada área estudiada, y navegar entre las distintas zonas estudiadas simplemente desplazándonos por el mapa. Esta mejora ya se contempló, pero debido que hablamos a una gran cantidad de animaciones concurrentes, esto afectaría considerablemente al rendimiento de la aplicación, y la solución fue acotar estas animaciones separando las áreas en varias zonas de estudio.

Ahora que se ha conseguido minimizar el coste computacional al hacer la carga dinámica de barcos en la fase 4, quizás sí que se podría prescindir del nivel intermedio de mapas, es decir, tener un mapa peninsular y de este acceder directamente al mapa acústico de cada área estudiada, en vez de acotarlo por zonas de interés.

- **Paso de *actionscript 2* a *actionscript 3*:** Uno de los procesos costosos, pero que mejorarían el rendimiento y sobretodo la estructuración del código, sería pasar de *actionscript 2* a *actionscript 3*. Al conseguir un código más limpio y modular, se facilitaría en gran medida la interpretación del código, pensando en mejoras de código, incluso por nuevos desarrolladores. Ya que todo el código estaría contenido en librerías actionscript y, a su vez, estructurado por clases.

El paso principal sería extraer todo el código que hay dentro de los archivos Flash y crear librerías actionscript a partir de estos.

Actualmente por cada área estudiada hay dos archivos de tipo Flash, uno para el área en sí y otro para las zonas estudiadas de cada área. Esto podría evitarse, dejando dos únicos archivos Flash, uno de áreas y otros de zonas de interés únicos para todas las áreas de interés.

5.2.3. Posible método de desarrollo alternativo

Pensando en el mapa acústico, un sistema que actualmente es inviable debido a la limitada o incluso nula facilidad con la que se dispone de los datos de tráfico marítimo, sería la creación de una aplicación *Mashup* Ref.17.

Esta aplicación podría constar de un mapa desarrollado bajo la API de Google Maps que mostrase el nivel de tráfico marítimo que hay en cada uno de los puertos. Simulando mediante imágenes que representasen la contaminación en base a los datos recogidos de cada puerto y/o embarcaciones. De este modo podría obtenerse un mapa con datos obtenidos incluso en tiempo real. Aunque para ello, se debería disponer de una base de datos que tuviese de forma unificada toda esta información, lo cual no existe a días de hoy, y menos aún, en tiempo real.

Mediante controles de tiempo, podría obtenerse la información según el día y hora escogida. De modo similar al que se hace en la aplicación actual, pero sin estar limitado a los 4 días estudiados de un año determinado.

Además, si se dispusiese de estos datos en tiempo real, podría hacerse un histórico de datos con los que se podrían hacer estadísticas, y comprobar la variación de contaminación acústica conforme pasa el tiempo.

6. BIBLIOGRAFÍA

1. http://www.fundacio.lacaixa.es/ambitos/lacaixaafavordelmar/mapaacustico_ca.html
2. <http://www.greensock.com/>
3. <http://www.adobe.com/es/products/flash/>
4. <http://www.adobe.com/es/products/photoshop/>
5. <http://code.google.com/intl/es/apis/maps/documentation/flash/>
6. http://es.wikipedia.org/wiki/HTML_5
7. <http://slides.html5rocks.com/>
8. <http://code.google.com/p/tweener/>
9. <http://blog.greensock.com/tweening-speed-test/>
10. <http://www.monografias.com/trabajos29/contaminacion-acustica/contaminacion-acustica.shtml>
11. <http://www.sonystyle.com.mx/lounge/>
12. <http://smokescreen.us/>
13. <http://listentothedeep.net/acoustics/index.html>
14. <http://www.sonsdemar.eu/sonsdemar.php>
15. <http://www.localizatodo.com/>
16. <http://www.lab.upc.es/>
17. [http://es.wikipedia.org/wiki/Mashup_\(aplicaci%C3%B3n_web_h%C3%ADbrida\)](http://es.wikipedia.org/wiki/Mashup_(aplicaci%C3%B3n_web_h%C3%ADbrida))
18. <http://seamap.env.duke.edu/>
19. <http://www.w3.org/>
20. <https://creately.com>
21. http://es.wikipedia.org/wiki/Curva_de_B%C3%A9zier
22. <http://www.w3.org/TR/2000/CR-SVG-20001102/paths.html>
23. <http://www.lab.upc.es/index2.php?id=2&web=personal&lang=es>
24. <http://www.silverlight.net/>
25. http://www.microsoft.com/expression/products/blend_overview.aspx
26. <http://www.adobe.com/es/products/flex/>