



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación :

INGENIERO TÉCNICO DE TELECOMUNICACIÓN,
ESPECIALIDAD EN SONIDO E IMAGEN

Título del proyecto:

DISEÑO E IMPLEMENTACIÓN DEL BACK-END DEL SITIO WEB
DEL PROYECTO AURALIZARTE. HTML5, JQUERY, PHP,
SQLITE

Eduardo García Lara

Tutor: Ricardo San Martín Murugarren

Pamplona, 21 de febrero de 2014

Tabla de contenido

Estado del arte	
La web antes.....	3
La web ahora.....	4
La web en el futuro (no muy lejano).....	5
Resumen	
Introducción.....	7
Capítulo 1: "ANÁLISIS DE REQUISITOS"	
Análisis de los requisitos de la web	8
Ideas rechazadas	9
Capítulo 2: "TECNOLOGÍAS A UTILIZAR"	
Tecnologías disponibles: CMS o diseño a medida.....	13
Lenguajes de programación web	14
Frameworks utilizados.....	16
Capítulo 3: "CONCEPTUALIZACIÓN"	
Base de datos relacional.....	17
Lado del servidor	19
Capítulo 4: "DISEÑO DE LA APLICACIÓN"	
Lado del cliente (panel de administración).....	27
<i>Introducción.....</i>	27
<i>Usabilidad y experiencia de usuario: Fundamentos , wireframe y diseño final.....</i>	28
Capítulo 5: "IMPLEMENTACIÓN"	
PHP, HTML y la función ECHO.....	33
Variables de formulario, POST y GET	34
Peticiones a la base de datos SQLITE	36
Estructuras de control IF y WHILE.....	39
Funciones.....	41
Include	42
Sesiones.....	42
Capítulo 6: "CONCLUSIONES Y LÍNEAS FUTURAS DE TRABAJO"	
Conclusiones sobre el trabajo realizado	44
Conclusiones personales.....	45
Líneas futuras de trabajo	46
Referencias bibliográficas	
ANEXO	

Estado del arte

No hace tanto tiempo, el diseño web no era lo que es ahora. La tecnología e internet han avanzado enormemente en los últimos 10 años y con ella las necesidades de los usuarios. Actualmente las páginas webs no solo tienen que satisfacer la curiosidad y necesidad de los internautas, sino también sus gustos estéticos. El umbral de satisfacción del usuario medio está muy por encima del uso y diseño de las antiguas webs, compuestas por títulos, listas y tablas, pero, ¿por qué?

La respuesta es un cambio, a lo largo del tiempo, en el modelo de negocio de las empresas en internet.

La web antes

Hace unos años, la función de una web era simple: ser una carta de presentación de una empresa. Pocas personas tenían el poder económico y medios para comprar un dominio en internet y crear un sitio web personal. Las empresas comenzaron publicando sus catálogos en internet, dando a conocer sus servicios. Podían incrementar la cartera de clientes y por tanto los beneficios.

Las páginas web estaban diseñadas con elementos simples de hipertexto (HTML). Los programadores web se preocupaban únicamente de que su código fuese limpio y que utilizase pocas clases e identificadores (The Smashing Magazine, 2013). En definitiva, eran páginas estáticas con poca navegación, en las que el mensaje sólo se transmitía en una sola dirección, de empresa a cliente.

Pronto muchas empresas se dieron cuenta que internet era el futuro de su negocio. La demanda hizo que surgiesen lenguajes de programación más avanzados que permitían no solo mejorar la estética de una web sino también la interacción del usuario con la misma.

Gracias a la aparición de CSS (hojas de estilo), los programadores podían realizar páginas más rápido, con una buena estética, más grandes y en definitiva mejor. CSS fue un cambio en la tecnología que muchos programadores y diseñadores esperaban y por ello se convirtió en un estándar web.

Lenguajes del lado del servidor como PHP permitieron crear páginas dinámicas, conectadas a una base de datos, en las que el usuario podía interactuar con la web (primeros foros por ejemplo). El canal de comunicación dejaba de ser unidireccional a ser bidireccional empresa-cliente-empresa.

Las empresas comenzaron a modificar su modelo de negocio, entendiendo el potencial de ofrecer sus servicios a través de internet, reduciendo los costes (recursos humanos, materiales, geográficos, etc.) y sin que el cliente se mueva de su casa. El cambio era inevitable: no tener una presencia online supondría para cualquier empresa una desventaja competitiva.

Esto hizo evolucionar las antiguas páginas web en pocos años a portales con cientos de páginas internas que generan billones de dólares al año.

La web ahora

En la actualidad cualquier web conocida es compleja, grande y genera muchas ganancias. Amazon, en el ejercicio de 2012, generó alrededor de 60 billones de dólares de beneficio (Bloomberg, 2014), Twitter registró más de 200 millones de usuarios activos (Twitter, 2013) y Google por ejemplo dio empleo a casi 54.000 personas (Google, 2013). Ya no tiene sentido tratar a las webs como se hacía quince años atrás.

La aparición de los Smartphones y las tabletas han hecho evolucionar el diseño web hacia la sencillez y la usabilidad, centrándose en el usuario. Existen infinidad de nuevas técnicas para conseguir una experiencia de usuario óptima. Cualquier web en la actualidad tiene que ser accesible desde un ordenador, una tableta o un Smartphone sin sacrificar sus características; esto podría suponer menos beneficios para la empresa. El programador de hoy ya no se puede ceñir únicamente a que su código sea limpio y legible.

Por todo ello los desarrolladores web tienen que adoptar otro cambio de actitud para tener en cuenta a las tres partes implicadas: el cliente, el usuario y el desarrollador.

Es fundamental mantener presente estos tres pilares, ya que para ellos la web es lo más importante.

- El cliente: es la persona o empresa que paga para que se realice la web que desea.
- El usuario: es la persona que utiliza la web del cliente que ha pagado para construirla.
- El desarrollador: es la persona que tiene que mantener y hacer crecer el portal web.

El cliente no entiende de semántica ni programación. No le importa las técnicas que se utilicen, ni si el código es reusable. Tampoco es su responsabilidad.

Lo que verdaderamente importa al cliente es cómo de rápido y eficiente se puede construir su web, si se puede actualizar fácilmente, si es compatible con todos los dispositivos y navegadores y, sobre todo, el coste de realizarla.

Al usuario tampoco le importa el código, muchas páginas web que visitamos diariamente no tienen un código muy ordenado. Incluso a los desarrolladores, en su papel como usuarios, no se preocupan de si Twitter o Google están bien escritos.

El usuario se preocupa de si la web es rápida y de si sus servicios (en caso de que los ofrezca) sean de confianza. El código de una web es como el cemento en un edificio, es muy importante, pero las personas no suelen fijarse en él.

El desarrollador quiere saber cómo de documentado está el código, si es fácil trabajar con él, si se puede actualizarlo y escalarlo, si puede realizar cambios rápidamente o cómo de fácil puede resolver los problemas planteados por el cliente.

Al usuario y al cliente no les importa si el desarrollador utiliza más código del necesario o si el código no es limpio. En realidad los únicos que se preocupan por esto son los desarrolladores.

El éxito del desarrollador actual radica en mantener estos tres pilares en armonía.

La web en el futuro (no muy lejano)

El boom de las redes sociales ha creado que los internautas estén interconectados entres sí sin importar el lugar geográfico. No solo están interconectados sino que comparten en todo momento sus gustos, aficiones, sentimientos y en definitiva sus necesidades. Las empresas lo saben y ya empiezan a sacar partido de ello modificando una vez más su modelo de negocio.

¿El futuro de las webs? Según Paul Adams (diseñador de Google, Youtube o Facebook), las webs del pasado giraban en torno al contenido (Newenham, 2013); las webs del futuro (no tan lejano) girarán en torno a las necesidades de las personas. Las páginas webs estarán personalizadas a cada usuario, mostrando únicamente sus preferencias y mostrando información que le interese tanto a él como a su círculo de amigos.

Ticketmaster es un ejemplo de ello. Cuando un usuario compra una entrada en Ticketmaster, muestra si sus amigos de Facebook han comprado entradas y, si coinciden en el espectáculo, dónde están sentados (Newenham, 2013).

Etsy, una tienda online de productos realizados a mano por artesanos independientes de todo el mundo, es capaz de aconsejarte regalos para tus amigos de Facebook dependiendo de sus gustos (Newenham, 2013).

En el pasado una web comunicaba lo que tenía que comunicar, para todos los usuarios por igual. En el futuro una web conocerá perfectamente lo que quieras saber sin necesidad de buscarlo, en todo momento y a través de cualquier dispositivo. El desarrollador web tiene que estar en continua actualización, tanto de técnica como de conocimientos, porque de él depende que los clientes, los usuarios y los propios desarrolladores estén satisfechos con los resultados de su trabajo.

Resumen

La presente memoria detalla el análisis, conceptualización, diseño e implementación del entorno de administración de la página web “auralizARTE”. Para ello se parte de los requisitos planteados por el cliente: el departamento de física y acústica de la Universidad Pública de Navarra.

La memoria se divide en seis capítulos:

- Capítulo 1: “ANÁLISIS DE REQUISITOS”
- Capítulo 2: “TECNOLOGÍAS A UTILIZAR”
- Capítulo 3: “CONCEPTUALIZACIÓN”
- Capítulo 4: “DISEÑO DE LA APLICACIÓN”
- Capítulo 5: “IMPLEMENTACIÓN”
- Capítulo 6: “CONCLUSIONES Y FUTURAS LÍNEAS DE TRABAJO”

Introducción

El proyecto “auralizARTE” parte de la idea de acercar al público en general, y en especial a los músicos e ingenieros acústicos, la técnica de la “auralización”. Dicha técnica permite generar archivos sonoros partiendo de datos numéricos extraídos del análisis de un auditorio para así realizar escuchas “virtuales” del mismo.

De esta manera el usuario puede experimentar cómo suena un auditorio, compararlos entre sí y adquirir conocimientos sobre esta técnica.

El proyecto está dividido en cuatro fases, de las cuales la tercera está dedicada a la creación de la página web.

Capítulo 1: “ANÁLISIS DE REQUISITOS”

Análisis de los requisitos de la web

Para que la web pueda cumplir con el objetivo de llegar al usuario, es necesario que en concepto sea sencilla de utilizar y clara en contenidos. Además debe cumplir con los estándares webs, usabilidad y legibilidad.

Se establece que la web debe contener tres apartados diferenciados:

- “auralizARTE”: Repositorio de auditorios.
- “educARTE”: Documentación divulgativa sobre la técnica de la auralización.
- “entrenARTE”: Sección comparativa de auditorios.

auralizARTE

Esta es la sección en la que los usuarios pueden visualizar el repositorio de auditorios. El usuario debe poder acceder de manera sencilla e intuitiva a la información de cada auditorio. En un primer momento se estableció un listado ordenado según países, provincias y ciudades. Esta idea permitiría que el usuario navegase a través del listado hasta el auditorio en cuestión.

Desde un punto de vista técnico, la ejecución es sencilla, pero no resulta muy amigable. La interacción con el usuario es monótona y poco atractiva.

Para mejorar la experiencia de usuario y además permitir que el visitante interactúe más tiempo con el portal se planteó el uso de un mapa (proporcionado por Google Maps). Los auditorios se presentan a modo de “chincheta” y el usuario puede ver tanto la localización geográfica, como utilizar un filtro para segmentar los auditorios según sus características acústicas.

Para esta solución, desde el panel de administración se debe poder realizar las siguientes operaciones:

- Añadir, borrar, editar auditorios y sus características acústicas, fotos, descripciones, auralizaciones, etc.
- Añadir la localización concreta del auditorio (latitud y longitud).
- Activar o desactivar la visualización del auditorio en el mapa.
- Editar la API key que permite el uso de la API de Google Maps.

educARTE

El objetivo de este apartado es dar a conocer la técnica de la auralización. Se estableció un sistema mediante el cual el administrador podía añadir artículos clasificados por categorías. Finalmente se decidió que el usuario debía acceder a una información más breve y concisa ya que el contenido no variaría demasiado en el tiempo.

La solución final consiste en un archivo con documentación fija, en dos idiomas y segmentada en varios apartados. Mediante el uso de un menú lateral, el usuario es capaz de saltar de un apartado a otro.

Desde el panel de administración se debe poder realizar las siguientes operaciones:

- Modificar el contenido del archivo de documentación.
- Poder modificarlo en varios idiomas.

entranARTE

Esta sección permite al usuario seleccionar dos auditorios de la base de datos y comparar sus características, tanto acústicas como geométricas. Como idea original se propuso que el usuario pudiese comparar las auralizaciones de cada auditorio, pero para ampliar su funcionalidad se permitió acceder a más contenido de la base de datos.

Este proceso se hace de manera transparente al administrador. No hace falta modificar nada desde el panel de administración ya que es el propio usuario el que realiza la llamada a la base de datos de manera individual dependiendo de su elección.

Ideas rechazadas

Visita “virtual” a través del auditorio para la sección auralizARTE

El primer concepto que se propuso para la web fue elaborar un portal mediante el cual el usuario fuese capaz de realizar una visita “virtual” por los diferentes auditorios de la base de datos. Podemos tomar como referencia los siguientes ejemplos:

- <http://www.auditorionacional.mcu.es/virtual/visita>
- <http://www.auditorio.com.mx/recorrido/auditorio.html>

Para realizar el estudio acústico de un auditorio, previamente hay que modelarlo en 3D. La idea se basa en utilizar el archivo del modelo generado para cargarlo en la web y que el usuario realice una “visita” utilizando los cursores para moverse por la escena. Una vez situado en ciertas zonas, podría realizar la escucha de la auralización concreta en ese punto del auditorio.

Para realizarlo es necesario utilizar tecnologías 3D compatibles con navegadores modernos. Un ejemplo de ello es WebGL (Diego Cantor, 2012) combinado con librerías Javascript como Three.js (Dirksen, 2013) o O3D (Paul, 2009). Otra posibilidad es utilizar la tecnología shockwave (Meyers, 2002) o flash de Adobe, generando previamente el archivo a reproducir en el navegador.

Ventajas

- Experiencia de usuario elevada.
- Interactividad con el medio.
- Intuitivo y didáctico.

Inconvenientes

- WebGL compatible únicamente con navegadores modernos.
- Necesario preprocesar el archivo shockwave o flash.
- La tecnología flash está en desuso.
- Necesario un modelo 3D con calidad suficiente para que la experiencia de usuario sea agradable.
- No está garantizado que el usuario pueda moverse con libertad sobre el modelo 3D utilizado en el análisis (modelo 3D de bajo detalle).
- Tamaño de la web y tiempos de carga elevados.
- Costes y recursos elevados, lo cual limita el tamaño de la base de datos.

Conclusión

El principal problema de la idea es la cantidad de recursos y coste que genera el conseguir una experiencia de usuario adecuada para cada auditorio. Esto implica que la base de datos no sea muy extensa como para satisfacer el objetivo principal del proyecto, dar a conocer el máximo número de auditorios.

Un problema secundario es la tecnología. Habría que modificar las librerías Javascript disponibles (puesto que no hay una específica que cumpla esta función)

para que el usuario interactúe con el medio, de modo que pueda “escuchar” las auralizaciones. En definitiva, esta aproximación no encaja tanto en un portal web como en un videojuego interactivo online (para los cuales existen librerías específicas más adecuadas).

Aplicación de convolución para la sección educARTE

La sección educARTE contiene documentación divulgativa sobre la técnica de la auralización. Se propuso una aplicación en la que el usuario pudiese cargar un archivo de audio alojado en su dispositivo, seleccionar una determinada respuesta al impulso y obtener el resultado audible de la convolución de ambos archivos.

Al realizar una búsqueda sobre técnicas de convolución de audio online, concluimos que solo existe una librería capaz de realizar esta operación de manera eficaz llamada Web Audio API (Smus, 2013).

Web Audio API está basada en Javascript, y utiliza el potencial del navegador del usuario para realizar operaciones de síntesis y procesado de audio. Es una API muy moderna y por lo tanto no es compatible con todos los navegadores.

Se realizaron diferentes pruebas piloto con la librería. El archivo que el usuario carga desde su dispositivo debe tener las mismas características (bitrate por ejemplo) que los archivos de audio de las respuestas al impulso almacenadas en el servidor, lo cual no garantiza que el sistema funcione siempre.

Ventajas

- Aplicación educativa para demostrar un caso práctico de auralización.
- Permite entender mejor al usuario la operación de convolución.
- La operación se realiza en el lado del cliente y no del servidor.
- Utiliza todo el potencial del dispositivo y del navegador del cliente.
- API muy versátil y potente.

Inconvenientes

- Web Audio API solo compatible con navegadores modernos.
- Las características de los archivos de audio utilizados para la convolución deben ser iguales.

Conclusión

A pesar de que la idea es buena y la API muy potente y versátil, es conveniente esperar a que sea compatible con la mayoría de navegadores. De esta manera todos los usuarios podrán utilizar la aplicación. Es una idea a implementar a futuro que depende del desarrollo de la API.

Capítulo 2: “TECNOLOGÍAS A UTILIZAR”

Tecnologías disponibles: CMS o diseño a medida

Prácticamente todas las páginas web están basadas en lo que se llaman gestores de contenidos o CMS. Un CMS es un programa que permite crear una estructura de soporte (también llamado framework (Apple, 2013)) para la creación y administración de contenidos por parte de los administradores, editores y demás usuarios.

Los CMS controlan una o varias bases de datos donde se aloja el contenido de la página web. El administrador o editor puede controlar el contenido del portal independientemente del diseño.

Actualmente existen infinidad de sistemas CMS desarrollados con software libre. Los más destacados son Wordpress, Joomla y Drupal.

Las principales ventajas de utilizar un gestor de contenidos son:

- Fácil instalación, gestión y personalización.
- Menor coste y trabajo de programación.
- Módulos, plugins y extensiones.
- Posicionamiento SEO.

Por el contrario, los CMS están diseñados para ajustarse a las necesidades típicas de una web. Principalmente la publicación de artículos tipo blog (Rowse, 2005), wikis (Nations, 2013), foros o comercio electrónico.

Las principales desventajas de utilizar un CMS son:

- Menor seguridad, puesto que la mayoría de CMS son de código abierto, existen comunidades que se dedican a encontrar vulnerabilidades del mismo.
- Menor velocidad de carga, al ser plataformas genéricas las consultas a base de datos son altas.

Si los requisitos de una web son diferentes a la mayoría, se puede optar por dos opciones:

- Añadir funcionalidad a los CMS mediante la creación de módulos o plugins.
- Crear un CMS a medida.

Para el caso particular de auralizARTE, las necesidades del portal son muy específicas. Se estudió crear una serie de módulos para mejorar la funcionalidad de un CMS (en concreto Drupal), pero la mayoría de las características del CMS se desaprovecharían.

La opción más adecuada fue crear un CMS a medida.

Las principales ventajas de esta solución son:

- Permite crear un panel de administración propio que cumpla únicamente con las necesidades del cliente.
- Mejora la usabilidad del panel de administración.
- Código propio y por lo tanto más seguro al no ser público.
- Mejor velocidad de carga que un CMS público (menor número de consultas a base de datos).

Los inconvenientes son:

- Menor flexibilidad y escalabilidad, hay que programar siempre las nuevas funcionalidades.
- Más caro ya que es más trabajo para el programador.

Lenguajes de programación web

Actualmente existen multitud de lenguajes de programación que permiten la creación de webs dinámicas (Web And Macros, 2007).

Lenguajes que interpreta el navegador:

- HTML
- CSS
- JavaScript

Lenguajes que interpreta el servidor:

- PHP
- ASP
- Java
- Python
- Ruby

Los lenguajes de programación del lado del navegador se utilizan para diseñar la interfaz de usuario, lo que el usuario visualiza cuando entra a una página web. Por el contrario los lenguajes del lado del servidor se encargan de realizar todas las operaciones necesarias cuando el usuario interactúa con la web (cálculos, llamadas a la base de datos, registros, mostrar la interfaz, etc.).

Para crear una página web dinámica es necesario utilizar una combinación de estos lenguajes.

En auralizARTE se han utilizado los siguientes lenguajes de programación:

Navegador:

- HTML5
- CSS3
- JavaScript

Servidor:

- PHP

HTML5 es la quinta revisión del lenguaje estándar para web. HTML5 establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web actuales, como son las etiquetas <footer>, <nav> o <article>.

El uso de HTML5 permite una transición a lo que se llama “web semántica”, lo cual mejora el posicionamiento en buscadores.

CSS3 (hojas de estilo) es el complemento de HTML5 que permite dar aspecto y formato a una web. En su tercera versión añade nuevas funciones capaces de crear animaciones, transiciones y efectos de texto, sustituyendo por completo el uso de la tecnología flash.

JavaScript es un lenguaje de programación orientado a objetos. Implementado como parte del navegador permite por ejemplo mejoras en la interfaz de usuario. Con este lenguaje podemos acceder a la estructura DOM de la web y modificar su contenido en tiempo real.

PHP es un lenguaje de código abierto del lado del servidor. Su curva de aprendizaje es muy corta y permite a los programadores crear aplicaciones complejas. Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado (escrito en PHP) que generará el contenido de manera dinámica. Además PHP permite la conexión con servidores de bases de datos como MySQL o SQLite.

Frameworks utilizados

Un framework es una estructura de software compuesta por librerías personalizables e intercambiables para el desarrollo de una aplicación. Gracias a un framework se acelera el proceso de desarrollo, se reutiliza código y mantiene una estructura de programación coherente.

Una librería es un conjunto de funciones y subprogramas disponibles que ayudan al programador a no reescribir código desde cero.

En el caso de auralizARTE se han utilizado varios frameworks del lado del cliente, tanto para el Frontend como para el Backend:

- Bootstrap 2.3.2
- jQuery
- jQuery-UI

Estas herramientas ayudan a la creación de la interfaz de usuario y la interacción del usuario con la web.

Bootstrap es un framework diseñado por la empresa Twitter y contiene varias librerías para el desarrollo de interfaces de usuario. Es totalmente personalizable y permite al programador añadir un sistema de cuadrícula ajustable a todo tipo de tamaños de pantalla. Además incorpora los elementos básicos de una web, (formularios, botones, pop ups, mensajes de alerta, etc.) sin necesidad de diseñarlos.

jQuery es un framework de código abierto basada en JavaScript. Permite al programador simplificar la manera de interactuar con los elementos del DOM e interactuar con la web. Añade efectos, eventos y permite desarrollar animaciones. Además es capaz de modificar las hojas de estilo CSS y añadir interacción con la técnica AJAX (Holdener, 2008).

jQuery-UI es una ampliación de jQuery que añade un conjunto de plugins, widgets y efectos visuales destinados exclusivamente a la interacción del usuario con la interfaz. Es un complemento perfecto para Bootstrap ya que añade mayor funcionalidad al framework, como por ejemplo un selector de calendario, selector de rangos (slider), elementos arrastrables, etc.

Capítulo 3: “CONCEPTUALIZACIÓN”

Para realizar un buen desarrollo web, primero hay que conceptualizar todos los elementos que intervienen en el funcionamiento de una web dinámica; base de datos, servidor, Backend y Frontend. Tras la conceptualización, se puede comenzar con el diseño tanto de funcionamiento como estético.

A lo largo de este capítulo se analizan las necesidades de la web para diseñar la base de datos, la estructura de archivos que controlan la web desde el servidor y el diseño estético del panel de administración del lado del cliente.

Base de datos relacional

Una base de datos relacional permite establecer interconexiones (relaciones) entre los datos que están guardados en tablas. A través de estas conexiones se pueden relacionar datos de diferentes tablas. Es el tipo de base de datos más común en una web dinámica.

Necesidades de la web

Para diseñar la base de datos de auralizARTE primero se realizó un listado de todos los datos que se necesitan guardar en base de datos. En concreto:

Auditorios:

- Nombre del auditorio
- País
- Provincia
- Ciudad
- Nombre de la fotografía
- Latitud
- Longitud
- Descripción
- Activado en mapa

Parámetros del auditorio:

- Capacidad del auditorio
- Volumen del auditorio
- TRmid

Datos de la gráfica TR30 del auditorio:

- Valor en 125Hz
- Valor en 250Hz
- Valor en 500Hz
- Valor en 1kHz
- Valor en 2kHz
- Valor en 4kHz
- Descripción de la gráfica

Archivos de audio de los auditorios:

- Nombre del archivo
- Descripción del archivo

Usuarios de la web

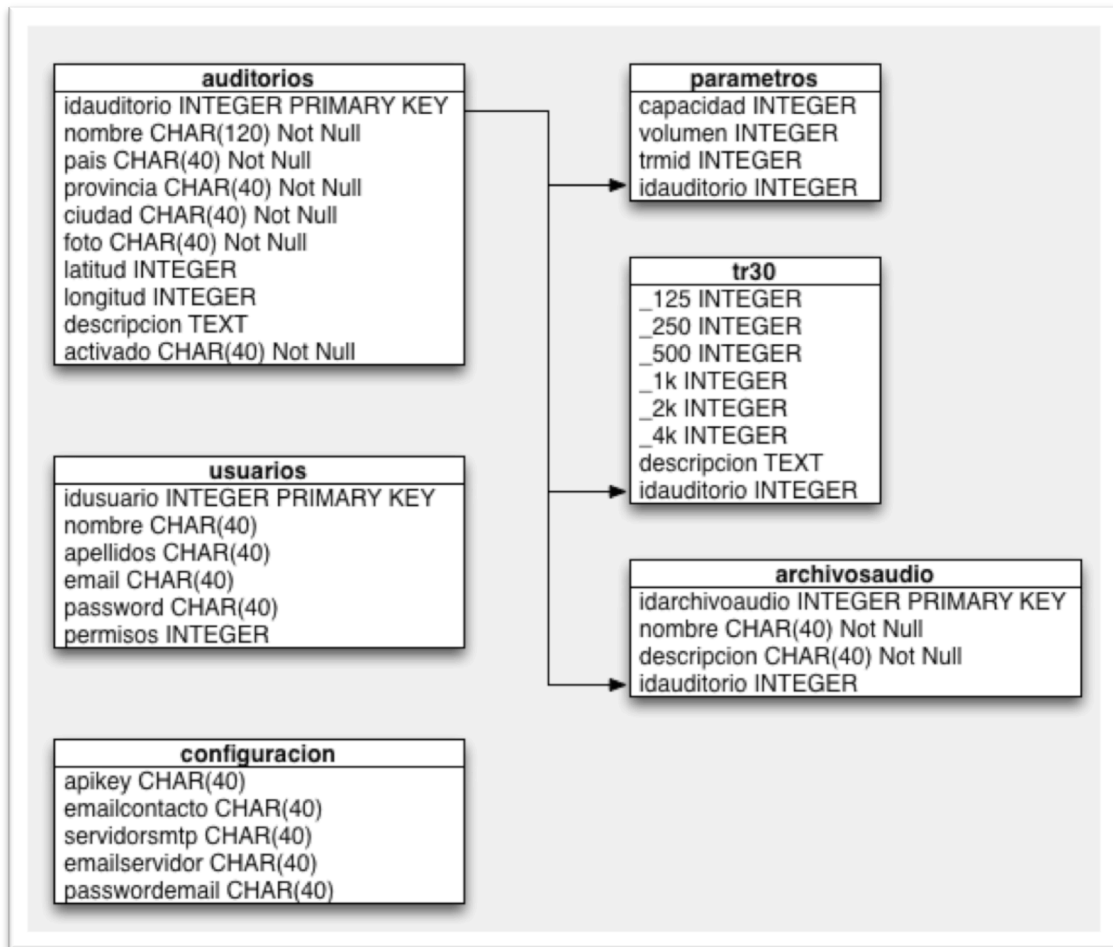
- Nombre
- Apellidos
- Email
- Contraseña
- Permisos de usuario

Configuración de la web

- Apikey de Google Maps
- Email de contacto
- Servidor smtp
- Email servidor
- Contraseña del correo

Diseño de la base de datos

Una vez realizado el listado, se pueden crear las tablas y relacionarlas entre sí con identificadores para generar un mapa de la base de datos. El único identificador que hay que relacionar es el del auditorio (idauditorio). De esta forma las tablas Parámetros, Gráfica TR30 y Archivos de audio quedan relacionadas con la tabla Auditorios.



Cabe destacar el tipo de dato utilizado en los identificadores (INTEGER), con el índice PRIMARY KEY. Este índice permite que en cada entrada de la tabla el identificador se autoincrementa en un valor automáticamente. Con esto conseguimos identificar los datos (por ejemplo un auditorio) de forma única.

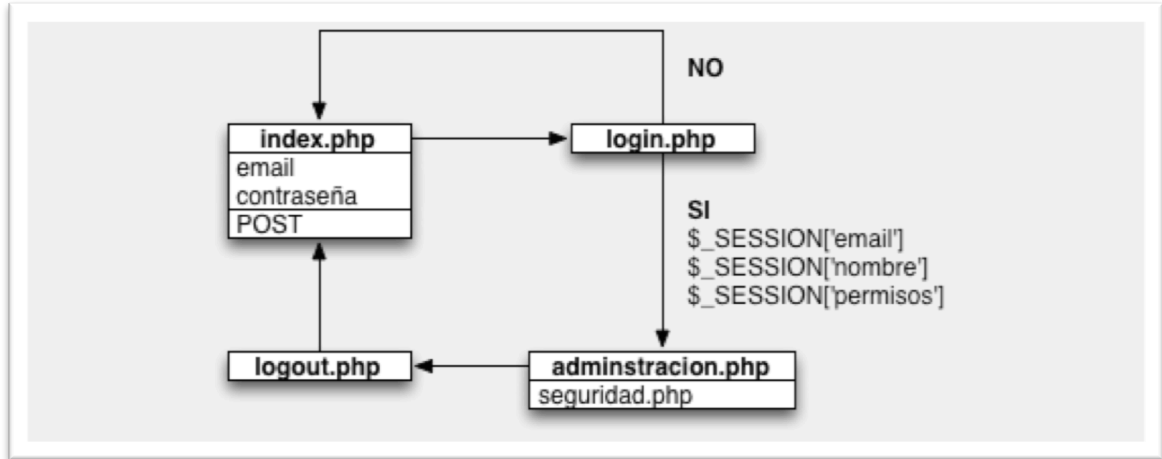
Lado del servidor

Para poder realizar una web dinámica capaz de realizar consultas a una base de datos y que el usuario pueda interactuar con ella a través de su navegador, es necesario realizar una estructura interna de funcionamiento. Esta estructura permite visualizar cómo trabajará internamente la web y programar posteriormente los scripts PHP necesarios para realizar las operaciones.

En el caso del panel de administración (Backend), se puede definir previamente un mapa con todos los archivos necesarios que contendrán los scripts PHP y sus acciones.

Acceso a la administración:

Para poder acceder al panel de administración se requerirá un “login”. De esta forma se evita el acceso de usuarios no identificados.



El archivo **index.php** es el script que pide el nombre de usuario y contraseña. Tras enviar por el método POST (W3Schools, 2013) los datos al script **login.php**, este evaluará si el usuario es correcto o no. Si es correcto creará las variables de sesión “email”, “nombre” y “permisos” y accederá correctamente al Backend (**administracion.php**). Si el usuario y contraseña no existe en la base de datos, **login.php** llamará a **index.php** para que el usuario vuelva a introducir un usuario y contraseña nuevo.

Además una vez que el usuario haya accedido correctamente a **administracion.php**, podrá salir del panel gracias al script **logout.php**, que eliminará las variables de sesión.

El archivo **seguridad.php** se encarga de comprobar que efectivamente existen las variables de sesión y que estas están activas. Si el usuario no existe, no conseguirá acceder manualmente a ningún archivo en el que se incluya este script de seguridad.

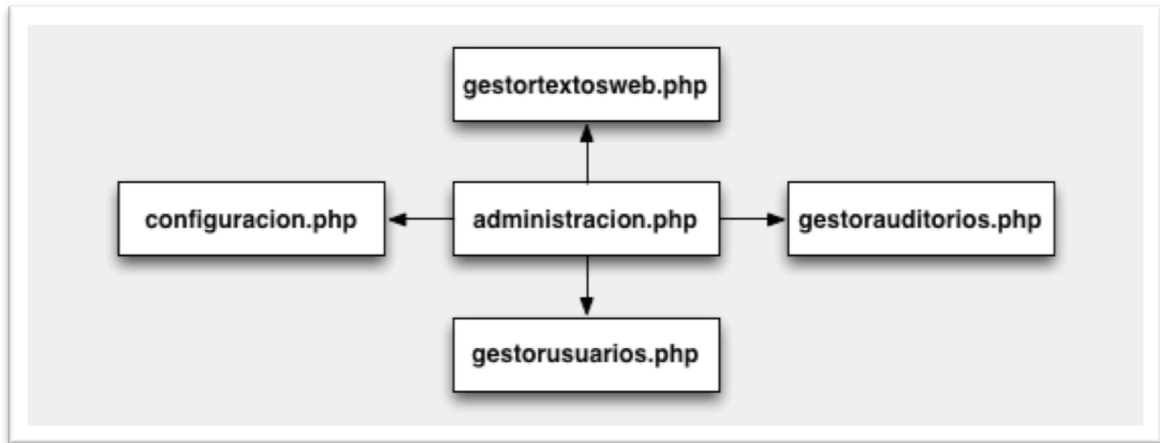
Administración

Una vez se acceda al panel de administración, se podrá navegar a través de él para realizar las diferentes operaciones. En concreto:

- Gestionar los auditorios.
- Gestionar los textos de la web.
- Gestionar los usuarios.

- Modificar la configuración.

Por tanto cada acción puede realizarse con scripts separados. El mapa sería el siguiente:



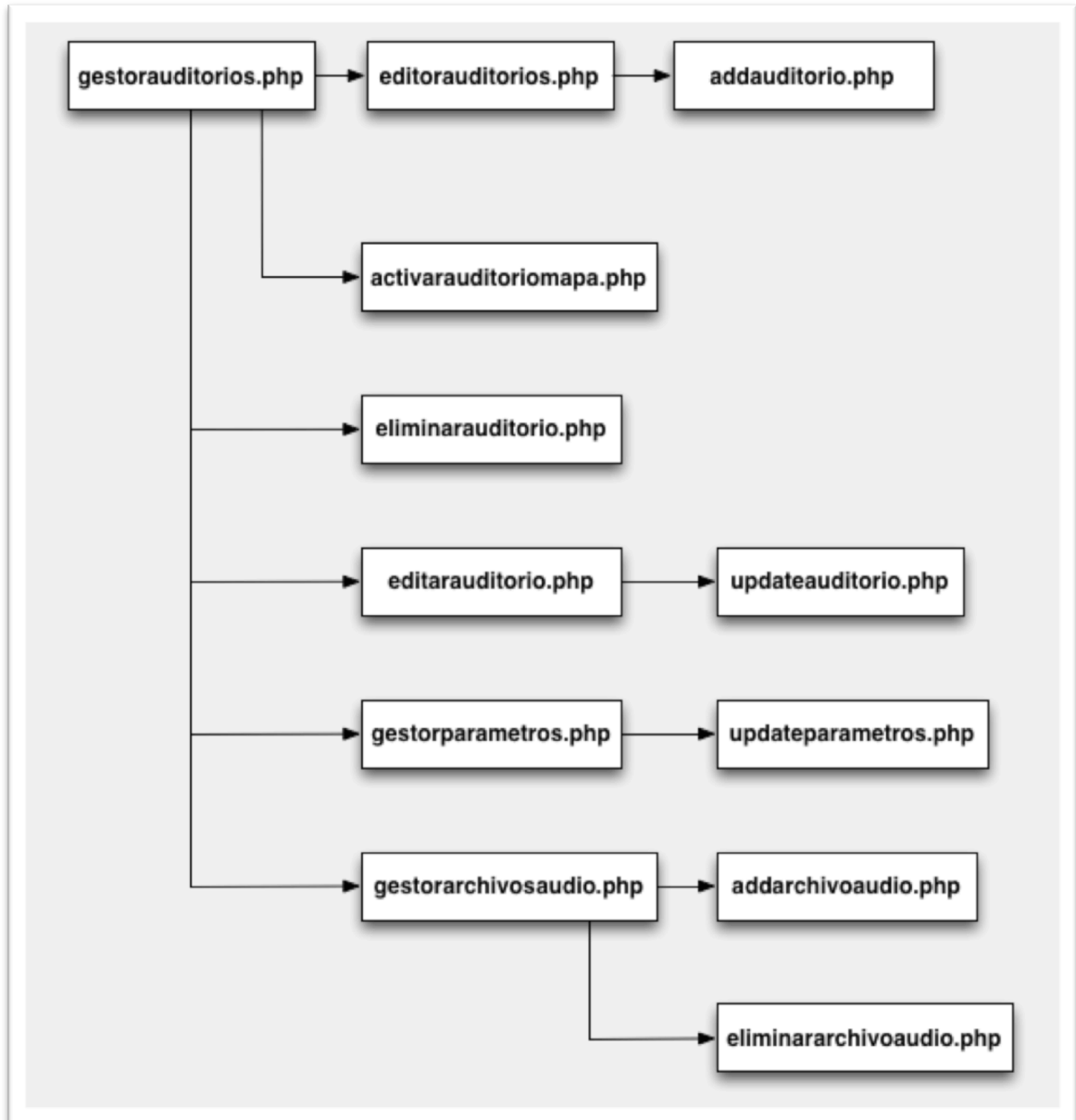
Una vez el usuario accede a **administracion.php**, este controlará cada acción por separado. Cada script está protegido con el script **seguridad.php**.

Gestor de auditorios

El gestor de auditorios tiene varias funciones:

- Añadir auditorios.
- Listar la base de datos de auditorios almacenados en base de datos.
- Activar o desactivar el auditorio en el mapa.
- Editar la información general de cada auditorio.
- Editar los parámetros geométricos y acústicos.
- Añadir o borrar archivos de audio.
- Borrar auditorios.

Cada función puede realizarse con scripts separados. El mapa sería el siguiente:



gestoauditorios.php: Es el script principal encargado de mostrar el listado de auditorios almacenados en base de datos y llamar a los demás scripts.

editorauditorios.php: Este script muestra el formulario con los datos iniciales necesarios para añadir un auditorio a la base de datos.

addauditorio.php: Toma los datos introducidos en el formulario de editorauditorios.php y los almacena en la base de datos. Además inicializa la descripción del auditorio, los parámetros geométricos del auditorio, los datos de la gráfica tr30 y crea la carpeta con el número de identificador único del auditorio en el servidor.

activarauditoriomapa.php: Desde el gestor de auditorios se puede modificar el estado del auditorio en el mapa (activado/desactivado). Este es el script encargado de modificar el estado y guardarlo en la base de datos.

eliminarauditorio.php: Cuando se pulse el botón borrar auditorio, se llamará a este script que eliminará por completo el auditorio de la base de datos. Además llama a la función **eliminar_directorio_auditorio** para eliminar la carpeta asociada al auditorio del servidor y todo su contenido.

editarauditorio.php: Este script permite modificar los datos generales del auditorio almacenados en base de datos. Una vez se modifican los datos, al guardar los cambios llamará al script `updateauditorio.php` para que guarde los cambios en la base de datos.

updateauditorio.php: Script que actualiza la base de datos con las modificaciones de los datos generales del auditorio.

gestorparametros.php: Al igual que `editarauditorio.php`, este archivo muestra los parámetros geométricos y la gráfica T30 para poder modificar los datos. Una vez se modifican los parámetros se llama al script `updateparametros.php` para almacenar los cambios en la base de datos.

updateparametros.php: Script que actualiza en la base de datos los parámetros geométricos y la gráfica T30.

gestorarchivosaudio.php: Este script muestra el listado de los archivos de audio asociados al auditorio. También permite escucharlos gracias al reproductor de audio de HTML5. El administrador o editor puede añadir una descripción al archivo para identificarlo correctamente. El script llama al script `addarchivoaudio.php` para crear la entrada de audio en la base de datos y subir el archivo al servidor. Cuando se elimina un archivo, el script llama a `eliminararchivoaudio.php` para borrar la entrada de la base de datos y el archivo físico de la carpeta del servidor.

addarchivoaudio.php: Este script analiza el archivo de audio que se quiere subir al servidor y, si cumple ciertas características, lo guarda en la carpeta del auditorio correspondiente. Además crea una entrada en la base de datos con el identificador único, el nombre del archivo, la descripción y el identificador del auditorio asociado.

eliminararchivoaudio.php: Este script se ejecuta al pulsar el botón borrar en el gestor de archivos de audio. Sencillamente borra la entrada de la base de datos del archivo de audio correspondiente y borra el archivo de la carpeta del servidor.

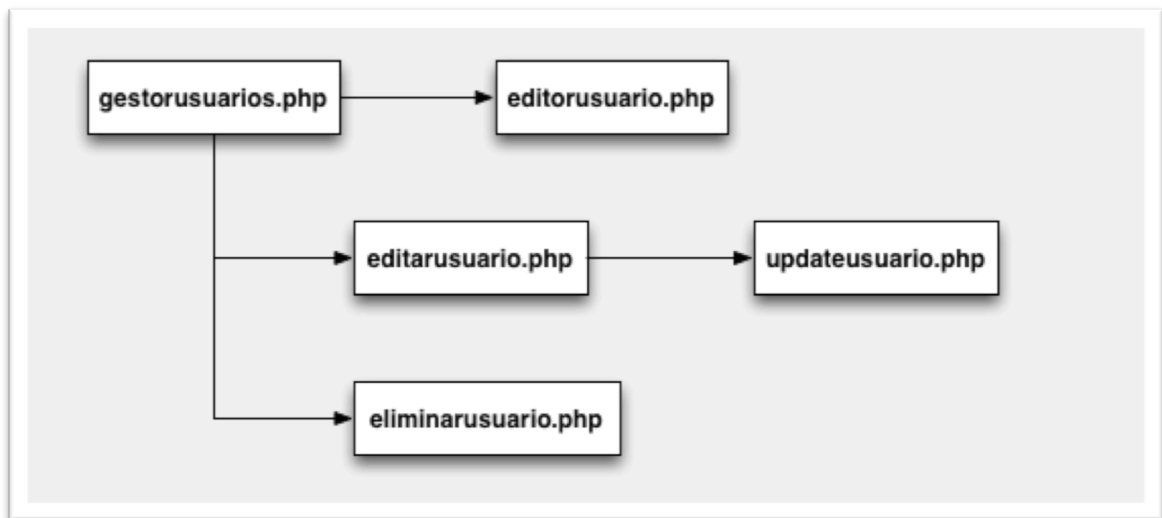
Gestor de usuarios

AuralizARTE no requiere que usuarios externos se registren en la web, pero sí que existan usuarios administradores y editores para controlar el panel de administración.

Las funciones principales para la gestión de usuarios es la siguiente:

- Añadir usuarios.
- Editar usuarios existentes.
- Eliminar usuarios.

El mapa de scripts es el siguiente:



gestorusuarios.php: Es el script capaz de mostrar todo el listado de usuarios creados en la base de datos. Indica además el tipo de permiso que tiene el usuario. A este script solo puede acceder un usuario con permisos de administrador y no de editor.

editorusuario.php: Este script se ejecuta al pulsar el botón de “Añadir usuario”. Permite añadir los campos necesarios para crear un usuario en la base de datos. Además contiene un validador de contraseñas.

editarusuario.php: Cuando el administrador quiere editar un usuario ya creado, este script muestra los datos del usuario almacenados en la base de datos para su modificación. Una vez modifica el usuario y guarda los cambios, **editarusuario.php** llama al script **updateusuario.php**.

updateusuario.php: Este script accede a la base de datos de la web y cambia los cambios que el administrador haya realizado sobre un usuario.

eliminarusuario.php: Al pulsar el botón eliminar usuario, este script elimina de la base de datos todos los campos relacionados con este usuario.

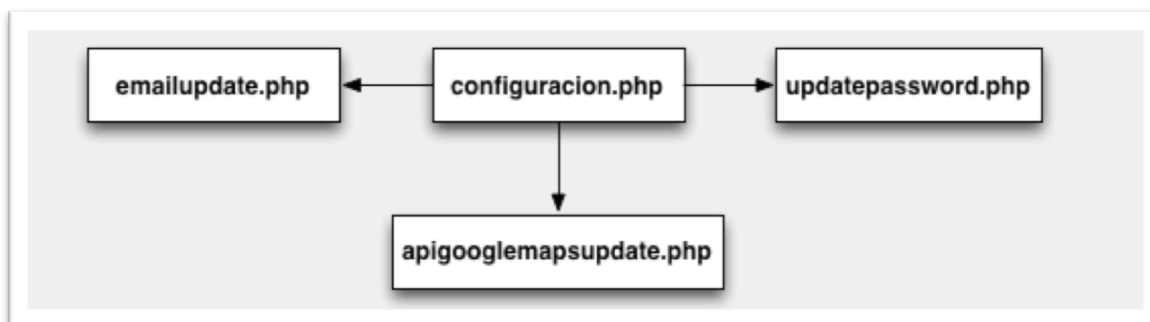
Configuración

Al gestor de configuración solo puede acceder por completo un usuario administrador, pero un usuario editor tiene un acceso parcial. El usuario editor tan solo tiene acceso al script `updatepassword.php` para modificar su contraseña de acceso, pero no al resto de opciones de configuración.

Las funciones que cumple la configuración son:

- Modificar la contraseña de acceso del usuario.
- Actualizar la configuración del formulario de contacto de la web.
- Actualizar la configuración de la clave de Google Maps.

El mapa de scripts es el siguiente:



configuracion.php: Es el script principal que muestra las diferentes configuraciones: modificar la contraseña de usuario, modificar la clave de Google Maps y modificar la configuración de correo para el formulario de contacto. Además filtra el contenido según los permisos de usuario.

updatepassword.php: Cuando el usuario modifica su contraseña y guarda los cambios, este script realiza la consulta en la base de datos para guardar la contraseña al usuario.

emailupdate.php: Este script guarda en la base de datos la configuración de correo electrónico que se utiliza en el formulario de contacto.

apigooglemapsupdate.php: Este script guarda la clave necesaria para Google Maps en la base de datos.

Textos de la web

El gestor de textos de la web permite modificar los textos estáticos de la web. El administrador puede cambiar los textos en castellano o en inglés. El usuario editor no tiene permisos para modificar este contenido. Los textos no están almacenados en base de datos, sino en archivos PHP. El administrador puede modificar estos archivos a través del panel de administración.

Las funciones básicas son:

- Modificar el contenido estático de la web en castellano e inglés.
- Modificar el contenido del apartado educARTE, en castellano e inglés.

El mapa de scripts es el siguiente:



gestortextosweb.php: Este script accede a los archivos `textosweb_es.php`, `textosweb_en.php`, `educarte_es.php` y `educarte_en.php` que contienen los textos estáticos de la web. Cuando se modifica un archivo, al pulsar el botón guardar cambios, `updatediccionario.php` o `updateeducarte.php` reescriben estos ficheros.

updatediccionario.php: Se encarga de guardar las modificaciones de los textos estáticos de la web (botones, títulos, menú, etc.) en los archivos **`textosweb_es.php`** y **`textosweb_en.php`**.

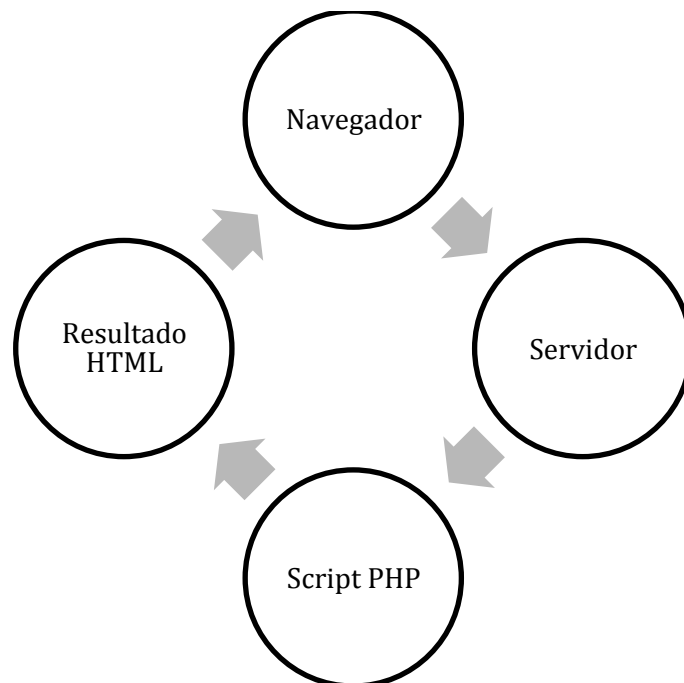
updateeducarte.php: Se encarga de guardar las modificaciones de los textos del apartado educarte de la web en los archivos **`educarte_es.php`** y **`educarte_en.php`**.

Capítulo 4: “DISEÑO DE LA APLICACIÓN”

Lado del cliente (panel de administración)

Introducción

El lado del cliente representa lo que el usuario ve cuando accede con su navegador al panel de administración. Es el interfaz que se genera dinámicamente tras una petición al servidor:



El proceso es el siguiente:

- El usuario, al introducir una dirección en el navegador, está conectando con el servidor de la web y realizando una petición, en este caso, mostrar la interfaz del panel de administración.
- El servidor recibe la petición, y procesa el script PHP (invisible e inaccesible al usuario) que, entre otros procesos, genera un resultado en HTML.
- Este resultado en HTML se envía al navegador, capaz de interpretarlo, y lo muestra al usuario en pantalla.

Este proceso se realiza cada vez que el usuario interactúa con la web y realiza una petición al servidor.

Para diseñar la interfaz del panel de administración se comenzó estudiando los requisitos y operaciones (conceptualización) que debía poder realizar el administrador en cada apartado de la web (gestión de auditorios, borrar, editar, etc.). A continuación se pudo definir un wireframe (diseño en dibujo de la interfaz) basado en las normas básicas de usabilidad y experiencia de usuario.

Usabilidad y experiencia de usuario: Fundamentos , wireframe y diseño final

¿Qué significa usabilidad? ¿Qué hace a un sitio “usable”? La usabilidad sólo significa el asegurarse que algo funciona bien. Es una de las características más importantes que hay que tener en cuenta a la hora de diseñar una web (Krug, 2006). Cualquier persona con capacidad normal debe poder utilizar el portal sin sentirse frustrado. Si se consigue este objetivo, podemos considerar que la usabilidad del sitio es adecuada.

Para cualquier persona si algo es complicado de utilizar, simplemente no la usará demasiado.

La manera en la que las personas leen una página en internet, difiere drásticamente de la manera en que leen otro tipo de publicaciones, como pueden ser una revista o un periódico.

Fundamentalmente:

- Los usuarios en internet no leen las páginas web, sino que las hojean.
- Si algo es complicado de utilizar, básicamente se deja de usar.
- El usuario considera las opciones de la web y elige la mejor, o generalmente elige la más razonable.
- El usuario no averigua el funcionamiento de las cosas, se las arregla para utilizarlas.

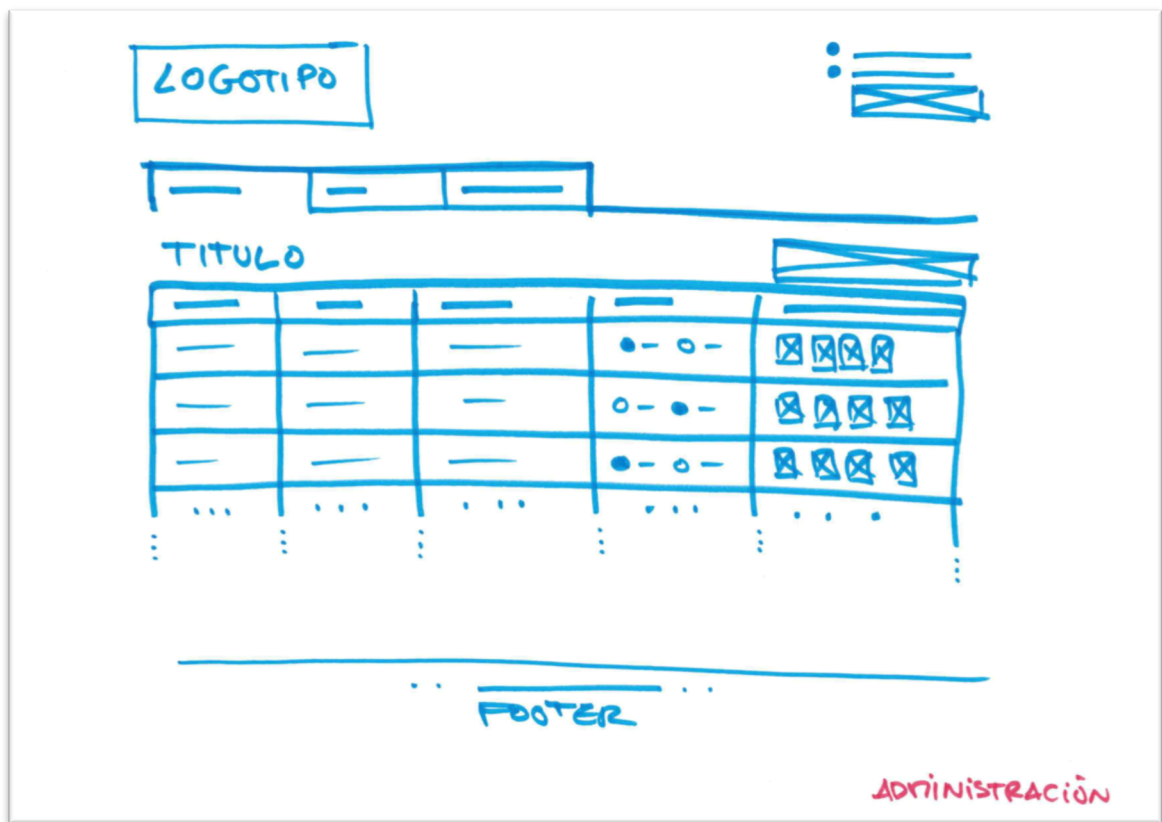
Por lo tanto, ¿qué es lo que hay que tener en cuenta para que un sitio web sea fácil de usar? La respuesta no se basa en que “nada importante esté a más de dos clics de distancia”, de hablar en el lenguaje del usuario o incluso de ser coherente. Se trata de **no hacer pensar al usuario**.

Para el caso del panel de administración de auralizarte, había que ser especialmente cuidadoso con la usabilidad, ya que cualquier persona debe poder utilizarlo sin necesidad de instrucciones. La función del panel no es la misma que el frontend, y por lo tanto debía estar enfocado totalmente a la funcionalidad.

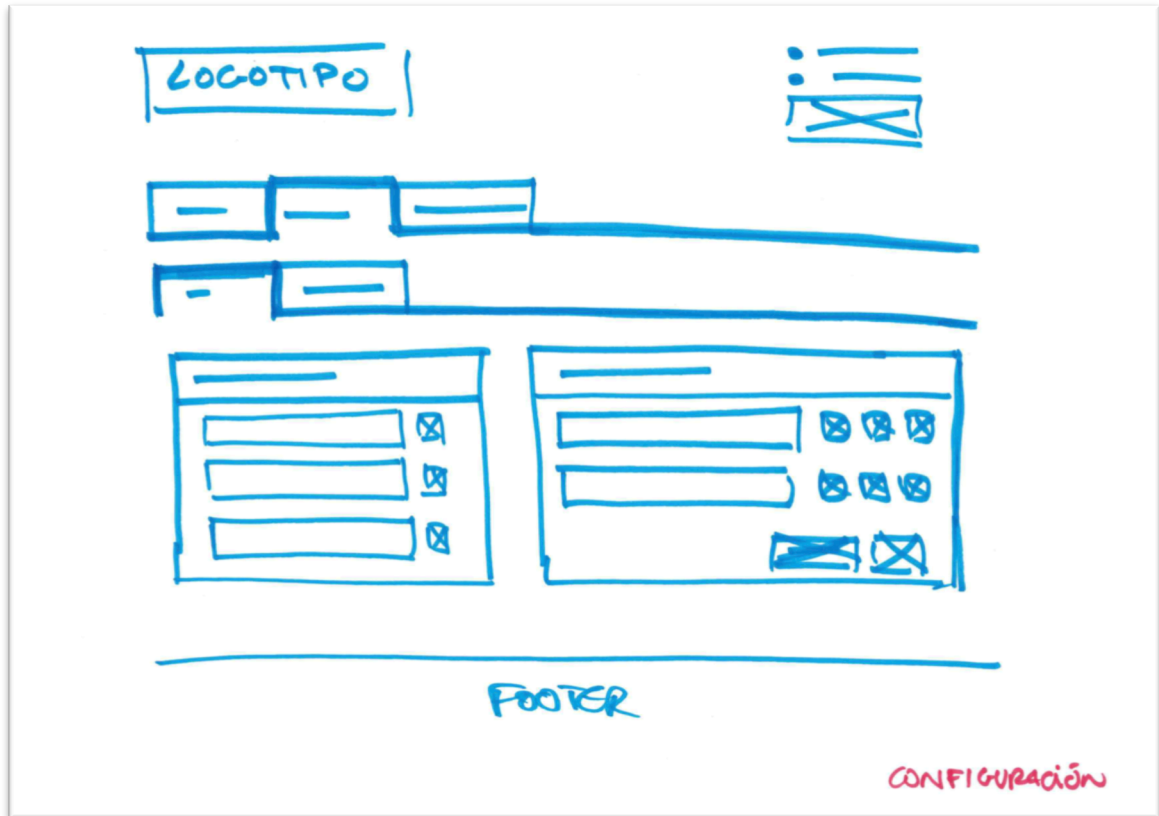
Antes de realizar un diseño final se realizó lo que se llama “Mockup” o wireframe. Básicamente es un boceto (digital o en papel) que sitúa la posición de los elementos de la interfaz antes de pasar a la programación.

Los bocetos se realizaron basándose directamente en los elementos de la librería de Bootstrap. De esta manera el proceso de diseño y programación se simplifican enormemente.

A continuación se muestran dos bocetos, uno del panel principal de administración de auditorios y otro de la configuración de la web.



Wireframe del panel de administración



Wireframe del panel de configuración con idea de pestañas anidadas

La estructura básica consiste en una cabecera con el título de la administración y una sección de saludo, permiso y cerrar sesión, el cuerpo de contenido y el pie de página.

El diseño final es el siguiente:

The screenshot displays the 'Administración AuralizARTE' web application. At the top, the title 'Administración AuralizARTE' is prominently displayed. To the right, the user is identified as 'Hola Eduardo Administrador' with a 'Cerrar sesión' button. Below the title, a navigation menu contains 'Auditorios', 'Textos de la web', 'Usuarios', and 'Configuración'. The main content area is titled 'Gestor de auditorios' and features a '+ Añadir auditorio' button. A table titled 'Listado de auditorios' contains two rows of data:

ID	Nombre	País	Provincia/Estado	Ciudad	Guardar estado	Editar
1	Auditorio Nacional de Música	España	Madrid	Madrid	<input checked="" type="checkbox"/> Activado <input type="checkbox"/> Desactivado	<input type="text"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
2	Baluarte	España	Navarra	Pamplona	<input checked="" type="checkbox"/> Activado <input type="checkbox"/> Desactivado	<input type="text"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

La utilización de Bootstrap como framework permite al programador no tener que diseñar todos los elementos de la web sino que, al poseer una librería básica de botones, listas o tablas, se diseña a la vez que se programa.

Cabecera

La cabecera o **header** es lo primero que un cliente visualiza cuando accede a una web. Por convención (heredada de las publicaciones impresas) la cabecera tiene que tener una jerarquía visual clara con respecto a las demás secciones. Esto permite al usuario situarse en el entorno. La cabecera permanece siempre visible sin importar en qué sección se encuentre el usuario.

Lo más importante ha de ser lo más prominente y por tanto resaltar el título “Administración AuralizARTE” hace que el usuario se sitúe rápidamente.

Además del título, se añade una sección en la parte derecha dentro de la cabecera, donde el usuario es identificado por la web y puede ver qué permisos tiene o la posibilidad de cerrar la sesión pulsando el botón de “cerrar sesión”.

Cuerpo de contenido

Para organizar el contenido por secciones se optó por utilizar pestañas. Esto tiene ciertas ventajas:

- Son claras y fáciles de entender.
- Son difíciles de perder de vista.
- Sugieren un espacio físico.

Puesto que la mayoría de contenidos se pueden apilar conforme se añaden a la base de datos, se planteó listar los elementos en una tabla, por ejemplo los auditorios, añadiendo una columna con botones que permitan modificar las características del elemento concreto.

La utilización de botones muy visibles es una característica que deja bien claro al usuario sobre dónde puede hacer clic. Además se complementan en su mayoría con iconos que evocan la función específica del botón (una papelera en un botón rojo por ejemplo). Esto permite omitir palabras innecesarias, como “borrar” o “añadir”, ahorrando espacio en la pantalla.

Pie de página

En el Frontend de una web, el pie de página muestra por convención los accesos rápidos a secciones de la web sin necesidad de que el usuario la conozca en su integridad. En el caso del panel de administración tiene una función muy simple que es mostrar la versión de desarrollo.

Capítulo 5: “IMPLEMENTACIÓN”

En este capítulo se presentan varias de las técnicas que se han aplicado en la implementación del panel de administración. Los ejemplos se apoyan con fragmentos de código fuente para una mejor comprensión.

PHP, HTML y la función ECHO

La implementación del portal Auralizarte se ha realizado mediante la combinación de varios lenguajes de programación web. Cada uno de estos lenguajes cumple con una función concreta, pero en su fundamento son PHP y HTML los que controlan el portal.

PHP y HTML interactúan mucho entre sí: PHP puede generar código HTML y HTML puede pasar información a PHP.

Hay dos maneras de mostrar código HTML en pantalla mediante el uso de PHP. Una es escribir el código HTML fuera de las etiquetas `<?php ... ?>`, por ejemplo:

```
<html>
  <?php
  //Código en PHP
  ?>
  <body>
    <b>Esto es un texto HTML en negrita</b>
  <?php
  //Más código en PHP
  ?>
</body>
</html>
```

Otra técnica es utilizar las funciones “**print**” o “**echo**” de PHP para “imprimir” en pantalla el código concreto. Esta técnica permite escribir todo el código en PHP.

```
<?php
echo “<html>”;
//Código PHP
print “<body><b>Esto es un texto HTML en negrita</b>”;
//Más código en PHP
echo “</body></html>”
```

?>

Cualquiera de estos dos métodos permite embeber código HTML en PHP.

Esto también se puede aplicar a otros lenguajes de programación como Javascript o jQuery. Al ser lenguajes de programación del lado del cliente, podemos mostrar código igual que como se hace con HTML.

Por ejemplo en el archivo **login.php**

```
//FUNCION ECHO QUE IMPRIME EN PANTALLA UN MENSAJE DE ERROR
echo '
<html>
  <head>
    <meta http-equiv="REFRESH" content="0;url=index.php">
  </head>
  <body>
    <script language="JavaScript">
      alert("Email o password incorrecto. Por favor vuelva a
intentarlo.");
    </script>
  </body>
</html>';
```

El código mostrado entre las etiquetas **<script>...</script>** indica código Javascript que cuando se ejecuta muestra una alerta en la pantalla del navegador. En este caso se puede observar que quien imprime en pantalla el código HTML junto con el código Javascript es la función “**echo**” de PHP.

En la implementación de Auralizarte se han utilizado las dos técnicas indistintamente.

Variables de formulario, POST y GET

Los formularios HTML son una utilidad perfecta para que un usuario pueda actualizar los datos de una base de datos. Gracias a la utilización de formularios HTML podemos enviar variables a PHP mediante dos métodos, el método **GET** y el método **POST**. Ambos métodos realizan la misma función pero de diferente forma.

El concepto de **GET** es obtener información del servidor. Este método obtiene datos que están alojados en un servidor, ya sea desde un archivo o desde una base de datos.

Las llamadas **GET** pueden ser indexadas o guardadas en la caché ya que los parámetros se transmiten mediante la URL y son visibles en el navegador. Una llamada **GET** se puede considerar como una solicitud de información que luego es procesada por el servidor. Este método se suelen utilizar en los hiperenlaces (links).

En cambio con el método **POST** el usuario puede enviar datos al servidor para que puedan ser procesados. Este método no envía datos a través de la URL, sino que los envía de manera “invisible” al usuario. Es especialmente útil para enviar datos sensibles al servidor tales como contraseñas o datos personales.

En el caso del panel de administración de Auralizarte, cuando se quiere editar la información completa de un auditorio por ejemplo, la consulta a la base de datos se realiza mediante el método **GET** en un link con una URL que identifica al auditorio concreto.

En el archivo **gestorauditorios.php** podemos encontrar el siguiente hiperenlace:

```
<a class='btn btn-mini' title='Editar auditorio'  
href='editarauditorio.php?idauditorio=".$fila['idauditorio']."&nombre=".$fila['nombre']."&pais=".$fila['pais']."&provincia=".$fila['provincia']."&ciudad=".$fila['ciudad']."&foto=".$fila['foto']."&latitud=".$fila['latitud']."&longitud=".$fila['longitud']."&descripcion=".$fila['descripcion']."'><i class='icon-pencil'></i></a>
```

La URL declarada en **href="..."** llama al archivo **editarauditorio.php** y pasa todos los parámetros característicos del auditorio, previa consulta en base de datos, para que puedan ser procesados en ese archivo. Esta dirección es visible al usuario cuando pulsa el botón de editar.

Si accedemos al archivo **editarauditorio.php** podemos encontrar lo siguiente:

```
//Variables GET de auditorio a editar  
$idauditorio = $_GET['idauditorio'];  
$nombre = $_GET['nombre'];  
$pais = $_GET['pais'];  
$provincia = $_GET['provincia'];  
$ciudad = $_GET['ciudad'];  
$latitud = $_GET['latitud'];  
$longitud = $_GET['longitud'];  
$foto = $_GET['foto'];  
$descripcion = $_GET['descripcion'];
```

En la cabecera del archivo se van declarando las variables que se van a utilizar en el script y mediante la función `$_GET` de PHP, almacenamos en ellas los datos recogidos en la URL.

En cambio para acceder al panel de administración se utiliza un formulario HTML que mediante el método **POST** envía a otro archivo el email y la contraseña de acceso. Estos datos por tanto no se almacenan ni se indexan en el navegador.

En el archivo **index.php** podemos encontrar el siguiente código HTML:

```
<form class="form-signin" action="login.php" method="post">
  <h2 class="form-signin-heading">Administración</h2>
  <input type="text" class="input-block-level" name="email"
placeholder="Email" required>
  <input type="password" class="input-block-level" name="password"
placeholder="Contraseña" required>
  <div align="center">
    <button class="btn btn-large btn-primary"
type="submit">Acceder</button>
  </div>
</form>
```

Como podemos observar se trata de un formulario que al pulsar el botón “Acceder” (`<button class="btn btn-large btn-primary" type="submit">Acceder</button>`) llama a al archivo `login.php` (`action="login.php"`) mediante el método POST (`method="post"`) y envía las variables introducidas en los inputs con nombre email (`name="email"`) y password (`name="password"`).

Si observamos el archivo **login.php** podemos apreciar el siguiente código:

```
//Variables POST
$email = $_POST['email'];
$password = $_POST['password'];
```

Se declaran las variables **\$email** y **\$password** para que puedan ser utilizadas en el archivo **login.php** y en ellas se almacenan los valores recogidos en el formulario mediante la función `$_POST` de PHP. Este proceso es invisible al usuario.

Peticiones a la base de datos SQLITE

SQLite es un sistema de gestión de base de datos que combina muchas de las funciones de MySQL junto con las ventajas y movilidad de poseer la base de datos en un archivo. Para utilizar SQLite es necesario que el servidor donde se aloje la web sea compatible con este sistema. Para comprobarlo podemos crear un archivo PHP con la función `phpinfo()`, que mostrará las características habilitadas en el servidor:

```
<?php  
phpinfo()  
?>
```

Si ejecutamos este archivo en el navegador, accederemos a toda la información del servidor. Para saber si SQLite está habilitado, hay que encontrar la siguiente tabla:

SQLite	
SQLite support	enabled
PECL Module version	2.0-dev \$Id: sqlite.c 306939 2011-01-01 02:19:59Z felipe \$
SQLite Library	2.8.17
SQLite Encoding	iso8859

En este caso deberá aparecer como “enabled”. Si no aparece la tabla anterior, es necesario instalar en el servidor el soporte para este sistema ya que sino el portal Auralizarte no funcionaría.

Para trabajar con SQLite hay que crear un archivo “.db” desde PHP en el cual se almacenarán los datos. Para el caso de Auralizarte se ha creado un proceso de instalación en el que se guía al usuario la primera vez que se ejecuta la web.

En el archivo `instalacion/bbddcreate.php` podemos observar lo siguiente:

```
//conexion  
$conexion = sqlite_open('../BBDD/auralizarte.db') or die ('No ha sido posible establecer la conexión, por favor vuelve a intentarlo.');
```

Al ejecutar el código anterior, SQLite, si no encuentra ningún problema, crea automáticamente un archivo en la carpeta BBDD llamado **auralizarte.db**. Este es el archivo que se utilizará para almacenar los datos de la web.

Para realizar una petición se utiliza el lenguaje SQL. Con este lenguaje se pueden crear tablas dentro del archivo donde almacenar los datos. Por ejemplo, en este mismo archivo podemos encontrar la siguiente petición:

```
//peticion
$peticion =
<<<SQL
CREATE TABLE auditorios (
idauditorio INTEGER PRIMARY KEY,
nombre CHAR(120) Not Null,
pais CHAR(40) Not Null,
provincia CHAR(40) Not Null,
ciudad CHAR(40) Not Null,
foto CHAR(40) Not Null,
latitud INTEGER,
longitud INTEGER,
descripcion TEXT,
activado CHAR(40) Not Null
);
SQL;
```

La variable **\$peticion** incluye una secuencia de comandos SQL que permitirán crear una tabla llamada “auditorios” en el archivo de la base de datos. En esta secuencia se indican los nombres de las columnas junto con el tipo de datos que albergará la columna.

Tras crear la variable petición podemos observar el siguiente código:

```
//ejecutar
$resultado = sqlite_exec($conexion,$peticion);
//cerrar
sqlite_close($conexion);
```

Se define la variable **\$resultado** como una función **sqlite_exec()** a la cual se le envían las variables **\$conexion** y **\$peticion**. Esta función se conectará a la base de datos definida en **\$conexion** y ejecutará la secuencia SQL definida en **\$peticion**, en este caso, crear la tabla “auditorios” con sus respectivas columnas.

Por último se invoca a la función **sqlite_close(\$conexion)** para cerrar el archivo de la base de datos. Esto no es obligatorio, pero se realiza para mayor seguridad.

Este tipo de peticiones se realizan frecuentemente en el panel de administración de Auralizarte por ejemplo para insertar un nuevo auditorio en la base de datos.

En lugar de realizar una petición con el comando **CREATE**, utilizaríamos el comando **INSERT**, por ejemplo en el archivo `addauditorio.php`:

```
$peticion =  
<<<SQL  
INSERT INTO auditorios VALUES  
(null,'$addnombreauditorio','$addpaisauditorio','$addprovinciaauditorio','$  
addciudadauditorio','Suba una foto jpg, gif o  
png','$addlatitudauditorio','$addlongitudauditorio','','No');  
SQL;
```

El comando inserta por orden cada uno de los campos separado por comas.

En cambio para realizar una consulta y poder mostrar los datos de la base de datos en pantalla se utiliza el comando **SELECT**. Por ejemplo en el archivo `gestorauditorios.php` podemos encontrar el siguiente código:

```
//Conectar a la base de datos  
$conexion = sqlite_open('BBDD/auralizarte.db');  
//Peticion  
$peticion = "SELECT * FROM auditorios ORDER BY nombre ASC;";  
//Resultado  
$resultado = sqlite_query($conexion,$peticion);
```

Primero se realiza la conexión con el archivo `auralizarte.db`, a continuación se define la variable `$peticion` como un conjunto de comandos que selecciona mediante el comando **SELECT** todos los datos de la tabla `auditorios` y los ordena por orden alfabético ascendente.

El resultado completo de la petición, en este caso todos los datos de los auditorios, se almacenan en la variable `$resultado` al ejecutar la función `sqlite_query()`.

Además de los comandos **CREATE** o **SELECT** se han utilizado otros comandos para operar con la base de datos como por ejemplo **UPDATE**, que permite modificar un campo ya creado en una tabla o **DELETE** para eliminar registros específicos.

Estructuras de control IF y WHILE

Las estructuras de control permiten controlar el flujo de un programa, tomar decisiones dependiendo de ciertas condiciones o realizar acciones de manera repetitiva. Las estructuras de control son muy comunes en la mayoría de lenguajes de programación, entre ellos PHP.

En Auralizarte se han utilizado principalmente las estructuras de control IF para ejecutar fragmentos de código dependiendo de unas condiciones y WHILE para realizar acciones repetitivas.

Por ejemplo, el control de acceso al contenido en el panel dependiendo de los permisos de usuario se controlan con un condicional IF. Podemos observar en el archivo administración.php el siguiente código:

```
if($_SESSION['permisos'] == 2){
    include 'gestorusuarios.php';
}else{
    echo '<div class="alert alert-block">
    <h4>¡Atención!</h4>
    No tiene permisos de administrador.
    </div>';
}
```

Si el usuario ha iniciado sesión correctamente, se crea una variable de sesión en el archivo **login.php** llamada “permisos” que tiene el valor 1 si es editor o 2 si es administrador de la web. Gracias al condicional IF se comprueba el permiso del usuario. Si el usuario es administrador, se ejecutará la función **include()** que añade el código de **gestorusuarios.php** mostrando el gestor de usuarios. En cambio si el usuario es editor, mostrará un mensaje de atención indicándole que no tiene permisos para acceder a dicha sección.

Esta comprobación de permisos de administrador también se utiliza en ciertos archivos donde el usuario editor no deba tener acceso. Así se evita que pueda acceder con tan solo escribir la dirección del archivo en la URL.

La estructura de control WHILE es el tipo de bucle más sencillo en PHP. Al igual que la IF depende de una condición. Mientras esta condición se cumpla, ejecuta el fragmento de código indefinidamente. En el momento en que la condición sea falsa, el bucle finaliza.

El bucle WHILE se ha utilizado en el panel de administrador de Auralizarte para mostrar el listado de auditorios o de usuarios. En el archivo gestorusuarios.php podemos encontrar el siguiente código:

```
//Conectar a la base de datos
$conexion = sqlite_open('BBDD/auralizarte.db');
//Petición
$peticion = "SELECT * FROM usuarios;";
//Resultado
```



```

$resultado = sqlite_query($conexion,$peticion);
//Mostrar resultado
while($fila = sqlite_fetch_array($resultado)){
echo " <tr><td>".$fila['idusuario']."</td>
//Código omitido para el ejemplo
}

```

En este caso podemos observar que, tras conectar a la base de datos, realizar la petición de seleccionar todos los datos de la tabla “usuarios” y ejecutar el resultado, el bucle WHILE ejecuta indefinidamente la función **sqlite_fetch_array()**. Esta función va almacenando los resultados de la consulta en el array **\$fila**. Cuando no encuentra más datos en **\$resultado**, devuelve un FALSE que hace que el bucle WHILE finalice.

De esta forma se puede construir una tabla HTML con los usuarios o los auditorios por ejemplo almacenados en la base de datos.

Funciones

En PHP existen funciones integradas como es la función “**echo**” anteriormente citada, pero existe la posibilidad de definir nuestras propias funciones. Se define una función propia cuando sabemos que cierto código se va a utilizar frecuentemente. En Auralizarte se han definido ciertas funciones por ejemplo para crear un directorio en el servidor. Cada vez que se añade un auditorio en la base de datos, se crea una carpeta de nombre el número de identificador de dicho auditorio.

En el archivo **funciones.php** podemos encontrar las funciones creadas para el panel de administración. Entre ellas la función “**crear_directorio_auditorio**”:

```

function crear_directorio_auditorio ($nombrecarpeta) {
//Creamos las carpetas fotos y audio con el nombre del parámetro de
entrada.
$directoriofotos =
mkdir("../archivos/auditorios/".$nombrecarpeta."/fotos",0777,true);
$directorioaudio =
mkdir("../archivos/auditorios/".$nombrecarpeta."/audio",0777,true);
};

```

Podemos observar cómo a la función se le envía un parámetro **\$nombrecarpeta** que definirá precisamente el nombre del directorio. La función integrada de PHP **mkdir** crea el directorio con permisos **0777**, es decir, de lectura, escritura y

ejecución. El valor **true** de la función **mkdir** permite que en la carpeta se puedan crear directorios anidados.

Include

Include es una función interna de PHP que permite insertar el código de un archivo en otro. Esto es de gran utilidad si hay código que se repite en todas las páginas. En la cabecera del panel de administración por ejemplo se definen los archivos de estilo CSS o los archivos Javascript que son necesarios para la interfaz. Como es un código que se repite en cada página, se puede separar en un archivo independiente e invocarlo con la función include donde sea necesario.

Si es hubiese que cambiar la ruta del archivo de estilos o añadir un nuevo archivo Javascript, con tan solo modificar el archivo **header.php** afectaría a todas las páginas por igual.

Podemos ver el uso de esta función en la mayoría de los archivos del panel de Auralizarte, por ejemplo en la cabecera del archivo **administracion.php**:

```
//Incluir header  
include 'temas/auralizarte/header.php';
```

Sesiones

En una aplicación web es muy frecuente necesitar recordar parámetros a lo largo de una sesión. En Auralizarte las sesiones se utilizan para recordar los datos del usuario que accede al panel de administración. Las variables de sesión son capaces de guardar durante un periodo de tiempo establecido dichos datos. Estas variables pueden ser utilizadas en diferentes momentos de la ejecución del código.

En el caso de Auralizarte y para tener un control de permisos y seguridad sobre el panel, las variables de sesión se crean cuando un usuario accede correctamente en el panel. Esto es en caso de que los datos insertados por el usuario coincidan con un usuario almacenado en la base de datos. El archivo **login.php** crea las variables de sesión email, nombre y permisos:

```
$_SESSION['email'] = $email;  
$_SESSION['nombre'] = $fila['nombre'];  
$_SESSION['permisos'] = $fila['permisos'];
```

Como hemos podido observar anteriormente, la variable `$_SESSION['permisos']` se utiliza frecuentemente en la estructura de control IF para saber si un usuario que ha accedido al panel es editor o administrador.

Capítulo 6: “CONCLUSIONES Y LÍNEAS FUTURAS DE TRABAJO”

En este capítulo se presentan las conclusiones obtenidas tanto a nivel de trabajo como a nivel personal. También se muestran futuras líneas de trabajo que podrían mejorar el portal.

Conclusiones sobre el trabajo realizado

La web de Auralizarte ha variado mucho desde su concepción hasta su versión final. No fue fácil plasmar la idea original en un portal atractivo para el usuario. La idea inicial sobre visitas virtuales en los auditorios no se podía abordar de una manera sencilla. Esta implicaba construir con una técnica compleja y costosa. Por esta razón hubo que pivotar hacia algo más sencillo. Plantear un mapa en el que mostrar los auditorios y con el que el usuario pudiese interactuar hizo que quedasen mejor definidos los objetivos que había que alcanzar.

La aplicación de convolución de la sección educARTE se simplificó a un listado de archivos de audio dentro de las características de un auditorio, pero la propuesta de poder comparar todas las características acústicas y geométricas entre dos auditorios hizo igual de atractiva esta sección de la web.

Los primeros meses del proyecto fueron los más difíciles. No quedaban claros los objetivos que se debían alcanzar. Una vez se definieron ciertas especificaciones, se pudo comenzar a trabajar sobre el Backend. Las características básicas del panel de administración tales como el acceso de usuario, el gestor de auditorios o el gestor de usuarios podían diseñarse sin necesidad de saber cómo iban a utilizarse los datos.

El proyecto se pudo dividir en dos vías de trabajo diferenciadas, Backend y Frontend. En ciertas ocasiones las dos vías confluían para determinar la manera en la que se mostrarían los datos al usuario en el Frontend.

Se puede afirmar que no es recomendable comenzar un desarrollo web sin que los requisitos del Frontend estén claros. Esto provocó por ejemplo que se desarrollase durante un mes un gestor de artículos que más tarde se desechó. La falta de especificaciones provocaron en más de una ocasión este tipo de errores.

Cuando quedó definido el Frontend, se pudo realizar apartados más concretos del Backend como por ejemplo el diccionario de la web o el panel de configuración. Estos apartados dependían directamente del Frontend y por tanto no se podían abordar.

Concluir que el proyecto del portal Auralizarte se prolongó más de lo debido por diferentes cuestiones: Un retraso en la falta de especificaciones en su inicio, lo cual generaba incertidumbre sobre los objetivos que se debían alcanzar, retrasos en las comunicaciones por ambas partes e indefinición de los ejecutantes del portal en la relación con el proyecto Auralizarte en su integridad y con el departamento de la universidad (la relación se limitó a ofrecer un servicio al departamento en lugar de una colaboración como cliente). Una vez quedaron claros los objetivos, el portal se pudo completar satisfactoriamente.

Conclusiones personales

Durante la realización de este proyecto he encontrado numerosos problemas y retos que he conseguido superar satisfactoriamente. Mi experiencia previa en desarrollo web ha ayudado enormemente, pero abordar todos los lenguajes de programación web a la vez no suele ser lo más común en un desarrollo profesional. Esto me ha ayudado no solo a ampliar mi conocimiento y técnica, sino a ser flexible y versátil en el terreno profesional. Crear un gestor de contenidos a medida y no utilizar un CMS o un framework PHP fue un reto a la vez que un proceso de aprendizaje muy valioso.

El proyecto planteaba desde el principio la dificultad de realizarlo a distancia entre dos personas. Esto ha hecho mejorar enormemente mis habilidades de comunicación y trabajo en equipo. No siempre es fácil mantener la asertividad para que los objetivos de un proyecto se cumplan, pero en este caso no resultó ser un problema.

En cuanto a mi visión sobre el proyecto, creo que habría que someterlo a un proceso de experimentación previa a su lanzamiento. Desde un principio se estableció un cliente objetivo (“músicos e ingenieros acústicos”) a los cuales presuntamente el contenido de la web de Auralizarte iba a ser de su interés. ¿Cómo se sabe que esto es cierto?

Lo más común en todo proyecto es que los emprendedores o el equipo emprendedor se centren en **“lo que saben hacer y les diferencia”** y por eso se cree que el cliente objetivo va a responder igual que piensa el equipo emprendedor. Se crea una visión del cliente idealizada, en la que se establece una estrategia deliberada con muy pocas probabilidades de éxito.

En lugar de centrarse en “**lo que sabemos hacer y nos diferencia**” habría que realizar un esfuerzo en pensar la “**propuesta de valor**” del proyecto, es decir, buscar una razón real por la que los clientes “músicos e ingenieros acústicos” van a entrar en el portal Auralizarte y lo van a recomendar a sus círculos. Esto solo se puede realizar mediante la experimentación en el mercado con clientes reales en lo que se llama “**estrategia emergente**”.

Por tanto y desde mi punto de vista, la página web de Auralizarte debería servir como “**producto mínimo viable**” (Ries, 2013) para comprobar sobre el mercado real si el cliente objetivo va a tener interés en acceder al portal e interesarse por sus contenidos. En caso de que no lo tenga, preguntarle por qué y aprender de ello para “pivotar” sobre la idea original.

Líneas futuras de trabajo

Este portal puede utilizarse como producto mínimo viable del proyecto Auralizarte. Por tanto habría que utilizarlo a modo de experimento para validar si el cliente objetivo tiene interés en el proyecto. Carecería de sentido mantener por ejemplo la sección entrenARTE si nadie utilizase la herramienta de comparación o la sección auralizARTE si nadie consultase el mapa de auditorios. Este proceso de análisis de resultados es fundamental realizarlo en un futuro próximo.

Desde el punto de vista de ingeniería se podrían implementar las siguientes funcionalidades siempre y cuando se validen antes con el cliente objetivo:

- **Aplicación de convolución:** Con el que el usuario puede experimentar cómo se oirían en los diferentes auditorios sus propios archivos de audio. Esta aplicación tendría sentido desarrollarla cuando la librería Web audio API sea compatible con todos los navegadores.
- **Visita virtual del auditorio:** Se podría desarrollar un sistema de visita virtual basado únicamente en fotografías del interior del auditorio en lugar de basarlo sobre un modelo 3D.
- **Dinamizar y socializar la web:** En general la web es poco dinámica en contenido. Se debería realizar un esfuerzo en implementar un apartado de artículos que se actualice periódicamente o una sección social en la que los usuarios puedan consultar sus dudas e inquietudes.
- **Monetizar la web:** En caso de que la web tuviese un número significativo de visitas, se podría monetizar mediante donaciones o patrocinadores que apoyen al proyecto.

Referencias bibliográficas

- W3Schools (2013) *W3Schools.com*, [Online], Available: http://www.w3schools.com/tags/ref_httpmethods.asp [2013].
- Web And Macros (2007) *Web And Macros*, [Online], Available: <http://www.webandmacros.com/webdinamica.htm> [2013].
- Apple (2013) *Apple Developer*, [Online], Available: <https://developer.apple.com/library/mac/documentation/macosx/Conceptual/BPFrameworks/Concepts/WhatAreFrameworks.html> [2013].
- Bloomberg (2014) *Bloomberg Businessweek*, [Online], Available: <http://investing.businessweek.com/research/stocks/earnings/earnings.asp?ticker=AMZN> [2014].
- Diego Cantor, B.J. (2012) *WebGL Beginner's Guide*, Birmingham: Packt Publishing Ltd.
- Dirksen, J. (2013) *Learning Three.js: The JavaScript 3D Library for WebGL*, Birmingham: Packt Publishing Ltd.
- Google (2013) *Google Investor Relations*, [Online], Available: <http://investor.google.com/financial/tables.html> [2014].
- Holdener, A.T. (2008) *Ajax: The Definitive Guide*, Sebastopol: O'Reilly Media.
- Krug, S. (2006) *No me hagas pensar. Una aproximación a la usabilidad en la Web*, Madrid: Pearson Educación, S.A.
- Luke Welling, L.T. (2009) *Desarrollo web con PHP y MySQL*, Madrid: Anaya Multimedia.
- Nations, D. (2013) *About.com*, [Online], Available: http://webtrends.about.com/od/wiki/a/what_is_a_wiki.htm [2013].
- Newenham, P. (2013) 'Cumulus Dublin design conference', Cumulus Dublin design conference, Dublin.
- Meyers, G. (2002) *Web Techniques*, [Online], Available: <http://people.apache.org/~jim/NewArchitect/webtech/2002/02/meyers/index.html> [2013].
- Paul, R. (2009) *Arstechnica*, [Online], Available: <http://arstechnica.com/information-technology/2009/04/google-releases-3d-graphics-plugin-for-browsers/> [2013].
- Ries, E. (2013) *El método LEAN STARTUP*, Nueva York: DEUSTO.
- Rowse, D. (2005) *Probblogger*, [Online], Available: <http://www.probblogger.net/archives/2005/02/05/what-is-a-blog/> [2013].
- Smus, B. (2013) *Web Audio API*, Sebastopol: O'Reilly Media.
- Twitter (2013) *Twitter*, [Online], Available: <https://blog.twitter.com/2013/celebrating-twitter7> [2014].
- The Smashing Magazine (2013) *The Smashing Book #4. New Perspectives On Web Design*, Greiburg: Smashing Magazine GmbH.
-

ANEXO

auralizARTE

Manual de usuario

Revisión:

v. 0.0.01

Fecha última versión:

octubre de 2013

HOJA DE CONTROL DOCUMENTAL

Resumen	Manual de usuario de auralizARTE		
Autor	Eduardo García, Pablo Diego	Fecha	25/10/2013
Revisada por	Ricardo San Martín	Fecha	
Anexos	N/A	Número de páginas	N/A

CONTROL DE VERSIONES

Versión	Fecha	Realizada por	Descripción
0.0.01	25/10/2013	Eduardo García	Versión inicial

Tabla de contenido

HOJA DE CONTROL DOCUMENTAL	2
1. Introducción	4
1.1 Finalidad	4
BACKEND	4
2. Instalación de la web.....	4
2.1 Servidor local o Hosting	4
2.2 Proceso de instalación del portal.....	5
3. Panel de administración	7
3.1 Acceso al panel de administración.....	7
4. Administrar auditorios.....	8
4.1 Añadir auditorios	8
4.2 Editar, parámetros acústicos, auralizaciones y eliminar auditorios	9
4.3 Activar o desactivar un auditorio.....	10
5. Textos de la web.....	10
6. Gestor de usuarios	11
6.1 Añadir un usuario (sólo con permisos de administrador).....	11
6.2 Editar y borrar usuarios (sólo con permisos de administrador).....	11
7. Configuración.....	12
7.1 Configurar contacto.....	12

1. Introducción

1.1 Finalidad

El propósito del presente documento es proporcionar una descripción detallada de las funcionalidades de la página web auralizARTE, que sirva de guía tanto para el administrador/editor de la web como para el usuario visitante.

El documento se divide en dos partes dependiendo del perfil de usuario y por lo tanto del uso que se haga de la web.

- A. *Backend* (usuario administrador/editor)
- B. *Frontend* (usuario visitante de la web)

En la primera parte (*Backend*) se aborda el proceso de instalación del portal y la gestión del mismo. Esta parte está destinada sólo a usuarios con permisos de administrador/editor (ver apartado 6 para más información).

La segunda parte (*Frontend*) explica el funcionamiento de la web desde el punto de vista del usuario visitante; los diferentes apartados del portal y cómo se debe interactuar con él.

BACKEND

2. Instalación de la web

Requisitos mínimos:

- Servidor Apache 2.2.21
- PHP 5.3.6
- SQLite 3.7.4

2.1 Servidor local o Hosting

Para realizar una instalación en un servidor local se recomienda utilizar uno de estos programas que cumplen con los requisitos mínimos de la web:

- A. WAMP (Windows)
- B. MAMP (Mac OSX)
- C. XAMPP (Windows, Mac OSX y Linux)

Para obtener información detallada sobre la instalación de cada uno de estos servidores consultar las siguientes direcciones:

- A. <http://www.wampserver.com/>

- B. <http://www.mamp.info/>
- C. <http://www.apachefriends.org/es/xampp.html>

Una vez instalado el servidor local, **debe copiar el contenido del archivo Auralizarte.zip** en el directorio correspondiente (varía en cada caso).

Tras ello podrá acceder al proceso de instalación del portal en la siguiente dirección:

<http://localhost/rutadelaweb/admin/instalacion/>

Nota: Dicha ruta dependerá del software utilizado. La ruta relativa es **admin/instalación/**

En caso de que haya contratado un servicio de **Hosting externo**, el proceso es similar. **Debe copiar el contenido del archivo Auralizarte.zip en la raíz del sitio** y acceder a la siguiente dirección:

<http://www.direcciondelaweb.com/admin/instalacion/>

2.2 Proceso de instalación del portal

Si ha realizado correctamente el proceso anterior, tras acceder a la dirección de instalación aparecerá la siguiente pantalla de bienvenida.



Tras pulsar el botón continuar se creará la base de datos y deberá rellenar el formulario para crear un **usuario administrador** del sitio.

Instalación AuralizARTE

✓ Base de datos creada correctamente.

Usuario administrador

Por favor rellene el siguiente formulario para crear el usuario administrador del sitio.

Nombre	<input type="text" value="Nombre"/>
Apellidos	<input type="text" value="Apellidos"/>
Email	<input type="text" value="Email"/>
Contraseña	<input type="password"/>
Repetir contraseña	<input type="password"/>

Finalizar

Tras pulsar el botón Finalizar aparecerá el siguiente mensaje.

Instalación AuralizARTE

✓ Instalación finalizada.

Para mayor seguridad, por favor elimine la carpeta **admin/instalacion**

Ir al panel de administración

Es importante que elimine la carpeta de instalación del servidor. Con esto habrá finalizado la instalación y podrá acceder al panel de administración haciendo clic en el botón o accediendo a la carpeta *admin* desde su navegador (ver apartado 3.1).

Nota: La base de datos es un archivo llamado “auralizarte.db” dentro de la carpeta “admin/BBDD”. Para realizar una copia de seguridad tan solo copie dicho archivo periódicamente en un lugar seguro. Si quiere restablecer una copia de seguridad, tan solo sobrescriba dicho archivo.

3. Panel de administración

En este apartado se detalla la dirección de acceso al panel de administración y una breve descripción de los diferentes apartados del mismo.

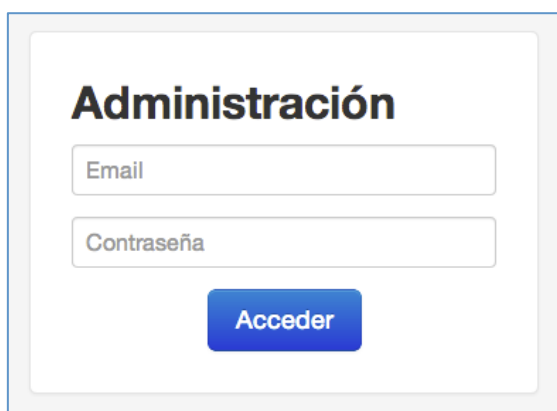
3.1 Acceso al panel de administración

Para acceder al panel de administración hay que ingresar en la dirección siguiente desde el navegador de internet:

<http://www.direcciondelaweb.com/admin>

En caso de que la instalación se realice en un servidor local, lo más común es que la dirección sea del tipo <http://localhost/>, por ejemplo <http://localhost/admin> o <http://localhost/rutadelaweb/admin>

Aparecerá el siguiente formulario:



Administración

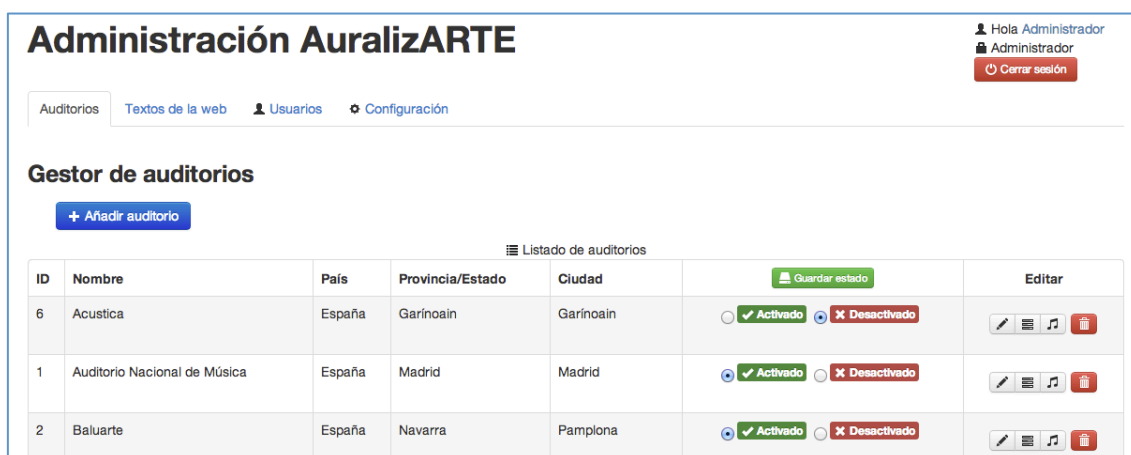
Email

Contraseña

Acceder

En él debe indicar el email y contraseña del usuario administrador que proporcionó durante el proceso de instalación.

Tras introducir los datos de acceso se mostrará el panel principal.



Administración AuralizARTE

Hola Administrador
Administrador
Cerrar sesión

Auditorios Textos de la web Usuarios Configuración

Gestor de auditorios

+ Añadir auditorio

Listado de auditorios

ID	Nombre	País	Provincia/Estado	Ciudad	Guardar estado	Editar
6	Acustica	España	Garínoin	Garínoin	<input type="radio"/> Activado <input checked="" type="radio"/> Desactivado	
1	Auditorio Nacional de Música	España	Madrid	Madrid	<input checked="" type="radio"/> Activado <input type="radio"/> Desactivado	
2	Baluarte	España	Navarra	Pamplona	<input checked="" type="radio"/> Activado <input type="radio"/> Desactivado	

En él podemos distinguir las siguientes secciones:

1. **Menú principal:** Desde donde se puede navegar a los apartados “Auditorios”, “Textos de la web”, “Usuarios” y “Configuración”. Cada apartado puede contener subapartados diferenciados por pestañas. En los siguientes capítulos se aborda en profundidad cada uno de ellos.
2. **Identificador y botón de cerrar sesión:** Permite visualizar el nombre de usuario con el que ha accedido, sus permisos (administrador o editor) y el botón de cerrar sesión con el que podrá volver al formulario de acceso al panel de administración.
3. **Gestor de contenido:** Variable en cada apartado, permite gestionar los diferentes contenidos de la web.

The screenshot shows the 'Administración AuralizARTE' interface. At the top right, there is a user profile box with the text 'Hola Administrador', 'Administrador', and a 'Cerrar sesión' button. Below this is a navigation menu with 'Auditorios', 'Textos de la web', 'Usuarios', and 'Configuración'. The main content area is titled 'Gestor de auditorios' and contains a '+ Añadir auditorio' button and a 'Listado de auditorios' table. The table has columns for ID, Nombre, País, Provincia/Estado, Ciudad, and actions (Guardar estado, Editar). The table lists three auditorios: ID 6 (Acustica, España, Garínoain, Garínoain), ID 1 (Auditorio Nacional de Música, España, Madrid, Madrid), and ID 2 (Baluarte, España, Navarra, Pamplona). Each row has 'Activado' and 'Desactivado' radio buttons and an 'Editar' button with icons for edit, list, music, and delete.

4. Administrar auditorios

Esta sección permite añadir, editar y borrar auditorios de la base de datos del portal. Estos auditorios son los que se mostrarán tanto en el mapa de AuralizARTE como en EntrenARTE.

4.1 Añadir auditorios

Para añadir un auditorio en la base de datos siga el siguiente procedimiento:

1. Pulse el botón “**Añadir auditorio**”.
2. Aparecerá un formulario con los datos básicos necesarios para crear una entrada en la base de datos.
 - **Nombre:** Indique el nombre del auditorio.
 - **País:** Indique el país donde se encuentra el auditorio.
 - **Provincia/Estado:** Indique la provincia o estado donde se encuentra el auditorio.
 - **Ciudad:** Indique la ciudad donde se encuentra el auditorio.

- **Población:** Indique la población donde se encuentra el auditorio.
- **Latitud:** Indique la latitud donde se posiciona el auditorio.
- **Longitud:** Indique la longitud donde se posiciona el auditorio.

Nota: Para obtener la Latitud y la Longitud del auditorio puede consultar el siguiente link: <http://itouchmap.com/latlong.html>

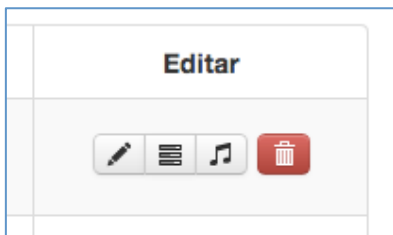
3. Pulse “**Guardar cambios**” para crear el auditorio en la base de datos.

Podrá observar como el auditorio se añade a la lista en estado “**desactivado**”.

Nota: Al añadir un auditorio nuevo se crea una carpeta dentro del servidor en la ruta “archivos/auditorios/”. Esta carpeta tiene como nombre el número identificador del auditorio (ID) y contiene tanto la foto del auditorio como los archivos de audio del mismo.

4.2 Editar, parámetros acústicos, auralizaciones y eliminar auditorios

Para completar todos los datos del auditorio puede pulsar los botones “Editar auditorio”, “Parámetros acústicos” o “Archivos de audio” situados en la parte derecha del listado.



Editar auditorio: Permite acceder al panel de edición para completar todos los datos del auditorio. Además de los datos iniciales, incluye la fotografía del auditorio y una descripción del mismo.



Parámetros acústicos: Permite acceder al panel de parámetros acústicos. En él se puede introducir la curva TR30 del auditorio, una descripción breve de la misma, y los valores “Capacidad del auditorio”, “Volumen total” y “TRmid”. **Estos últimos tres datos se utilizan en el filtro del mapa de la sección auralizARTE.**



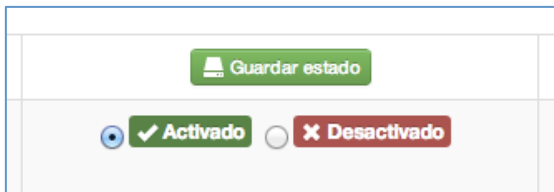
Archivos de audio: Permite acceder al gestor de archivos de audio. Puede añadir tanto archivos en formato WAV como MP3 así como una pequeña descripción del audio. Los archivos se guardan en una carpeta llamada “audio” dentro de la carpeta del auditorio correspondiente.



Eliminar auditorio: Si pulsa el botón eliminar se eliminará automáticamente el auditorio tanto de la base de datos como toda la carpeta que contiene sus archivos en el servidor.

4.3 Activar o desactivar un auditorio

Cuando añada un auditorio, **este aparece desactivado por defecto** y por lo tanto no aparece en el mapa de auralizARTE.



Una vez termine de editar todos los datos puede activar o desactivar auditorios del listado seleccionando “Activado” o “Desactivado”. A continuación debe pulsar el botón “Guardar estado”.

5. Textos de la web

Esta sección permite modificar y traducir los textos estáticos¹ de la web. El portal detecta automáticamente el idioma del navegador y cambia su idioma de manera transparente al usuario. **Si el navegador no está en castellano, el idioma que aparecerá será el inglés.**

La sección está dividida en dos pestañas, “Diccionario” y “Educarte”:

- **Diccionario:** Contiene los textos estáticos de la web, es decir, el nombre de los botones, títulos, subtítulos, y en general textos que poco o nada deben cambiar.
- **Educarte:** Contiene el texto de dicha sección. Por lo general es un contenido estático aunque puede actualizarse ocasionalmente.

Dentro de cada una de ellas podemos identificar otras dos pestañas con los dos idiomas disponibles, Español e Inglés.

Todo el contenido es código PHP y HTML, por lo que si no tiene conocimientos sobre ellos, consulte con un programador web.

¹ Se define como texto estático aquel que apenas varía en contenido, en este caso los textos de los botones o el contenido de educARTE por ejemplo.

IMPORTANTE: El contenido está estructurado con arrays llamados \$TEXT, por lo tanto es importante modificar únicamente el contenido del array:

\$TEXT['nombre del array'] = '**Contenido a modificar**';

6. Gestor de usuarios

Esta sección permite añadir, editar y borrar usuarios de la base de datos del portal. Estos usuarios pueden ser de dos tipos, administradores o editores.

- **Administrador:** Acceso completo a la gestión del portal.
- **Editor:** Acceso restringido al panel de administración. Sólo puede gestionar la sección de auditorios y modificar su contraseña de acceso dentro de la pestaña configuración.

6.1 Añadir un usuario (sólo con permisos de administrador)

Para añadir un usuario a la base de datos siga el siguiente procedimiento:

1. Acceda a la pestaña **Usuarios (sólo disponible si es administrador)**.
2. Pulse el botón **“Añadir usuario”**.
3. Aparecerá un formulario con los datos necesarios para crear un nuevo usuario:
 - **Nombre:** Indique el nombre del usuario.
 - **Apellidos:** Indique los apellidos del usuario.
 - **Email:** Indique el email del usuario. Necesario para iniciar sesión.
 - **Permisos de usuario:** Seleccione entre Editor (por defecto) y Administrador.
 - **Contraseña:** Elija una contraseña.
 - **Repetir contraseña:** Repita la contraseña anterior.
4. Pulse el botón **“Crear usuario”** para finalizar el proceso.

6.2 Editar y borrar usuarios (sólo con permisos de administrador)



Editar usuario: Para editar un usuario ya existente pulse el botón editar. Accederá al panel de edición donde podrá modificar los datos. Pulse el botón **“Guardar cambios”** cuando finalice.



Eliminar usuario: Elimina el usuario de la base de datos.

7. Configuración

La pestaña “Configuración” permite configurar ciertas características de la web. Estas son:

- **Nueva contraseña de acceso:** Permite modificar la contraseña de acceso al panel de administración de su usuario.
- **Mapa Auralizarte:** En este panel debe introducir una clave API válida para que funcione el mapa de auralizarte. Durante la instalación se asigna una por defecto, pero es recomendable cambiarla por otra válida en la dirección <https://developers.google.com/maps/>
- **Contacto:** Desde este panel puede configurar el servidor de correo por defecto que se utilizará para que un usuario visitante contacte con los responsables de la web a través del formulario de contacto.

7.1 Configurar contacto

Para que un usuario se pueda poner en contacto con los responsables de la web debe configurar primero este panel.

- **Email de contacto:** Indique el email que utilizará como contacto. Es el email al cual recibirá los mensajes de los usuarios.
- **Servidor SMTP:** Indique el servidor SMTP a utilizar. Dependerá del servicio de email que contrate. Algunos ejemplos:
 - **Gmail:** smtp.gmail.com
 - **Hotmail:** smtp.live.com
- **Email servidor:** Este es el email que se utilizará como medio para enviar los correos electrónicos. Suele ser del tipo noreply@dominio.com
- **Contraseña email servidor:** Indique la contraseña del email servidor anterior.

DISEÑO E IMPLEMENTACIÓN DEL BACKEND DEL SITIO WEB DEL PROYECTO AURALIZARTE

Eduardo García Lara

21 de febrero de 2014

INDICE

1. ESTADO DEL ARTE
2. OBJETIVOS
3. DESARROLLO E IMPLEMENTACIÓN
4. DEMOSTRACIÓN
5. FUTURAS LÍNEAS DE TRABAJO Y CONCLUSIONES

ESTADO DEL ARTE

ESTADO DEL ARTE

- Las páginas web de antes
 - Carta de presentación de las empresas.
 - Elementos simples de hipertexto.
 - Páginas estáticas simples sin apenas navegación.



Welcome to Amazon.com Books!

*One million titles,
consistently low prices.*

(If you explore just one thing, make it our personal notification service. We think it's very cool!)

SPOTLIGHT! -- AUGUST 16TH

These are the books we love, offered at Amazon.com low prices. The spotlight moves **EVERY** day so please come often.

ONE MILLION TITLES

Search Amazon.com's [million title catalog](#) by author, subject, title, keyword, and more... Or take a look at the [books we recommend](#) in over 20 categories... Check out our [customer reviews](#) and the [award winners](#) from the Hugo and Nebula to the Pulitzer and Nobel... and [bestsellers](#) are 30% off the publishers list...

EYES & EDITORS, A PERSONAL NOTIFICATION SERVICE

Like to know when that book you want comes out in paperback or when your favorite author releases a new title? Eyes, our tireless, automated search agent, will send you mail. Meanwhile, our human editors are busy previewing galleys and reading advance reviews. They can let you know when especially wonderful works are published in particular genres or subject areas. Come in, [meet Eyes](#), and have it all explained.

YOUR ACCOUNT

Check the status of your orders or change the email address and password you have on file with us. Please note that you **do not** need an account to use the store. The first time you place an order, you will be given the opportunity to create an account.

Shop by Department

Search All

Go

Hello. Sign in Your Account

Try Prime

Cart

Wish List

- Unlimited Instant Videos
- MP3s & Cloud Player
20 million songs, play anywhere
- Amazon Cloud Drive
5 GB of free storage
- Kindle Books & E-readers
- Kindle Fire Tablets
- Appstore for Android
Get WordsWorth for free today
- Digital Games & Software
- Books & Audible
- Movies, Music & Games
- Electronics & Computers
- Home, Garden & Tools
- Beauty, Health & Grocery
- Toys, Kids & Baby
- Clothing, Shoes & Jewelry
- Sports & Outdoors
- Automotive & Industrial
- Full Store Directory

¿Estás comprando desde España? Visita amazon.es > Entrar aquí
Shopping from Spain?

Instant Video MP3 Store Cloud Player Kindle Cloud Drive Appstore for Android Digital Games & Software Audible Audiobooks

kindle fire HDX

And the revolutionary "Mayday" button

From \$229 > Shop now



Women in Wartime Clothing Trends FREE Two-Day Shipping

Amazon Fashion

WOMEN'S JACKET + COATS

Stay warm in down puffers, wool peacoats, and more.

> Shop Women's Coats
> Shop All Clothing



St. Patrick's Day > Shop now



Unlimited access to thousands of TV shows
Prime Instant Video

Start Free Trial



Splash of Color
Unlocked iPhone 5c

> Shop now



amazon basics

Go-Anywhere Speaker > Learn more



Related to Items You've Viewed

You viewed Customers who viewed this also viewed



Best Sellers
Watches : Luxury Watches
Updated hourly

ESTADO DEL ARTE

- ¿Qué ha provocado esta evolución estética y tecnológica?

UN CAMBIO EN EL
MODELO DE NEGOCIO
DE LAS EMPRESAS EN INTERNET

ESTADO DEL ARTE

- Ventajas de internet para las empresas:
 - Amplían su cartera de clientes
 - Reducen costes (RRHH, geográficos)
 - El cliente no se mueve de su casa
 - **AUMENTAN SUS BENEFICIOS**
- Esta demanda de las empresas en internet hizo evolucionar y aparecer nuevos lenguajes de programación.

ESTADO DEL ARTE

- **ANTES:** El desarrollador web **se centraba en la programación:**
 - Código limpio y simple
- **AHORA:** El desarrollador web de ahora **se centra en el cliente, en el usuario y el desarrollador:**
 - Estética
 - Usabilidad
 - Accesibilidad (navegadores y dispositivos móviles)

OBJETIVOS

OBJETIVOS

- OBJETIVO PRINCIPAL
 - **Desarrollar un Backend usable y accesible para el portal del proyecto Auralizarte**
- OBJETIVOS ESPECÍFICOS
 - **Cumplir los requisitos del cliente** (departamento de acústica)
 - **Crear** un sistema de gestión de usuarios interno
 - **Crear** un sistema de configuración
 - **Crear** un sistema de seguridad
 - **Crear** un proceso de instalación

DESARROLLO E IMPLEMENTACIÓN

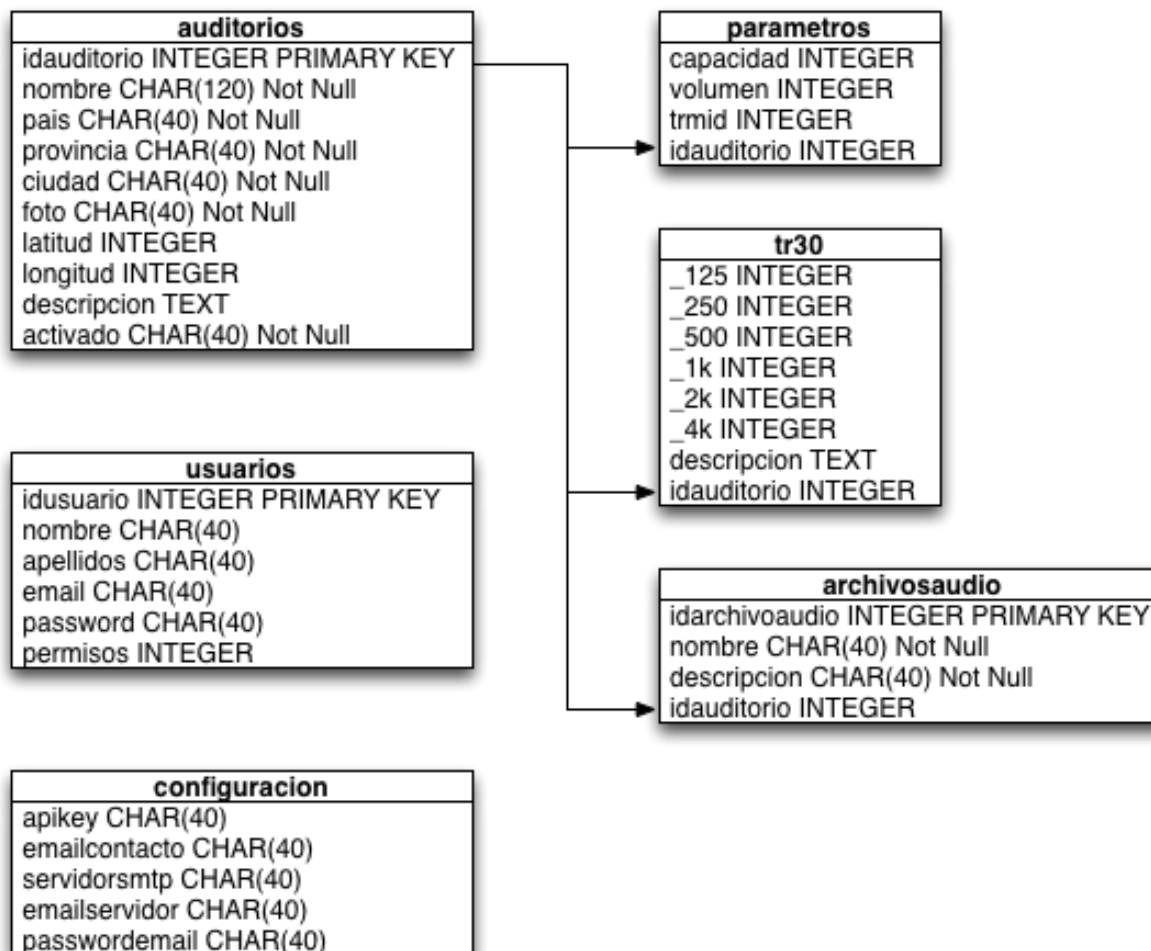
Tecnologías utilizadas

- **PHP** (lenguaje del servidor)
- **SQLite** (base de datos)
- **Bootstrap** (interfaz)
- **jQuery y Javascript** (interacción)

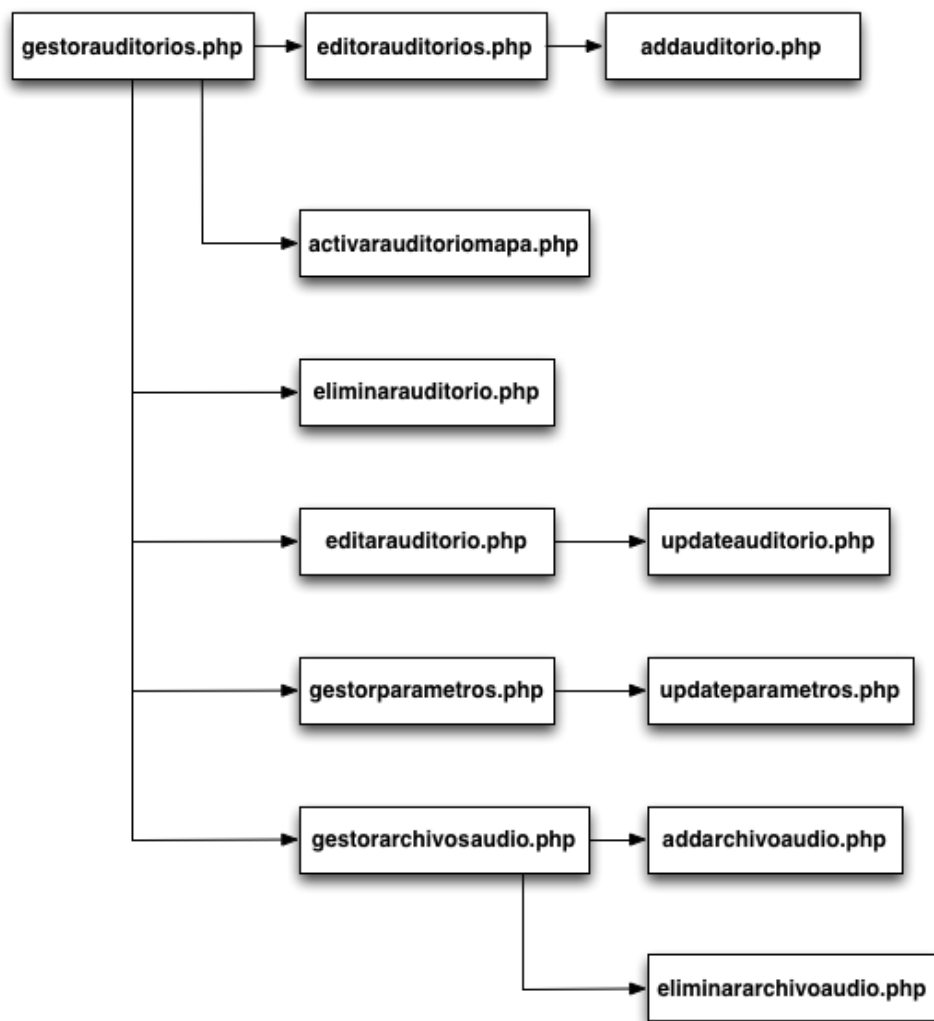
Requisitos del cliente

- **auralizARTE**: Repositorio de auditorios.
- **educARTE**: Documentación sobre la técnica de auralización.
- **entrenARTE**: Comparación entre auralizaciones.

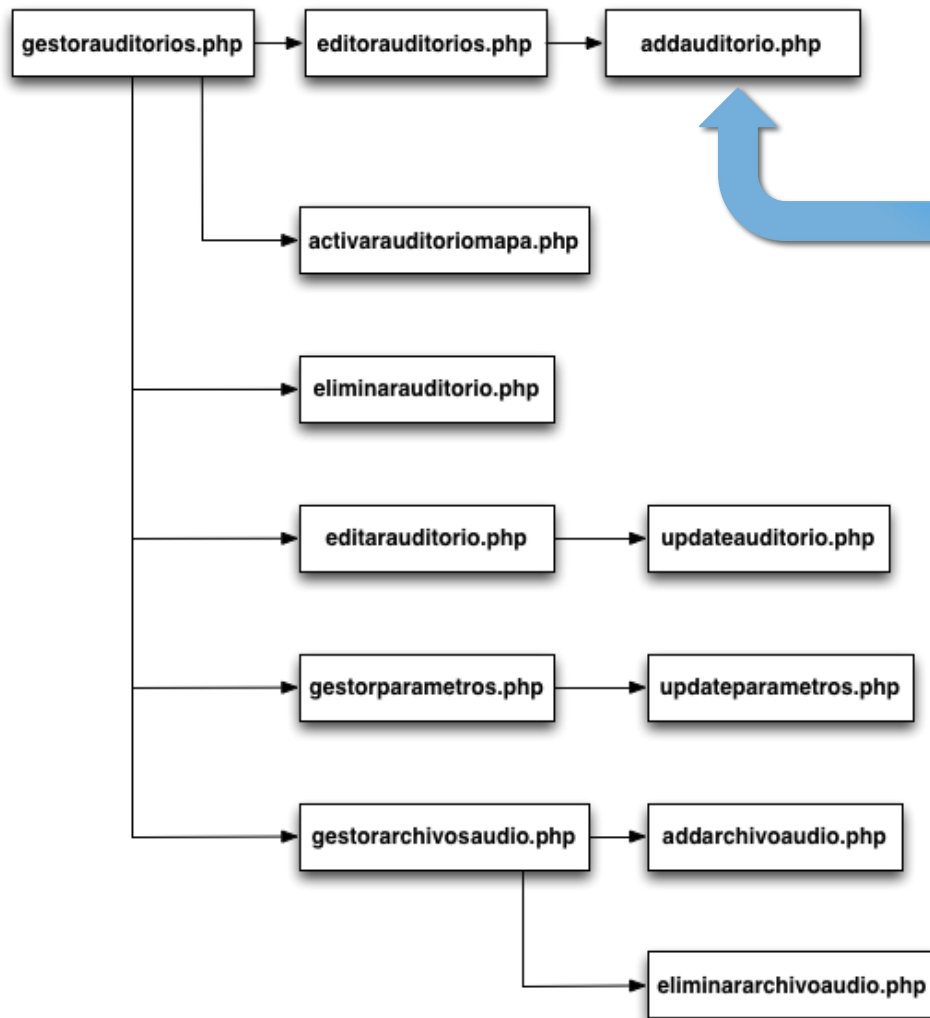
Base de datos: Diseño



auralizARTE: Gestor de auditorios



auralizARTE



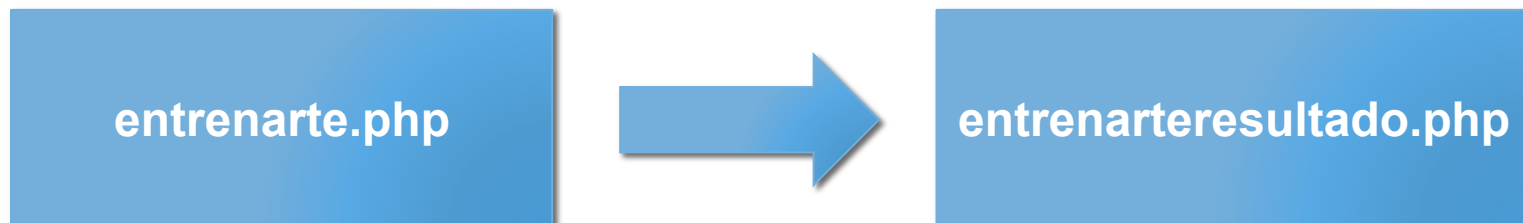
- Incluye archivo de seguridad.
- Incluye archivo de funciones.
- Conecta con la base de datos.
- Recupera los nuevos datos.
- Inserta datos en la BBDD.
- Obtiene el idauditorio.
- Crea una carpeta única en el servidor
- Inicializa los parámetros y la curva TR30.
- Cierra la conexión.
- Refresca la página con un mensaje.

educARTE: Acciones desde el panel

- **Modificar** el contenido del archivo de documentación.
- Modificarlo en **varios idiomas**.

entrenARTE

- **No hay acciones realizables desde el panel, es automático.**
- Las acciones las realiza el usuario.
- Los resultados se procesan internamente para mostrarlos en pantalla.



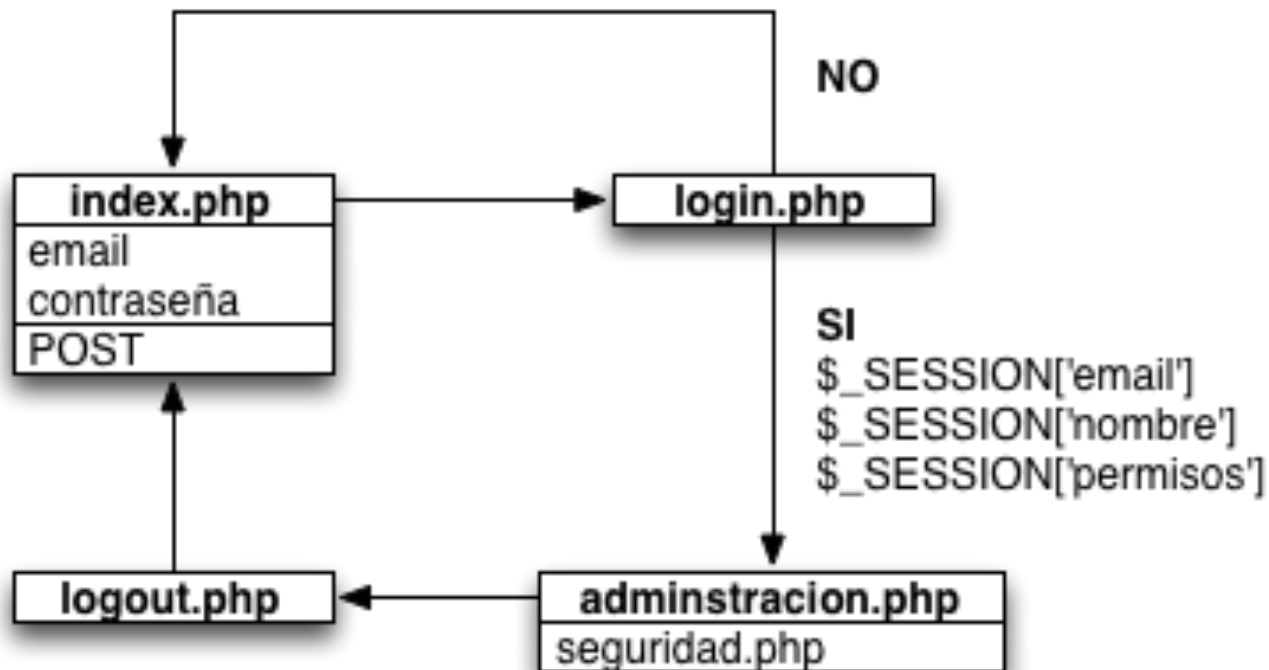
Gestión de usuarios interno

- **Añadir** usuarios de tipo administrador o editor
- **Editar** los usuario existentes
- **Eliminar** usuarios

Configuración de la web

- **Modificar** la contraseña de acceso
- **Modificar** la API Key de Google Maps
- **Configurar** el servicio de contacto de la web

Seguridad de la web: Acceso



DEMOSTRACIÓN

Futuras líneas de trabajo

- Realizar una aplicación de convolución
- Visitas virtuales en el auditorio con fotografías
- Dinamizar y socializar la web, blog, foro, etc.
- Testar cómo interactúa el usuario

Conclusiones

- Centrarse en “**la proposición de valor**”.
- **Producto mínimo viable.**
- Experimentación y obtención de **feedback**.

MUCHAS GRACIAS