DOCUMENTACION :-(

El cuestionario del Siglo XIX nos pondrá a prueba de la información que aparece en la pagina del siglo XIX, se accede desde opción del menú de la pagina ¿que sabes del siglo XIX? en el podremos elegir 3 dificultades y nos hará 10 preguntas aleatorias cada vez que hagamos el cuestionario. Una vez realizado podremos corregirlo para comprobar nuestros conocimientos.

**Componentes:**

*Almacenamiento:*

Para el almacenamiento de las preguntas hemos optado por alojar la información en una base de datos. En ella se ha creado una tabla:

- wp\_preguntas -> Esta guardara las preguntas.

Y las tres tienen los siguientes campos:

- id (int)

- enunciado (varchar)

- respuesta1 (varchar)

- respuesta2 (varchar)

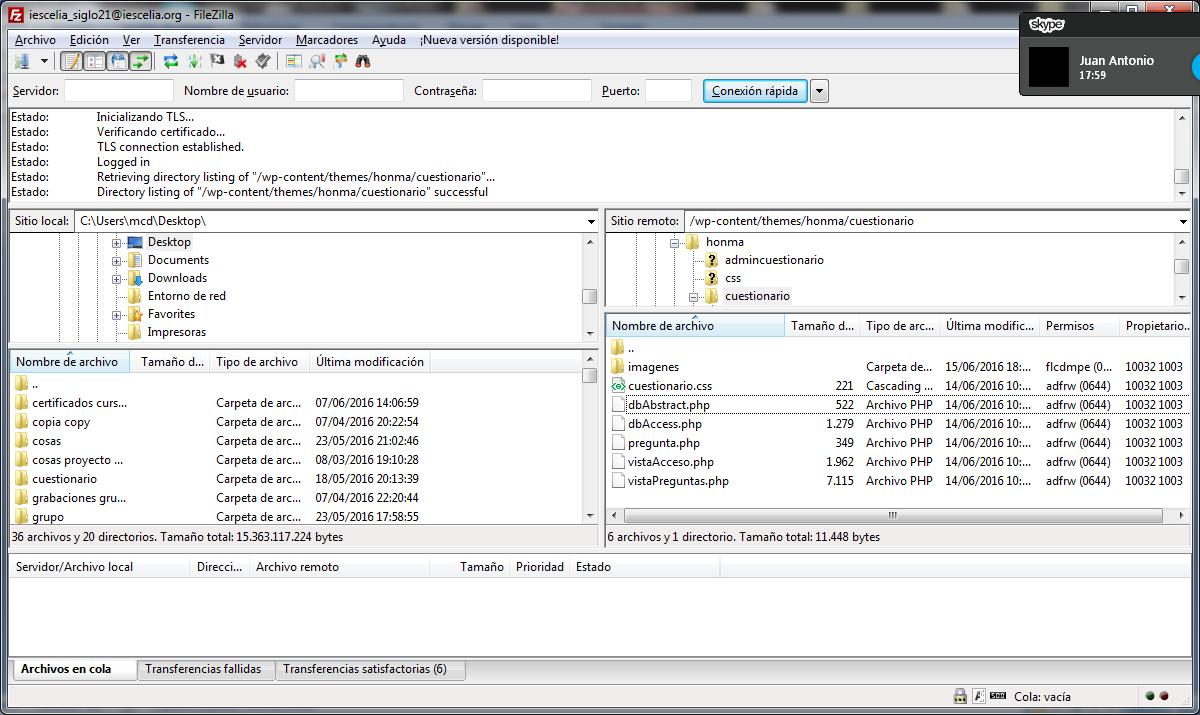
- respuesta3 (varchar)

- dificultad (varchar) que se almacenan las preguntas fáciles, normales y difícil.

- imagen (varchar)

*Archivos:*

A partir de la plantilla usada de wordpress alojada en wp-content/themas/honma/cuestionario.php y desde esa plantilla se ha incluido el código correspondiente para relacionar los demás archivos creados en la carpeta cuestionario. Los archivos se encuentran en wp-content/themas/honma/cuestionario junto con una carpeta imágenes.



**Funcionamiento:**

*Resumen:*

El cuestionario empieza con una pantalla de login, donde de se introduce un nombre y se escoge la dificultad, una vez que has rellenado el formulario te lleva a la pantalla de responder las preguntas del cuestionario que estará compuesta por 10 preguntas. El usuario solo podrá corregir el cuestionario una vez contestado las mismas y pulsando el botón de corregir.

Una vez corregido aparecerá un mensaje que nos mostrará una evaluación sobre el cuestionario y el número de aciertos junto a un enlace donde nos da la oportunidad de volver a empezar de nuevo, justo debajo estará la tabla mostrando todas las preguntas correctas y las incorrectas. Pulsándole a cada pregunta nos llevara a ella para poder ver la corrección y ver la respuesta correcta en el caso de que has fallado la respuesta. En cada pregunta aparecerá un enlace para volver a la corrección de abajo.

*Patrón de arquitectura:*

El modelo usado ha sido MVC(modelo vista controlador).

*Flujo de ejecución*

Como se ha expuesto anteriormente, el archivo principal es cuestionario.php este comprueba si viene informada la variable "empezar", si no es así, llamara a la vistaAcceso.php que mostrara el formulario de acceso, .

Funciones:

dbAccess.php

|  |  |
| --- | --- |
| dbAccess() | |
| Datos de entrada |  |
| Datos de salida |  |
| Descripción | Crea un objeto de la clase dbAbstract |

|  |  |
| --- | --- |
| crearConexion() | |
| Datos de entrada |  |
| Datos de salida |  |
| Descripción | Crea una conexión a la base de datos. |

|  |  |
| --- | --- |
| getIdPreguntas() | |
| Datos de entrada | * dificultad (String) |
| Datos de salida | * idPreguntas (Array) |
| Descripción | Hace una consulta a la base de datos por dificultad y almacena en un array todas los id de las preguntas. |

|  |  |
| --- | --- |
| getPreguntas() | |
| Datos de entrada | * n (String) |
| Datos de salida | * pregunta (object) |
| Descripción | Realiza una consulta a la base de datos por el id que se le pasa a la función, obteniendo toda la información y almacenándola en un objeto. |

dbAbstract.php

|  |  |
| --- | --- |
| crearConexion() | |
| Datos de entrada | * servidor(String) * usuario(String) * clave(String) * dbname(String) |
| Datos de salida |  |
| Descripción | Contiene el método para crear la conexión de la base de datos por los parámetros de entrada. |

|  |  |
| --- | --- |
| cerrarConexion() | |
| Datos de entrada |  |
| Datos de salida |  |
| Descripción | Cierra la conexión a la base de datos. |

|  |  |
| --- | --- |
| consulta() | |
| Datos de entrada | * sql (String) |
| Datos de salida | * query (consulta sql) |
| Descripción | Realiza una consulta a la base de datos. |

|  |  |
| --- | --- |
| getResultado() | |
| Datos de entrada | * resultado (consulta sql) |
| Datos de salida | * resultado (array asociativo) |
| Descripción | Obtiene una fila de resultado como un array asociativo |

pregunta.php

|  |  |
| --- | --- |
| function\_\_construct | |
| Datos de entrada | * e (String) * res1 (String) * res2 (String) * res3 (String) * foto (String) |
| Datos de salida |  |
| Descripción | Construye un objeto pregunta. |

Atributos:

- enunciado: Se guarda el enunciado de la pregunta.

- r1: Esta será siempre la respuesta correcta.

- r2: Respuesta alternativa.

- r3: Respuesta alternativa.

- foto: nombre del archivo de la imagen.

vistaAcceso.php

|  |  |
| --- | --- |
| mostrar() | |
| Datos de entrada |  |
| Datos de salida | * Datos del formulario (también establece variable de sesión empezar) |
| Descripción | Crea un formulario en pantalla para insertar el nombre y la dificultad. |

vistaPreguntas.php

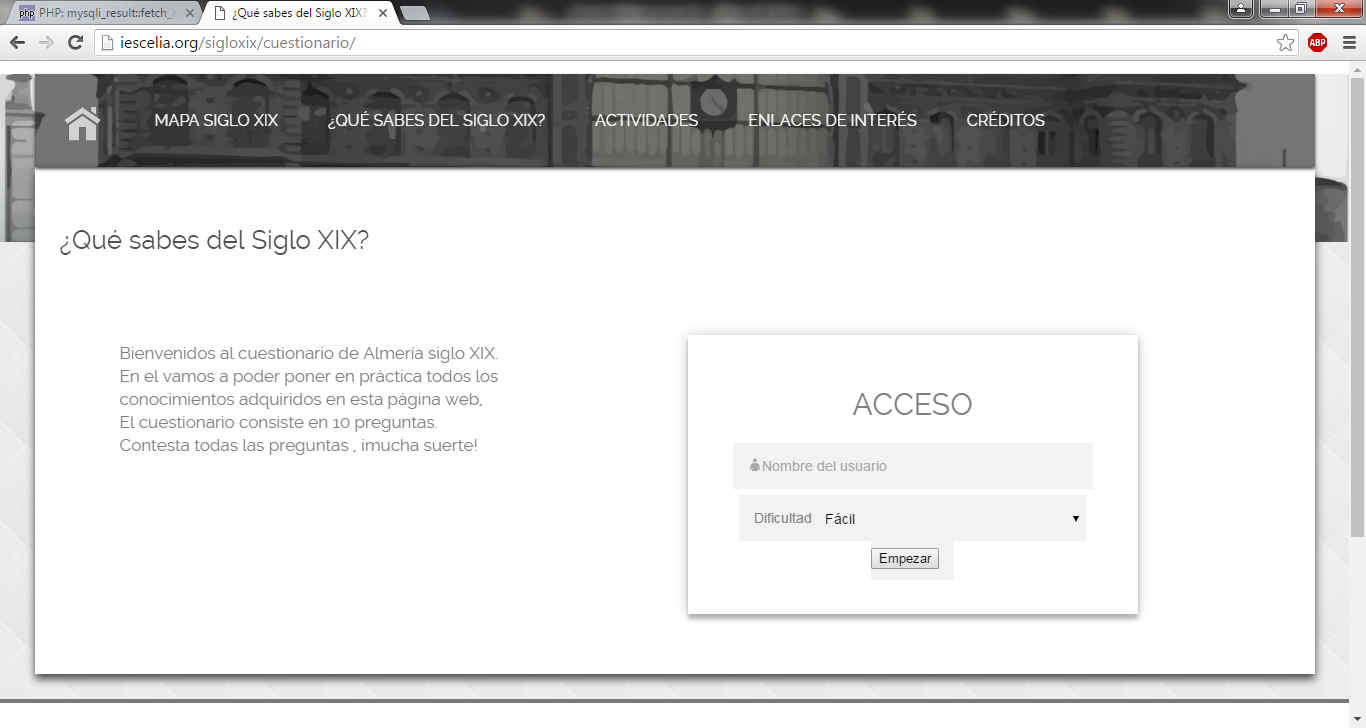
|  |  |
| --- | --- |
| mostrar() | |
| Datos de entrada | * juego(objeto que contiene toda la información de las preguntas) |
| Datos de salida |  |
| Descripción | Crea las capas que contienen las preguntas. |

Se genera una capa por cada pregunta, que contiene el enunciado, las respuestas y la imagen en el caso de que tenga. Las respuestas son capas sensibles al click del ratón seleccionando con el color naranja la respuesta que creemos que es la correcta. Cada vez que pulses una respuesta diferente el color naranja se trasladara a la seleccionada, no dejando nunca elegir dos respuestas en cada pregunta. Al final de la última pregunta estará el botón de corregir que lanzara un evento javascript para corregir las preguntas. Estas se corregirán consultado el id de la pregunta, ya que sabemos que la respuesta correcta siempre es "r1", si la seleccionada es la correcta la pondremos en color verde, si no pondremos la pondremos en rojo y marcaremos en verde la respuesta correcta.

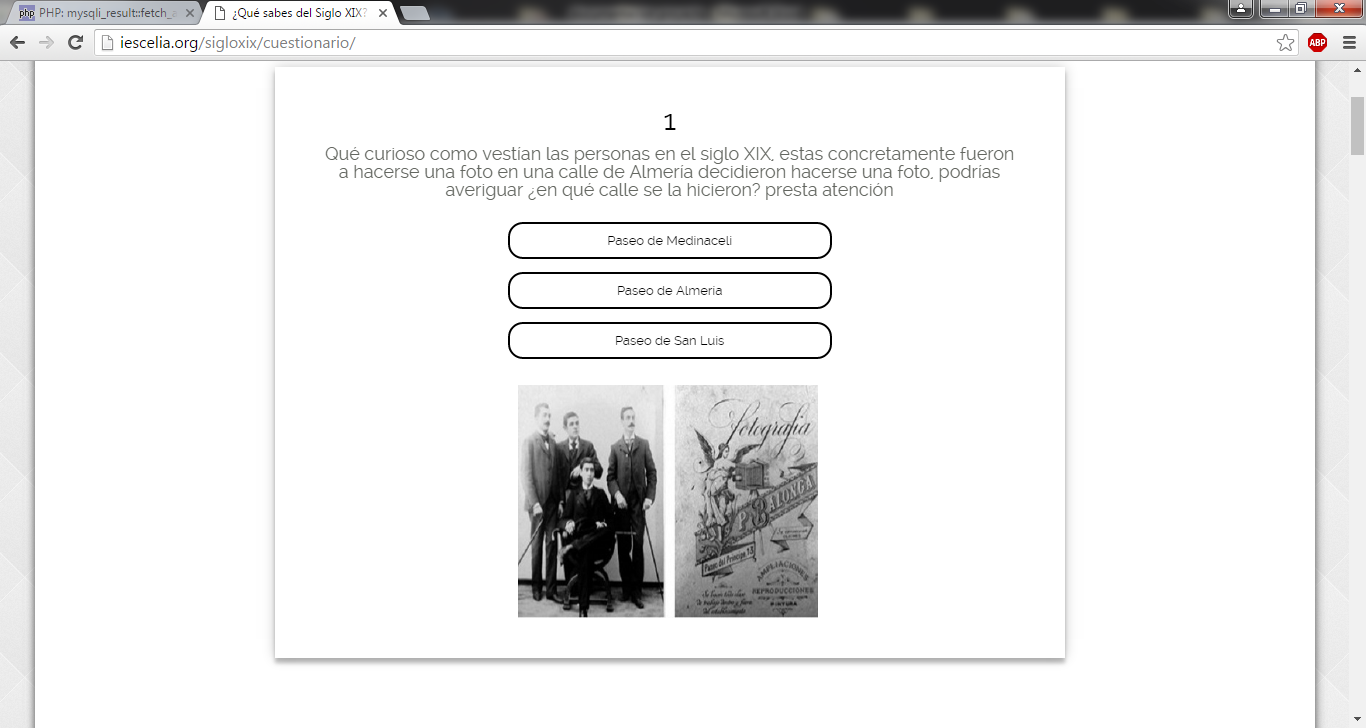
cuestionario.php

Este archivo es el que une el cuestionario con la plantilla principal de wordpress. Este se encarga de llamar a las vistas y a los controladores, a continuación explicaremos el flujo del código:

- Por defecto mostrara la vistaAcceso para escoger nombre y dificultad.



- Si viene de vistaAcceso, se cargaran las 10 preguntas que se mostraran en vistaPreguntas. Se pretende que de la base de datos en la que tenemos más de 10 preguntas, cada vez que juguemos aparezca una combinación diferente de preguntas, para ello almacenamos todos los id de las preguntas de la dificultad seleccionada y se extraen 10 id aleatorios del array, con los cuales se realiza las consulta para obtener dichas preguntas.

- Una vez que tengamos las 10 preguntas se llamara a vistaPreguntas.