

NIA: 10388732

Leyendas de Europa

[Volver al índice](#)

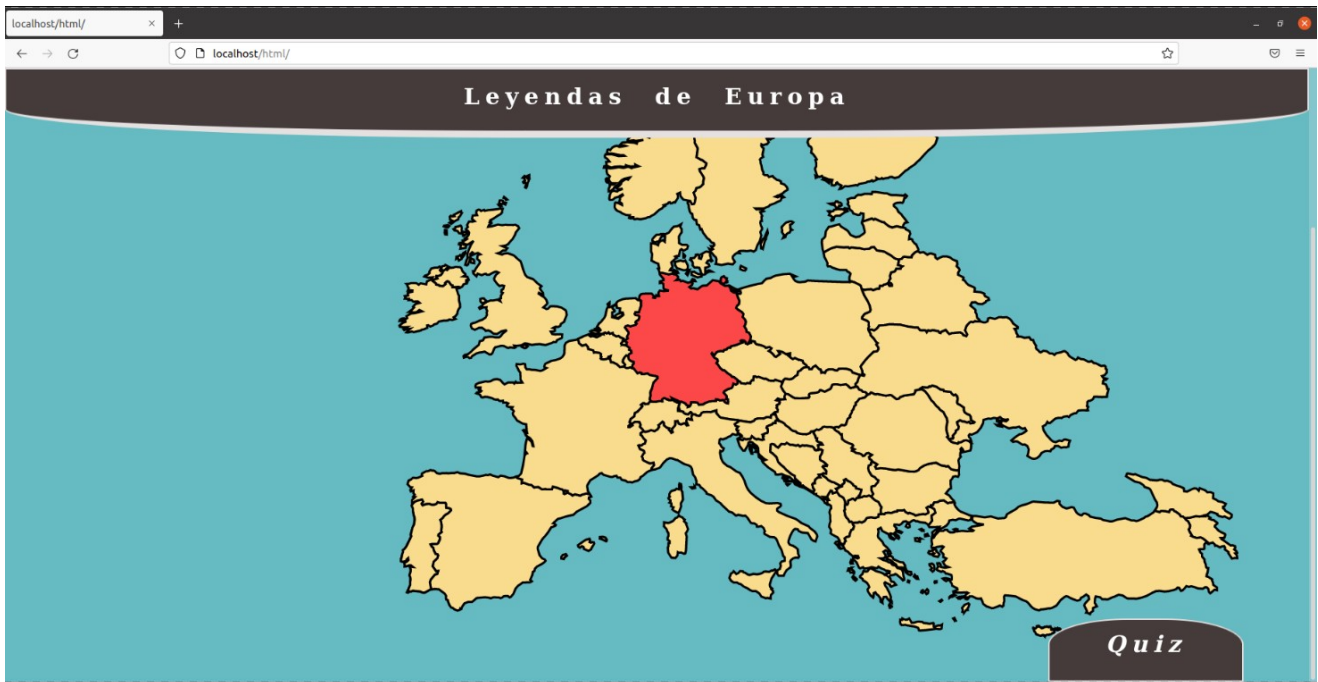
Índice

1. [Introducción](#) **página 3**
2. [Objetivos](#) **página 4**
3. [Planificación temporal](#) **página 5**
5. [Desarrollo](#) **página 8**
 - 5.1 [Base de datos](#) **página 5**
 - 5.2 [Página principal](#) **página 6**
 - 5.3 [Página quiz](#) **página 9**
 - 5.4 [Página de administración](#) **pagina 12**
6. [Conclusiones](#) **página 16**
7. [Integración de los módulos de todo el ciclo](#) **página 17**
8. [Bibliografía](#) **página 18**
9. [Anexos](#)
 - 9.1. [Script de la creación de la base de datos](#) **página 20**
 - 9.2 [Mostrar leyendas](#) **pagina 23**
 - 9.3 [Funcionamiento quiz](#) **página 28**
 - 9.4 [Funcionamiento admin](#) **página 34**

[Volver al índice](#)

Introducción

El concepto inicial de este proyecto era crear una pagina web donde poder ver de forma interactiva mitos y leyendas de los diferentes países de Europa.



En cuanto a la implementación decidí desde un inicio que la pagina principal seria un gran mapa de Europa seccionado por países los cuales poder seleccionar para ver sus leyendas.

La pagina web esta alojada en un entorno LAMP y utiliza las siguientes tecnologías:

- Lenguajes de programación:
 - PHP
 - JavaScript
- Lenguajes de marcas:
 - HTML
 - CSS
- Base de datos:
 - Mysql
- Otras tecnologías:
 - AJAX

[Volver al índice](#)

Objetivos

generales:

- Reducir la carga del servidor utilizando JavaScript y AJAX siempre que se pueda.

Pagina principal

- Crear un mapa interactivo donde al seleccionar un país salga un pop-up con la leyenda de dicho país.
- Poder abrir y cerrar distintas leyendas sin cambiar de pagina.
- Que el contenido de cada leyendas se adecué al formato del pop-up independientemente de su tamaño.

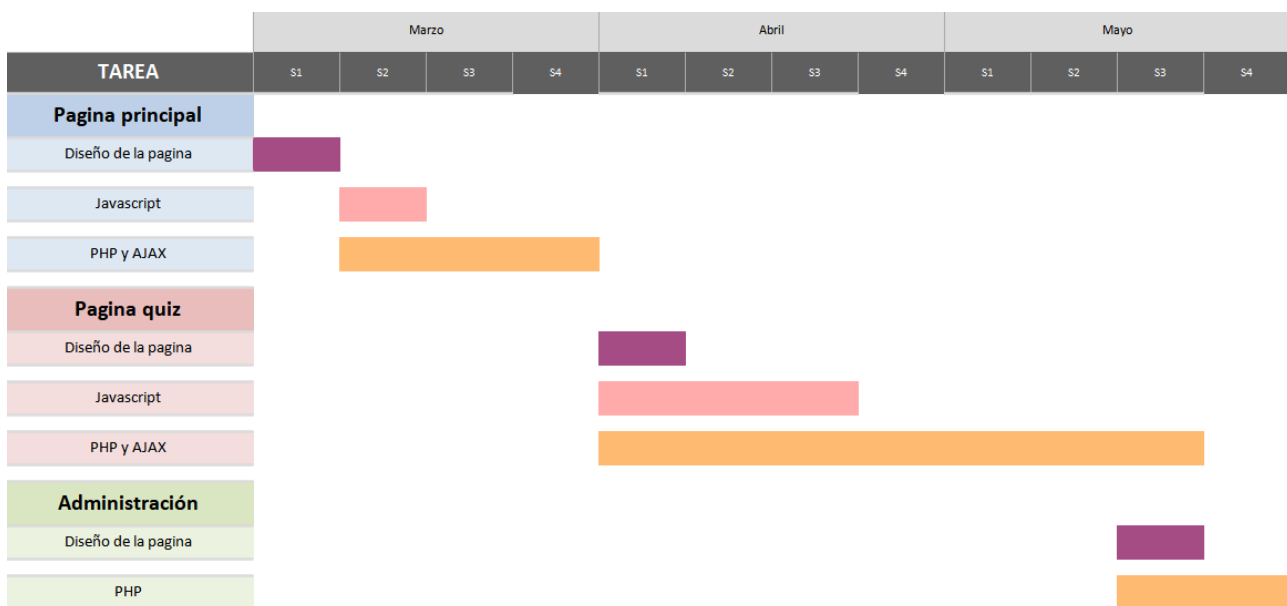
Pagina Quiz

- Crear un quiz sobre las leyendas de la pagina principal.
- Cambiar de pregunta sin tener que salir de la pagina inicial del quiz.
- Que las preguntas sean aleatorias y no se repitan.
- Sumas puntos al acertar y restar vidas al fallar.
- Que el quiz finalice al quedarse sin vidas o responder las cuarenta preguntas.
- Guardar y mostrar la puntuación con un nombre de usuario.
- Mostrar los cinco usuarios con mejor puntuación.

[Volver al índice](#)

Planificación temporal

Cabe recalcar que antes de iniciar la pagina web a inicios de Marzo tuve que recopilar las leyendas de cada país, así como encontrar una imagen para cada leyenda y arreglar dichas imágenes para que salgan bien en la pagina, sin embargo, al no ser parte como tal de la pagina web no las e añadido al diagrama.

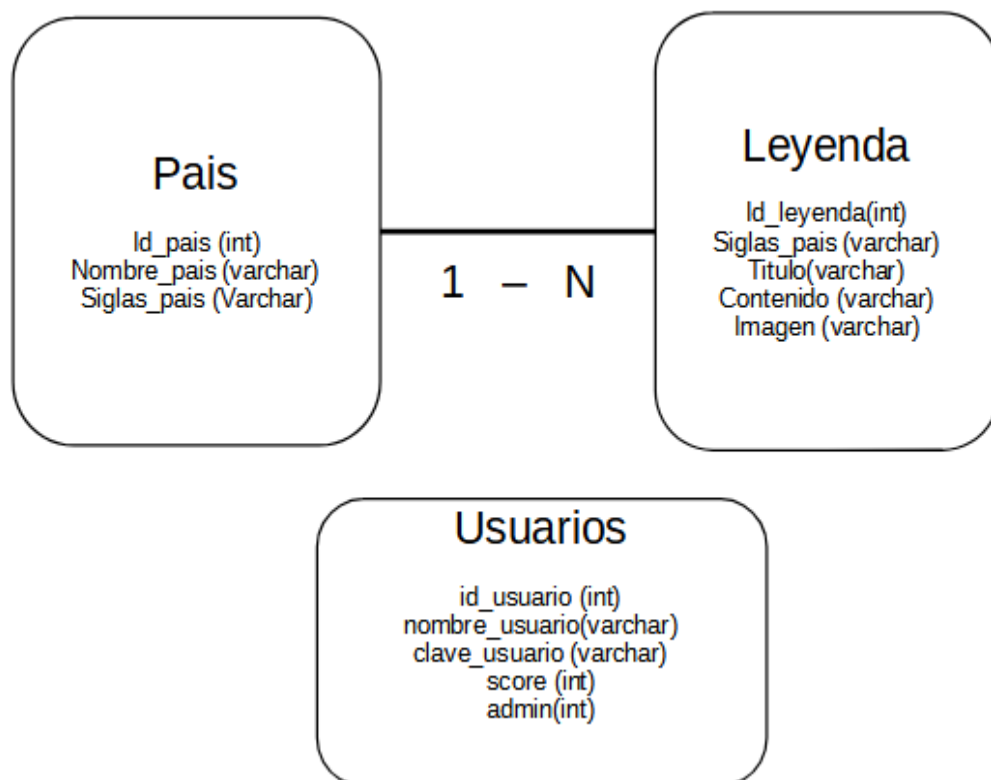


[Volver al índice](#)

Desarrollo

Creación de la bases de datos

Modelo relacional de la base de datos:



script de creación de la base de datos vacía: [Anexo 1](#)

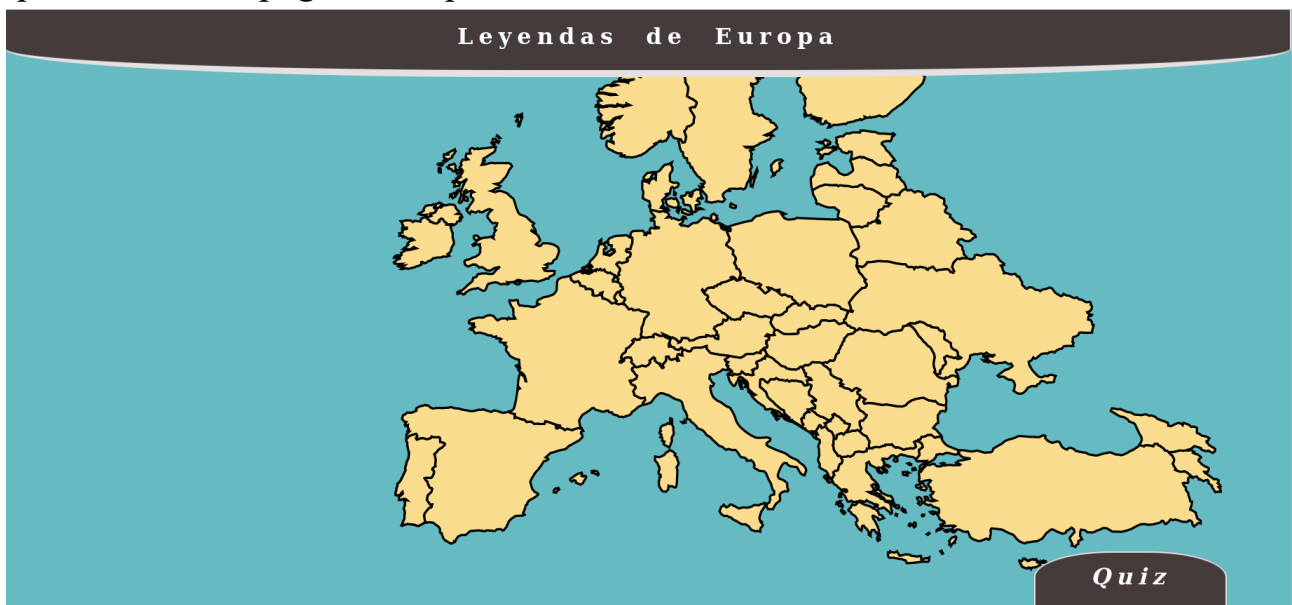
[Volver al índice](#)

Pagina principal

Esta pagina parte de un mapa de Europa en svg que tenia guardados la posición y tamaño exactos de cada país.

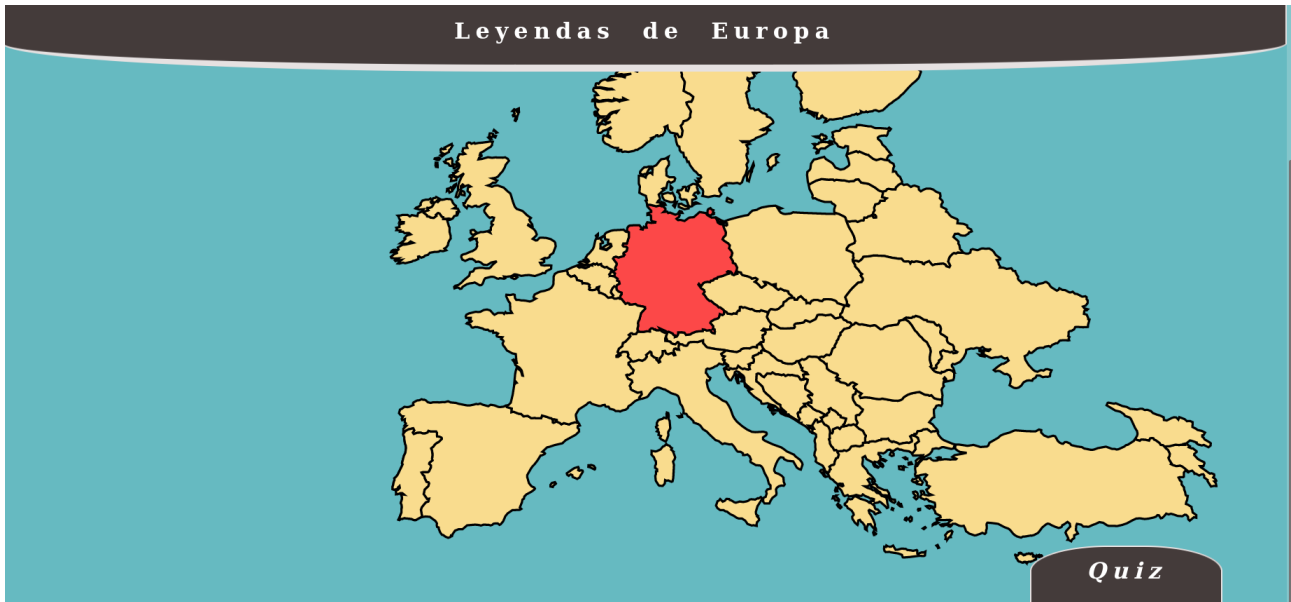


Le añadí algo de formato, una cabecera con el titulo de la pagina y una pestaña que llevara a la pagina del quiz.

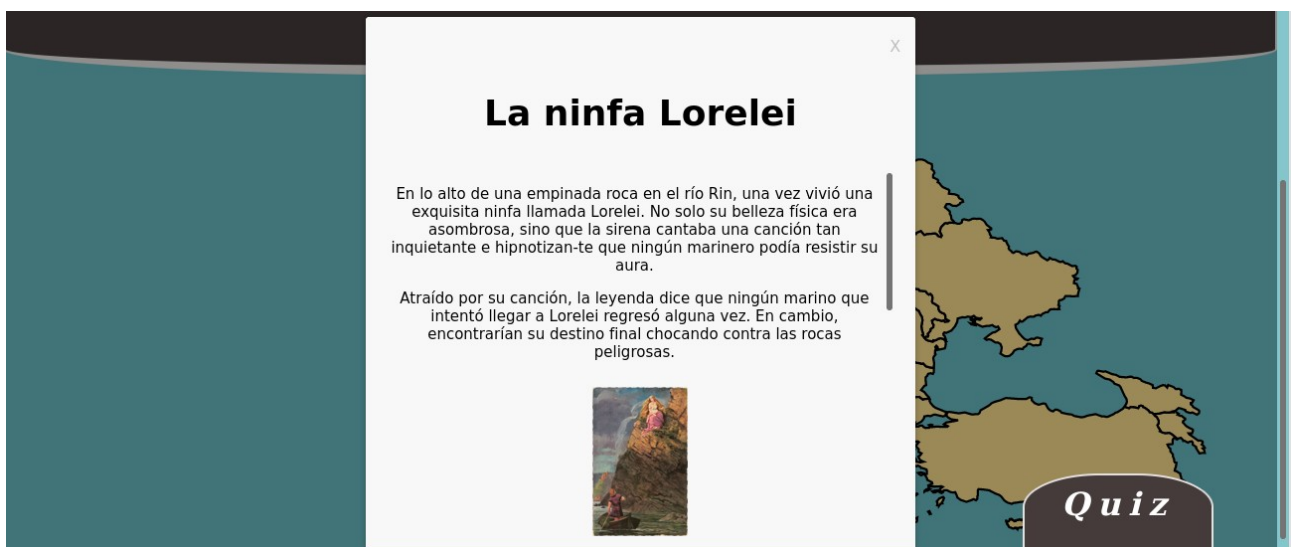


[Volver al índice](#)

Si pasamos el ratón por encima de un país este se cambiara de color como se muestra en la imagen.



Si hacemos clic en un país nos abrirá un pop-up con la leyendas del país seleccionado.



[Volver al índice](#)

Por ultimo si hacemos clic en “Quiz”, nos mandara a la pagina del quiz.

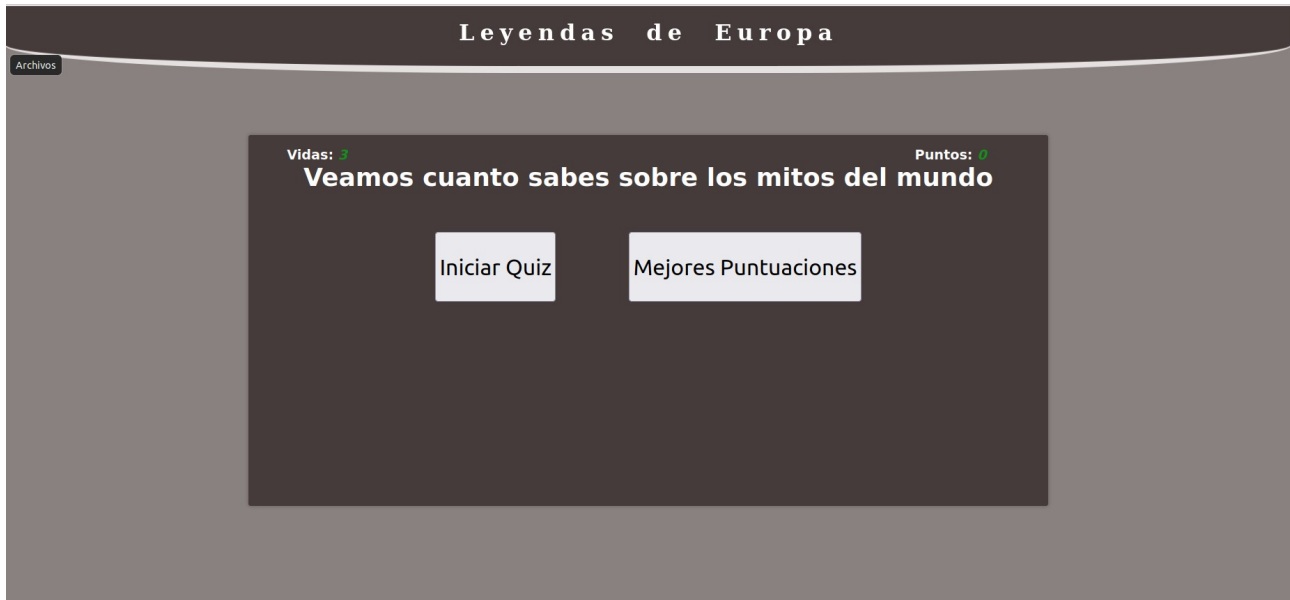


Funcionamiento: [Anexo2](#)

[Volver al índice](#)

Pagina quiz

Como se puede ver aquí tenemos dos opciones: “Iniciar Quiz” y “Mejores puntuaciones”.

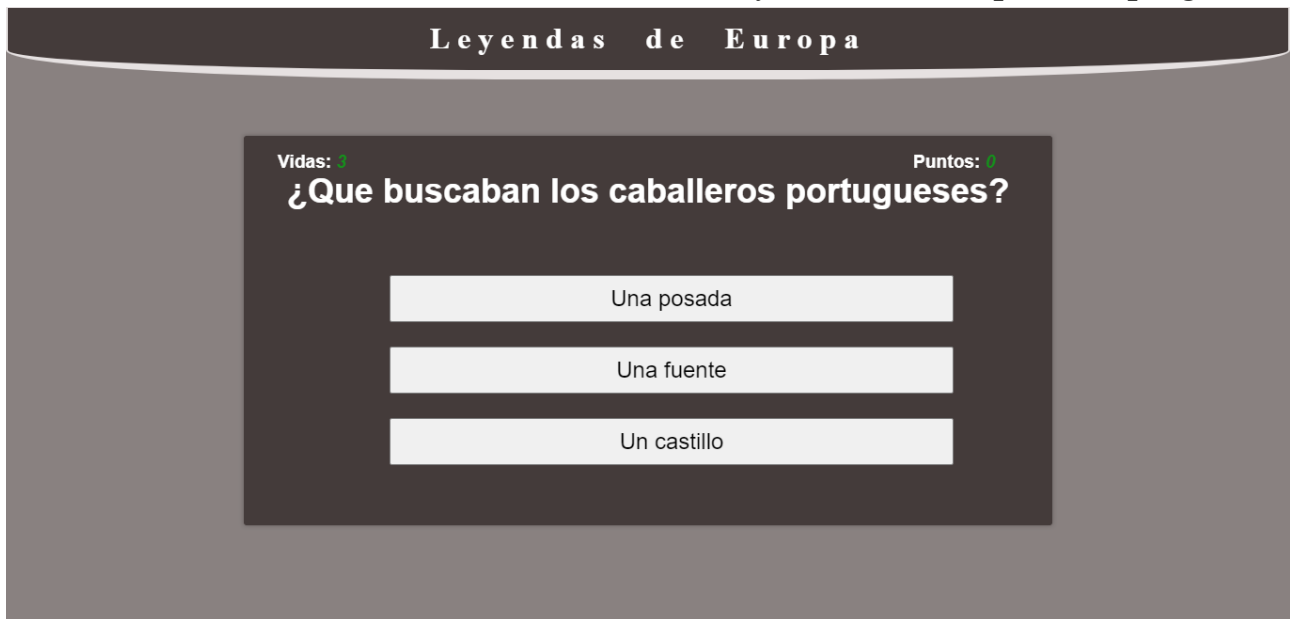


Si hacemos clic en “Mejores Puntuaciones” nos sacara un listado de los 6 usuarios con mayor puntuación registrados.



[Volver al índice](#)

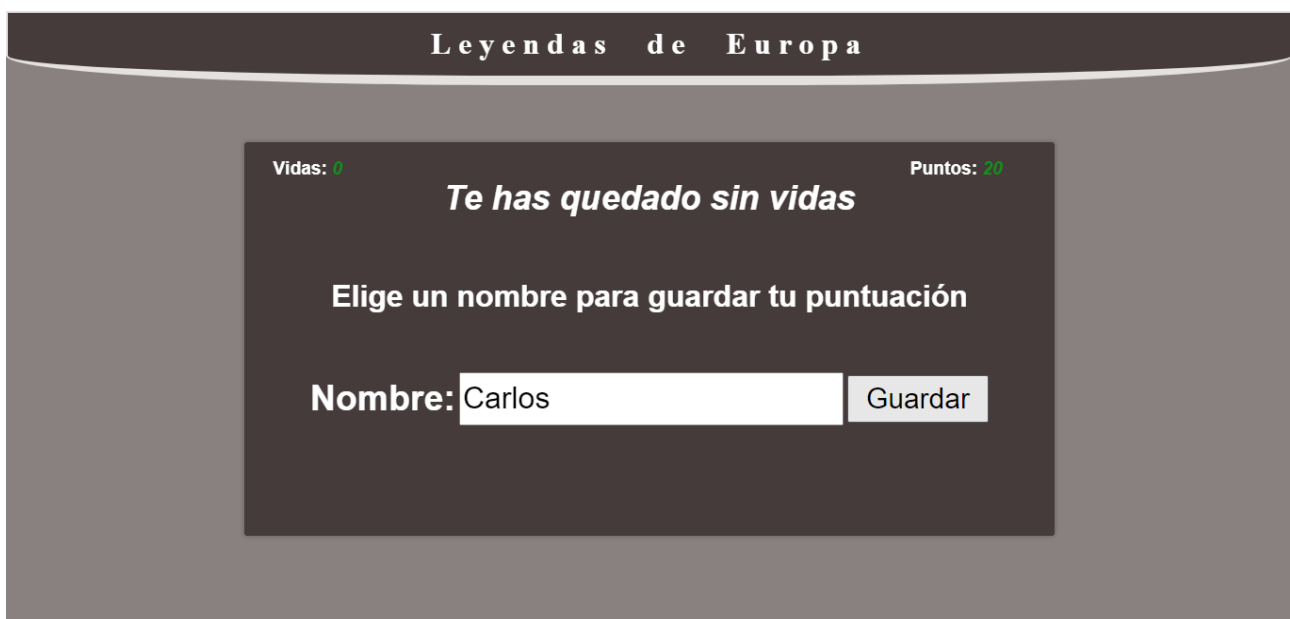
Si hacemos clic en “Iniciar Quiz” este iniciara y nos saldrá la primera pregunta.



The screenshot shows a web interface titled "Leyendas de Europa". At the top, it displays "Vidas: 3" and "Puntos: 0". The main question is "¿Que buscaban los caballeros portugueses?". Below the question are three answer options in white boxes: "Una posada", "Una fuente", and "Un castillo".

Al seleccionar una respuesta nos sumara un punto si es correcta y nos restara una vida si es incorrecta y luego mostrara la siguiente pregunta, esto se repetirá hasta que nos quedemos sin vidas o contestemos todas las preguntas (40 en total).

Cuando una de esas dos condicione se cumpla nos pedirá un nombre para guardar nuestra puntuación.



The screenshot shows the end screen of the "Leyendas de Europa" quiz. It displays "Vidas: 0" and "Puntos: 20". The main message is "Te has quedado sin vidas". Below this, it says "Elige un nombre para guardar tu puntuación". There is a text input field labeled "Nombre:" containing the text "Carlos", and a "Guardar" button next to it.

[Volver al índice](#)

Por ultimo podemos volver a la pagina principal haciendo clic en la cabecera o volver al inicio de esta pagina dándole a “Reintentar”.



Funcionamiento: [Anexo3](#)

[Volver al índice](#)

Pagina Administración



En esta pagina tenemos cuatro opciones:

1. Añadir Leyenda

Esta nos mostrara un formulario donde podremos rellenar los campos necesarios para añadir una nueva leyendas a la base de datos.



una vez todos los campos estén rellenos le damos a “Guardar” y nos mostrara un mensaje de que la leyenda se a añadido.

[Volver al índice](#)

Desde aquí podremos volver a la pantalla inicial de administración o ir nuevamente al formulario anterior para añadir una nueva leyenda



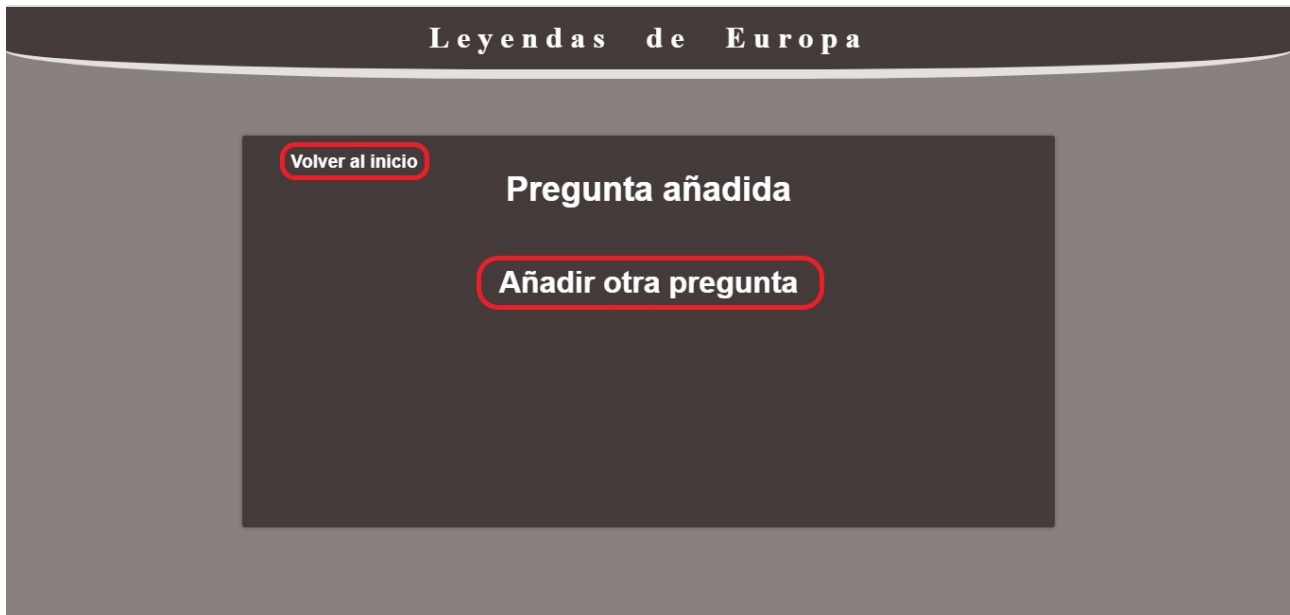
2. Añadir Pregunta

Al igual que en “Añadir leyenda”, aquí nos mostrara un formulario que debemos rellenar con el enunciado de la pregunta y las posibles respuestas.



[Volver al índice](#)

Nuevamente al darle a “Guardar” son mostrara un mensaje de que la pregunta a sido añadida y podremos volver al inicio o ir a añadir una nueva pregunta



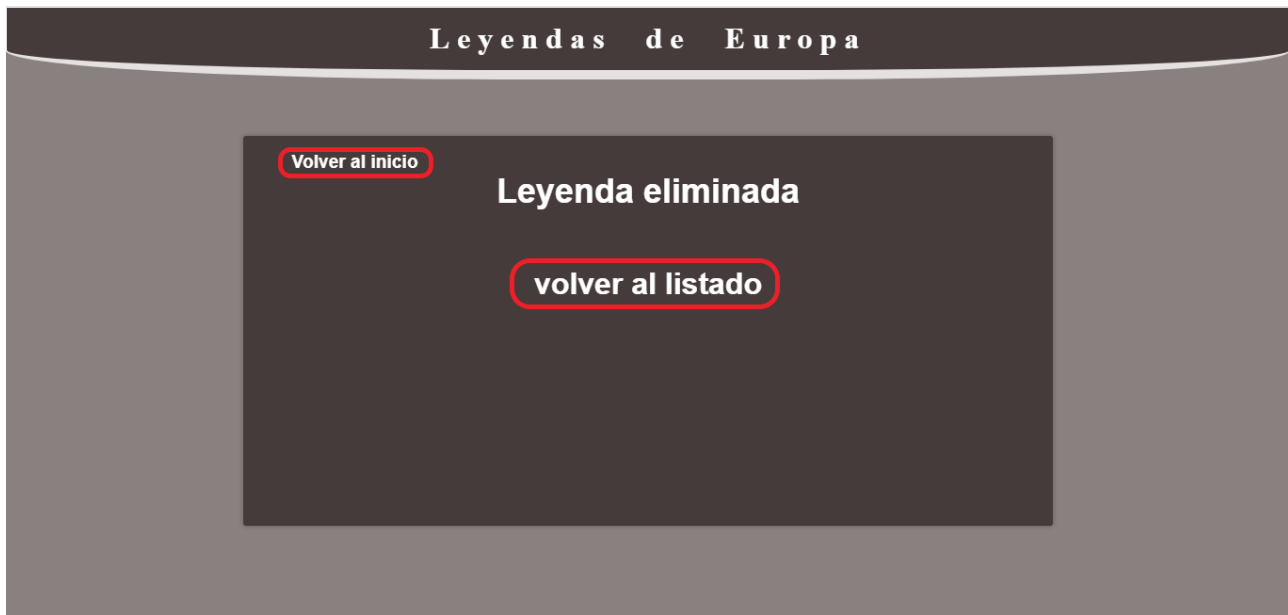
3. Eliminar Leyenda

En esta pagina veremos un listado de todas las leyenda



junto a cada enunciado esta la opción “borrar”, si hacemos clic en ella eliminara la pregunta de su fila y nos mostrara un mensaje para que sepamos que la entrada a sido eliminada de la base de datos.

[Volver al índice](#)

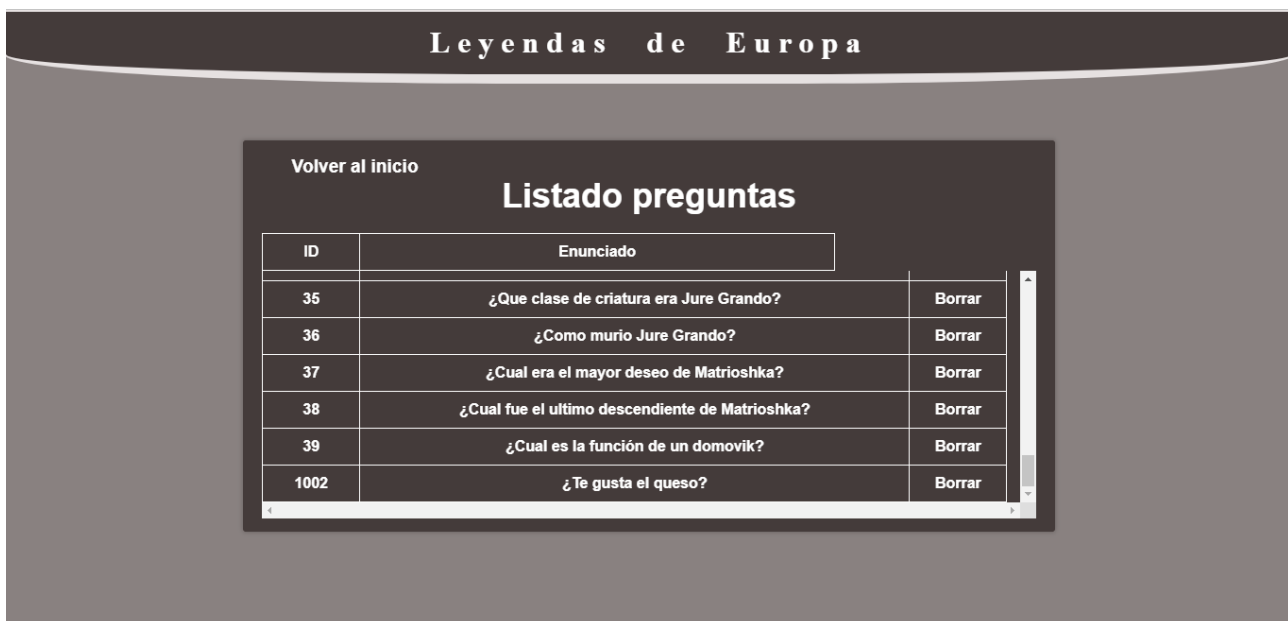


Desde esta pagina podremos volver al listado para eliminar mas leyendas o ir al inicio de la pagina de administración.

4. Eliminar Pregunta

Exactamente igual que eliminar leyendas, salvo que esta vez en este caso nos mostrara los enunciados de las preguntas.

Funcionamiento: **Anexo 4**



[Volver al índice](#)

Conclusiones

Lo primero dejar claro que muchos aspectos de la pagina son mejorables, por ejemplo, el quiz no tiene animaciones al acertar o fallar, y las imágenes de las leyendas se ven pequeñas en pantallas con buena resolución.

Por otro lado, la pagina tiene cosas que me parecen muy buenas, como poder ver las leyendas en una misma pagina o que las preguntas del quiz se actualices usando AJAX.

Mi conclusión final es que han sido completados todos los objetivos que me impuse para este proyecto.

[Volver al índice](#)

Integración de los módulos

Diseño de interfaces WEB

- El HTML utilizado:
 - Si bien es cierto que el html de esta pagina es bastante básico cabe recalcar el uso del atributo “hidden” que a sido de gran ayuda.
- El CSS utilizado:
 - El diseño de la pagina y la posición de los distintos elemento, así como también las animación de entrada de las leyendas.

Desarrollo WEB en entorno cliente

- El JavaScript utilizado:
 - Como ya comente en los objetivos esta pagina tiene una gran cantidad de JavaScript que utiliza para cosas muy variadas
- La tecnología AJAX:
 - El poder realizar varias peticiones PHP sin necesidad de cambiar de pagina a sido muy relevante en este proyecto

Desarrollo WEB en entorno servidor

- El PHP utilizado:
 - Esta pagina usa tanto PHP convencional como PHP en segundo plano(con AJAX), también se utiliza en las peticiones a la base de datos permitiendo gestionar esta desde la propia pagina.

[Volver al índice](#)

Bibliografía

[Mapa de Europa](#)

[Tutorial mapa interactivo](#)

[Colores hexadecimal](#)

[Tutorial AJAX](#)

Leyendas

[Fonte da Moura](#)

[Los amantes de Teruel](#)

[El hombre de la máscara de hierro](#)

[Rómulo y Remo \(La fundación de Roma\)](#)

[El puente del diablo](#)

[Los cisnes de Brujas](#)

[El holandés errante](#)

[La leyenda de Melusina](#)

[Jack-o'-lantern](#)

[Excalibur, la espada mágica](#)

[Yule Lads](#)

[Lorelei](#)

[Volver al índice](#)

[Beowulf](#)

[Krampus](#)

[Heykal](#)

[Los Caballeros de Sitno](#)

[Rey Matjaž](#)

[El primer vampiro europeo, Jure Grando](#)

[Matrioshka](#)

[Domovik](#)

[Volver al índice](#)

Anexos

Anexo1

Script de la creación de la base de datos

```
--
-- Base de datos: `leyendas_europa`
--
-----
--
-- Estructura de tabla para la tabla `leyenda`
--
CREATE TABLE `leyenda` (
  `id_leyenda` int NOT NULL,
  `siglas_pais` varchar(3) NOT NULL,
  `titulo_leyenda` varchar(50) NOT NULL,
  `contenido` text,
  `imagen` varchar(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
-----
--
-- Estructura de tabla para la tabla `pais`
--
CREATE TABLE `pais` (
  `id_pais` int NOT NULL,
  `nombre_pais` varchar(30) NOT NULL,
  `siglas_pais` varchar(3) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
-----
--
-- Estructura de tabla para la tabla `pregunta`
--
CREATE TABLE `pregunta` (
  `id_pregunta` int NOT NULL,
```

[Volver al índice](#)

```
`enunciado` text CHARACTER SET utf8mb4 COLLATE
utf8mb4_0900_ai_ci NOT NULL,
`respuesta1` varchar(50) CHARACTER SET utf8mb4 COLLATE
utf8mb4_0900_ai_ci NOT NULL,
`respuesta2` varchar(50) CHARACTER SET utf8mb4 COLLATE
utf8mb4_0900_ai_ci NOT NULL,
`respuesta3` varchar(50) CHARACTER SET utf8mb4 COLLATE
utf8mb4_0900_ai_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

```
--
-- Estructura de tabla para la tabla `usuario`
--
```

```
CREATE TABLE `usuario` (
  `id_usuario` int NOT NULL,
  `nombre_usuario` varchar(30) NOT NULL,
  `score` int NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

```
--
-- Índices para tablas volcadas
--
```

```
--
-- Indices de la tabla `leyenda`
--
```

```
ALTER TABLE `leyenda`
  ADD PRIMARY KEY (`id_leyenda`),
  ADD UNIQUE KEY `siglas_pais` (`siglas_pais`),
  ADD KEY `siglas_pais_leyendas` (`siglas_pais`);
```

```
--
-- Indices de la tabla `pais`
--
```

[Volver al índice](#)

```
ALTER TABLE `pais`
  ADD PRIMARY KEY (`id_pais`),
  ADD UNIQUE KEY `nombre_pais` (`nombre_pais`),
  ADD UNIQUE KEY `siglas_pais` (`siglas_pais`);

--
-- Indices de la tabla `pregunta`
--
ALTER TABLE `pregunta`
  ADD PRIMARY KEY (`id_pregunta`);

--
-- Indices de la tabla `usuario`
--
ALTER TABLE `usuario`
  ADD PRIMARY KEY (`id_usuario`),
  ADD UNIQUE KEY `nameUsu` (`nombre_usuario`);

--
-- AUTO_INCREMENT de las tablas volcadas
--

--
-- AUTO_INCREMENT de la tabla `usuario`
--
ALTER TABLE `usuario`
  MODIFY `id_usuario` int NOT NULL AUTO_INCREMENT;

--
-- Restricciones para tablas volcadas
--

--
-- Filtros para la tabla `leyenda`
--
ALTER TABLE `leyenda`
  ADD CONSTRAINT `leyenda_ibfk_1` FOREIGN KEY (`siglas_pais`)
```

[Volver al índice](#)

```
REFERENCES `pais` (`siglas_pais`);
COMMIT;
```

Anexo2

Mostrar leyendas

Para ello e utilizado el siguiente código:

Pagina principal

```
<!-- Ventana Leyenda-->
<div class="contenedor">
  <div class="overlay" id="overlay">
    <div class="popup" id="popup">
      <a href="#" id="btn-cerrar-popup" class="btn-cerrar-popup">X</a>
      <h3 class="popup-content-title"></h3>
      <div class="popup-content-text"></div>
      <div class="popup-content-img"></div>
    </div>
  </div>
</div>
```

CSS

```
.overlay {
  /*oscurece el fondo */
  background: rgba(0, 0, 0, 0.397);
  /* declara la posicion de la ventana */
  position: fixed;
  top: 0;
  bottom: 0;
  left: 0;
  right: 0;
  /* la alineacion del contenido */
```


[Volver al índice](#)

```
align-items: center;
justify-content: center;
display: flex;
/* la oculta */
visibility: hidden;
}
```

```
/* muestra la ventana */
.overlay.active {
  visibility: visible;
}
```

```
.popup {
  /* le da color y formato */
  background: #F8F8F8;
  box-shadow: 0px 0px 5px 0px rgba(0,0,0,0.3);
  border-radius: 3px;
  font-family: 'Montserrat', sans-serif;
  padding: 20px;
  text-align: center;
  /* declara el tamaño de la ventana */
  width: 40%;
  height: 90%;
  /* Animacion de la ventana */
  transition: .3s ease all;
  transform: scale(0.7);
  opacity: 0;
}
```

```
/* formato y posición para el botón que cierra la ventana */
.popup .btn-cerrar-popup {
  font-size: 16px;
  line-height: 16px;
  position: relative;
  top: 0;
  left: 50%;
  transition: .3s ease all;
```

[Volver al índice](#)

```
    color: #BBBBBB;
}

.popup .btn-cerrar-popup:hover {
    color: #000;
}
```

```
/* formato para el titulo de la leyenda */
.popup h3 {
    font-size: 36px;
    font-weight: 600;
    position: relative;
    opacity: 0;
}
```

```
/* formato para el texto de la leyenda */
.popup-content-text {
    font-size: 15px;
    font-weight: 300;
    margin-bottom: 20px;
    position: relative;
    opacity: 0;
/* tamaño que ocupa */
    height: 35%;
/* en caso de necesitar mas espacio le pone un scroll */
    overflow: auto;
}
```

```
/* declara el tamaño para las imágenes */
img{
    width: auto;
    position: relative;
    max-height: 150px;
    bottom: 10px;
```

[Volver al índice](#)

```
}
```

JavaScript

```
/* Rellena el pop-up con la información de la base de datos */
function fillLeyen(leyendaTitleNew, leyendaTextNew, leyendaImgNew) {
    leyendaTitle.innerHTML = leyendaTitleNew;
    leyendaText.innerHTML = leyendaTextNew;
    leyendaImg.innerHTML = leyendaImgNew;
}

/* abre el pop-up y le da formato */
function abreLeyenda(e) {
    /* guarda el id del pais seleccionado */
    var idPais = e.target.getAttribute('id');
    /* crea la peticion */
    var xhr = new XMLHttpRequest();
    /* crea variables para guardar los resultados */
    var newTitle = "";
    let newText = "";
    let newImg = "";
                fillLeyen(newTitle, newText, newImg);
            }
            /* En caso de que no reciba informacion de la base de datos
            rellena el pop-up con placeholders */
            else {
                newTitle = "Proximamente";
                newText = "";
                newImg = "";
            }
        }
    }
}
```

[Volver al índice](#)

```

        fillLeyen(newTitle, newText, newImg);
    }
}
}
/* Manda la peticion al php con la informacion necesaria */
var url = "Pedir_leyendaDB.php?siglas_pais=" + idPais;
xhr.open("GET", url, true);
xhr.send();
/* Hace visible el pop-up */
overlay.classList.add('active');
popup.classList.add('active');
}
/* Añade el evento de abrir ventana a todos los paises del mapa */
arrayCountry.forEach(element => {
    element.addEventListener('click', abreLeyenda, false);
});
/* Añade el evento a la 'X' para cerrar el pop-up */
btnCerrarPopup.addEventListener('click', function (e) {
    e.preventDefault();
    overlay.classList.remove('active');
    popup.classList.remove('active');
});

```

PHP

```

<?php
header("Content-Type: application/json; charset=UTF-8");
//Conexion a la base de datos
include './conDB.php';
/* si la conexion falla muestra un error */
if ($conexion->connect_error){
    die("Error en la conexion: " + $conexion->connect_error);
}else{
    //Crea la consulta SQL
    $sql = "SELECT titulo_leyenda, contenido, imagen FROM leyenda WHERE
        siglas_pais = '$_GET['siglas_pais']'.";
    $resultado = $conexion->query($sql);
    if ($resultado = $conexion->query($sql)) {
        while ($row = $resultado->fetch_assoc()) {

```

[Volver al índice](#)

```

        $data1 = $row["titulo_leyenda"];
        $data2 = $row["contenido"];
        $data3 = $row["imagen"];
    }
}
// guarda el resultado en un objeto
$salida = array(
    "titulo_leyenda" => $data1,
    "contenido" => $data2,
    "imagen" => $data3,
);
//convierte el objeto en un JSON
echo json_encode($salida);
}
$conexion->close();
?>

```

Anexo3

Funcionamiento quiz

Pagina principal

En esta pagina están los siguientes elementos:

- El contenedor de la pregunta [*class="contenedor-preg"*]
- El contenedor de las respuestas [*class="contenedor-res"*], este contiene:
 - Los botones de las respuestas [*class="button-quiz-respuesta"*] ocultos.
 - El formulario de registro [*class="registro"*] oculto.
 - Una tabla con los seis usuarios con la mejor puntuación [*class="scoreTabla"*] oculto.
 - El botón de “Reintentar” [*class="button-reset"*] oculto.

JavaScript

```

/* Contador de vidas */
var newNumVidas = 3;
/* Contador de puntos */
var newNumPuntos = 0;
/* respuesta correcta para comparar */
var respuestaCorrecta;

```

[Volver al índice](#)

/ Controla si la respuesta seleccionada es la correcta y dependiendo de si lo es o no, suma un punto o resta una vida */*

```
function contVidas_Puntos(e){
    botonSelect = e.target.innerHTML.toString();
    let resCorrecta = respuestaCorrecta;
    console.log(botonSelect + " == " + resCorrecta);
    if(botonSelect == resCorrecta){
        newNumPuntos++;
        numPuntos.innerHTML = newNumPuntos;
    }
    else if(preg_mostradas.length >= 1){
        newNumVidas--;
        numVidas.innerHTML = newNumVidas;
    }
}
```

/ Rellena la pregunta y los botón de la respuesta con el contenido de la base de datos */*

```
function fillQuiz(newQuizPregunta, newQuizRespuesta1, newQuizRespuesta2,
newQuizRespuesta3) {
    quizPregunta.innerHTML = newQuizPregunta;
```

/ para randomizar las posición de las respuesta, genera un numero entre 0 y 2 y dependiendo de este numero cambia la posición de las respuestas */*

```
var rand = Math.floor(Math.random() * 3);
respuestaCorrecta = newQuizRespuesta1;
if(rand == 0){
    arrayRes[0].innerHTML = newQuizRespuesta1;
    arrayRes[1].innerHTML = newQuizRespuesta2;
    arrayRes[2].innerHTML = newQuizRespuesta3;
}
else if(rand == 1){
    arrayRes[0].innerHTML = newQuizRespuesta3;
    arrayRes[1].innerHTML = newQuizRespuesta2;
    arrayRes[2].innerHTML = newQuizRespuesta1;
}
else if(rand == 2){
```

[Volver al índice](#)

```

        arrayRes[0].innerHTML = newQuizRespuesta3;
        arrayRes[1].innerHTML = newQuizRespuesta1;
        arrayRes[2].innerHTML = newQuizRespuesta2;
    }
}

/* numero de la proxima pregunta */
var num_pregunta;
/* Array con los numeros de las preguntas ya mostradas */
var preg_mostradas = new Array();
/* Controlador del inicio del quiz */
var inicio = true;

/* Actualiza las preguntas */
function refreshQuiz(e) {
    /* Al inicio esconde todos lo que haya en el contenedor de la respuesta y muestra los botones*/
    if(inicio == true){
        arrayRes.forEach(element => {
            element.removeAttribute("hidden");
        });
        botonQuiz.hidden = true;
        botonScore.hidden = true;
        score.hidden = true;
        inicio = false;
    }

    //Cantidad de preguntas en la base de datos
    var max_preg = 40;
    //Selecciona un numero de pregunta al azar
    num_pregunta = Math.floor(Math.random() * max_preg);

    //Si el array esta vacio simplemente añade el numero de la pregunta a este
    if(preg_mostradas.length == 0){
        preg_mostradas.push(num_pregunta);
    }
    /*Si el array ya contiene algo comprueba que el nuevo numero de la pregunta no haya salido ya, de esta forma las preguntas no se repiten */
    else {
        //Si ya a salido el numero lo vuelve a generar y lo comprueba de nuevo

```

[Volver al índice](#)

```

while(preg_mostradas.includes(num_pregunta)){
    num_pregunta = Math.floor(Math.random() * 40);
}
preg_mostradas.push(num_pregunta);
}
//Texto de boton seleccionado
var botonSelect =e.target.innerHTML.toString();
/* Comprueba que no nos hemos quedado sin vidas y que no se han acabado las
pregunto */
if(preg_mostradas.length < max_preg-1 && newNumVidas > 0){
/* crea la peticion para el PHP */
var xhr = new XMLHttpRequest();
/* variables para guardar la respuesta del PHP */
var newQuizPregunta = "";
var newQuizRespuesta1 = "";
var newQuizRespuesta2 = "";
var newQuizRespuesta3 = "";
xhr.onreadystatechange = function () {
/* comprueba que el servidor este listo para recibir la peticion */
if (this.readyState == 4 && this.status == 200) {
/* Guarda la respuesta en un objeto */
var array = JSON.parse(this.responseText);
/* Guarda el contenido del objeto en un array */
var resArray = [array.respuesta1,array.respuesta2,array.respuesta3];
/* Comprueba si la respuesta del PHP no esta vacia */
if(array.enunciado != null){
/* Rellena con la informacion de la base de datos */
newQuizPregunta = array.enunciado;
newQuizRespuesta1 = array.respuesta1;
newQuizRespuesta2 = array.respuesta2;
newQuizRespuesta3 = array.respuesta3;
fillQuiz(newQuizPregunta, newQuizRespuesta1, newQuizRespuesta2,
newQuizRespuesta3, botonSelect);
}

/* En caso de que no reciba informacion de la base de datos mostrará un
error */
else {
newQuizPregunta = "Error";

```


[Volver al índice](#)

```

        fillQuiz(newQuizPregunta, newQuizRespuesta1, newQuizRespuesta2,
            newQuizRespuesta3, botonSelect);
    }
}

```

/ Manda la peticion al php con la informacion necesaria */*

```

var url = "Pedir_PreguntaDB.php?id_pregunta=" + num_pregunta;
xhr.open("GET", url, true);
xhr.send();
}

```

/ En caso de quedarnos sin vidas muestra un mensaje y el formulario para guarda la puntuacion */*

```

else if(newNumVidas <= 0){
    quizPregunta.innerHTML = "<i>Te has quedado sin vidas</i>";
    arrayRes.forEach(element => {
        element.hidden = true;
    });
    registro.hidden = false;
}

```

/ En caso de haber completado el quiz sin quedarse sin vidas muestra un mensaje y el formulario para guarda la puntuacion */*

```

else if(preg_mostradas.length <= 40){
    quizPregunta.innerHTML = "<i>¡Increible!, has contestado a todas las preguntas</i>";
    arrayRes.forEach(element => {
        element.hidden = true;
    });
    registro.hidden = false;
}
}

```

/ Toma el contenido del formulario de registro y los puntos y hace una peticion al PHP */*

```

function mandarUsuario(){
    var xhr = new XMLHttpRequest();
    var newNomUsu = nomUsu.value;
    // Manda la petición con los datos necesarios

```

[Volver al índice](#)

```

    var url = "Score_usu.php?nomUsu=" + newNomUsu + "&score=" +
newNumPuntos;
    xhr.open("GET", url, true);
    xhr.send();
    /* Una vez mandada la petición esconde el formulario de registro y muestra el
       boton "Reintentar" */
    registro.hidden = true;
    reinicio.hidden = false;
    quizPregunta.innerHTML = "¿Quieres volver a intentarlo?";
}

```

```

//Añade el evento al boton inicial
botonQuiz.addEventListener("click", refreshQuiz, false);

```

```

//Añade los eventos a los botones de las respuestas
var arrayRes = document.querySelectorAll('.button-quiz-respuesta');
arrayRes.forEach(element => {
    element.addEventListener('click', contVidas_Puntos, false);
    element.addEventListener('click', refreshQuiz, false);
});

```

```

//Añade el evento al boton de resgistro de usuario
botonRegistro.addEventListener('click', mandarUsuario, false);
PHP (Pedir_pregunta)

```

```

<?php
    header("Content-Type: application/json; chaset=UTF-8");
    //conexion a la base de datos
    include './conDB.php';

```

```

//Muestra un error si no se a podido conectar a la base de datos

```

```

if ($conexion ->connect_error){
    die("Error en la conexion: "+$conexion->connect_error);
}else{

```

```

    //Conexión correcta

```

```

    //Creamos la consulta SQL

```

```

    $sql = "SELECT enunciado,respuesta1,respuesta2,respuesta3 FROM pregunta
    WHERE id_pregunta = '$_GET[id_pregunta]'.";

```

```

    //Guardamos el resutado en variables

```

[Volver al índice](#)

```

$resultado = $conexion ->query($sql);
if ($resultado = $conexion->query($sql)) {
    while ($row = $resultado->fetch_assoc()) {
        $data1 = $row["enunciado"];
        $data2 = $row["respuesta1"];
        $data3 = $row["respuesta2"];
        $data4 = $row["respuesta3"];
    }
}
//Guardamos todo en un objeto
$salida = array(
    "enunciado" => $data1,
    "respuesta1" => $data2,
    "respuesta2" => $data3,
    "respuesta3" => $data4,
);
//Pasamos el objeto a JSON y lo mandamos al JavaScript
echo json_encode($salida);
}
//Cerramos la conexión a la base de datos
$conexion->close();
?>

```

PHP (Score_usu)

```

<?php
//conexion a la base de datos
include './conDB.php';

if ($conexion ->connect_error){
    die("Error en la conexion: " . $conexion->connect_error);
}else{
    //Conexión correcta
    //Creamos la consulta SQL
    $sql = "INSERT INTO usuario(`nombre_usuario`, `score`) VALUES ('" .
        $_GET['nomUsu']. "', '" . $_GET['score']. "')";
    //Ejecutamos la consulta
    $resultado = $conexion ->query($sql);
}

```

[Volver al índice](#)

```
//Cerramos la conexion a la base de datos
$conexion->close();
?>
```

Anexo4

Funcionamiento admin

Añadir

Formulario html

```
<form method='POST' enctype="multipart/formdata" action='add_Pregunta2.php'>
    <p>Enunciado: <input type='text' name='enunciado'></p>
    <p>Respuesta1: <input type='text' name='respuesta1'></p>
    <p>Respuesta2: <input type='text' name='respuesta2'></p>
    <p>Respuesta3: <input type='text' name='respuesta3'></p>
    <p><input type='submit' value='Guardar' name='comprobarinsertar'></p>
</form>
```

Ejecución

```
<?php
/* Inicia la conexion con la base de datos */
include "./conDB.php";
/*Guarda los datos recibidos del formulario en variables */
$enunciado = $_POST["enunciado"];
$respuesta1 = $_POST["respuesta1"];
$respuesta2 = $_POST["respuesta2"];
$respuesta3 = $_POST["respuesta3"];
//Crea la sentencia sql
$sql = "INSERT INTO pregunta(`enunciado`, `respuesta1`, `respuesta2`, `respuesta3`)
VALUES ('".$enunciado."', '".$respuesta1."', '".$respuesta2."', '".$respuesta3."')";
//Ejecuta la sentencia
$resultado = $conexion -> query($sql);
//Cierra la conexión
$conexion->close();
?>
```

Eliminar

Listado

```
<?php
include("conDB.php");
if ($conexion -> connect_error){
    die("Error en la conexion: "+$conexion->connect_error);
}else{
    //Conexión correcta
```

[Volver al índice](#)

```

$sql = "SELECT id_pregunta,enunciado FROM pregunta";
$resultado = $conexion ->query($sql);
if ($resultado = $conexion->query($sql)) {
    /* Cada pasada del bucle crea una nueva fila hasta que no
    queden resultados */
    while ($row = $resultado->fetch_assoc()) {
        echo '<tr>';
        echo '<td>'.$row["id_pregunta"].'</td>';
        echo '<td>'.$row["enunciado"].'</td>';
        * En cada pasada genera el "Borrar" unico de esa
        columna */
        echo "<td><a href=borrar_Pregunta2.php?id_pregunta=".
            $row['id_pregunta']. ">Borrar<a></td>";
        echo '</tr>';
    }
}
echo "</table>";
}
mysqli_close($conexion);
?>

```

Ejecución

```

<?php
/* Inicia la conexion con la base de datos */
include './conDB.php';
//Crea la sentencia sql
$sql = "DELETE FROM pregunta WHERE id_pregunta = '".$_GET['id_pregunta']."'";
echo $sql;
//Ejecuta la sentencia
$resultado = $conexion ->query($sql);
//Cierra la conexion
$conexion->close();
?>

```

Para añadir y eliminar leyendas es el mismo concepto simplemente cambiando los nombres de los campos.