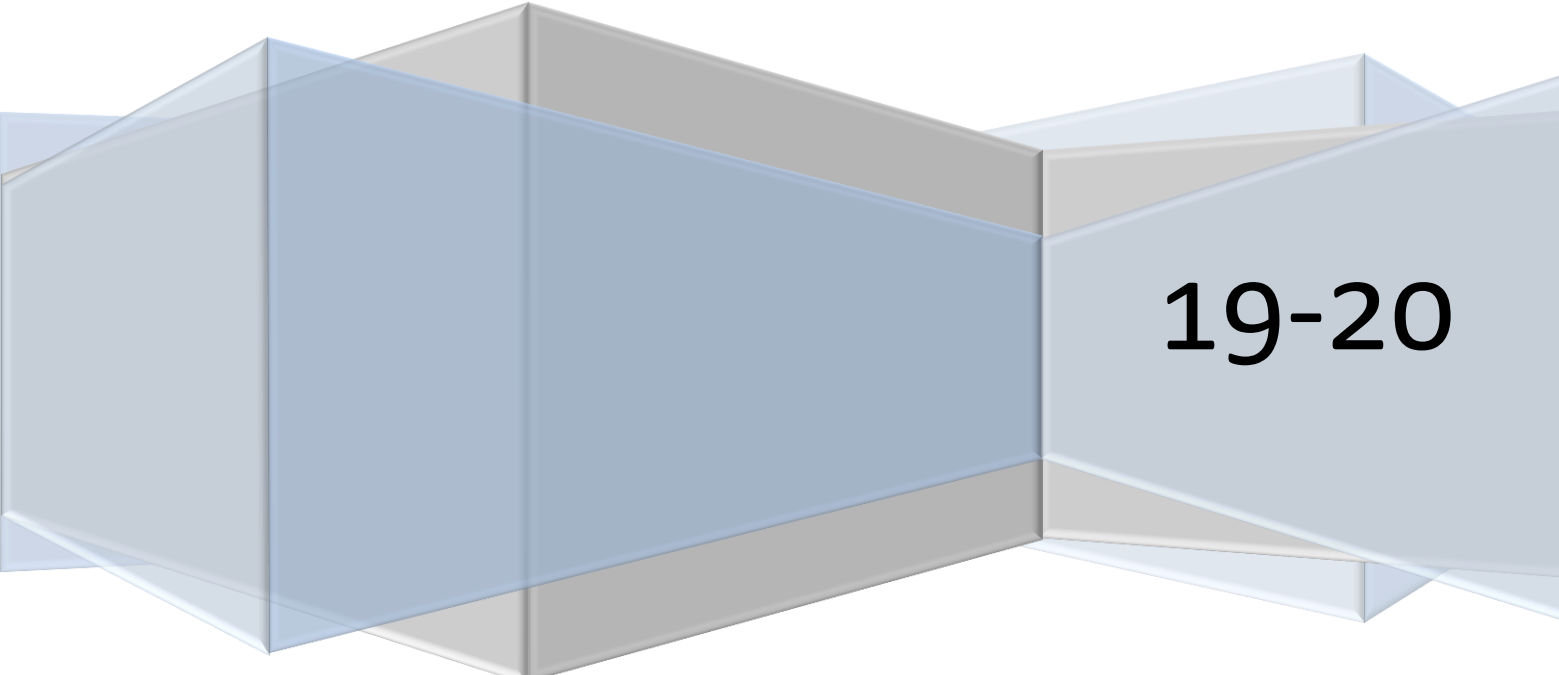


IES Gregorio Prieto

Proyecto de Desarrollo de Aplicaciones Multiplataforma

Desarrollo de Aplicaciones Multiplataforma

Harold Alcalde Solarte

A 3D graphic consisting of several overlapping, semi-transparent blue and grey polyhedral shapes, resembling a modern architectural structure or a stylized book cover. The shapes are arranged in a way that creates a sense of depth and perspective.

19-20

1 Contenido

2	Introducción.....	1
2.1	Descripción, objetivos y finalidad del proyecto	1
2.2	Requisitos.....	1
2.2.1	Requisitos funcionales	1
2.2.2	Requisitos no funcionales.....	1
2.3	Restricciones	1
3	Análisis del entorno.....	2
3.1	Descripción de las oportunidades de negocio.	2
3.2	Prevención de riesgos laborales.....	3
4	Análisis del sistema actual.....	4
5	Solución propuesta	4
5.1	Tecnologías existentes para el desarrollo y puesta a punto de la aplicación.	4
5.2	Recursos humanos y materiales necesarios para el desarrollo del proyecto	5
6	Planificación temporal para el desarrollo del proyecto	5
6.1	Adquisición del Software y del Hardware.....	5
6.2	Desarrollo de la aplicación	6
6.3	Fase de pruebas.....	6
6.4	Lista de modificaciones o ampliaciones futuras que se podrían aplicar a la aplicación proyectada.....	10
6.5	Plan de mantenimiento de la aplicación.....	10
7	Estudio de la viabilidad del proyecto	11
8	Documentación del diseño e implementación de la solución adoptada	11
8.1	Prototipo de la aplicación. Diseño de interfaces.....	11
8.2	Diagrama de flujo entre escenas.....	17
8.3	Diseño de la base de datos. Diagrama Entidades/Relación	17
8.4	Scripts PHP	17
8.5	Scripts de C#.....	19
8.6	Aplicación en CodeIgniter con librería Grocery CRUD.....	20
8.7	Funcionamiento de la aplicación.....	22
9	Puesta a punto en Hostinger	25
10	Resultado Final.....	26
11	Bibliografía y fuentes de información	27

2 Introducción

2.1 Descripción, objetivos y finalidad del proyecto

Este proyecto llamado HaroldEduca es un juego de preguntas y respuestas online enfocado en los alumnos de ciclos formativos del área de informática. La idea es que los alumnos aprendan y pongan a prueba sus conocimientos de una forma divertida y relajada desde cualquier lugar y en cualquier momento.

Para la primera versión del juego el usuario podrá crear una cuenta en el juego, poder jugar en las diferentes categorías que existen y ver sus puntuaciones a modo de ranking.

El objetivo principal de este proyecto es crear un juego en Unity para dispositivos móviles Android, que permita a los usuarios aprender de una manera simple y relajada a través de su dispositivo móvil.

2.2 Requisitos

2.2.1 Requisitos funcionales

1. La aplicación dispone de una base de datos en MySQL donde se tendrán almacenados los registros de las siguientes tablas: Preguntas, Respuestas, Usuarios, Partidas, Categorías.
2. Se permitirá al usuario crear una cuenta e iniciar sesión con ella.
3. Habrá un administrador del sistema que se encarga de las operaciones CRUD a la base de datos a través de CodeIgniter CRUD.
4. Se permitirá al usuario elegir la categoría en la que desea jugar.
5. Cada partida constará de un número concreto de preguntas.
6. Al final de cada partida se mostrará un resumen del número de aciertos, fallos y puntuación obtenida.

2.2.2 Requisitos no funcionales

1. La aplicación será amigable e intuitiva para el usuario.
2. La aplicación deberá tener una interfaz clara y libre de información no relevante.
3. La aplicación deberá tener accesos rápidos y fáciles de recordar a las acciones más habituales del usuario.
4. La aplicación deberá funcionar en cualquier dispositivo móvil Android.
5. La aplicación debe permitir que se puedan añadir, modificar, actualizar o eliminar más preguntas, categorías, respuestas.
6. Las contraseñas de los usuarios almacenadas en la base de datos no se mostrarán en texto plano.

2.3 Restricciones

1. El nombre de usuario y su contraseña deben ser mínimo de 4 caracteres.
2. No pueden existir dos usuarios con el mismo nombre.

3. Se mostraran solo categorías que tengas categorías hijas o que tengan un mínimo de preguntas para poder comenzar una partida.
4. Se seleccionaran solo preguntas que tengan cuatro respuestas.

3 Análisis del entorno

3.1 Descripción de las oportunidades de negocio.

Actualmente hay muchos juegos de este tipo, pero la diferencia que puede haber entre HaroldEduca y los demás es que se centra en un determinado público, en este caso a los alumnos de los ciclos formativos de grado superior de informática. A través de la interfaz creada, para la inserción, modificación y borrado de las distintas tablas con CodeIgniter CRUD, será sencillo introducir datos por el administrador que en este caso será el profesor del módulo. Así mismo tanto el profesor como los propios alumnos pueden ayudar a alimentar la base de datos con sus preguntas.

Es un producto dedicado a un grupo pequeño de alumnos pero fácil de adaptar a otros grupos, como por ejemplo a los diferentes cursos de la ESO o el bachillerato, ya que se puede tener un administrador añadiendo las preguntas y respuestas que han ido creando los alumnos.

Por lo que el producto tendría muchas oportunidades de negocios en institutos, colegios, clases particulares, etc.

Ejemplo de inserción de datos con CodeIgniter: En este caso de la tabla categorías

[Categoría](#) [Pregunta](#) [Respuesta](#)

Add Categoría		Export Print	
Show 25 entries	Search:		
Categoría	Categoría Padre	Actions	
1º ASIR	ASIR	View	Clone Edit Delete
1º DAM	DAM	View	Clone Edit Delete
1º DAW	DAW	View	Clone Edit Delete
2º ASIR	ASIR	View	Clone Edit Delete
2º DAM	DAM	View	Clone Edit Delete
2º DAW	DAW	View	Clone Edit Delete
Acceso a datos	2º DAM	View	Clone Edit Delete

Add Categoría

Categoría* :

Categoría Padre :

Select Categoría Padre

1º ASIR

1º DAM

1º DAW

2º ASIR

2º DAM

2º DAW

Acceso a datos

Admin. de SO

ASIR

Reservado de datos

Save

Save and go back to

Add Categoría

Export

Print

Show 25 entries

Search:

Categoría	Categoría Padre	Actions			
Admin. de SO	2º ASIR	<div>View</div>	<div>Clone</div>	<div>Edit</div>	<div>Delete</div>
Seguridad y Alta disponibilidad	2º ASIR	<div>View</div>	<div>Clone</div>	<div>Edit</div>	<div>Delete</div>
Servicios red e Internet	2º ASIR	<div>View</div>	<div>Clone</div>	<div>Edit</div>	<div>Delete</div>
<div>Search Categoría</div>	<div>2º ASIR</div>	<div>Clear filtering</div>			

Displaying 1 to 3 of 3 items (filtered from 27 total entries)

First

Previous

1

Next

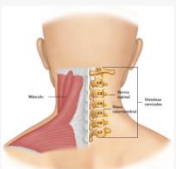

Last

3.2 Prevención de riesgos laborales

Problemas más comunes:



DOLOR CERVICAL

Asociado al mantenimiento de posturas con el cuello girado, demasiado flexionado o extendido. Su principal causa es una mala colocación del ordenador.



DOLOR DE ESPALDA

La postura sentada provoca esfuerzos entre las vértebras, los ligamentos y la musculatura de la espalda. Es recomendable disponer de una silla que proporcione un buen apoyo y favorezca los cambios de postura

MOLESTIAS EN LOS HOMBROS

Asociadas a un sobre esfuerzo de la musculatura escapular, producido por no apoyar los antebrazos al teclear o manejar el ratón, así como por una elevación excesiva de los hombros a causa de una mala regulación de la mesa.

FATIGA VISUAL

CAUSAS

Suele tener su origen en una mala iluminación (alumbrado deficiente y contrastes de luz inadecuados).


Si ante la presencia de síntomas, la persona sigue forzando la vista, los músculos del ojo se agotan y deriva en cefalea (dolor de cabeza).

SÍNTOMAS

- Vista cansada (Visión borrosa o doble)
- Hipersensibilidad a la luz
- Picores e irritación en conjuntiva y párpados
- Lagrimos
- Dolor de cabeza y mareos

¿COMO PODEMOS EVITARLA?

- Trabajar con una pantalla anti-reflejo y ajustar el brillo adecuado.
- Evitar el uso de pantallas con alto contraste.
- Evitar el uso de pantallas con alto contraste.
- Evitar el uso de pantallas con alto contraste.
- Hacer descansos y practicar ejercicios visuales. Se recomienda alejarse de la pantalla por espacio de quince segundos cada una o dos horas.
- Para relajar los músculos oculares, podemos fijar la vista en un objeto lo más distante posible por espacio de cinco o diez segundos.



Postura adecuada del trabajador:

- De frente a la pantalla, la línea de visión debe ser paralela al plano horizontal. Evite trabajar con la cabeza o el tronco girados.
- Trabajar con la espalda apoyada en el respaldo y en posición vertical.
- Los brazos deben trabajar en una postura relajada con los antebrazos y manos en una postura lo más paralela al suelo posible.
- Muslos en posición horizontal o ligeramente inclinados hacia abajo
- Parte inferior de las piernas prácticamente en ángulo recto respecto de los muslos.

- Los pies apoyados horizontalmente sobre el suelo.

4 Análisis del sistema actual

En este caso la aplicación se empieza desde cero, además no se integrará o se complementará con otras aplicaciones.

Funcionamiento: Nuestra aplicación de Unity se conectará a nuestra base de datos en MySQL a través de scripts PHP que son los encargados de realizar las consultas a la base de datos. Se accederá a la base de datos en:

- el inicio de sesión.
- En la creación de una nueva cuenta.
- Cuando se muestran las categorías del juego.
- Antes de empezar las partidas se cargan las preguntas y respuestas
- Al terminar la partida para guardar el resultado de la partida.
- Al mostrar el ranking con las puntuaciones.

En la partida se mostrará una pregunta y hasta que no se contesta no pasa a la siguiente, en la parte superior derecha se mostrará la puntuación que lleva en ese momento. Más adelante se explicará con detalle cada una de las pantallas y su funcionamiento.

5 Solución propuesta

5.1 Tecnologías existentes para el desarrollo y puesta a punto de la aplicación.

Para el desarrollo de esta aplicación contábamos con dos opciones para el entorno de desarrollo de la aplicación, la primera en Android Studio y la segunda Unity3D, a continuación veremos porque nos hemos decantado por Unity.

Se eligió **Unity** (Lenguaje de programación C#) porque los elementos gráficos son más vistosos y su entorno de desarrollo es más sencillo, intuitivo y fácil de usar, además al desarrollador del proyecto se le dio la libertad de elegir y decidió que el proyecto se debía desarrollar en Unity3D.

Lo mismo sucedió a la hora de elegir un **gestor de bases de datos**, hay muchos gestores de bases de datos en el mercado (PostgreSQL, Oracle, SQLite, etc.) Se descartó SQLite porque no es muy potente como PostgreSQL o MySQL, teniendo en cuenta la escalabilidad de la base de datos es mejor tener un gestor robusto y fiable.

Por otra parte tenemos el servidor que en este caso se usa **XAMPP**, que es un servidor independiente de plataforma de código libre. Te permite instalar de forma sencilla Apache para que tu ordenador se utilice como servidor, sin importar el sistema operativo del ordenador.

XAMPP también incluye servidores de bases de datos (MySQL Y SQLite) con sus gestores phpMyAdmin y phpSQLiteAdmin. En nuestro caso usamos XAMPP para tener nuestro servidor **MySQL** y para poder acceder a los **scripts PHP** que realizan las consultas a la base de datos.

Por último, tenemos **CodeIgniter** que es un framework para el desarrollo de aplicaciones en PHP que utiliza el MVC. Permite a los programadores Web mejorar la forma de trabajar y hacerlo a mayor velocidad, pero lo importante en este caso es la librería **Grocery CRUD** que nos permite crear en cuestión de minutos un CRUD para nuestra base de datos.

5.2 Recursos humanos y materiales necesarios para el desarrollo del proyecto

Vamos a listar las diferentes aplicaciones software que se han utilizado para el desarrollo del proyecto y el precio que tendrían si se fuera a comercializar el producto.

1. **Ordenador** de 8GB de RAM para el buen funcionamiento del entorno de desarrollo Unity. (800 €).
2. **Sistema Operativo:** Windows10(110€).
3. **Conexión a Internet** ya que el ordenador se usara como servidor del juego. (50€/mes).
4. **Unity:** Es un motor de videojuego multiplataforma está disponible como entorno de desarrollo para Windows, MacOS, Linux. Programado en C#.(Licencia plus: 35 \$/mes)
5. **XAMPP:** Es un servidor independiente de plataforma de código libre. Permite la instalación de forma sencilla de Apache y MySQL, convierte tu ordenador en un servidor.(Gratis)
6. **Visual Studio 2017:** Se usa Visual Studio por que se ha usado durante el curso y se tiene un manejo avanzado del entorno de desarrollo.(Gratis)
7. **CodeIgniter:** Framework para el desarrollo de aplicaciones en PHP que utiliza el MVC.(Gratis)
8. **Grocery CRUD:** Librería para la creación de una interfaz para un manejo fácil de la base de datos.(Gratis)
9. **Balsamiq mockups:** Herramienta de prototipado, utilizado para la propuesta del proyecto (2 proyectos -> 9\$/mes).
10. **SublimeText** como editor de texto para la creación de scripts en PHP.(Gratis).
11. **Tiempo** empleado por el desarrollador: 2 meses (700 €/mes).

En total es costo de la primera versión del proyecto durante los dos primeros meses es de **2503 €**.

6 Planificación temporal para el desarrollo del proyecto

6.1 Adquisición del Software y del Hardware

Lo único que debemos tener es un dispositivo móvil Android con Internet para poder jugar y descargarnos la aplicación desde los siguientes enlaces:

1. Google Drive: https://drive.google.com/file/d/1g19oXaiLjLy-jGkOc0HvOe_Agas1sRoY/view?usp=sharing
2. Mega: <https://mega.nz/#!wQVAASJS!yxvDhtiSQiawqCxmoLEqT0RD1hkTYh48icNVTjXlyxg>

6.2 Desarrollo de la aplicación

Una vez elegidas las tecnologías con las que se iba a trabajar se hizo una especie de metodologías ágil. La metodología SCRUM se centra en ajustar sus resultados y responder a las exigencias reales y exactas del cliente. De ahí, que se vaya revisando cada entregable, ya que los requerimientos van variando a corto plazo. El tiempo mínimo para un Sprint es de una semana y el máximo es de cuatro semanas.

Cada vez que se terminaba una tarea se revisaba y se hacían pruebas para comprobar el funcionamiento de dicha tarea.

En este caso, se hicieron Sprints o hitos de una semana, empezando por la creación de la base de datos a partir del modelo entidad/relación que es lo más importante.

La siguiente tarea fue la de generar una escena de Inicio de sesión o Registro para la aplicación, se crearon los scripts para comprobar credenciales o para la inserción de un nuevo usuario en la base de datos.

Continuamos con la escena de Categorías y su flujo que nos permite ver las categorías que puede elegir el usuario para poder empezar una partida. Se creó la interfaz gráfica y los scripts para la obtención de las categorías según su categoría padre.

Durante la siguiente semana se hicieron las comprobaciones de Internet, música ambiente del juego, la escena de configuración, configuración del inicio de sesión y otros cambios más pequeños.

En la siguiente semana se creó la escena partida que recibe una lista de preguntas con sus respuestas y las va mostrando según se van contestando, según si aciertas o fallas se muestra el color de acierto (verde) o error (rojo) y un sonido para cada uno. A continuación se pasó a crear la escena de resumen de partida, donde se muestran los fallos y aciertos del usuario, la puntuación final y se guarda en la base de datos el resumen de la partida.

En la última semana se hizo la inserción de categorías, preguntas y respuestas en la base de datos, pruebas de la aplicación, documentación, etc.


6.3 Fase de pruebas

Como se ha comentado en el apartado anterior las pruebas se fueron realizando cada vez que se acaba una tarea.


Validación de campos

Para el inicio de sesión o registro se comprueban la longitud de los campos y que ambos campos estén completos. En el inicio de sesión se comprueba que el usuario introducido exista en la base de datos y que las contraseñas coincidan. En el registro se comprueba que el usuario introducido no exista, en caso contrario se limpia ambos campos.


HaroldEduca




Completa los campos




HaroldEduca




Usuario no existe



HaroldEduca



Password Incorrecto



HaroldEduca



Username ya existe.Intentelo de nuevo



HaroldEduca



Completa el campo Password



Pruebas de categorías vacías:

Cuando se selecciona una categoría que no tiene hijas o que no tiene suficientes preguntas para empezar una partida



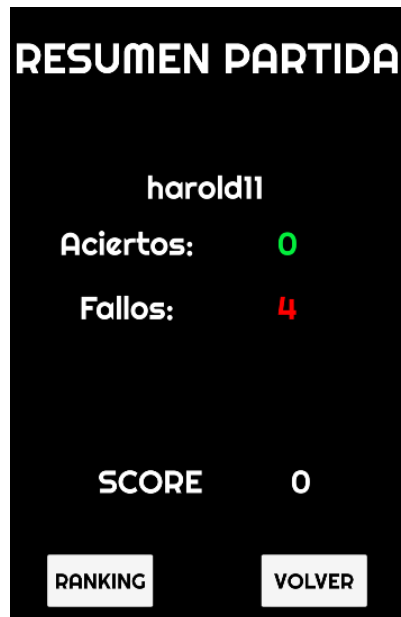
Prueba de acceso a la base de datos de varios dispositivos: Se prueba la aplicación a la misma vez desde varios dispositivos y se comprueba que funciona correctamente.

Prueba de seleccionar dos respuestas: Cuando se selecciona una respuesta inmediatamente salta el evento onclick() del botón y se activa una pantalla transparente para que no se pueda seleccionar otra mientras se carga la siguiente pregunta(2 segundos).

Prueba de inicio de sesión: Una vez se inicia sesión si se sale de la aplicación y se vuelve a entrar la sesión sigue activa, si se quiere cerrar sesión está el botón LOG OUT.



Pruebas de puntuación negativa: No se puede obtener puntuación negativa, ni se mostrará puntuación negativa. La puntuación más baja que se puede tener es cero.



Pruebas de conexión a Internet en el Login:

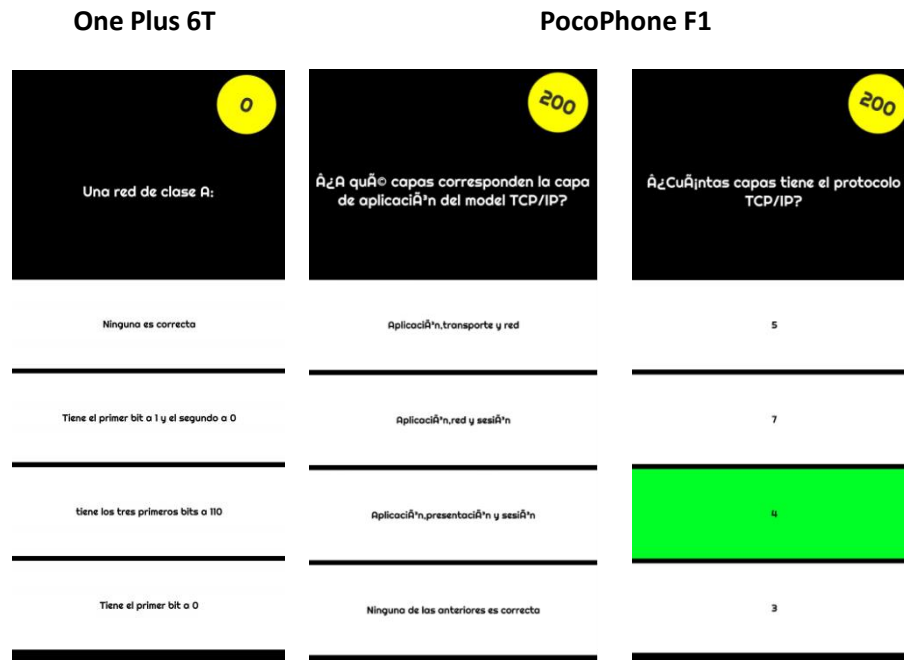
En el inicio de sesión se comprueba que el dispositivo tenga Internet si no es así se muestra el siguiente mensaje:



Pruebas de la aplicación en varios dispositivos móviles:

Se ha probado la aplicación en un dispositivo móvil One plus 6T con Android 9 , móvil bq Aquaris M5 con Android 5 y con un PocoPhone F1 con Android 9. Podemos observar que el

juego se adapta a las pantallas de ambos móviles de forma correcta. La aplicación está hecha para ser soportada hasta Android 4.1 (Jelly Bean)



6.4 Lista de modificaciones o ampliaciones futuras que se podrían aplicar a la aplicación proyectada.

1. Que el usuario pueda seleccionar una foto de perfil.
2. Crear una configuración de usuario, para la modificación de la contraseña, que se pueda añadir un email.
3. Añadirle un tiempo a cada respuesta.
4. Poder seleccionar la dificultad con la que se quiere jugar, el campo está en la base de datos, pero no se ha podido desarrollar la idea por falta de tiempo.
5. Crear un sistema de puntuación mejor usando la variable del tiempo de respuesta de cada pregunta para calcular la puntuación por cada pregunta.
6. Implementar un servidor que funcione continuamente y con dirección fija, ya que en esta primera versión se tiene que cambiar la IP dentro de un script del proyecto según la red en la que se esté conectado.
7. Recuperación de la cuenta a través del correo.
8. Crear la aplicación para PC, en estos momentos la interfaz está hecha para dispositivos móviles, si se genera la aplicación para PC se ve desproporcionada.

6.5 Plan de mantenimiento de la aplicación

A medida que se aumentan los usuarios la conexión a la base de datos será más lenta por lo que se tendrá que montar la base de datos y el servidor web en un servidor que soporte un amplio número de conexiones. Por otra parte, se tendrá que crear un script para la limpieza de la tabla partidas si el registro llega a ser muy grande. Se deberá eliminar las partidas con mayor antigüedad.

7 Estudio de la viabilidad del proyecto

Con el plazo de tiempo que se tiene para el desarrollo de la aplicación (40 horas del módulo) vemos que no es viable la realización del proyecto ni siquiera en su primera versión.

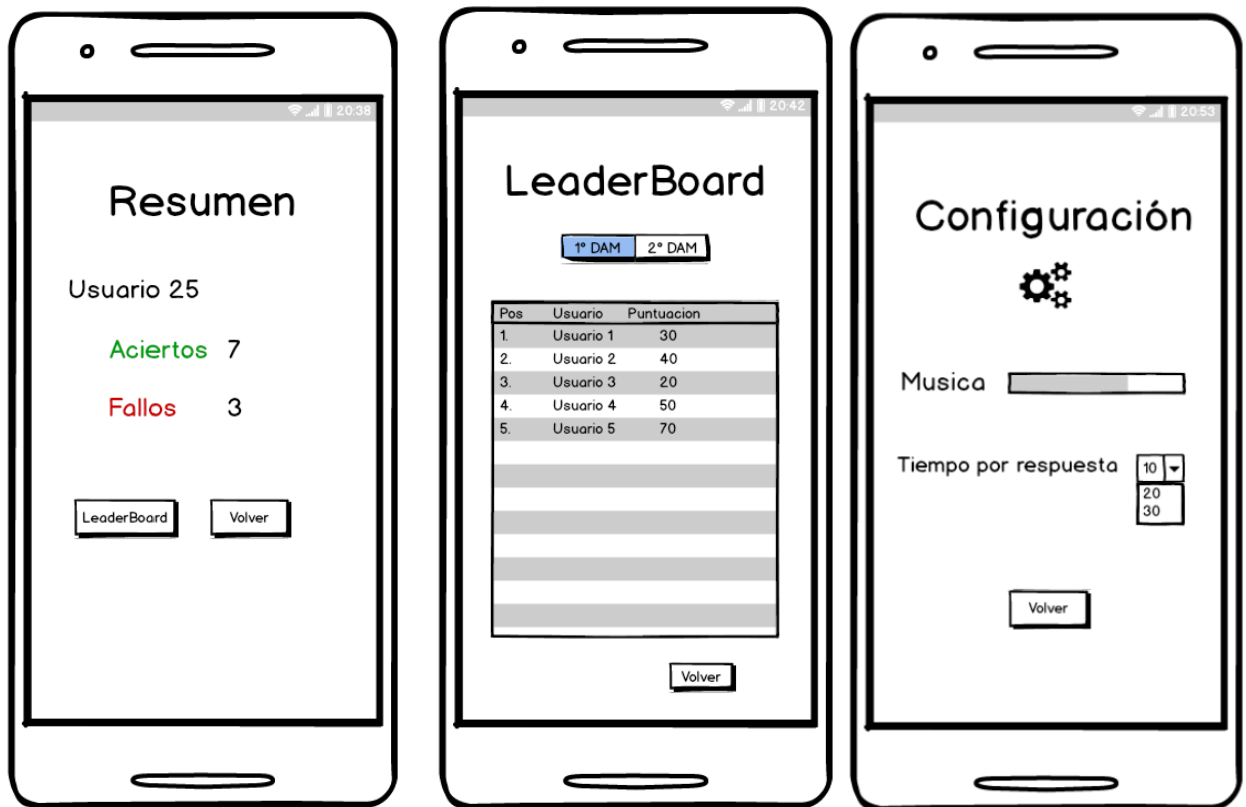
El desarrollo de la primera versión de este pequeño proyecto lleva bastante tiempo, en este caso se ha excedido el tiempo que se tiene para su desarrollo, además la primera complicación y la más importante es que el número de personas trabajando en el proyecto es una. Con más tiempo para la realización del proyecto y aplicando las mejoras futuras mencionadas anteriormente y con ayudas económicas que ayuden a cubrir los gastos del proyecto (licencias de las herramientas a utilizar, personal, servidor, etc.), **HaroldEduca** puede ser un proyecto que se puede lanzar al mercado.

8 Documentación del diseño e implementación de la solución adoptada

8.1 Prototipo de la aplicación. Diseño de interfaces

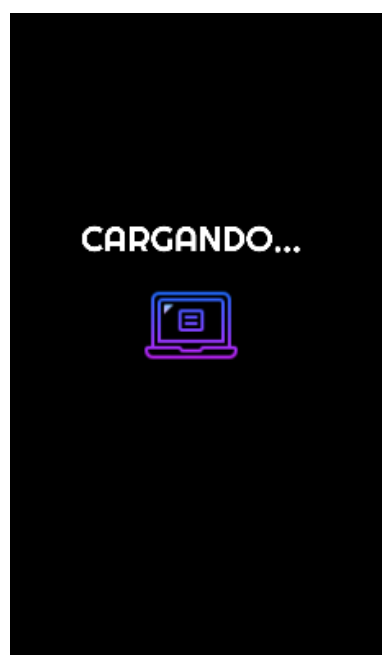
Lo primero que se hizo antes de empezar a desarrollar el proyecto fue un prototipo de las principales pantallas o escenas que iba a tener nuestro juego, además de una serie de requisitos para poder empezar la aplicación.





Diseño de las interfaces del juego una vez se ha terminado la primera versión:

- **Ventana de carga:** Se comprueba si hay una sesión iniciada o no, si la hay se pasa directamente a la escena del Menu sino se muestra la escena de Login.



- Escena de Login



HaroldEduca



LOG IN **SIGN IN**



The login screen features a light blue background. At the top, the text 'HaroldEduca' is displayed in a large, bold, black font. Below it is a black circle containing a white computer monitor icon. There are two white input fields: the first is labeled 'Username(4 chrs min)' and the second is labeled 'Password(4 chrs min)'. Below these fields are two green buttons with black text: 'LOG IN' and 'SIGN IN'. At the bottom left, there is a yellow circular icon with a white left-pointing arrow.

- Escena Menu



 harold11

MENU

CATEGORIAS

RANKING

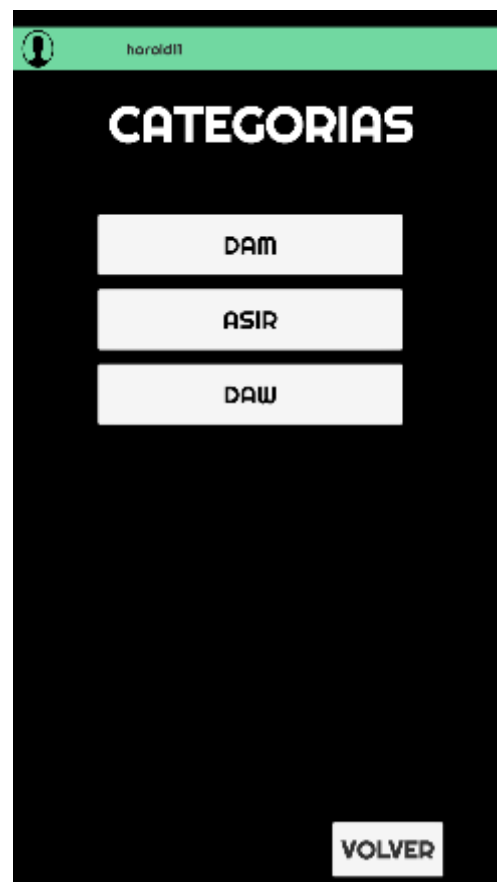
CONFIGURACIÓN

INFORMACIÓN

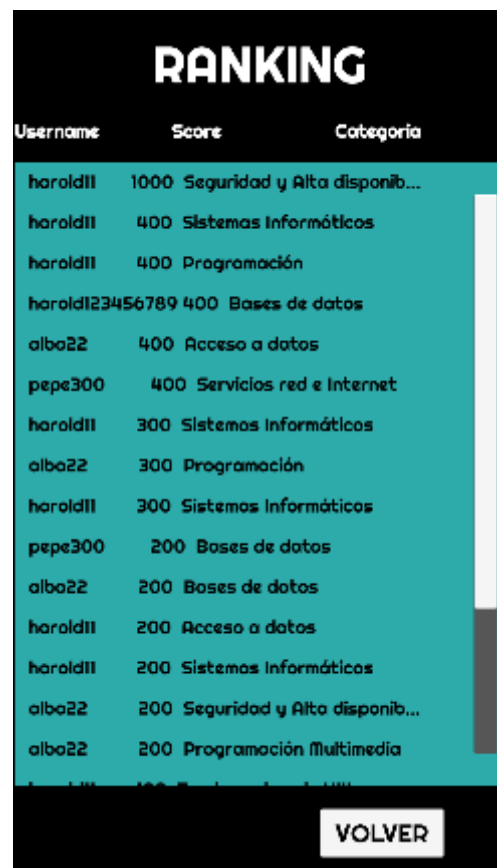
LOG OUT **SALIR**

The menu screen has a black background. At the top, there is a green header bar containing a small user icon and the text 'harold11'. Below the header, the word 'MENU' is written in large, bold, white capital letters. Underneath 'MENU' are four white rectangular buttons with black text, stacked vertically: 'CATEGORIAS', 'RANKING', 'CONFIGURACIÓN', and 'INFORMACIÓN'. At the bottom of the screen are two more white rectangular buttons with black text: 'LOG OUT' and 'SALIR'.

- Escena categorías



- Escena ranking



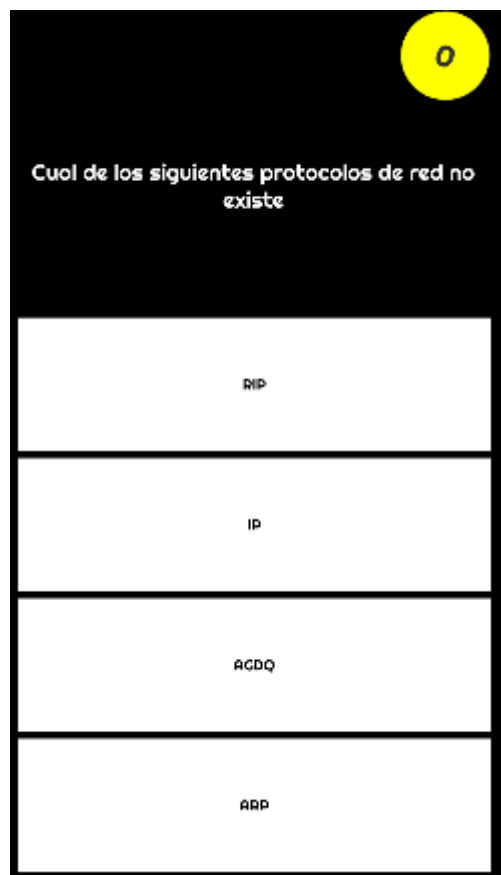
- Escena de configuración



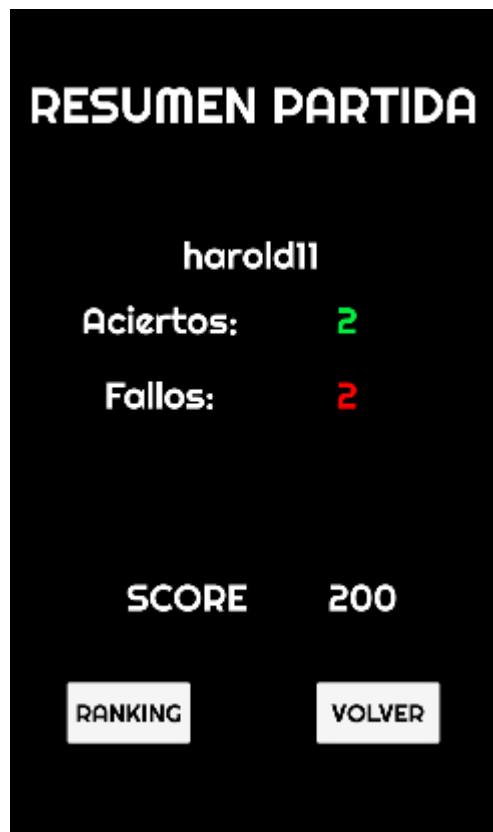
- Escena de información



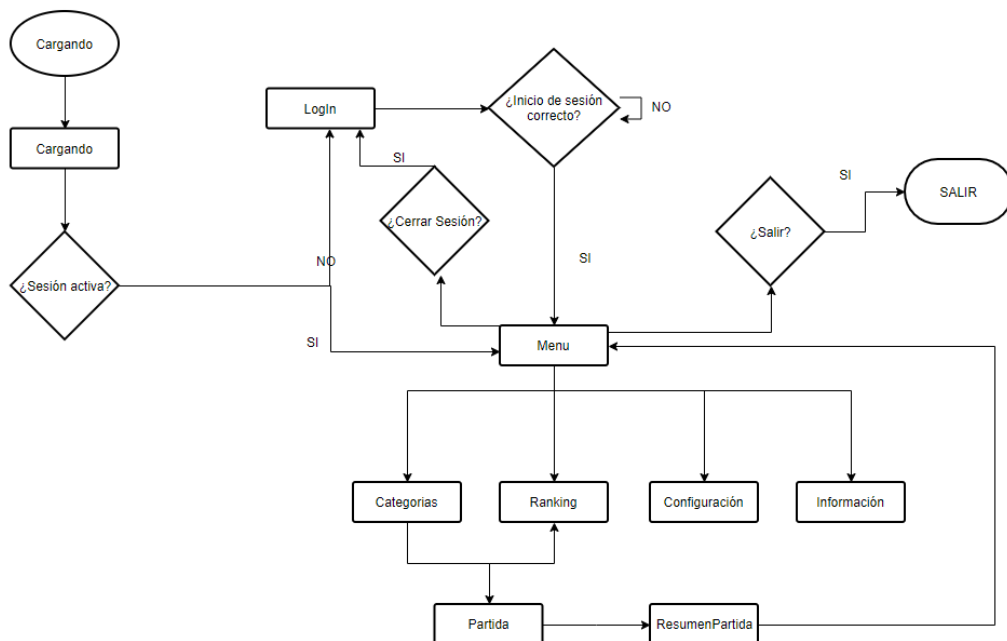
- Escena partida



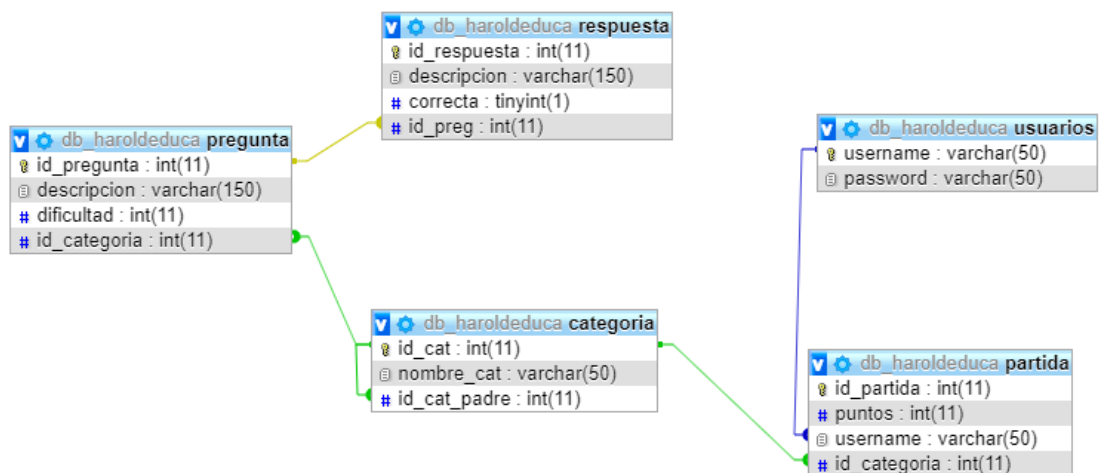
- Escena ResumenPartida



8.2 Diagrama de flujo entre escenas















8.3 Diseño de la base de datos. Diagrama Entidades/Relación



8.4 Scripts PHP

En esta sección se va a mostrar un ejemplo de los scripts en PHP que son los que permiten el acceso a la base de datos, es decir, desde el juego se llama a estos scripts y estos son los encargados del acceso a la base de datos.

En la siguiente imagen se muestran todos los ficheros .PHP que se usan en el proyecto.

	Codeigniter	05/08/2019 11:02	Carpeta de archivos	
	config.php	15/11/2019 16:43	Archivo PHP	1 KB
	GetCategoriasBien.php	04/12/2019 19:58	Archivo PHP	3 KB
	GetIdCategoria.php	02/12/2019 12:20	Archivo PHP	1 KB
	GetLoggedUser.php	27/11/2019 19:54	Archivo PHP	1 KB
	GetNombreCategoria.php	06/12/2019 12:15	Archivo PHP	1 KB
	GetPreguntas.php	30/11/2019 21:37	Archivo PHP	2 KB
	GetRanking.php	05/12/2019 20:09	Archivo PHP	1 KB
	GetRespuestas.php	30/11/2019 21:17	Archivo PHP	1 KB
	GuardarPartida.php	02/12/2019 11:32	Archivo PHP	1 KB
	Login.php	05/12/2019 21:29	Archivo PHP	1 KB
	RegisterUser.php	05/12/2019 21:26	Archivo PHP	1 KB

Ahora vamos a ver el script de Login.php

Lo primero que hace el script es inicializar las variables globales que tenemos las cuales son: DB_HOST que es la IP del servidor, DB_USER el usuario que se conecta a la base de datos, la contraseña y el nombre de la base de datos.

Lo siguiente que hace es hacer la conexión si todo va bien lee los parámetros que le llegan desde el juego (usuario y password que hemos introducido), comprueba que existe el usuario en la tabla usuarios y después comprueba que la password obtenida de la consulta es igual que la que escribimos en el juego. Devuelve un mensaje que tratamos en el juego.

```

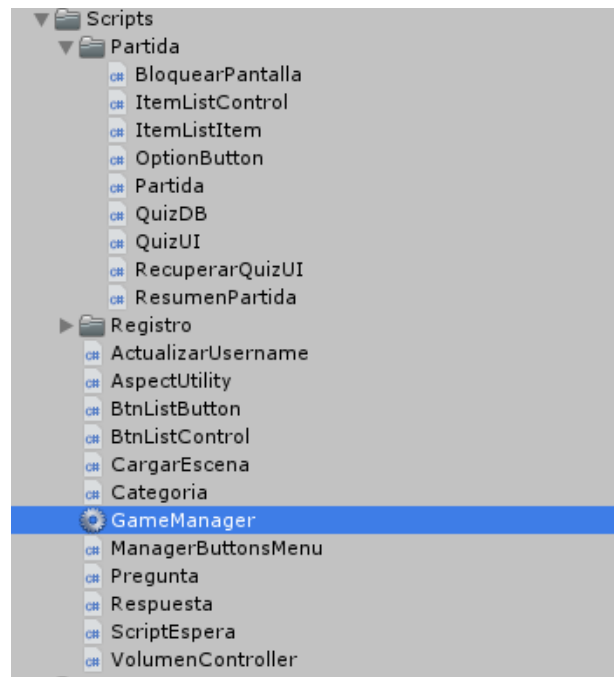
1  <?php
2
3  // Create connection
4  require_once 'config.php';
5  $conn = new mysqli(DB_HOST, DB_USER, DB_PASSWORD, DB_DATABASE);
6
7  // Check connection
8  if ($conn->connect_error) {
9      die("Connection failed: " . $conn->connect_error);
10 }
11
12 //Variables que envia el juego
13 $loginUser = $_POST["loginUser"];
14 $loginPass = $_POST["loginPass"];
15
16 $sql = "SELECT password FROM usuarios WHERE username = '" . $loginUser . "'";
17 $result = $conn->query($sql);
18
19 if ($result->num_rows > 0) {
20     while($row = $result->fetch_assoc()) {
21         if ($row["password"] == $loginPass) {
22             echo "Login success";
23         }else{
24             echo "Password incorrecto";
25         }
26     }
27 } else {
28     echo "Usuario no existe";
29 }
30 $conn->close();
31
32 ?>

```

8.5 Scripts de C#

En este caso son los scripts para controlar la interfaz de usuario, para hacer la llamada los ficheros PHP, etc.

El script más importante es el *GameManager.cs* porque se le asigna a un objeto que estará activo durante todo el juego y es el que usaremos para controlar todo el juego. Se obtiene de un asset llamada **Universe**, que se puede descargar de la Asset Store de Unity.



En el *GameManager.cs* tenemos algunos métodos importantes como:

- CambiarEscena(): permite la transición entre escenas.
- CheckConexion(): Comprobar que hay conexión a internet.
- GetCategoriasBien(): Para obtener las categorías según su padre.
- GetPreguntas(): Devuelve una lista con las preguntas de la categoría que se le pasa.
- ObtenerRespuestas(): Devuelve una lista de respuestas de la pregunta que se le pasa.
- NextPregunta(): Obtiene los datos de la siguiente pregunta y actualiza la pregunta y los botones en la escena Partida.
- Encriptar(): Enviar la contraseña encriptado para que en el caso de que haya fugas de información no se pueda usar.
- GuardarPartida(): Recoge nombre del usuario, aciertos, fallos y puntuación y llama al script de PHP para hacer la inserción en la tabla Partidas.

Otro asset que se usa es **JSON .net**, también es gratuita y sirve para convertir un json en objetos de una determinada clase. En este caso se usa para las categorías, preguntas, respuestas y partidas.

8.6 Aplicación en CodeIgniter con librería Grocery CRUD.

Vamos a ver los scripts que se usan para crear el manejo a la base de datos de una forma muy sencilla con CodeIgniter y Grocery CRUD.

La vista principal son tres enlaces a las tablas Categoría, Pregunta y Respuesta, cada uno nos lleva al CRUD de cada tabla.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="utf-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <?php
7 foreach($css_files as $file): ?>
8     <link type="text/css" rel="stylesheet" href="<?php echo $file; ?>" />
9 <?php endforeach; ?>
10 </head>
11 <body>
12     <div>
13         <a href='<?php echo site_url('HaroldEduca/categoria_management')?'>'>Categoría</a>
14         <a href='<?php echo site_url('HaroldEduca/pregunta_management')?'>'>Pregunta</a>
15         <a href='<?php echo site_url('HaroldEduca/respuesta_management')?'>'>Respuesta</a>
16     </div>
17     <div style='height:20px;'></div>
18     <div style="padding: 10px">
19         <?php echo $output; ?>
20     </div>
21     <?php foreach($js_files as $file): ?>
22         <script src="<?php echo $file; ?>"></script>
23     <?php endforeach; ?>
24 </body>
25 </html>
```

Ahora vemos que genera este código.

[Categoría](#) [Pregunta](#) [Respuesta](#)

Seguimos con el código para mostrar la información de las tablas, ver cada registro, actualizar, eliminar, añadir, etc. Aquí lo que se hace es mostrar información y crear las relaciones entre tablas si es que existen.

```
public function categoria_management()
{
    try{
        $crud = new grocery_CRUD();

        $crud->set_theme('datatables');
        $crud->set_relation('id_cat_padre','categoria','nombre_cat');
        $crud->display_as('nombre_cat','Categoría');
        $crud->display_as('id_cat_padre','Categoría Padre');
        $crud->set_table('categoria');
        $crud->set_subject('Categoría');
        $crud->required_fields('nombre_cat');
        $crud->columns('nombre_cat','id_cat_padre');

        $output = $crud->render();

        $this->_example_output($output);
    }catch(Exception $e){
        show_error($e->getMessage().' --- '.$e->getTraceAsString());
    }
}
```

La vista que genera este código es la siguiente:

[Categoria Pregunta Respuesta](#)

Add Categoria

ExportPrint

Show 25 entries

Search:

Categoria	Categoria Padre	Actions			
1º ASIR	ASIR	View	Clone	Edit	Delete
1º DAM	DAM	View	Clone	Edit	Delete
1º DAW	DAW	View	Clone	Edit	Delete
2º ASIR	ASIR	View	Clone	Edit	Delete
2º DAM	DAM	View	Clone	Edit	Delete
2º DAW	DAW	View	Clone	Edit	Delete
Acceso a datos	2º DAM	View	Clone	Edit	Delete
Admin. de SO	2º ASIR	View	Clone	Edit	Delete
ASIR		View	Clone	Edit	Delete
Bachillerato		View	Clone	Edit	Delete
Bases de datos	1º DAM	View	Clone	Edit	Delete

Cuando se va a añadir una categoría, las opciones de la categoría padre son categorías ya existentes, ya que en el código se ha creado la relación.

[Categoria Pregunta Respuesta](#)

Add Categoria

Categoria* :

Categoria Padre :

Select Categoria Padre

1º ASIR

1º DAM

1º DAW

2º ASIR

2º DAM

2º DAW

Acceso a datos

Admin. de SO

ASIR

Bachillerato

Save

Save and go back to

Por lo tanto vemos que con tres métodos, uno para cada tabla, podemos crear un CRUD fácil, sencillo y vistoso para el administrador de la aplicación.

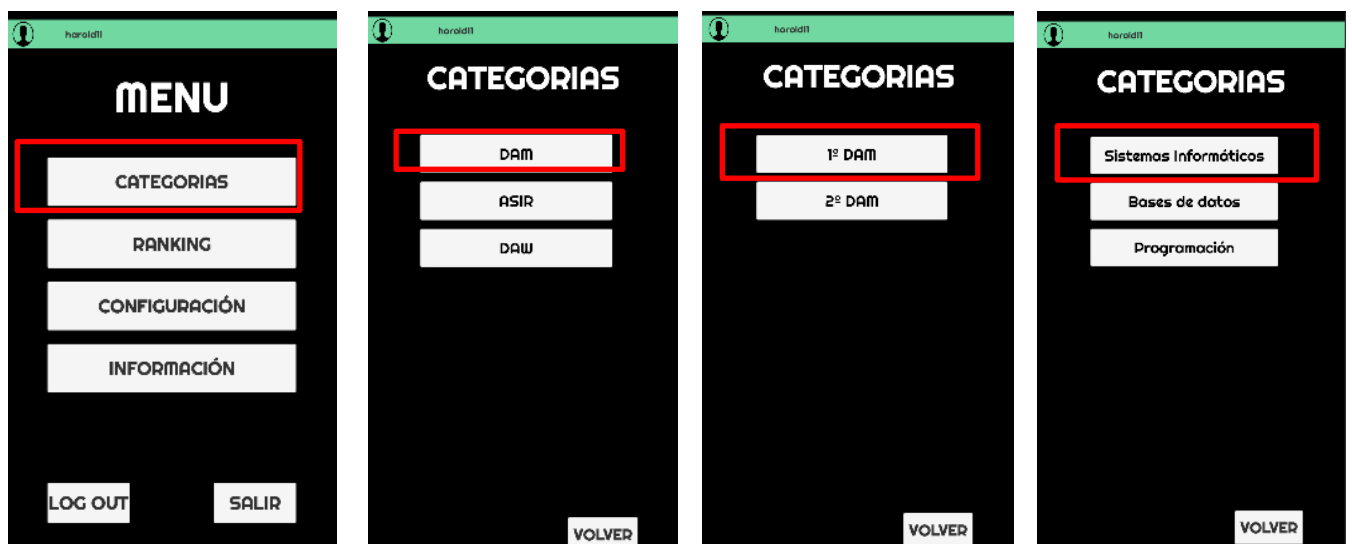
8.7 Funcionamiento de la aplicación

Se mostrará y se explicará una interacción entre el usuario y la aplicación a través de imágenes.

1. **Crear cuenta (Sign In):** El nombre del usuario no debe existir ya en la base de datos, los campos tienen un mínimo de caracteres (4), una vez se validan ambos campos se hace la inserción en la base de datos, se guarda la sesión y se mantiene activa a menos de que el usuario cierre sesión. Si ya se ha creado un usuario anteriormente hay que usar el botón de **Log In**.



2. **Seleccionar categoría:** En el menú se selecciona CATEGORIAS donde se mostrarán las categorías disponibles. Solo se mostrarán las categorías que tengan hijas o que tengan más de cuatro preguntas, es decir, que se pueda empezar una partida de esa categoría



Una vez se llega a una categoría con preguntas empieza la partida.

3. **Partida:** Se van mostrando las preguntas según se van contestando, en la parte superior derecha aparece la puntuación en cada momento de la partida, si se acierta una pregunta cambia el color del botón a verde y si se falla a rojo. El juego se espera dos segundos hasta pasar a la siguiente pregunta. Cuando se responde una pregunta durante esos dos segundos se activa un panel transparente que no permite que selecciones otra respuesta.



Una vez se responden todas las preguntas, se pasa al resumen de la partida:

4. **Resumen de la partida:** Lo primero que se hace aquí es la inserción de la partida en la base de datos, a continuación se muestra el nombre del jugador, fallos, aciertos y puntuación. Desde aquí podemos y a ver el ranking o volver al menú.



5. **Ranking:** Se puede acceder desde el menú principal o después de que se termina una partida. Muestra las mejores puntuaciones de todas las partidas que se han jugado hasta el momento.



The screenshot shows a 'RANKING' screen with a black background and a teal-colored list of scores. The list has three columns: 'Username', 'Score', and 'Categoría'. The scores are sorted in descending order. At the bottom of the screen, there is a white button labeled 'VOLVER'.

Username	Score	Categoría
harold11	1000	Seguridad y Alta disponib...
harold11	400	Sistemas Informáticos
harold11	400	Programación
harold123456789	400	Bases de datos
pepe300	400	Servicios red e Internet
albo22	400	Acceso a datos
harold11	300	Sistemas Informáticos
harold11	300	Sistemas Informáticos
albo22	300	Programación
albo22	200	Bases de datos
pepe300	200	Bases de datos
albo22	200	Programación Multimedia
harold11	200	Acceso a datos
harold11	200	Sistemas Informáticos
harold11	200	Sistemas Informáticos

6. **Configuración:** Desde el menú podemos acceder a la configuración global del juego, donde podemos subir o bajar el volumen a la música de fondo.



7. **Información:** Podemos ver la información acerca del proyecto y un email de contacto con el desarrollador del proyecto.



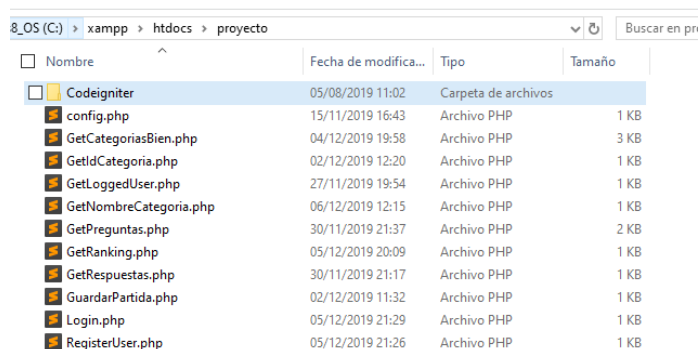
La interfaz de la aplicación es muy intuitiva y clara, el usuario desde la primera vez que la use sabrá moverse a través de ella. En todas las escenas tiene un botón de volver que permite regresar al menú.

9 Puesta a punto en Hostinger

Hostinger es un servicio que nos permite que nuestra base de datos y nuestros archivos PHP sean visibles por cualquier persona, esto no sucede si utilizamos nuestro ordenador con XAMPP como servidor, ya que solo tendrán acceso a nuestros recursos aquellos dispositivos que estén en la misma red.

Hostinger, al igual que XAMPP, nos permite tener una base de datos MySQL y poder manejarla a través de phpmyadmin, así que lo que haremos es exportar la base de datos desde XAMPP e importarla a nuestra nueva base de datos en Hostinger.

Por otra parte, tenemos CodeIgniter y los scripts PHP que teníamos para el acceso a la base de datos. Lo que haremos es subir estos archivos Hostinger a la carpeta 'public_html/proyecto'.



Nombre	Fecha de modifica...	Tipo	Tamaño
Codeigniter	05/08/2019 11:02	Carpeta de archivos	
config.php	15/11/2019 16:43	Archivo PHP	1 KB
GetCategoriasBien.php	04/12/2019 19:58	Archivo PHP	3 KB
GetIdCategoria.php	02/12/2019 12:20	Archivo PHP	1 KB
GetLoggedUser.php	27/11/2019 19:54	Archivo PHP	1 KB
GetNombreCategoria.php	06/12/2019 12:15	Archivo PHP	1 KB
GetPreguntas.php	30/11/2019 21:37	Archivo PHP	2 KB
GetRanking.php	05/12/2019 20:09	Archivo PHP	1 KB
GetRespuestas.php	30/11/2019 21:17	Archivo PHP	1 KB
GuardarPartida.php	02/12/2019 11:32	Archivo PHP	1 KB
Login.php	05/12/2019 21:29	Archivo PHP	1 KB
RegisterUser.php	05/12/2019 21:26	Archivo PHP	1 KB

A continuación, debemos actualizar los archivos de configuración, es decir, cambiar el nombre de la base de datos, el host, el usuario, etc.

Por último debemos ir a la carpeta /Assets/Scripts y buscar el Script GameManager.cs para buscar la variable `IP_SERVER` y asignarle la IP del nuevo servidor y volver a generar el APK del juego.

```
private static string IP_SERVER = "192.168.1.15";  
//private static string IP_SERVER = "192.168.70.49";  
//private static string IP_SERVER = "haroldeduca.000webhostapp.com";
```

El **primer problema** que nos encontramos es que las versiones de PHP no coinciden y desde Hostinger no podemos tocar la configuración de PHP ya que es un sitio gratuito y para poder configurarlo debemos pagar. Esto provoca que no se muestren bien los caracteres especiales en nuestra aplicación.

Ejemplo: **Hostinger**



XAMPP desde local



Desde los scripts de acceso a los datos de la base de datos se usa la función `utf_encode()` para los caracteres especiales y es lo que se usa en el servidor XAMPP, pero en Hostinger no lo reconoce.

Se comprueba también que ambas bases de datos tienen el mismo charset y cotejamiento, los scripts para acceder a los datos son los mismos, por lo que no se ha podido encontrar una solución al problema.

10 Resultado Final

Se van a generar dos instaladores (APKs) para el proyecto, la primera será para que se acceda a los servicios del hosting, que se llamara `haroldeduca.apk` y los profesores puedan probar el funcionamiento de la aplicación.

La segunda será para acceder a los servicios en XAMPP, se utilizara para la presentación en clase del proyecto y se llamara `haroldeducaInstituto.apk`.

`haroldeduca.apk` enlace:

<https://mega.nz/#!wQVAASJS!yxvDhtiSQiawqCxmoLEqTORD1hkTYh48icNVTjXlyxg>

`haroldeducaInstituto.apk` enlace: <https://mega.nz/#!wZMwGCID!aJ9HfrDB-9c7ip9pddjaLAGJDnWeqhovuyQT-7XOggE>

11 Bibliografía y fuentes de información

CodeIgniter:

- https://codeigniter.com/user_guide/index.html
- <https://www.grocerycrud.com/examples/>
- <https://www.youtube.com/watch?v=16t6VT3la1s&feature=youtu.be>

Llamada a scripts PHP:

- <https://docs.unity3d.com/ScriptReference/Networking.UnityWebRequest.html>
- <https://docs.unity3d.com/ScriptReference/Networking.UnityWebRequest.Post.html>

Consultas generales:

- <https://stackoverflow.com/>
- <https://answers.unity.com/index.html>
- <https://www.youtube.com/>

Generar listas dinámicas basadas en consultas MySQL:

- <https://www.youtube.com/watch?v=2TYLBusJKjc&t=599s>

Escena de partida:

- <https://www.youtube.com/watch?v=eKzKntYG7Pc>